Vol. 36, No. 4 (2026) | DOI: 10.37190/ord/231153

OPEN ACCESS

Operations Research and Decisions

www.ord.pwr.edu.pl

OPERATIONS RESEARCH AND DECISIONS QUARTERLY

Multi-component hybrid deep learning model for railway passenger demand forecasting

Iqbal Kharisudin*[1] Merlinda Lavenia[2]

[1]*Department of Statistics and Data Science, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Semarang, Indonesia*

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

**Corresponding author, email address: iqbalkharisudin@mail.unnes.ac.id*

Abstract

Railway transportation plays a critical role in supporting sustainable mobility in Indonesia, yet significant fluctuations in passenger demand often lead to congestion and operational challenges. This study presents a systematic evaluation of decomposition-based forecasting frameworks for railway passenger demand prediction by integrating Seasonal-Trend Decomposition using Loess (STL), Empirical Mode Decomposition (EMD) applied to residual components, and Fuzzy C-Means (FCM) clustering. Using the Argo Muria train service as a case study, multiple deep learning models, including LSTM, GRU, RNN, CNN, and BiLSTM, are trained on decomposed components, and their forecasts are combined linearly. Model performance is evaluated using a rolling-origin strategy across multiple stations. At the primary destination station, Semarang-Gambir, the best configuration achieves an MAE of 19.88, RMSE of 26.79, sMAPE of 8.97, and R^2 of 0.84. Consistent results across stations demonstrate the framework's robustness and generalization capability.

Keywords: *signal decomposition, fuzzy clustering, time series forecasting, railway passenger demand, sustainable transportation*

1. Introduction

Over recent decades, Indonesia has witnessed substantial growth and advancement across various sectors, with the transportation industry emerging as a critical driver of national development [2]. Within this sector, public transport networks in major cities experience fluctuations in passenger demand [21]. Among various modes of public transportation, trains are a preferred option for many Indonesian people [25]. Nevertheless, the system concurrently faces multifaceted and evolving challenges characterized by growing complexity and variability [30]. The passenger flow in intercity and urban railways exhibits similar characteristics, including spatial and temporal disparities as well as abrupt increases in passenger

Received 12 March 2026, accepted 16 August 2026, published online 16 August 2026
ISSN 2391-6060 (Online)/

demand [17]. Due to the irregular patterns and complex temporal dynamics of passenger flow, forecasting passenger demand during large-scale events poses significantly greater challenges than under normal operating conditions [38]. More broadly, short-term passenger flow forecasting is inherently difficult because passenger movement exhibits nonlinear behavior and is highly influenced by dynamic environmental factors [16]. Passenger flows in railway transportation systems exhibit cyclical fluctuations as well as substantial noise resulting from stochastic factors [5, 30, 33]. Urban special events, including holidays and large gatherings, can introduce additional variability, causing significant deviations in passenger flows [37]. These flows are also characterized by pronounced seasonal and trend patterns, while the residual component often contains stochastic fluctuations that are challenging to model accurately [4]. Even when time-series data lack an apparent trend, they often indicate the presence of an underlying trend component [6]. Owing to the intrinsic nonlinear characteristics of railway passenger flow patterns [8] and the inherently irregular behavior of the residual component, hybrid models that integrate multiple techniques have been proposed to effectively handle residual uncertainty.

A critical aspect of hybrid forecasting approaches is the selection of an appropriate decomposition technique, as accurate component separation is essential for effective residual management and estimation accuracy. Techniques such as Seasonal-Trend Decomposition Based On Loess (STL) [29], Empirical Mode Decomposition (EMD) [10], and Wavelet Transform (WT) [33] have been widely applied to enhance sequential data analysis [35]. Among these methods, STL is particularly effective for decomposing time-series data into trend, seasonal, and residual components due to its flexibility in handling nonlinear and nonstationary patterns [20]. To further capture intrinsic oscillatory modes and complex residual fluctuations, EMD is commonly employed as a complementary decomposition technique in transport-related studies [33]. Subsequently, Fuzzy C-Means (FCM) clustering is applied to characterize residual patterns, reduce uncertainty, and improve feature quality for forecasting models [3].

Although decomposition-based hybrid forecasting has been widely explored in previous studies, most existing works primarily focus on improving predictive accuracy through model combinations, while limited attention has been given to understanding the individual contribution of each decomposition and residual-handling stage. In particular, residual components are commonly treated as stochastic noise and are therefore either discarded or modeled without further structural analysis. To address this gap, this study conducts a systematic residual-aware evaluation of decomposition-based forecasting frameworks under a unified experimental setting. Multiple configurations, including STL, STL-EMD, and STL-EMD-FCM with and without Sample Entropy-based feature extraction, are comparatively analyzed through an ablation-driven framework. This enables the quantitative assessment of the contribution of residual decomposition, clustering mechanisms, and feature extraction strategies to forecasting performance and robustness. Therefore, the contribution of this study lies not in proposing a new hybrid architecture, but in providing empirical and methodological insight into how different decomposition design choices influence forecasting behavior in highly dynamic railway passenger flow series.

2. Literature review

In recent years, numerous approaches have been developed for short-term passenger flow prediction, which can generally be categorized into three main groups: traditional statistical models, machine learn-

Table 1. Summary of decomposition and forecasting techniques across studies.

Study Paper	Decomposition	Dataset	Forecasting Technique
Chen et al. (2024) [4]	STL, SGMD	Beijing, Guangzhou, and Shenzhen Subway	ARIMA, SARIMA, LSSVR, KELM, LSTM
Wang et al. (2024) [29]	STL	MTA NYC Subway	GaLSTM, LSTM, SVM, GNN
Sun et al. (2024) [27]	STL	URT Hangzhou	ARIMA, SVM, GRU, GCN
Qin et al. (2024) [23]	EMD	Daxing Airport Station	HWAM, ARIMA, GRU, Prophet
Song et al. (2024) [26]	CEEMDAN	Harbin Metro Station	BiLSTM, Attention LSTM, CNN-GRU
Yuan et al. (2024) [36]	EEMD	Wuhan-Guangzhou HSR	MSVM, GSVM
Guan et al. (2024) [8]	EEMD, CEEMDAN	Beijing-Shanghai HSR	ARIMA, SVM, LSTM, BiLSTM
Xie et al. (2024) [33]	EMD, EEMD, EWT	Xi'an Metro Station	ARIMA, SVM, LSTM, GRU
Ma et al. (2024) [19]	-	Beijing Subway Station	RNN, GRU, CNN, LSTM, Attention
Yin et al. (2025) [35]	STL	Indoor Air Quality	LSTM, Attention LSTM

ing techniques, and deep learning approaches [7, 24]. Traditional models often struggle to capture the dynamic inter-station spatial relationships, leading to reduced prediction accuracy [34]. In contrast, Deep learning has advanced significantly in transportation [18], particularly sequential data model[39], such as LSTM [1], GRU [27], RNN [22], CNN [13], and their variants have demonstrated strong capability in modeling sequential dependencies and nonlinear dynamics in large-scale passenger flow data [19, 32].

Nevertheless, these models often struggle when learning directly from raw time-series signals in which trend, seasonal, and irregular components remain intertwined. Consequently, modern forecasting frameworks increasingly rely on decomposing the time series and modeling trend, seasonal, and residual components separately, allowing deep learning models to more effectively capture nonlinear dynamics and long-range temporal dependencies. This progression highlights the growing importance of decomposition techniques and residual analysis in enhancing predictive accuracy. Building upon the decomposition and a thorough analysis of the residual component, earlier studies have incorporated hybrid approaches that pair decomposition methods with deep learning-driven residual modeling to fully utilize residual information in the forecasting process. For instance, [4] proposed an STL-SGMD-Fuzzy framework integrated with LSTM, LSSVR, and KELM to improve forecasting accuracy and stability. STL-based hybrid models have also been applied in urban rail forecasting, including genetic-algorithm-optimized LSTM models [29] and STL-GCN-GRU architectures for capturing spatio-temporal dependencies [27]. Beyond STL, [23] developed an HWAM-EMD-GRU framework for airport rail passenger forecasting using decomposition-based component prediction. Similar decomposition-aware deep learning strategies have also demonstrated effectiveness in other time-series applications, such as gold price forecasting [15] and short-term electricity load forecasting [14]. However, most existing studies primarily focus on integrating decomposition methods with forecasting models, while limited attention has been given to systematically analyzing the contribution of each modeling stage. In particular, the structural role of residual components remains relatively underexplored, as they are often treated as stochastic noise rather than potentially informative signals.

3. Methodology

A schematic overview of this study addresses limitations of previous approaches by proposing a structured passenger flow prediction framework with multiple modeling stages, residual processing, cluster-

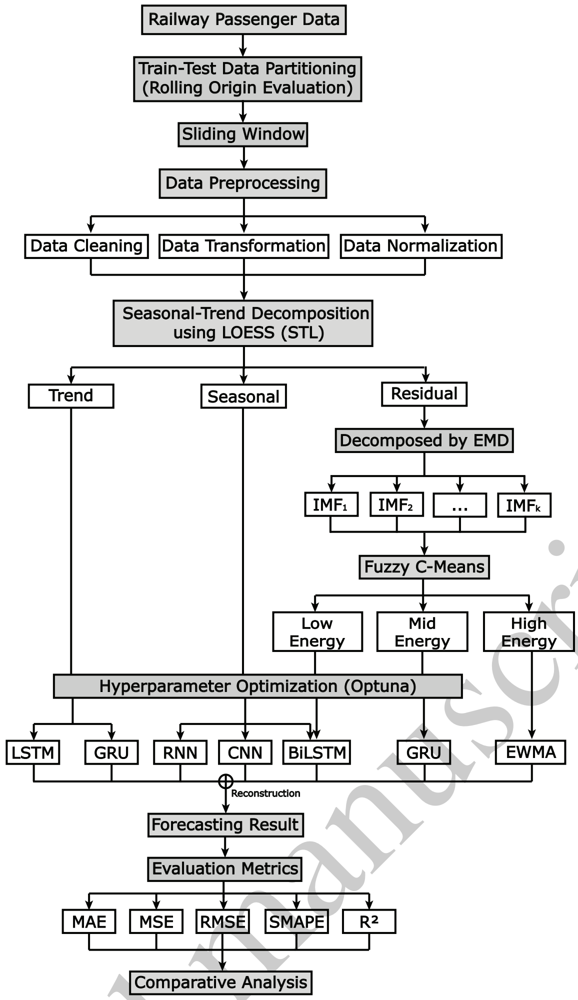


Figure 1. Workflow overview.

ing, and component-wise modeling strategies. The methodology consists of six main steps: (1) data preprocessing, (2) time series decomposition using STL, (3) residual processing through EMD and clustering, (4) component-wise model construction, (5) reconstruction of the forecasted components, and (6) performance evaluation and comparative analysis. The modeling workflow is organized into a series of structured and systematic stages as illustrated in Figure 1. The proposed framework follows a multi-stage decomposition and residual processing strategy. First, the original time series is decomposed using STL into trend, seasonal, and residual components. While the trend and seasonal components are modeled directly, the residual component is further analyzed due to its complex and non-stationary nature. Specifically, EMD is applied exclusively to the residual component to extract intrinsic mode functions (IMFs). These IMFs are subsequently grouped into low-, mid-, and high-energy components using FCM clustering for further modeling.

3.1. Dataset overview and description

The dataset analyzed in this study was obtained from an intercity railway service operating between Semarang Tawang Station and Gambir Station in Jakarta. The railway corridor follows the northern Java railway line, connecting major economic and urban centers along the northern coast of Java Island. Along the route, the train stops at Pekalongan, Tegal, Cirebon, Bekasi, and Jatinegara Stations before terminating at Gambir Station. To illustrate the temporal characteristics of passenger demand across all destination stations, a line plot of daily passenger volumes is presented in Figure 2.

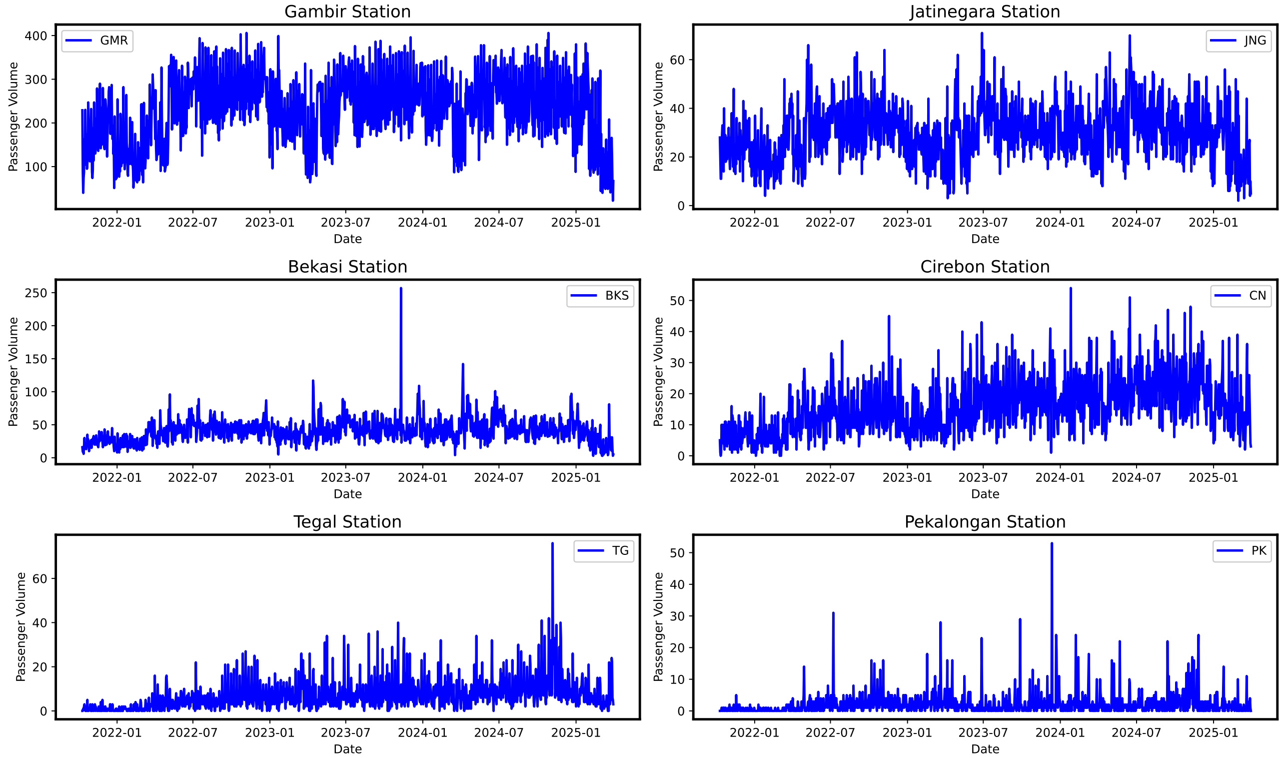


Figure 2. Daily passenger volumes across all destination stations

These destination stations represent diverse regional and metropolitan mobility characteristics that may contribute to heterogeneous passenger demand patterns. The estimated travel distances and travel times between Semarang Tawang Station and each destination station are summarized in Table 2, providing contextual information for passenger flow forecasting. To support the analysis of passenger demand

Table 2. Route characteristics and estimated travel distances.

