SHM PUBLISHER
SURYA HIJAU MANFAAT

Journal of Soft Computing Exploration

Homepage: https://shmpublisher.com/index.php/joscex

JOSCEX

p-ISSN: 2746-7686
e-ISSN: 2746-0991

.

LSTM with temporal encoding for irregular time series forecasting in power consumption

Eko Verianto[1], Muhammad Arif Alfian[2]
[1]Information Systems, Universitas Cendekia Mitra Indonesia, Indonesia
[2]Informatics, Universitas Cendekia Mitra Indonesia, Indonesia

Article Info

Article history:

Received Mar 9, 2026
Revised Mar 23, 2026
Accepted Mar 26, 2026

Keywords:

LSTM
Temporal encoding
Irregular time series
Forecasting
Power consumption

ABSTRACT

Power consumption data obtained from sensors are often recorded at irregular time intervals due to network disruptions, device errors, or power outages, resulting in irregular time series that make forecasting difficult. This study aims to develop an electricity consumption forecasting model based on Long Short-Term