Destination Station	Estimated Distance (km)	Estimated Travel Time	Potential Impact on Passenger Demand
Pekalongan	89	1-2 hours	Short-distance travel tends to generate relatively low demand
Tegal	150	2-3 hours	Moderate regional demand may occur
Cirebon	225	3-4 hours	Interregional travel creates moderate demand
Bekasi	421	4-5 hours	Long-distance travel increases passenger demand
Jatinegara	437	5-6 hours	Transit passengers may contribute to demand
Gambir	440	5-6 hours	Final destination of the route, leading to higher passenger demand

along this route, the dataset used in this study was officially obtained from PT Kereta Api Indonesia (Persero). The dataset spans from 10 October 2021 to 31 March 2025 and contains daily passenger volumes for six destination stations: Gambir (GMR), Jatinegara (JNG), Bekasi (BKS), Cirebon (CN), Tegal (TG), and Pekalongan (PK), with Semarang Tawang Station serving as the departure point. A total of 1,269 daily observations were collected. The dataset exhibits heterogeneous and nonlinear passenger flow patterns across destinations, motivating the use of advanced forecasting approaches. Descriptive statistics of the datasets are summarized in Table 3.

The original dataset was recorded in a transactional format, where each row represents a passenger departure event for a specific destination station and departure time. After aggregation into a daily mul-

Table 3. Descriptive statistics of datasets.

Station	Length	Non-null	Missing	Min	Max	Mean	Std Dev	Q1 (25%)	Q2 (50%)	Q3 (75%)
Gambir	1269	1269	0	22	406	227.00	77.75	174.00	230.00	287.00
Jatinegara	1269	1269	0	2	71	29.52	11.27	22.00	29.00	37.00
Bekasi	1269	1269	0	3	257	40.37	18.09	28.00	39.00	50.00
Cirebon	1269	1265	4	0	54	15.73	8.70	9.00	14.00	21.00
Tegal	1269	1150	119	0	76	6.91	6.87	2.00	5.00	9.00
Pekalongan	1269	870	399	0	53	2.16	3.62	0.00	1.00	3.00

tivariate time series, some destination stations contained missing entries on certain dates because no departures to those destinations occurred. These missing entries represent structural zeros, reflecting a genuine absence of passenger activity rather than missing or unobserved data. Therefore, the absence of records for a given destination and date unambiguously indicates zero departures, and such entries were appropriately imputed as zeros, consistent with the interpretation of structural zero demand. The number of missing entries prior to transformation is summarized in Table 3. In addition, to further visualize the distribution of daily passenger volumes across all destination stations, a boxplot is presented in Figure 3. This boxplot clearly highlights the median, interquartile range, and potential outliers, providing a clear view of the variability and nonlinearity in passenger flow, which reinforces the need for advanced forecasting methods.

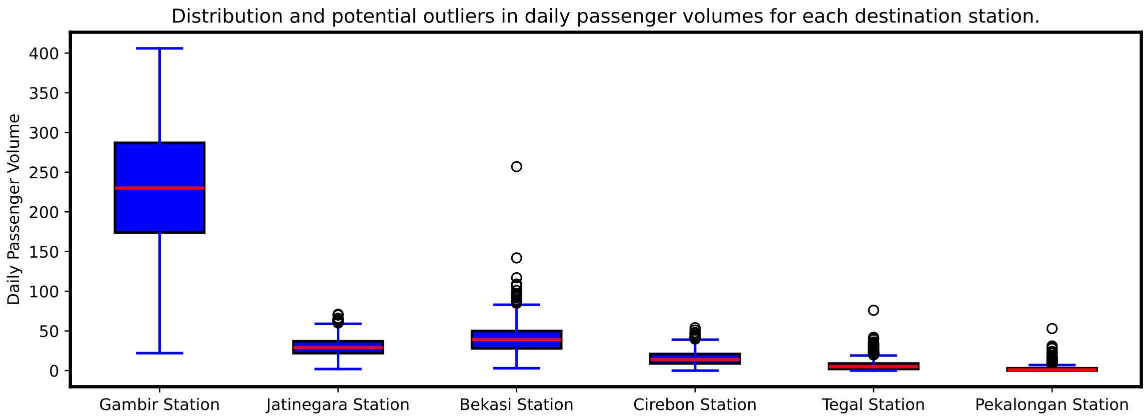


Figure 3. Boxplot of daily passenger volumes for all destination stations.

3.2. Seasonal-trend decomposition using loess

Seasonal-Trend Decomposition using Loess (STL) is a robust method for decomposing time series based on nonparametric local regression. Let X_t denote the observed series at time t, where $t = 1, \dots, n$ and n represents the number of observations. The series can be expressed as

$$X_t = T_t + S_t + R_t, \tag{1}$$

where T_t, S_t, and R_t represent the trend, seasonal, and residual components, respectively. The decomposition is performed iteratively. At iteration k, where k denotes the iteration index, the series is first detrended as $X'_t = X_t - T_t^{(k)}$, where X'_t denotes the detrended series and $T_t^{(k)}$ represents the trend estimate at iteration k. The trend component is initialized as $T_t^{(0)} = 0$. The detrended series is partitioned into seasonal subseries and smoothed using locally estimated scatterplot smoothing (LOESS) to produce an intermediate seasonal component $C_t^{(k+1)}$. A low-pass filter is then applied to remove low-energy leakage, yielding $L_t^{(k+1)}$, and the seasonal component is updated as $S_t^{(k+1)} = C_t^{(k+1)} - L_t^{(k+1)}$. The series is subsequently deseasonalized and smoothed again via LOESS to update the trend component $T_t^{(k+1)}$. This process is repeated until convergence, resulting in the final components $T_t = T_t^{(k+1)}$, $S_t = S_t^{(k+1)}$, and $R_t = X_t - T_t - S_t$. Because of its flexibility, STL is widely employed in time series analysis [27].

3.3. Empirical mode decomposition

Empirical Mode Decomposition (EMD) is a data-driven approach for decomposing nonlinear and non-stationary signals into a finite set of Intrinsic Mode Functions (IMFs), which represent localized oscillatory modes [23]. The decomposition is performed iteratively through a sifting process, where the signal is separated into upper and lower envelopes using cubic spline interpolation. The mean envelope is defined as $x_m(t) = \frac{x_u(t)+x_l(t)}{2}$ and used to extract a detail component $h(t) = x(t) - x_m(t)$ [31]. This process is repeated until an IMF is obtained. The signal is then progressively decomposed, yielding

$$x(t) = \sum_{i=1}^{n} \text{IMF}_i(t) + r_n(t), \tag{2}$$

where $x(t)$ denotes the original signal, $\text{IMF}_i(t)$ is the i-th IMF, $r_n(t)$ is the residual trend, n is the number of IMFs, and t represents the time index. To characterize each IMF, a feature vector $\mathbf{x}_i = [f_i^{\text{dom}}, E_i]$ is defined, where $f_i^{\text{dom}} = \arg\max_f |\mathcal{F}\{\text{IMF}_i(t)\}|$ denotes the dominant frequency with $\mathcal{F}(\cdot)$ as the Fourier transform, and $E_i = \frac{1}{N}\sum_{t=1}^{N} \text{IMF}_i^2(t)$ represents the energy of the IMF, with N being the signal length. These features are subsequently used in the clustering stage.

3.4. Sample entropy

Sample Entropy (SampEn) is employed to quantify the complexity of each Intrinsic Mode Function (IMF) obtained from EMD. Given a signal $\text{IMF}i(t)t = 1^T$, where i denotes the IMF index, t represents the time index, and T is the signal length, SampEn is defined as

$$\text{SampEn}(m, r) = -\ln\left(\frac{A}{B}\right), \tag{3}$$

where m is the embedding dimension and r is the tolerance threshold used to determine similarity between vectors. Here, A and B denote the number of matched vector pairs of length $m + 1$ and m, respectively, whose distances are within the tolerance threshold r. Lower SampEn values indicate more regular behavior, whereas higher values reflect greater complexity. To provide an overview of the IMF complexity characteristics, Figure 4 presents the SampEn values of $\text{IMF}_1 - \text{IMF}_9$ across all stations using the complete residual series. A decreasing trend in SampEn values can generally be observed from IMF_1 to IMF_9, indicating a transition from more irregular to more regular component behavior.

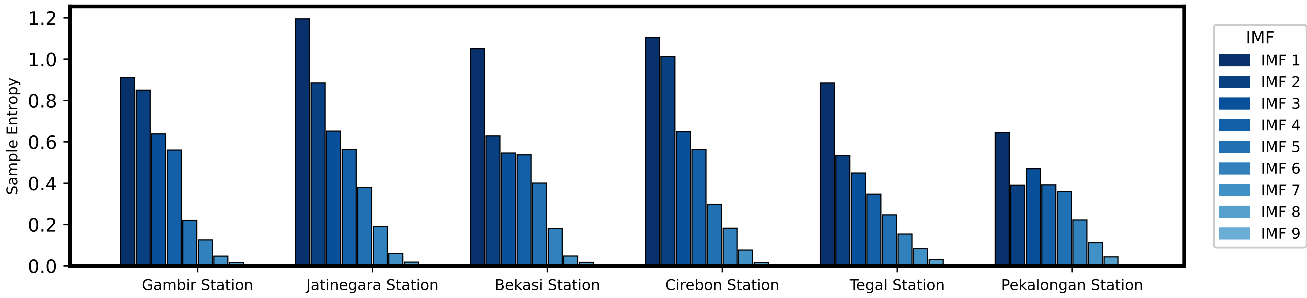


Figure 4. Sample entropy values of $\text{IMF}_1 - \text{IMF}_9$ across different stations.

3.5. Fuzzy C-means clustering

Fuzzy C-Means (FCM) extends k-means clustering by incorporating fuzzy membership, allowing each data point to belong to multiple clusters with varying degrees. Given a dataset $X = \{x_1, \ldots, x_n\}$ containing n data points and c clusters, the partition matrix $U = [u_{ij}]$ and cluster centroids v_i are obtained by minimizing

$$J = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} \|x_j - v_i\|^2, \tag{4}$$

subject to $\sum_{i=1}^{c} u_{ij} = 1$, where u_{ij} denotes the membership degree of data point x_j in cluster i, and $m > 1$ controls the fuzziness of the clustering process. Here, $\|x_j - v_i\|^2$ represents the squared Euclidean distance between data point x_j and centroid v_i. The optimization is solved iteratively using Lagrange multipliers until convergence, yielding the final centroids and membership matrix. For consistency across experimental settings, the resulting clusters are reordered according to the average energy of the IMFs assigned to each cluster. The average cluster energy is defined as

$$\bar{E}_c = \frac{1}{|\mathcal{C}_c|} \sum_{i \in \mathcal{C}_c} E_i, \tag{5}$$

where $\mathcal{C}_c$ denotes the set of IMFs assigned to cluster c, $|\mathcal{C}_c|$ represents the number of IMFs in that cluster, and E_i denotes the energy of IMF i. The IMFs within each cluster are then aggregated to form reconstructed components:

$$R_c(t) = \sum_{i \in \mathcal{C}_c} \text{IMF}_i(t), \tag{6}$$

where $R_c(t)$ denotes the reconstructed component associated with cluster c at time step t. The overall residual reconstruction is expressed as $r(t) = R_1(t) + R_2(t) + R_3(t)$, where $r(t)$ denotes the reconstructed residual signal. To maintain consistent interpretation across experiments, Cluster 1, Cluster 2, and Cluster 3 correspond to relatively low-, medium-, and high-energy groups, respectively. This relabeling is used solely for interpretability and does not influence the clustering process itself.

3.6. TPE-based hyperparameter optimization

Hyperparameter optimization aims to identify the configuration that minimizes prediction error. Let $\boldsymbol{\theta} \in \Theta$ denote the hyperparameter vector within the search space Θ, with the optimization objective defined as

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\Theta} \mathcal{L}(\boldsymbol{\theta}), \tag{7}$$

where $\boldsymbol{\theta}^*$ denotes the optimal hyperparameter configuration and $\mathcal{L}(\boldsymbol{\theta})$ represents the validation loss function. This optimization is conducted using Optuna, which formulates the task as a black-box optimization problem and employs the Tree-structured Parzen Estimator (TPE), where the conditional probability density of hyperparameters is modeled as

$$p(\boldsymbol{\theta} \mid \mathcal{L}) = \begin{cases} l(\boldsymbol{\theta}), & \mathcal{L} < \mathcal{L}^*, \\ g(\boldsymbol{\theta}), & \mathcal{L} \geq \mathcal{L}^*, \end{cases} \tag{8}$$

where $\mathcal{L}^*$ is a quantile threshold separating low-loss and high-loss regions, with $l(\boldsymbol{\theta}) = p(\boldsymbol{\theta} \mid \mathcal{L} < \mathcal{L}^*)$ and $g(\boldsymbol{\theta}) = p(\boldsymbol{\theta} \mid \mathcal{L} \geq \mathcal{L}^*)$. The TPE selects new candidates by maximizing the ratio $l(\boldsymbol{\theta})/g(\boldsymbol{\theta})$, which is consistent with the expected improvement criterion $\mathrm{EI}(\boldsymbol{\theta}) = \mathbb{E}[\max(\mathcal{L}^* - \mathcal{L}(\boldsymbol{\theta}), 0)]$, where $\mathbb{E}[\cdot]$ denotes the expectation operator. This strategy balances exploration and exploitation during the optimization process.

3.7. Statistical Models

To provide a comprehensive benchmark, several classical statistical models are employed as baseline methods, including Seasonal Naive (SN), Exponential Smoothing (ETS), Seasonal Autoregressive Integrated Moving Average (SARIMA), and Prophet. These models are widely used in time series forecasting due to their strong interpretability and well-established theoretical foundations.

3.7.1. Seasonal naive

The Seasonal Naive (SN) model assumes that future observations repeat past seasonal patterns. The forecast for time $t + h$ is given as

$$\hat{y}_{t+h} = y_{t+h-m}, \tag{9}$$

where $\hat{y}_{t+h}$ denotes the forecast value at forecasting horizon h, and y_{t+h-m} represents the observed value from the previous seasonal cycle. Here, t denotes the current time index, h represents the forecasting horizon, and m denotes the seasonal period. Despite its simplicity, SN serves as a strong baseline in seasonal time series forecasting.

3.7.2. Exponential smoothing

Exponential smoothing (ETS) is a widely used statistical forecasting approach that models a time series through recursively updated level, trend, and seasonal components [11, 12]. In this study, the Holt–Winters exponential smoothing implementation provided by the `statsmodels` library is employed as one of the statistical benchmark models.

To provide a fair comparison while maintaining computational efficiency, four candidate model configurations are evaluated, consisting of additive or multiplicative seasonal components, each with or without an additive trend. An internal validation split (80% training and 20% validation) is constructed exclusively from the training data to evaluate these candidate configurations. The configuration yielding the lowest validation RMSE is selected and subsequently used throughout the rolling-origin evaluation. During model fitting, the smoothing parameters are automatically estimated by maximum likelihood.

For completeness, the additive Holt–Winters formulation, which represents the most commonly used ETS specification, is summarized below [11]. The one-step-ahead forecast is given by

$$\hat{y}_{t+1|t} = \ell_t + b_t + s_{t-m+1}, \tag{10}$$

where $\hat{y}_{t+1|t}$ denotes the one-step-ahead forecast generated at time t, ℓ_t is the estimated level component, b_t is the estimated trend component, s_{t-m+1} denotes the seasonal component corresponding to the next seasonal position, and m is the seasonal period.

The state components are recursively updated as

$$\ell_t = \alpha\left(y_t - s_{t-m}\right) + (1-\alpha)\left(\ell_{t-1} + b_{t-1}\right), \tag{11}$$

$$b_t = \beta\left(\ell_t - \ell_{t-1}\right) + (1-\beta)b_{t-1}, \tag{12}$$

$$s_t = \gamma\left(y_t - \ell_t\right) + (1-\gamma)s_{t-m}, \tag{13}$$

where y_t denotes the observed value at time step t, while α, β, and γ are the smoothing parameters associated with the level, trend, and seasonal components, respectively, with values ranging from 0 to 1. During model selection, the structural configuration (trend and seasonal components) is determined using the validation data, whereas the model parameters are estimated automatically during fitting according to the Holt–Winters exponential smoothing framework [11, 12].

3.7.3. Prophet

Prophet is an additive forecasting model that decomposes a time series into trend, seasonal, and holiday components. The model is designed to capture nonlinear trends and multiple seasonal patterns while maintaining computational efficiency and interpretability.

$$y_t = g(t) + s(t) + h(t) + \varepsilon_t, \tag{14}$$

where y_t denotes the observed value at time t, $g(t)$ represents the long-term trend component, $s(t)$ captures periodic seasonal patterns, $h(t)$ models the effects of holidays or special events, and ε_t denotes the random error term [28]. The trend component $g(t)$ can be modeled using a piecewise linear growth function

$$g(t) = (k + \mathbf{a}(t)^\top \boldsymbol{\delta})t + (m + \mathbf{a}(t)^\top \boldsymbol{\gamma}), \tag{15}$$

where k is the growth rate, m is the offset parameter, $\mathbf{a}(t)$ is a vector indicating changepoints, and $\boldsymbol{\delta}$ and $\boldsymbol{\gamma}$ represent the corresponding rate and offset adjustments. The seasonal component is modeled using a Fourier series

$$s(t) = \sum_{n=1}^{N} \left(a_n \cos\left(\frac{2\pi n t}{P}\right) + b_n \sin\left(\frac{2\pi n t}{P}\right) \right), \tag{16}$$

where P denotes the seasonal period, N is the order of the Fourier series, and a_n and b_n are coefficients estimated from the data.

3.7.4. SARIMA

The Seasonal Autoregressive Integrated Moving Average (SARIMA) model extends ARIMA to incorporate seasonality. It is denoted as SARIMA$(p, d, q)(P, D, Q)_m$ and defined by

$$\Phi_p(B)\Phi_P(B^m)(1-B)^d(1-B^m)^D y_t = \Theta_q(B)\Theta_Q(B^m)\varepsilon_t, \tag{17}$$

where B denotes the backshift operator, such that $By_t = y_{t-1}$, and y_t represents the observed value at time step t. Here, p, d, and q denote the non-seasonal autoregressive, differencing, and moving average orders, respectively, while P, D, and Q represent the corresponding seasonal orders. Furthermore, m denotes the seasonal period, and ε_t represents the white noise error term. The polynomials $\Phi_p(B)$ and $\Phi_P(B^m)$ correspond to the non-seasonal and seasonal autoregressive components, whereas $\Theta_q(B)$ and $\Theta_Q(B^m)$ denote the non-seasonal and seasonal moving average components, respectively.

3.8. Deep learning architecture

A variety of deep learning architectures have been developed for sequential data modeling and forecasting, each offering distinct advantages and modeling capabilities.

3.8.1. Recurrent neural network

Recurrent Neural Networks (RNN) are designed to model sequential data by capturing temporal dependencies through recurrent connections. At each time step t, the hidden state $h_t \in \mathbb{R}^M$ is computed as

$$h_t = \sigma_h(W_i x_t + W_h h_{t-1} + b_h), \tag{18}$$

where x_t denotes the input vector at time step t, h_{t-1} represents the hidden state from the previous time step, and M denotes the number of hidden units. Furthermore, σ_h denotes a nonlinear activation function. The network output is then given by $y_t = f(W_o h_t + b_o)$, where f is an activation function. Here, W_i, W_h, and W_o are the input, recurrent, and output weight matrices, respectively, while b_h and b_o denote bias vectors. All parameters are learnable and optimized during training.

3.8.2. Long short-term memory

Long Short-Term Memory (LSTM) networks extend recurrent architectures by incorporating gated memory cells that enable effective modeling of long-term dependencies. The core component of LSTM is the cell state c_t, which is regulated by input, forget, and output gates. At each time step, the gating mechanisms are computed as $i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$, $f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$, and $o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$, while the candidate cell state is given by $\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c)$, and the cell state is updated as

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \tag{19}$$

and the hidden state is obtained as $h_t = o_t \odot \tanh(c_t)$, where x_t denotes the input vector at time step t, h_{t-1} and c_{t-1} represent the hidden state and cell state from the previous time step, respectively, and h_t denotes the hidden state at the current time step. Here, W_i, W_f, W_o, and W_c are weight matrices, while b_i, b_f, b_o, and b_c are bias vectors. The notation $[h_{t-1}, x_t]$ denotes the concatenation of the previous hidden state and current input vector. Furthermore, σ and $\tanh$ denote activation functions, and $\odot$ represents element-wise multiplication.

3.8.3. Bidirectional LSTM

BiLSTM extends the conventional LSTM by processing sequences in both forward and backward directions, enabling richer temporal representation. It consists of two parallel recurrent layers operating from $t = 1, \ldots, T$ (forward) and $t = T, \ldots, 1$ (backward), where t denotes the time step and T represents the sequence length. The hidden states from both directions are combined and mapped to the output as

$$y_t = W_{hy}^b h_t^b + W_{hy}^f h_t^f + b_y, \tag{20}$$

where y_t denotes the output at time step t, h_t^b and h_t^f represent the backward and forward hidden states, respectively, and the superscripts b and f indicate backward and forward directions. Here, W_{hy}^b and W_{hy}^f are weight matrices associated with the backward and forward hidden states, while b_y denotes the bias vector. All parameters are learnable during training [9].

3.8.4. Gated recurrent unit

Gated Recurrent Unit (GRU) is a variant of Recurrent Neural Networks (RNN) designed to capture long-term dependencies in sequential data. Unlike LSTM, GRU combines the input and forget gates into a single *update* gate, resulting in a simpler architecture while maintaining the ability to model temporal dependencies. The gating mechanism is defined as follows: the update gate $z_t = \sigma(x_t W^z + h_{t-1} U^z + b_z)$ controls the information flow from the previous state, while the reset gate $r_t = \sigma(x_t W^r + h_{t-1} U^r + b_r)$ determines how much past information is retained. The candidate hidden state is computed as $\tilde{h}_t = \tanh(r_t \odot (h_{t-1} U) + x_t W + b)$, and the final hidden state is updated as

$$h_t = (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{t-1} \tag{21}$$

where x_t denotes the input vector at time step t, h_{t-1} and h_t represent the previous and current hidden states, respectively, and σ and $\tanh$ denote activation functions. Here, W^z, W^r, and W denote input weight matrices, while U^z, U^r, and U are recurrent weight matrices. Furthermore, b_z, b_r, and b are bias terms, and $\odot$ denotes element-wise multiplication.

3.8.5. Convolutional neural network

The fundamental operation of a Convolutional Neural Network (CNN) is conducted within the convolutional layer, where a kernel (or filter) slides over the input data to execute the convolution process. This operation can be mathematically expressed as

$$S(i) = \sum_{m=0}^{M-1} X(i+m) K(m) \tag{22}$$

where $S(i)$ denotes the resulting feature map at position i, X represents the input data, and $X(i+m)$ denotes the input value at position $i+m$. Furthermore, K is the convolutional kernel, $K(m)$ represents the kernel value at index m, and M indicates the kernel size. Here, i denotes the current position of the convolution operation, while m represents the kernel index used during the summation process. CNNs employ feature extractors in the form of convolutional filters or kernels.

3.9. Exponential weighted moving average

EWMA is a widely applied technique for smoothing time series data. It assigns greater importance to recent observations while gradually decreasing the influence of older values, allowing it to effectively track underlying trends as well as subtle variations in the series. The formula is calculated as follows:

$$E_t = \lambda\mu_0 + (1-\lambda)E_{t-1} \tag{23}$$

where E_t denotes the EWMA value at time step t, and E_{t-1} represents the EWMA value from the previous time step. Furthermore, μ_0 denotes the mean value of the validation sequence, while $\lambda \in (0, 1)$ represents the weighting factor that controls the influence of past residuals on the current EWMA value. Here, larger values of λ assign greater weight to recent observations, whereas smaller values place more emphasis on historical information.

3.10. Evaluation metrics

Model performance is evaluated using several standard error metrics. Let $\{y_t\}_{t=1}^{T}$ denote the actual values and $\{\hat{y}_t\}_{t=1}^{T}$ denote the predicted values.

Table 4. Error metrics for model evaluation.

MAE	RMSE	MSE	sMAPE	R^2
$\frac{1}{T}\sum_{t=1}^{T}\lvert y_t-\hat{y}_t\rvert$	$\sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t-\hat{y}_t)^2}$	$\frac{1}{T}\sum_{t=1}^{T}(y_t-\hat{y}_t)^2$	$\frac{100}{T}\sum_{t=1}^{T}\frac{\lvert y_t-\hat{y}_t\rvert}{(\lvert y_t\rvert+\lvert\hat{y}_t\rvert)/2}$	$1-\frac{\sum_{t=1}^{T}(y_t-\hat{y}_t)^2}{\sum_{t=1}^{T}(y_t-\bar{y})^2}$

where t denotes the time index, T represents the total number of observations, and $\bar{y} = \frac{1}{T}\sum_{t=1}^{T} y_t$ denotes the mean of the observed values.

3.11. Model and Experimental Configuration

All deep learning models in this study share a unified architecture, consisting of an input layer with sliding window representation, a neural layer (RNN, LSTM, GRU, BiLSTM, or Conv1D), a dropout layer for regularization, and a dense output layer for prediction. The architecture of the model is illustrated in Figure 5. Based on this architecture, the corresponding hyperparameter configuration used for training the deep learning models is summarized in Table 5. To ensure consistency and a fair comparison, all deep learning models are optimized using the Tree-structured Parzen Estimator (TPE)-based Optuna framework under a unified hyperparameter search space, with the optimized configurations reported in Table 19. The same search space is applied uniformly across all deep learning models and components, including both the original and decomposed time series. Each model was optimized using its standard tuning strategy to ensure fair comparison within its methodological class.

The experimental setup is designed to ensure robust evaluation of forecasting performance. A multi-step forecasting strategy is employed using a rolling-origin evaluation scheme. The detailed configuration of the experimental and reproducibility settings is summarized in Table 6. The forecasting horizon of

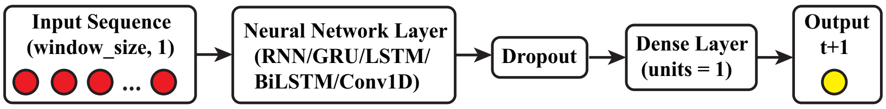


Figure 5. General architecture of the deep learning models used in this study.

Table 5. Hyperparameter configuration of deep learning models

Parameter	Value	Type	Description
Epochs	50	Fixed	Number of training iterations over the dataset
Batch Size	16	Fixed	Number of samples processed before updating model weights
Optimizer	Adam	Fixed	Optimization algorithm for updating neural network weights
Loss Function	MSE	Fixed	Objective function used to minimize prediction error
Validation Split	0.2	Fixed	Proportion of training data used for validation
Shuffle	False	Fixed	Maintains temporal order of time series data during training
Units	[32, 64, 128]	Tuned	Number of hidden units in the neural network layer
Learning Rate	[0.001, 0.0001, 0.0005]	Tuned	Controls the step size of weight updates during optimization
Dropout Rate	[0.1, 0.2, 0.3]	Tuned	Fraction of neurons randomly deactivated to prevent overfitting
Activation Function	ReLU, Tanh	Tuned	Nonlinear transformation applied to hidden layer outputs

254 is selected to reflect medium- to long-term prediction scenarios. The sliding window size is set to 14 to capture two consecutive weekly patterns in the daily time series, while the STL seasonal period is set to 7 to model weekly seasonality. The number of clusters is set to three to represent low-, mid-, and high-energy components, capturing multi-scale temporal dynamics. Sample entropy is employed as the clustering feature to characterize the complexity and irregularity of each intrinsic mode function, enabling more structured and meaningful component-wise modeling.

Table 6. Experimental and reproducibility settings

Parameter	Value	Description
Input Shape	(14, 1)	Defines the input sequence length and feature dimension
Sliding Window	14	Number of lagged observations used as input
Forecasting Strategy	multi-step forecasting	Predictions are generated step-by-step and recursively extended across the forecasting horizon
Evaluation Method	Rolling origin	Model is evaluated using progressive time-based splits
Training Size	800	Number of observations used in the initial training window
Forecast Horizon	254	Number of observations predicted in each evaluation window
Rolling Step Size	20	Forward shift of the sliding window
STL Seasonal Period	7	Captures weekly seasonality in daily time series data
Sample Entropy (m, r)	$m = 2, r = 0.2\sigma$	Controls similarity threshold and tolerance level
Number of Clusters	3	Represents low-, mid-, and high-energy components
Optuna Trials	50	Number of hyperparameter optimization iterations
Validation Split	0.2	Fraction of training data used for validation during optimization
Random Seed	42	Ensures reproducibility of experimental results
Early Stopping	Patience = 5	Prevents overfitting by stopping training early

4. Results and discussion

This section presents the experimental results and focuses on evaluating the forecasting performance of each model across different signal components derived from the decomposed railway passenger time series representing origin-destination passenger flows between Semarang Station and six major destination stations: Gambir, Jatinegara, Bekasi, Cirebon, Tegal, and Pekalongan. Most forecasting models were optimized using Optuna with a Tree-structured Parzen Estimator (TPE) based hyperparameter optimization strategy, while the high-energy residual component was handled using EWMA smoothing. Model performance was evaluated using RMSE, MAPE, MAE, and R^2 metrics. All experiments were conducted using Python (versions 3.12 and 3.13) with TensorFlow and scikit-learn libraries on two machines with different hardware configurations, equipped with an Intel Core Ultra 7 155H processor and an AMD Ryzen 7 5800U processor, both with 16 GB RAM. The experiments were distributed across the two machines due to computational constraints. Despite minor differences in software and hardware environments, all experiments follow the same configuration and evaluation protocol, ensuring consistency and reproducibility of the results. On average, the full experimental pipeline requires approximately 75 minutes per station, while hyperparameter optimization using Optuna takes approximately 245 minutes per station.

To reduce the risk of information leakage, all preprocessing and model development procedures were performed within each rolling-origin evaluation fold after temporal separation of the training and testing partitions. Scaling, clustering, hyperparameter optimization, and model fitting were conducted exclusively using the training data available up to the corresponding forecast origin. The decomposition stage was performed independently within each evaluation fold. No information from the testing partition was used during scaling, clustering model construction, hyperparameter optimization, or forecasting model training.

For each rolling-origin fold, the training subset was first separated from the testing subset according to the forecast origin. STL decomposition was then applied independently to the training and testing partitions, followed by EMD decomposition of the corresponding residual components. IMF features were extracted, after which the scaler was fitted using only the training IMFs and subsequently applied to the testing IMFs. Fuzzy C-Means clustering was learned exclusively from the scaled training IMFs, and the learned clustering structure was then used to assign the testing IMFs through the FCM prediction procedure. Hyperparameter optimization and forecasting model training were performed exclusively using the training data, after which forecasts were generated for the testing horizon.

Since EMD is adaptive and data-dependent, the number of IMFs generated from the training and testing residual components may differ. To ensure dimensional consistency in the experimental framework, only the common number of IMFs shared by both decompositions was retained for subsequent forecasting experiments. While this procedure does not influence feature extraction, clustering model construction, hyperparameter optimization, or forecasting model training, it relies on decomposition characteristics derived from the testing subset and is therefore acknowledged as a methodological limitation of the current study.

Overall, the adopted protocol prevents conventional information transfer between training and testing partitions during preprocessing, clustering, hyperparameter optimization, and model fitting. However, because STL and EMD are retrospective decomposition methods, their application within the evaluation framework may not fully satisfy a strictly causal forecasting setting. This limitation should be considered when interpreting the reported results. To investigate the impact of feature representation, an ablation study is conducted by comparing two clustering strategies: (1) without Sample Entropy (SampEn) and (2) with SampEn-based features. Clustering is performed based on the selected feature representations, allowing the contribution of SampEn to be explicitly evaluated. The clustered IMFs are subsequently remapped to the original time dimension, where each time step is assigned according to its corresponding cluster membership. This enables each cluster-specific component to be modeled separately in the forecasting stage. While weighted reconstruction based on FCM membership degrees (soft clustering) can be employed, where each IMF contributes to multiple clusters proportionally to its membership values, this approach is not adopted in this study. Instead, a hard assignment is used for reconstruction to maintain clearer separation between components and to reduce additional modeling complexity. To maintain consistent labeling across experiments, the cluster indices are reordered based on the average energy of the IMFs. It should be noted that this reordering is used solely for interpretability and does not influence the clustering process itself. In this context, Cluster 1, Cluster 2, and Cluster 3 represent clusters reordered according to their average energy levels for interpretation purposes. These clusters should not be interpreted as purely frequency-based components. The clustering results vary depending on the feature representation used, highlighting the influence of Sample Entropy in capturing distinct residual patterns.

Table 7. Rolling origin performance evaluation of trend component.

		Training Data					Testing Data				
Station	**Model**	**MAE**	**RMSE**	**MSE**	**sMAPE**	R^2	**MAE**	**RMSE**	**MSE**	**sMAPE**	R^2
Gambir	LSTM	4.94	6.62	45.38	2.35	0.98	4.74	6.67	47.13	2.00	0.95
	GRU	3.99	4.88	25.15	1.98	0.99	3.95	5.53	32.54	1.68	0.96
Jatinegara	LSTM	0.58	0.72	0.53	2.08	0.99	0.71	1.16	1.54	2.12	0.95
	GRU	0.52	0.66	0.44	1.87	0.99	0.66	1.10	1.36	2.01	0.96
Bekasi	LSTM	1.22	1.55	2.44	3.12	0.98	1.51	2.42	7.09	3.25	0.95
	GRU	0.81	0.97	1.00	2.13	0.99	1.05	2.11	7.21	2.21	0.95
Cirebon	LSTM	0.46	0.56	0.32	3.85	0.99	0.53	0.78	0.72	2.62	0.95
	GRU	0.44	0.54	0.30	3.70	0.99	0.50	0.77	0.67	2.52	0.95
Tegal	LSTM	0.23	0.30	0.09	6.31	0.99	0.33	0.53	0.30	3.31	0.97
	GRU	0.19	0.23	0.06	6.45	0.99	0.31	0.58	0.37	3.03	0.97
Pekalongan	LSTM	0.13	0.19	0.04	9.57	0.98	0.15	0.26	0.08	7.24	0.97
	GRU	0.10	0.14	0.02	9.36	0.99	0.13	0.25	0.08	6.16	0.96

With the optimized hyperparameters, the deep learning models are applied to each decomposed component following the proposed component-wise modeling strategy. For the residual components, different strategies are applied, where low- and mid-energy components are modeled using deep learning approaches, while high-energy components are handled using EWMA smoothing due to their irregular behavior. All reported performance metrics are averaged across rolling origin evaluation folds. To quantitatively evaluate the forecasting performance, Table 7 summarizes the performance metrics for the trend component across all destination stations for both training and testing datasets. Overall, both models are capable of capturing long-term temporal dependencies in the trend component. GRU generally achieves lower error values across most stations, for example at Bekasi where the RMSE decreases from 2.42 to 2.11. However, LSTM slightly outperforms GRU in certain cases, such as at Gambir, indicating that GRU provides better overall generalization while LSTM remains effective for modeling smoother trends. Following the analysis of the trend component, Table 8 presents the performance comparison of RNN, CNN, and BiLSTM models for the seasonal component across six destination stations. The results indicate that RNN achieves the best performance for most stations, while LSTM outperforms other models at the Pekalongan station. For instance, at the Gambir station, RNN achieves an MAE of 8.39, an RMSE

Table 8. Rolling origin performance evaluation of seasonal component.

		Training Data					Testing Data				
Station	**Model**	**MAE**	**RMSE**	**MSE**	**sMAPE**	R^2	**MAE**	**RMSE**	**MSE**	**sMAPE**	R^2
Gambir	RNN	6.44	8.28	68.93	39.15	0.96	8.39	13.71	196.90	41.81	0.91
	CNN	6.03	7.77	60.49	37.27	0.97	8.42	13.49	191.45	41.62	0.91
	BiLSTM	9.29	12.19	149.29	51.26	0.92	12.43	18.68	369.95	52.67	0.82
Jatinegara	RNN	1.16	1.48	2.20	51.11	0.92	1.62	2.64	7.31	57.63	0.77
	CNN	1.14	1.44	2.09	50.93	0.92	1.70	2.75	8.03	59.64	0.75
	BiLSTM	1.82	2.31	5.36	69.36	0.81	2.45	3.58	13.53	78.83	0.58
Bekasi	RNN	1.59	2.15	4.64	57.68	0.91	2.11	3.66	14.13	57.56	0.74
	CNN	1.60	2.19	4.80	57.44	0.91	2.21	3.87	16.35	57.52	0.70
	BiLSTM	2.63	3.53	12.57	82.34	0.76	3.26	5.15	29.74	81.94	0.46
Cirebon	RNN	0.84	1.10	1.22	56.55	0.93	1.23	1.84	3.44	59.64	0.83
	CNN	0.87	1.12	1.27	58.04	0.92	1.31	1.90	3.67	61.91	0.82
	BiLSTM	1.39	1.79	3.24	76.86	0.81	1.89	2.56	6.67	84.96	0.68
Tegal	RNN	0.59	0.78	0.61	53.84	0.96	0.92	1.41	2.02	50.49	0.90
	CNN	0.59	0.79	0.62	54.37	0.96	0.97	1.46	2.16	53.97	0.89
	BiLSTM	0.98	1.31	1.74	73.15	0.88	1.45	1.99	4.00	73.98	0.80
Pekalongan	RNN	0.35	0.54	0.30	61.38	0.91	0.51	1.03	1.55	56.13	0.59
	CNN	0.42	0.66	0.44	68.96	0.87	0.64	1.08	1.37	69.89	0.63
	BiLSTM	0.67	1.04	1.13	92.15	0.67	0.85	1.30	1.87	87.00	0.48

of 13.71, and an R^2 value of 0.91, demonstrating its strong capability in capturing seasonal patterns.

These findings indicate that the trend and seasonal components are relatively easier to model due to their smoother dynamics. In contrast, the residual component exhibits highly irregular and complex behavior, requiring more advanced modeling strategies. Given the additive nature of EMD, IMFs are reconstructed within each cluster through summation, which provides a straightforward and physically interpretable representation without introducing overlap between components. To further investigate this behavior, Table 9 presents the performance evaluation of the EMD-based residual IMFs. Overall, the results show that low-order IMFs (e.g., IMF 1 and 2) exhibit significantly higher error values and lower R^2, indicating their highly irregular and noise-like characteristics. In contrast, higher-order IMFs achieve substantially lower errors and near-perfect R^2 values, suggesting that they are smoother and easier to model. This suggests that IMFs associated with slower oscillatory behavior tend to be more predictable. Negative or low R^2 values observed in lower-order IMFs are expected due to their highly irregular and noise-dominated nature, making them inherently more difficult to model accurately.

Building upon this observation, Tables 10 and 11 present the performance evaluation of clustered residual components without and with SampEn-based features, respectively. Overall, the results indicate that clustering helps reduce the complexity of the residual components, as reflected by improved error metrics compared to individual IMFs. Furthermore, the inclusion of sample entropy features generally leads to better performance in several cases. For example, at Gambir (low cluster), the RMSE decreases from 7.75 to 5.24, while at Bekasi, improvements are also observed across MAE and RMSE in both clusters. However, the improvement is not uniform across all stations. In certain cases, such as Cirebon and Tegal, the performance slightly degrades or shows negative R^2, indicating that highly irregular residual patterns remain challenging to model. This suggests that while SampEn enhances feature representation by capturing signal complexity, its effectiveness depends on the underlying characteristics of the residual components. Table 12 presents the performance of reconstructed residual components using different

Table 9. Rolling origin performance evaluation of EMD-based residual IMFs using RNN

Station	IMF	Training Data					Testing Data				
		MAE	**RMSE**	**MSE**	**sMAPE**	R^2	**MAE**	**RMSE**	**MSE**	**sMAPE**	R^2
Gambir	IMF 1	8.53	13.53	183.45	67.80	0.69	19.34	26.25	696.01	113.83	0.09
	IMF 2	2.54	3.58	12.87	49.80	0.93	4.35	6.03	36.66	65.89	0.80
	IMF 3	0.63	0.87	0.76	21.21	0.99	0.54	1.08	1.53	25.00	0.97
	IMF 4	0.18	0.27	0.08	12.74	1.00	0.17	0.39	0.20	18.15	0.97
	IMF 5	0.06	0.09	0.01	9.07	1.00	0.09	0.28	0.11	12.65	0.95
	IMF 6	0.06	0.09	0.01	8.48	1.00	0.06	0.18	0.04	11.31	0.84
	IMF 7	0.03	0.04	0.00	8.73	1.00	0.04	0.12	0.02	11.30	0.94
	IMF 8	0.01	0.01	0.00	8.62	0.99	0.05	0.10	0.01	10.57	0.97
Jatinegara	IMF 1	2.92	3.72	13.85	100.13	0.50	3.74	4.71	22.21	114.27	0.27
	IMF 2	0.77	1.05	1.12	57.61	0.86	1.05	1.49	2.27	65.77	0.79
	IMF 3	0.18	0.24	0.06	25.57	0.98	0.19	0.28	0.08	29.30	0.96
	IMF 4	0.03	0.03	0.00	11.98	1.00	0.05	0.09	0.01	17.97	0.97
	IMF 5	0.01	0.01	0.00	6.42	1.00	0.02	0.06	0.01	11.83	0.88
	IMF 6	0.00	0.01	0.00	3.69	1.00	0.01	0.02	0.00	8.75	0.95
	IMF 7	0.00	0.00	0.00	5.29	1.00	0.01	0.01	0.00	7.02	0.98
	IMF 1	3.79	6.97	49.64	85.60	0.38	5.51	7.39	56.12	118.52	0.06
Bekasi	IMF 2	1.00	1.54	2.37	54.58	0.91	1.45	1.98	4.00	65.30	0.84
	IMF 3	0.32	0.50	0.25	26.44	0.98	0.30	0.54	0.32	31.56	0.96
	IMF 4	0.07	0.10	0.01	16.84	1.00	0.09	0.19	0.04	17.30	0.97
	IMF 5	0.02	0.03	0.00	9.35	1.00	0.04	0.09	0.01	10.77	0.98
	IMF 6	0.01	0.01	0.00	6.25	1.00	0.02	0.07	0.01	11.94	0.86
	IMF 7	0.01	0.01	0.00	8.66	1.00	0.02	0.05	0.00	6.19	-1.08
Cirebon	IMF 1	1.70	2.26	5.13	80.26	0.63	3.31	4.22	17.80	111.47	0.18
	IMF 2	0.50	0.65	0.43	60.11	0.89	0.67	0.89	0.80	62.87	0.83
	IMF 3	0.10	0.14	0.02	21.52	0.99	0.11	0.23	0.06	20.91	0.97
	IMF 4	0.02	0.03	0.00	11.66	1.00	0.03	0.11	0.02	13.24	0.96
	IMF 5	0.01	0.01	0.00	5.28	1.00	0.01	0.04	0.00	6.81	0.95
	IMF 6	0.00	0.01	0.00	7.58	1.00	0.01	0.02	0.00	10.98	0.82
	IMF 7	0.00	0.00	0.00	6.41	1.00	0.01	0.03	0.00	9.89	0.86
Tegal	IMF 1	1.44	2.05	4.20	106.18	0.38	2.58	3.73	14.32	121.18	0.21
	IMF 2	0.43	0.63	0.40	61.70	0.88	0.57	0.83	0.69	61.44	0.84
	IMF 3	0.08	0.12	0.01	30.24	0.98	0.16	0.28	0.10	34.43	0.93
	IMF 4	0.02	0.03	0.00	20.53	0.99	0.06	0.10	0.01	20.81	0.97
	IMF 5	0.01	0.01	0.00	10.77	1.00	0.13	0.23	0.09	22.46	0.83
	IMF 6	0.00	0.00	0.00	11.38	1.00	0.17	0.25	0.12	32.92	0.69
	IMF 7	0.00	0.00	0.00	10.35	1.00	0.07	0.10	0.01	32.66	0.07
Pekalongan	IMF 1	0.82	1.40	1.98	89.41	0.66	1.41	1.95	3.81	119.48	0.24
	IMF 2	0.22	0.38	0.15	52.45	0.94	0.35	0.54	0.30	59.27	0.83
	IMF 3	0.07	0.12	0.01	27.72	0.98	0.08	0.18	0.04	30.79	0.93
	IMF 4	0.02	0.03	0.00	17.04	1.00	0.02	0.07	0.01	18.69	0.90
	IMF 5	0.01	0.02	0.00	11.78	1.00	0.01	0.05	0.00	14.34	0.82
	IMF 6	0.00	0.01	0.00	10.51	1.00	0.01	0.02	0.00	10.20	0.97
	IMF 7	0.00	0.01	0.00	12.82	1.00	0.01	0.02	0.00	9.08	0.79

decomposition strategies. Overall, EMD generally achieves lower error values compared to STL across most stations, indicating its effectiveness in capturing nonlinear residual patterns. Clustering further provides partial improvements, although performance remains inconsistent.

To further evaluate the effectiveness of the proposed approach, several baseline models are considered, including Seasonal Naive, SARIMA, as well as deep learning models such as LSTM and GRU trained directly on the original time series. Table 15 shows that baseline models generally exhibit higher error

Table 10. Rolling origin performance evaluation of clustered residual components without SampEn-based features.

		Training Data					Testing Data				
Station	**Cluster**	**MAE**	**RMSE**	**MSE**	**sMAPE**	R^2	**MAE**	**RMSE**	**MSE**	**sMAPE**	R^2
Gambir	Low	4.67	6.23	38.92	72.41	0.80	5.89	7.75	61.00	86.71	0.65
	Mid	15.66	20.56	456.10	129.54	0.23	16.47	21.24	485.94	132.57	0.18
Jatinegara	Low	0.91	1.18	1.40	66.48	0.83	1.03	1.34	1.80	70.04	0.79
	Mid	2.30	2.81	13.34	96.43	0.50	2.40	2.87	13.24	99.35	0.50
Bekasi	Low	2.70	3.80	14.71	106.38	0.56	3.25	4.19	18.12	118.28	0.18
	Mid	5.06	7.35	64.67	129.03	0.19	4.20	5.37	35.74	137.78	0.01
Cirebon	Low	0.76	1.01	1.04	80.13	0.75	1.36	1.69	2.89	105.99	0.42
	Mid	2.24	2.80	9.35	112.35	0.31	2.29	2.89	9.20	126.66	0.15
Tegal	Low	0.85	1.18	1.66	95.20	0.63	1.42	1.93	3.87	121.48	0.12
	Mid	0.68	0.95	1.82	77.91	0.69	1.24	1.74	3.32	131.63	-0.09
Pekalongan	Low	0.83	1.40	2.09	133.25	0.28	0.89	1.25	1.72	141.59	0.16
	Mid	1.38	2.19	5.16	145.49	0.13	1.19	1.67	3.20	149.18	0.06

Table 11. Rolling origin performance evaluation of clustered residual components with SampEn-based features.

		Training Data					Testing Data				
Station	**Cluster**	**MAE**	**RMSE**	**MSE**	**sMAPE**	R^2	**MAE**	**RMSE**	**MSE**	**sMAPE**	R^2
Gambir	Low	2.97	3.92	50.94	64.35	0.81	3.95	5.24	63.14	91.63	0.60
	Mid	12.99	17.09	365.59	117.00	0.34	13.87	18.00	396.41	125.94	0.23
Jatinegara	Low	2.00	2.44	11.72	89.71	0.57	2.02	2.42	10.52	100.50	0.53
	Mid	2.56	3.16	13.72	103.70	0.46	2.74	3.32	13.58	112.18	0.41
Bekasi	Low	1.45	2.07	14.77	87.98	0.71	1.66	2.14	8.72	109.59	0.42
	Mid	5.08	7.34	64.37	135.98	0.19	4.38	5.62	37.53	143.67	0.04
Cirebon	Low	0.60	0.76	2.19	65.41	0.79	1.03	1.30	2.68	119.46	-0.10
	Mid	2.34	2.93	9.67	119.86	0.28	2.63	3.31	11.07	131.75	-0.10
Tegal	Low	0.95	1.30	2.88	93.54	0.54	1.37	1.99	4.47	133.06	-0.61
	Mid	1.20	1.68	3.40	105.61	0.44	1.49	2.04	4.30	128.99	0.12
Pekalongan	Low	0.23	0.38	0.56	73.45	0.82	1.37	2.17	5.13	146.76	0.13
	Mid	0.39	0.56	0.66	125.10	0.15	1.21	1.69	3.13	151.41	0.05

values and less stable performance across stations, highlighting the limitations of direct modeling on raw data. Notably, SARIMA exhibits exceptionally poor performance for the Tegal station, resulting in substantially larger error values and a highly negative R^2. This behavior suggests that the linear assumptions of SARIMA were unable to adequately capture the underlying demand dynamics at this station, leading to unstable forecasting performance. Table 16 presents the performance of models using STL decomposition. The results indicate that separating trend and seasonal components improves forecasting performance, although the improvements remain limited and inconsistent. For clarity, the corresponding model configurations are summarized in Table 13. Table 17 further shows that incorporating EMD enhances performance by better capturing nonlinear and multi-scale residual patterns. However, variability across stations still persists, indicating that additional refinement is required. Building upon this, Table 18 shows that FCM clustering provides further improvements in several cases by organizing residuals into more structured components. Nevertheless, the gains remain inconsistent, suggesting that clustering alone is not sufficient to fully capture residual complexity.

To address this limitation, the complete framework incorporates sample entropy to enhance the feature representation of clustered residual components. The final passenger flow forecast is obtained by aggregating the predicted values of all components. Table 14 presents the performance of the proposed

Table 12. Rolling origin performance evaluation of reconstructed residual components.

		Training Data					Testing Data				
Station	**Residual**	**MAE**	**RMSE**	**MSE**	**sMAPE**	R^2	**MAE**	**RMSE**	**MSE**	**sMAPE**	R^2
Gambir	STL	15.65	20.57	425.68	112.13	0.47	20.67	26.84	721.65	121.10	0.22
	EMD	8.93	13.90	193.60	72.48	0.76	19.71	26.53	711.54	112.99	0.24
	Clustered (No SampEn)	16.13	21.53	472.42	112.67	0.42	19.06	24.59	613.71	123.18	0.34
	Clustered (SampEn)	15.55	20.82	449.49	109.32	0.44	18.18	23.89	578.22	118.03	0.37
Jatinegara	STL	3.44	4.41	19.49	112.87	0.42	4.29	5.32	28.39	119.22	0.30
	EMD	2.97	3.78	14.28	101.19	0.57	3.88	4.91	24.11	111.11	0.40
	Clustered (No SampEn)	3.04	3.81	16.08	103.20	0.51	3.42	4.17	18.79	106.04	0.54
	Clustered (SampEn)	4.06	4.99	24.91	129.97	0.25	4.33	5.17	26.82	128.40	0.34
Bekasi	STL	5.47	8.08	65.55	116.44	0.35	5.91	7.76	60.71	122.00	0.30
	EMD	3.87	7.03	50.31	85.02	0.50	5.58	7.54	58.47	110.88	0.34
	Clustered (No SampEn)	5.81	8.35	72.55	122.08	0.28	6.29	8.14	67.54	129.53	0.21
	Clustered (SampEn)	6.23	9.08	82.52	130.05	0.18	5.93	7.66	60.37	126.83	0.33
Cirebon	STL	2.39	3.06	9.39	107.48	0.46	3.04	3.88	15.07	105.63	0.44
	EMD	1.77	2.33	5.44	83.22	0.69	3.28	4.15	17.22	108.25	0.36
	Clustered (No SampEn)	2.47	3.13	10.13	110.25	0.41	3.40	4.12	17.02	123.76	0.36
	Clustered (SampEn)	2.74	3.46	12.00	119.39	0.31	3.53	4.29	18.54	126.50	0.31
Tegal	STL	1.65	2.39	5.73	116.46	0.38	2.48	3.70	14.24	114.27	0.36
	EMD	1.46	2.05	4.21	102.48	0.55	2.50	3.59	13.31	117.62	0.39
	Clustered (No SampEn)	1.35	1.88	3.79	99.23	0.59	2.66	3.69	14.10	122.88	0.36
	Clustered (SampEn)	1.78	2.46	6.06	118.38	0.35	2.61	3.71	14.24	126.32	0.35
Pekalongan	STL	1.13	1.88	3.55	115.27	0.42	1.37	2.03	4.14	115.49	0.30
	EMD	0.83	1.42	2.02	94.89	0.67	1.42	1.97	3.90	115.81	0.34
	Clustered (No SampEn)	1.48	2.49	6.23	144.49	0.01	1.63	2.33	5.46	148.09	0.09
	Clustered (SampEn)	1.48	2.40	5.76	138.17	0.09	1.55	2.18	4.85	138.59	0.19

Table 13. Summary of model architectures.

Code	Trend	Seasonal	Notation
M1	LSTM	RNN	LSTM-RNN
M2	LSTM	CNN	LSTM-CNN
M3	LSTM	BiLSTM	LSTM-BiLSTM
M4	GRU	RNN	GRU-RNN
M5	GRU	CNN	GRU-CNN
M6	GRU	BiLSTM	GRU-BiLSTM

hybrid model. Overall, the results show that the inclusion of sample entropy leads to mixed performance across stations. While improvements are observed in several cases, in others the performance remains similar or slightly degrades compared to models without sample entropy. This suggests that although sample entropy provides additional information about the complexity of residual components, its effectiveness depends on the underlying characteristics of the data. Nevertheless, the complete framework achieves competitive and stable performance overall, indicating that the combination of decomposition, clustering, and entropy-based feature representation remains effective for modeling complex passenger flow patterns. This overall performance is further illustrated in Figure 6 and 7, based on the last rolling origin fold. The time series plots show that the model closely follows the actual passenger demand patterns across stations, while the scatter plots indicate a strong agreement between predicted and observed values, as reflected by the concentration of points along the diagonal line.

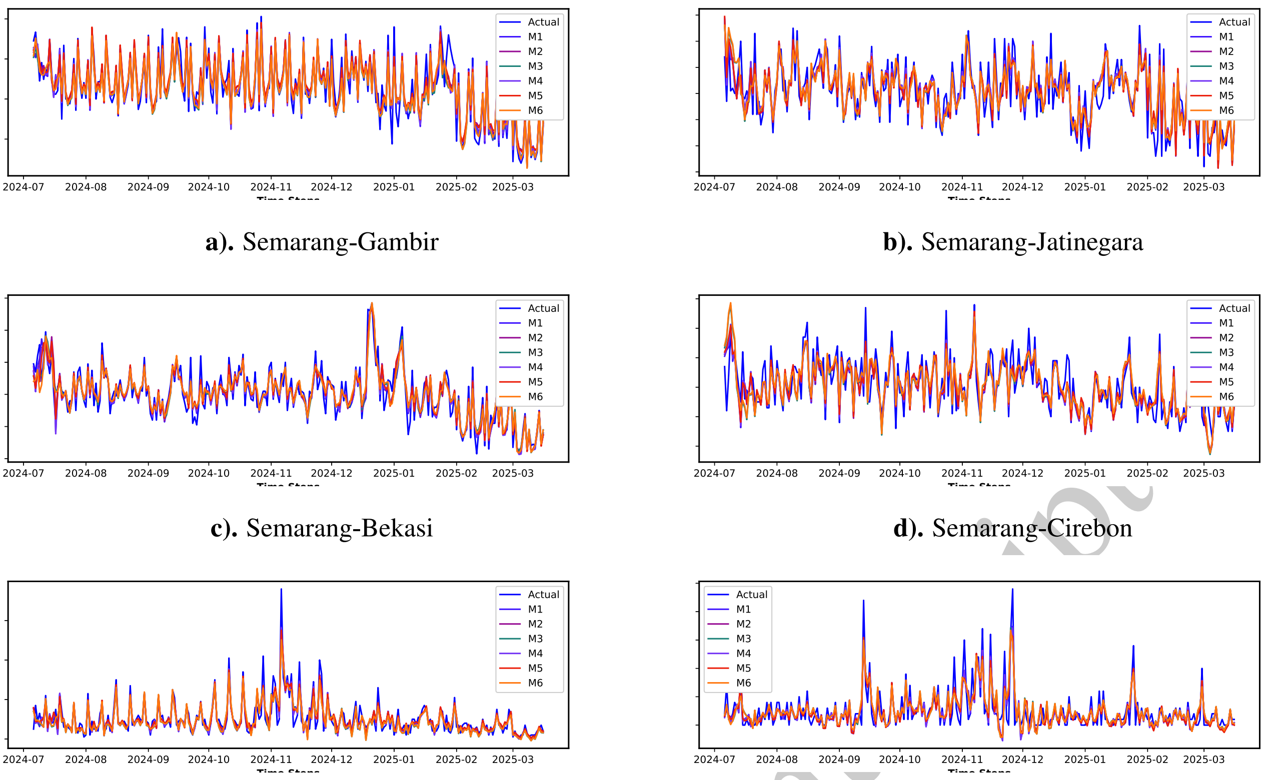
Table 14. Rolling origin performance evaluation of STL-EMD-FCM with sample entropy.

Station	Model	Training Data					Testing Data				
		MAE	RMSE	MSE	sMAPE	R^2	MAE	RMSE	MSE	sMAPE	R^2
Gambir	M1	16.75	22.15	502.18	8.47	0.91	19.82	27.04	743.67	8.94	0.84
	M2	17.10	22.69	528.02	8.57	0.91	20.39	27.59	773.94	9.22	0.83
	M3	20.15	26.66	722.20	9.98	0.87	23.97	32.09	1051.04	10.63	0.77
	M4	16.76	21.93	494.30	8.46	0.91	19.88	26.79	728.67	8.97	0.84
	M5	16.95	22.35	516.79	8.47	0.91	20.32	27.26	755.86	9.21	0.83
	M6	19.69	26.06	693.86	9.76	0.88	23.65	31.64	1019.98	10.49	0.77
Jatinegara	M1	3.96	4.95	24.55	15.59	0.79	4.47	5.61	31.58	15.01	0.75
	M2	4.01	5.02	25.20	15.80	0.79	4.56	5.75	33.12	15.30	0.73
	M3	4.63	5.83	33.97	17.98	0.71	5.33	6.73	45.55	17.65	0.63
	M4	3.99	5.00	25.02	15.70	0.79	4.49	5.66	32.21	15.10	0.74
	M5	4.03	5.05	25.52	15.87	0.78	4.58	5.78	33.54	15.39	0.73
	M6	4.68	5.89	34.66	18.18	0.71	5.37	6.78	46.31	17.81	0.63
Bekasi	M1	6.20	9.06	82.07	16.82	0.73	6.36	8.57	75.20	15.36	0.75
	M2	6.30	9.24	85.42	17.00	0.72	6.55	8.82	80.02	15.73	0.74
	M3	7.20	10.40	108.18	19.38	0.65	7.43	10.12	105.49	17.55	0.65
	M4	6.13	9.00	81.02	16.63	0.74	6.37	8.60	76.03	15.21	0.75
	M5	6.24	9.18	84.35	16.81	0.73	6.56	8.86	80.94	15.67	0.73
	M6	7.10	10.31	106.42	19.11	0.66	7.39	10.12	105.45	17.32	0.65
Cirebon	M1	2.72	3.49	12.16	25.03	0.81	3.61	4.53	20.62	18.61	0.70
	M2	2.78	3.56	12.71	25.42	0.80	3.67	4.60	21.27	18.85	0.69
	M3	3.15	4.06	16.46	27.87	0.74	4.14	5.22	27.33	21.20	0.60
	M4	2.72	3.48	12.10	24.95	0.81	3.62	4.54	20.66	18.67	0.70
	M5	2.77	3.55	12.61	25.26	0.80	3.68	4.60	21.26	18.87	0.69
	M6	3.16	4.07	16.54	27.99	0.74	4.16	5.24	27.57	21.29	0.60
Tegal	M1	1.77	2.45	5.99	59.15	0.83	2.72	3.80	14.92	34.34	0.75
	M2	1.80	2.49	6.18	58.74	0.83	2.78	3.89	15.57	34.27	0.74
	M3	2.12	2.92	8.52	66.97	0.76	3.19	4.45	20.28	39.16	0.66
	M4	1.77	2.46	6.04	58.35	0.83	2.74	3.84	15.22	34.56	0.75
	M5	1.79	2.49	6.23	58.12	0.82	2.79	3.92	15.82	34.46	0.74
	M6	2.12	2.93	8.60	66.66	0.76	3.20	4.48	20.61	39.34	0.66
Pekalongan	M1	1.45	2.35	5.55	103.64	0.58	1.59	2.35	5.75	99.30	0.57
	M2	1.49	2.40	5.79	105.32	0.56	1.67	2.41	5.92	101.59	0.55
	M3	1.65	2.72	7.43	109.37	0.44	1.83	2.65	7.18	106.14	0.46
	M4	1.46	2.37	5.62	102.97	0.58	1.60	2.36	5.81	98.99	0.56
	M5	1.50	2.42	5.87	105.13	0.56	1.68	2.42	5.99	101.41	0.55
	M6	1.65	2.74	7.50	108.80	0.43	1.84	2.66	7.25	106.07	0.45

5. Conclusion

This study presents a systematic evaluation of decomposition-based forecasting frameworks for railway passenger demand, focusing on the contribution of each modeling stage. Using an ablation-driven approach, multiple configurations, including STL, STL-EMD, and STL-EMD-FCM with and without sample entropy are examined under a unified experimental setting. The results, evaluated using a rolling origin framework, show that decomposition-based strategies generally improve predictive performance compared to direct modeling approaches. In particular, EMD enhances the representation of nonlinear residual patterns, while clustering helps structure complex components. However, these improvements are not consistently observed across all stations, indicating the data-dependent nature of the approach.

The inclusion of sample entropy further enriches feature representation, although its impact varies across cases, with both improvements and slight degradations observed. Overall, the study provides



a). Semarang-Gambir **b).** Semarang-Jatinegara

c). Semarang-Bekasi **d).** Semarang-Cirebon

e). Semarang-Tegal **f).** Semarang-Pekalongan

Figure 6. Comparison of actual and predicted passenger volumes across all destination stations.

practical insights into how decomposition, clustering, and feature extraction influence forecasting performance and stability. Furthermore, empirical evaluation on the KA Argo Muria railway service demonstrates strong predictive capability, with R^2 values of 0.84 at the Semarang-Gambir station. Nevertheless, this work is limited to a univariate setting and does not consider external variables. Future research will explore multivariate models, alternative decomposition techniques, and more advanced architectures to improve forecasting robustness. Empirical evaluation on the KA Argo Muria railway service demonstrates that the proposed framework achieves strong predictive performance.

Acknowledgement

The authors are grateful to two anonymous reviewers for their valuable comments and suggestions made on the previous draft of this manuscript. This work has been supported by PT Kereta Api Indonesia (Persero), Operational Area 4 (Semarang) for providing the rail traffic data essential used in this research.

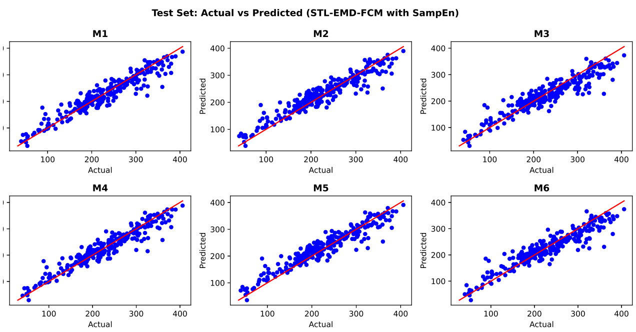
Funding

The authors received no specific funding for this work.

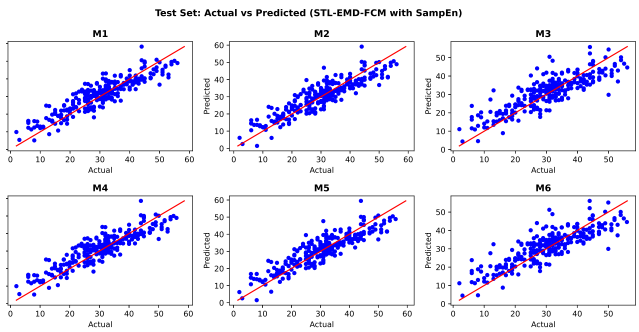
References

[1] Abri, S., and Abri, R. Deep learning methods for LSTM-based personalized search: a comparative analysis. *International Journal of Machine Learning and Cybernetics 16* (4 2025), 2747–2759.

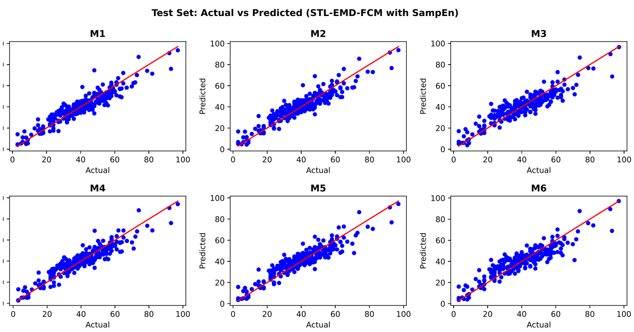
[2] Alhadi, Z., Pratama, D. F., Yusran, R., Permana, I., Eriyanti, F., Suryanef, Khaidir, A., and Riandini, O. Effectiveness of traffic signal system policy for railway transportation safety on the Padang City-Pariaman City line, West Sumatra,



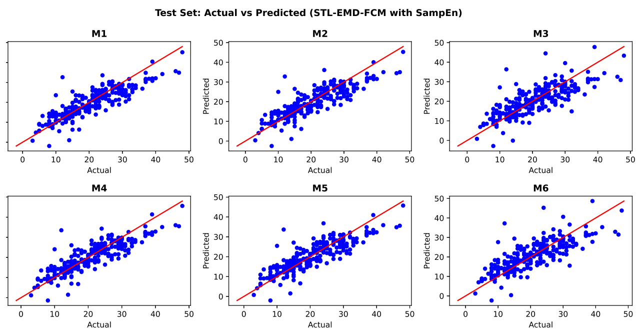
a). Semarang-Gambir



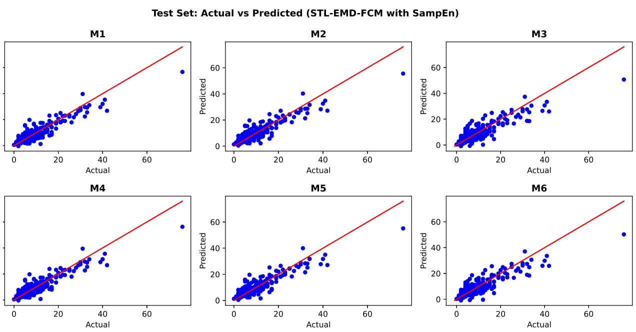
b). Semarang-Jatinegara



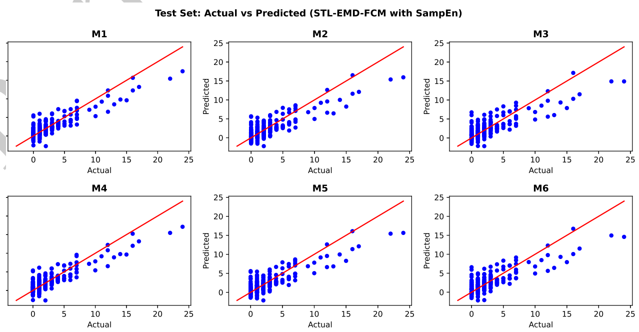
c). Semarang-Bekasi



d). Semarang-Cirebon



e). Semarang-Tegal



f). Semarang-Pekalongan

Figure 7. Scatter plot of actual and predicted passenger volumes across all destination stations.

Indonesia. *International Journal of Transport Development and Integration 9* (3 2025), 69–79.

[3] Alsalem, K. A hybrid time series forecasting approach integrating fuzzy clustering and machine learning for enhanced power consumption prediction. *Scientific Reports 15* (12 2025).

[4] Chen, D., Li, H., Sun, S., Bai, J., and Huang, Z. Point and interval forecasting approach for short-term urban subway passenger flow based on residual term decomposition and fuzzy information granulation. *Applied Soft Computing 166* (11 2024).

[5] Chen, Z., Chen, J., Zheng, H., and Gao, T. Enhancing high-speed railway timetable resilience: a two-level spatiotemporal network model focused on disturbance absorption. *Urban Rail Transit 11* (3 2025), 67–89.

[6] Danbatta, S. J., Muhammad, A., Varol, A., and Abdurrahaman, D. T. Forecasting monthly rainfall using hybrid time-series models and monte carlo simulation amidst security challenges: A case study of five districts from Northern Nigeria. *Environment, Development and Sustainability 27* (6 2025), 13815–13837.

[7] Fang, H., Chen, C. H., Hwang, F. J., Chang, C. C., and Chang, C. C. Metro station functional clustering and dual-view recurrent graph convolutional network for metro passenger flow prediction. *Expert Systems with Applications 247* (8 2024).

[8] Guan, X., Li, M. Z., Qin, J., and Wang, C. Short-term high-speed rail passenger flow forecasting integrated extended empirical mode decomposition with multivariate and bidirectional support vector machine. *Expert Systems with Applications 298* (3 2026), 129870.

[9] Hopfe, D. H., Lee, K., and Yu, C. Short-term forecasting airport passenger flow during periods of volatility: comparative investigation of time series vs neural network models. *Journal of Air Transport Management 115* (3 2024).

[10] Huang, H., Mao, J., Kang, L., Lu, W., Zhang, S., and Liu, L. A dynamic graph deep learning model with multi-

variate empirical mode decomposition for network-wide metro passenger flow prediction. *Computer-Aided Civil and Infrastructure Engineering 39* (9 2024), 2596–2618.

[11] Hyndman, R. J., and Athanasopoulos, G. *Forecasting: Principles and Practice*, 3 ed. OTexts, Melbourne, Australia, 2021.

[12] Hyndman, R. J., Koehler, A. B., Ord, J. K., and Snyder, R. D. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting 18*, 3 (2002), 439–454.

[13] Kasimu, A., Zhou, W., Meng, Q., Wang, Y., Wang, Z., Zhang, Q., and Peng, Y. Performance evaluation of pretrained deep learning architectures for railway passenger ride quality classification. *Alexandria Engineering Journal 118* (4 2025), 194–207.

[14] Kharisudin, I., Arissinta, I. O., Aulia, S. A., Dani, A. Q., and Wijaya, G. K. Comparative study of LSTM-based models with hyperparameter optimization for short-term electricity load forecasting. *BAREKENG: Jurnal Ilmu Matematika dan Terapan 20* (2026), 0105–0122.

[15] Kharisudin, I., and Oktaviani, N. Y. Optimizing deep learning techniques with stacking BiLSTM and BiGRU models for gold price prediction. *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska 15* (9 2025), 123–130.

[16] Li, Z., Wang, X., Cai, H., and Xu, H. Novel hybrid spatiotemporal convolution neural network model for short-term passenger flow prediction in a large-scale metro system. *Journal of Transportation Engineering, Part A: Systems 150* (5 2024).

[17] Liu, J. J., Canca, D., Lv, H., and Ni, S. Demand-adapted train timetabling with coupling-decoupling operations on a bidirectional intercity railway line. *Computers and Industrial Engineering 189* (3 2024).

[18] Long, S., Xie, J., and Cai, M. MFMM: a passenger flow prediction model for rail transit systems considering multiple external environmental factors at stations. *Transportmetrica A: Transport Science* (2025).

[19] Ma, C., Zhang, B., Li, S., and Lu, Y. Urban rail transit passenger flow prediction with ResCNN-GRU based on self-attention mechanism. *Physica A: Statistical Mechanics and its Applications 638* (3 2024).

[20] Ma, G., Xu, K., Zhang, Y., Zhang, L., and Chen, Z. A novel deep learning model for air quality index prediction integrating time series decomposition and intelligent optimization. *Results in Engineering 27* (9 2025).

[21] Patel, M., Patel, S. B., and Swain, D. Data-driven decision support by utilizing machine learning to predict passenger flow for route and station optimization. *Journal of The Institution of Engineers (India): Series B* (2025).

[22] Peng, F. L., Qiao, Y. K., and Yang, C. A LSTM-RNN-based intelligent control approach for temperature and humidity environment of urban utility tunnels. *Heliyon 9* (2 2023).

[23] Qin, Q., Wang, Z., Li, B., and Huang, A. The HWAM-EMD-GRU forecasting model for short-term passenger flow in an airport light rail transit line. *Urban Rail Transit 10* (6 2024), 178–187.

[24] Qiu, H., Zhang, J., Yang, L., Han, K., Yang, X., and Gao, Z. Spatial-temporal multi-task learning for short-term passenger inflow and outflow prediction on holidays in urban rail transit systems. *Transportation* (2025).

[25] Sokhibi, A., Suhardi, B., Laksono, P. W., and Sugiono, S. Executive carriages toilet of Joglosemarkerto Indonesian's train using universal design approach. *Journal of Applied Engineering and Technological Science (JAETS) 6* (6 2025), 1148–1165.

[26] Song, Z., Feng, S., and Li, G. Urban rail transit passenger flow prediction with gaussian process regression based on CEEMDAN. *Journal of Transportation Engineering, Part A: Systems 151* (9 2025).

[27] Sun, J., Ye, X., Zhou, Y., Li, R., and Liu, L. A deep learning model combining STL, GCN, and GRU for urban mass transit station short-term passenger flow prediction. In *Proceedings of CICTP 2024* (12 2024), American Society of Civil Engineers, pp. 1133–1141.

[28] Wang, X., Liu, H., Zhou, J., and Du, Y. Multilevel SVMD-AD-Prophet model for predicting the track irregularity of high-speed railways. *Journal of Infrastructure Systems 31* (9 2025).

[29] Wang, Z., Li, J., and Hei, X. A subway passenger flow prediction based on long short-term memory combined with seasonal and trend decomposition using loess algorithm and genetic algorithm. In *ICNC-FSKD 2024 - 20th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery* (2024), Institute of Electrical and Electronics Engineers Inc.

[30] Wen, X., Si, B., Wei, Y., and Cui, H. Resilience assessment of urban rail transit systems: a literature review. *Public Transport 17* (10 2025), 657–681.

[31] Wu, B., Tang, W., Zhou, Z., Tan, Y., Chen, S., and Feng, Y. An improved adaptive EMD-SVD method for railway bridge dynamic LG-strain processing under moving trainloads. *Structures 80* (10 2025).

[32] Wu, F., Zheng, C., Du, M., Ma, G., and Wu, Z. Adaptive passenger flow prediction in metro systems: an incremental learning framework for evolving patterns. *Journal of Transportation Engineering, Part A: Systems 151* (8 2025).

[33] Xie, X., Wang, Y., Jiang, Y., Cheng, Y., and Peng, H. Metro short-term section passenger flow inherent patterns extraction and prediction: a novel denoising-based method. *Physica A: Statistical Mechanics and its Applications 657* (1 2025).

[34] Xie, Y., Yang, J., Zhang, H., Hou, Y., and Li, Z. Dynamic and multigraph fusion spatio-temporal graph convolution network for short-term forecasting of urban rail transit passenger flow. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations* (2025).

[35] Yin, H., Jin, D., Hong, H., Moon, J., and Gu, Y. H. IAQ-STL-ML: a novel indoor air quality prediction pipeline using meta-learning framework with STL decomposition. *Environmental Technology and Innovation 38* (5 2025).

[36] Yuan, Y., Jiang, X., Zhang, P., and Lai, C. S. Short-term high-speed rail passenger flow prediction by integrating ensemble empirical mode decomposition with multivariate grey support vector machine. *Engineering Applications of Artificial Intelligence 136* (10 2024).

[37] Zhang, D. PFFM-SE: a passenger flow forecasting model for urban rail transit based on multimodal fusion of AFC and social media sentiment under special events. *Transportation* (2025).

[38] Zhang, J., Mao, S., Zhang, S., Yin, J., Yang, L., and Gao, Z. Ef-former for short-term passenger flow prediction

during large-scale events in urban rail transit systems. *Information Fusion 117* (5 2025).

[39] Zuo, J., He, M., Yu, Z., and Wang, J. Short-term passenger flow prediction of urban rail transit on bayesian optimization-bidirectional long short-term memory with causal temporal pattern attention. *International Journal of Transportation Science and Technology* (1 2025).

6. Appendix

Table 15. Rolling origin performance evaluation of baseline models (full series).

		Training Data					Testing Data				
Station	**Model**	**MAE**	**RMSE**	**MSE**	**sMAPE**	R^2	**MAE**	**RMSE**	**MSE**	**sMAPE**	R^2
Gambir	SN	45.03	61.20	3746.62	21.94	0.34	64.10	79.70	6635.29	28.45	-0.45
	ETS	33.88	45.50	2070.61	16.24	0.64	53.04	66.34	4710.36	23.49	-0.02
	PROPHET	34.51	45.52	2074.09	16.87	0.63	65.14	77.43	6438.10	26.52	-0.40
	SARIMA	34.85	47.25	2232.97	16.80	0.61	44.73	57.50	3338.80	19.51	0.27
	LSTM	46.85	57.68	3330.28	22.11	0.41	49.70	60.39	3667.58	21.24	0.20
	GRU	47.57	58.33	3406.37	22.37	0.40	50.99	62.03	3866.69	21.75	0.15
	RNN	30.47	40.49	1641.13	14.57	0.71	39.79	52.21	2734.94	17.41	0.40
	CNN	36.36	47.60	2269.47	17.62	0.60	40.34	53.00	2815.21	17.50	0.38
	BiLSTM	44.34	55.02	3030.77	21.00	0.47	47.17	58.33	3411.46	20.20	0.25
Jatinegara	SN	9.10	11.93	142.50	32.53	-0.21	12.25	15.29	241.80	39.35	-0.96
	ETS	6.96	9.05	81.93	25.07	0.31	9.60	12.14	150.03	30.73	-0.21
	PROPHET	6.80	8.89	79.14	24.54	0.33	10.62	12.80	169.67	33.59	-0.35
	SARIMA	7.52	9.75	94.99	27.20	0.19	13.16	15.59	361.25	36.32	-1.98
	LSTM	7.62	9.69	93.93	27.52	0.20	8.61	10.69	114.40	27.72	0.08
	GRU	7.50	9.61	92.31	27.12	0.22	8.59	10.70	114.57	27.68	0.08
	RNN	7.14	9.18	84.39	25.92	0.28	8.42	10.42	108.73	27.25	0.12
	CNN	6.43	8.18	67.07	23.47	0.43	8.73	10.73	115.31	28.03	0.07
	BiLSTM	7.56	9.65	93.15	27.28	0.21	8.57	10.67	113.88	27.56	0.08
Bekasi	SN	14.31	21.72	472.03	34.89	-0.53	18.17	23.44	563.72	40.09	-0.91
	ETS	10.15	15.03	226.13	25.74	0.27	15.99	20.38	433.48	34.90	-0.47
	PROPHET	9.84	14.45	208.93	25.26	0.32	13.81	17.45	305.45	30.30	-0.03
	SARIMA	11.02	16.35	267.48	28.09	0.13	13.35	17.83	320.82	29.55	-0.06
	LSTM	10.50	15.40	237.25	26.77	0.23	11.41	14.93	224.29	25.81	0.26
	GRU	10.50	15.34	235.46	26.80	0.24	11.45	14.97	225.47	25.91	0.25
	RNN	10.26	15.20	230.99	26.04	0.25	11.23	14.98	226.07	25.33	0.25
	CNN	10.91	15.67	245.66	27.58	0.20	11.56	15.14	230.76	26.03	0.24
	BiLSTM	10.43	15.30	234.07	26.54	0.24	11.33	14.99	226.40	25.61	0.25
Cirebon	SN	6.29	8.21	67.50	50.24	-0.07	9.21	11.60	139.46	43.41	-1.03
	ETS	4.71	6.17	38.07	37.99	0.40	6.71	8.50	72.76	31.83	-0.05
	PROPHET	4.71	6.10	37.22	37.96	0.41	6.64	8.22	67.72	31.82	0.02
	SARIMA	5.19	6.81	46.36	40.66	0.27	6.85	8.62	74.64	32.83	-0.08
	LSTM	5.44	6.88	47.40	42.30	0.25	6.42	8.19	67.02	31.19	0.03
	GRU	5.37	6.87	47.15	41.84	0.25	6.41	8.22	67.63	31.16	0.02
	RNN	4.94	6.37	40.65	39.09	0.36	6.56	8.47	71.80	31.91	-0.04
	CNN	4.95	6.44	41.54	39.25	0.34	6.35	8.08	65.41	30.84	0.05
	BiLSTM	5.19	6.64	44.16	40.70	0.30	6.39	8.12	65.95	31.10	0.04
Tegal	SN	3.93	5.99	35.90	80.42	-0.01	6.37	9.29	89.93	69.29	-0.58
	ETS	2.94	4.56	20.83	69.19	0.41	4.65	7.48	58.33	47.92	0.06
	PROPHET	3.09	4.56	20.78	73.93	0.41	4.43	7.12	53.11	45.71	0.15
	SARIMA	2459.40	3524.86	136346210.06	81.16	-3867514.84	11770.20	11953.98	1568384478.66	112.22	-46303204.60
	LSTM	3.55	5.13	26.33	76.30	0.26	5.48	9.05	89.65	54.51	-0.37
	GRU	3.43	4.99	24.95	73.93	0.30	4.78	7.70	61.48	48.39	0.00
	RNN	2.96	4.48	20.09	68.02	0.43	4.86	7.62	59.59	51.15	0.02
	CNN	2.96	4.41	19.45	67.50	0.45	4.67	7.30	55.14	48.39	0.11
	BiLSTM	3.22	4.91	24.09	70.75	0.32	5.03	8.22	71.89	50.65	-0.12
Pekalongan	SN	2.21	4.76	22.65	99.91	-0.69	3.54	6.32	60.58	118.07	-3.55
	ETS	1.77	3.47	12.08	108.96	0.10	2.27	3.58	12.81	99.46	0.03
	PROPHET	1.80	3.41	11.63	106.82	0.13	2.21	3.56	12.69	102.22	0.04
	SARIMA	2.40	4.79	22.95	132.52	-0.72	2.88	4.33	20.13	131.62	-0.54
	LSTM	1.84	3.56	12.71	102.26	0.05	2.19	3.62	13.11	98.08	0.01
	GRU	1.99	3.59	12.89	103.11	0.03	2.24	3.59	12.91	98.36	0.02
	RNN	1.83	3.46	11.97	102.99	0.10	2.32	3.68	13.58	101.43	-0.03
	CNN	1.75	3.46	11.95	103.04	0.11	2.25	3.59	12.90	100.17	0.02
	BiLSTM	1.80	3.55	12.59	102.42	0.06	2.21	3.59	12.95	98.37	0.02

Table 16. Rolling origin performance evaluation of STL-based baseline models.

		Training Data					Testing Data				
Station	**Model**	**MAE**	**RMSE**	**MSE**	**sMAPE**	R^2	**MAE**	**RMSE**	**MSE**	**sMAPE**	R^2
Gambir	M1	15.39	19.79	393.84	7.59	0.93	21.26	28.15	799.34	9.50	0.82
	M2	16.10	20.87	437.28	7.94	0.92	22.15	29.22	859.43	9.89	0.81
	M3	19.56	25.43	649.28	9.47	0.89	25.78	33.92	1162.55	11.36	0.74
	M4	14.43	18.70	351.97	7.18	0.94	20.33	27.30	753.24	9.13	0.83
	M5	15.11	19.74	392.07	7.49	0.93	21.32	28.33	810.24	9.57	0.82
	M6	18.51	24.19	586.95	9.02	0.90	24.91	33.00	1100.39	10.99	0.76
Jatinegara	M1	3.10	4.01	16.11	11.97	0.86	4.19	5.37	28.97	14.24	0.77
	M2	3.22	4.14	17.18	12.42	0.85	4.34	5.57	31.18	14.86	0.75
	M3	3.90	5.06	25.58	14.71	0.78	5.18	6.64	44.38	17.13	0.64
	M4	3.09	4.02	16.13	11.85	0.86	4.21	5.39	29.23	14.25	0.77
	M5	3.19	4.13	17.06	12.22	0.86	4.35	5.57	31.23	14.83	0.75
	M6	3.91	5.08	25.83	14.70	0.78	5.21	6.67	44.77	17.19	0.64
Bekasi	M1	5.08	7.47	56.23	13.36	0.82	5.81	7.99	64.97	13.82	0.78
	M2	5.24	7.70	59.75	13.69	0.81	6.04	8.32	70.37	14.39	0.76
	M3	6.16	8.90	79.57	16.17	0.74	7.04	9.72	96.17	16.41	0.68
	M4	5.14	7.58	57.88	13.47	0.81	5.97	8.21	68.67	14.17	0.77
	M5	5.31	7.81	61.36	13.82	0.80	6.21	8.53	74.16	14.74	0.75
	M6	6.16	8.95	80.51	16.09	0.74	7.14	9.87	99.01	16.57	0.67
Cirebon	M1	2.22	2.79	7.82	20.81	0.88	2.96	3.85	14.85	15.57	0.78
	M2	2.33	2.92	8.59	21.76	0.86	3.05	3.94	15.53	16.04	0.78
	M3	2.82	3.60	12.97	25.15	0.79	3.72	4.76	22.68	19.27	0.67
	M4	2.19	2.76	7.65	20.52	0.88	2.95	3.84	14.74	15.53	0.79
	M5	2.30	2.89	8.38	21.46	0.87	3.04	3.92	15.38	15.94	0.78
	M6	2.82	3.59	12.93	25.14	0.80	3.73	4.77	22.78	19.30	0.67
Tegal	M1	1.54	2.20	4.83	51.33	0.86	2.44	3.65	13.82	29.60	0.78
	M2	1.61	2.28	5.22	52.36	0.85	2.54	3.76	14.70	30.59	0.76
	M3	1.91	2.75	7.59	58.43	0.79	2.98	4.39	19.89	35.33	0.68
	M4	1.55	2.22	4.92	50.84	0.86	2.44	3.69	14.22	29.52	0.77
	M5	1.62	2.30	5.31	52.11	0.85	2.53	3.81	15.05	30.43	0.76
	M6	1.91	2.78	7.72	58.10	0.78	2.98	4.43	20.32	35.28	0.67
Pekalongan	M1	1.03	1.73	3.04	101.91	0.77	1.38	2.15	5.00	95.69	0.63
	M2	1.10	1.80	3.29	104.91	0.75	1.48	2.22	5.12	99.67	0.62
	M3	1.31	2.20	4.92	110.57	0.63	1.65	2.48	6.30	104.84	0.52
	M4	1.03	1.74	3.05	101.41	0.77	1.37	2.15	4.98	95.05	0.63
	M5	1.10	1.81	3.30	104.39	0.75	1.48	2.22	5.12	99.00	0.62
	M6	1.30	2.20	4.92	109.89	0.63	1.65	2.48	6.30	104.11	0.53

Table 17. Rolling origin performance evaluation of STL-EMD-based baseline models.

		Training Data					Testing Data				
Station	**Model**	**MAE**	**RMSE**	**MSE**	**sMAPE**	R^2	**MAE**	**RMSE**	**MSE**	**sMAPE**	R^2
Gambir	M1	12.06	16.54	273.95	6.01	0.95	21.17	28.92	845.20	9.71	0.81
	M2	12.17	16.76	281.25	6.05	0.95	21.78	29.47	875.92	9.99	0.81
	M3	15.30	20.55	422.79	7.51	0.93	25.38	33.86	1159.40	11.41	0.74
	M4	11.57	15.99	257.55	5.75	0.95	20.84	28.56	824.04	9.57	0.82
	M5	11.58	16.14	261.51	5.74	0.95	21.41	29.05	851.69	9.82	0.81
	M6	14.43	19.62	385.94	7.13	0.93	24.85	33.35	1122.18	11.18	0.75
Jatinegara	M1	2.92	3.74	14.03	11.62	0.88	3.98	5.30	28.19	13.66	0.77
	M2	3.01	3.85	14.87	11.92	0.87	4.13	5.48	30.14	14.25	0.76
	M3	3.72	4.75	22.59	14.44	0.81	4.97	6.50	42.44	16.56	0.66
	M4	2.94	3.76	14.18	11.69	0.88	4.01	5.33	28.51	13.75	0.77
	M5	3.01	3.85	14.87	11.90	0.87	4.14	5.49	30.26	14.29	0.76
	M6	3.75	4.79	22.96	14.60	0.81	5.01	6.53	42.90	16.72	0.65
Bekasi	M1	4.14	7.16	52.28	11.24	0.83	6.13	8.51	74.29	14.97	0.75
	M2	4.22	7.31	54.43	11.40	0.82	6.30	8.80	79.60	15.38	0.74
	M3	5.20	8.32	69.89	14.03	0.77	7.23	10.03	103.70	17.25	0.66
	M4	4.03	7.11	51.52	10.91	0.83	6.05	8.53	74.96	14.72	0.76
	M5	4.10	7.26	53.64	11.04	0.82	6.23	8.83	80.36	15.22	0.74
	M6	5.06	8.23	68.42	13.67	0.78	7.14	10.02	103.51	16.95	0.66
Cirebon	M1	1.85	2.41	5.85	18.45	0.91	3.40	4.32	18.72	18.34	0.73
	M2	1.91	2.48	6.15	18.83	0.90	3.45	4.38	19.25	18.56	0.72
	M3	2.33	3.02	9.16	22.23	0.85	4.00	5.09	25.96	21.16	0.62
	M4	1.84	2.41	5.81	18.42	0.91	3.40	4.33	18.80	18.35	0.73
	M5	1.88	2.46	6.07	18.73	0.90	3.45	4.39	19.29	18.58	0.72
	M6	2.34	3.04	9.26	22.49	0.85	4.02	5.11	26.25	21.29	0.62
Tegal	M1	1.47	2.04	4.16	51.95	0.88	2.60	3.70	14.06	32.80	0.77
	M2	1.52	2.10	4.42	52.27	0.88	2.67	3.79	14.80	32.83	0.76
	M3	1.82	2.50	6.27	59.51	0.82	3.07	4.37	19.66	37.44	0.68
	M4	1.46	2.04	4.18	51.05	0.88	2.61	3.71	14.14	32.94	0.77
	M5	1.52	2.11	4.44	51.76	0.87	2.67	3.80	14.83	32.98	0.75
	M6	1.82	2.51	6.32	59.01	0.82	3.08	4.39	19.76	37.60	0.67
Pekalongan	M1	0.85	1.39	1.95	86.72	0.85	1.46	2.14	4.75	97.36	0.64
	M2	0.89	1.44	2.08	87.69	0.84	1.54	2.20	4.90	99.44	0.63
	M3	1.11	1.77	3.15	95.90	0.76	1.71	2.44	6.06	103.53	0.54
	M4	0.85	1.39	1.93	85.72	0.85	1.45	2.14	4.75	96.78	0.64
	M5	0.89	1.44	2.07	87.47	0.84	1.54	2.20	4.91	98.93	0.63
	M6	1.10	1.76	3.12	95.09	0.76	1.70	2.44	6.07	102.98	0.54

Table 18. Rolling origin performance evaluation of STL-EMD-FCM baseline models without Sample Entropy.

		Training Data					Testing Data				
Station	**Model**	**MAE**	**RMSE**	**MSE**	**sMAPE**	R^2	**MAE**	**RMSE**	**MSE**	**sMAPE**	R^2
Gambir	M1	17.10	22.39	508.39	8.64	0.91	20.23	27.27	754.26	9.07	0.83
	M2	17.53	23.04	537.72	8.79	0.91	20.96	27.96	792.23	9.42	0.82
	M3	20.70	27.21	747.00	10.22	0.87	24.62	32.55	1079.23	10.87	0.76
	M4	16.87	22.01	492.34	8.53	0.91	20.20	26.95	735.86	9.08	0.84
	M5	17.21	22.61	518.33	8.62	0.91	20.84	27.60	770.75	9.39	0.83
	M6	20.10	26.51	710.49	9.96	0.87	24.27	32.07	1044.78	10.73	0.77
Jatinegara	M1	3.06	3.85	16.05	12.14	0.86	3.71	4.85	24.41	12.45	0.80
	M2	3.11	3.93	16.70	12.38	0.86	3.84	5.02	26.07	12.99	0.79
	M3	3.75	4.77	24.03	14.56	0.80	4.56	5.96	36.57	15.10	0.71
	M4	3.08	3.88	16.27	12.16	0.86	3.73	4.88	24.69	12.50	0.80
	M5	3.10	3.93	16.77	12.29	0.86	3.85	5.02	26.14	12.99	0.79
	M6	3.77	4.81	24.47	14.64	0.79	4.59	5.98	36.99	15.20	0.70
Bekasi	M1	5.76	8.22	69.79	15.66	0.77	6.67	8.92	80.48	16.04	0.73
	M2	5.79	8.32	71.70	15.62	0.77	6.80	9.11	84.19	16.30	0.72
	M3	6.63	9.44	91.15	17.86	0.71	7.57	10.24	106.94	17.85	0.64
	M4	5.77	8.26	70.85	15.66	0.77	6.74	9.05	82.50	16.09	0.72
	M5	5.81	8.36	72.72	15.60	0.77	6.89	9.25	86.30	16.42	0.71
	M6	6.61	9.45	91.49	17.77	0.70	7.59	10.32	108.09	17.77	0.64
Cirebon	M1	2.45	3.12	10.06	22.73	0.84	3.47	4.35	18.95	18.06	0.73
	M2	2.49	3.18	10.37	23.09	0.84	3.53	4.40	19.44	18.32	0.72
	M3	2.90	3.74	14.31	25.81	0.77	4.00	5.05	25.55	20.63	0.63
	M4	2.47	3.15	10.26	22.89	0.84	3.50	4.38	19.26	18.19	0.72
	M5	2.51	3.20	10.52	23.25	0.83	3.55	4.43	19.71	18.38	0.71
	M6	2.95	3.79	14.64	26.29	0.77	4.03	5.10	26.07	20.78	0.62
Tegal	M1	1.40	1.91	3.87	50.15	0.89	2.75	3.80	14.89	34.83	0.76
	M2	1.43	1.95	4.00	50.45	0.89	2.79	3.87	15.50	34.62	0.75
	M3	1.71	2.38	5.85	56.31	0.83	3.14	4.36	19.54	38.35	0.68
	M4	1.41	1.93	3.92	49.90	0.89	2.76	3.82	15.04	34.96	0.75
	M5	1.44	1.96	4.06	50.47	0.89	2.80	3.88	15.61	34.66	0.75
	M6	1.72	2.40	5.94	56.31	0.83	3.15	4.38	19.73	38.50	0.68
Pekalongan	M1	1.44	2.40	5.80	100.78	0.56	1.66	2.47	6.27	98.39	0.53
	M2	1.49	2.46	6.05	102.36	0.54	1.74	2.53	6.47	100.23	0.51
	M3	1.65	2.78	7.77	107.27	0.41	1.89	2.77	7.79	104.08	0.41
	M4	1.45	2.43	5.92	100.53	0.55	1.67	2.48	6.36	98.24	0.52
	M5	1.50	2.48	6.17	102.54	0.54	1.75	2.54	6.56	100.10	0.50
	M6	1.65	2.80	7.88	107.00	0.41	1.90	2.78	7.88	104.22	0.40

Table 19. Optimized hyperparameter.

Model	**Param**	**Gambir**	**Jatinegara**	**Bekasi**	**Cirebon**	**Tegal**	**Pekalongan**
ETS	Trend	None	None	None	Add	None	None
	Seasonal	Add	Add	Add	Add	Add	Add
Prophet	Seasonality Mode	Additive	Additive	Multiplicative	Additive	Additive	Additive
	Changepoint Prior Scale	0.5	0.5	0.05	0.05	0.5	0.05
SARIMA	Order (p, d, q)	(2,0,0)	(2,1,0)	(0,0,0)	(1,2,2)	(2,2,2)	(0,1,0)
	Seasonal Order (P, D, Q, s)	(1,1,1,7)	(1,1,1,7)	(1,1,1,7)	(0,0,0,7)	(1,0,1,7)	(0,0,1,7)
LSTM	Activation	ReLU	Tanh	Tanh	Tanh	Tanh	ReLU
	Learning Rate	0.0005	0.001	0.001	0.001	0.001	0.001
	Units	32	128	64	128	128	128
	Dropout	0.1	0.2	0.2	0.2	0.2	0.1
GRU	Activation	ReLU	ReLU	ReLU	Tanh	Tanh	Tanh
	Learning Rate	0.0005	0.001	0.001	0.001	0.001	0.001
	Units	32	64	64	32	128	64
	Dropout	0.1	0.1	0.1	0.2	0.2	0.2
RNN	Activation	ReLU	ReLU	Tanh	ReLU	ReLU	ReLU
	Learning Rate	0.0005	0.0001	0.0001	0.0005	0.0005	0.0005
	Units	64	64	64	32	32	32
	Dropout	0.1	0.1	0.1	0.1	0.1	0.1
CNN	Activation	Tanh	ReLU	Tanh	Tanh	ReLU	ReLU
	Learning Rate	0.0005	0.0005	0.0005	0.0001	0.0005	0.0001
	Units	128	128	128	32	32	32
	Dropout	0.2	0.1	0.2	0.1	0.2	0.1
BiLSTM	Activation	Tanh	ReLU	Tanh	Tanh	ReLU	ReLU
	Learning Rate	0.0005	0.0005	0.0005	0.0005	0.0005	0.0001
	Units	32	128	32	128	128	64
	Dropout	0.1	0.2	0.1	0.2	0.2	0.1
Trend STL (LSTM)	Activation	ReLU	Tanh	Tanh	Tanh	Tanh	Tanh
	Learning Rate	0.001	0.001	0.001	0.001	0.001	0.001
	Units	64	128	32	128	128	128
	Dropout	0.2	0.1	0.1	0.1	0.1	0.1
Trend STL (GRU)	Activation	Tanh	Tanh	Tanh	Tanh	Tanh	Tanh
	Learning Rate	0.001	0.001	0.001	0.0005	0.001	0.001
	Units	64	32	128	32	128	128
	Dropout	0.1	0.2	0.1	0.1	0.2	0.2
Seasonal STL (RNN)	Activation	Tanh	Tanh	Tanh	Tanh	Tanh	Tanh
	Learning Rate	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005
	Units	128	128	128	128	128	128
	Dropout	0.2	0.2	0.2	0.2	0.2	0.2
Seasonal STL (CNN)	Activation	Tanh	Tanh	Tanh	Tanh	Tanh	ReLU
	Learning Rate	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005
	Units	128	128	128	128	128	128
	Dropout	0.1	0.1	0.1	0.1	0.2	0.2
Seasonal STL (BiLSTM)	Activation	Tanh	Tanh	Tanh	Tanh	Tanh	Tanh
	Learning Rate	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005
	Units	128	64	128	128	64	64
	Dropout	0.1	0.1	0.2	0.2	0.1	0.2
Residual STL (RNN)	Activation	ReLU	ReLU	ReLU	Tanh	Tanh	ReLU
	Learning Rate	0.0005	0.0001	0.0005	0.0005	0.0005	0.0005
	Units	128	128	64	128	32	64
	Dropout	0.1	0.1	0.1	0.1	0.2	0.1
Low Residual without SampEn (BiLSTM)	Activation	Tanh	ReLU	ReLU	ReLU	ReLU	ReLU
	Learning Rate	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005
	Units	128	128	128	128	128	64
	Dropout	0.1	0.1	0.1	0.1	0.1	0.2
Mid Residual without SampEn (GRU)	Activation	ReLU	ReLU	Tanh	ReLU	ReLU	ReLU
	Learning Rate	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005
	Units	32	128	128	64	64	128
	Dropout	0.2	0.2	0.1	0.2	0.2	0.1
Low Residual with SampEn (BiLSTM)	Activation	Tanh	Tanh	Tanh	Tanh	Tanh	Tanh
	Learning Rate	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005
	Units	128	128	128	128	128	128
	Dropout	0.1	0.1	0.1	0.1	0.1	0.1
Mid Residual with SampEn (GRU)	Activation	ReLU	ReLU	Tanh	Tanh	ReLU	ReLU
	Learning Rate	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005
	Units	32	128	64	128	64	128
	Dropout	0.1	0.1	0.1	0.2	0.1	0.1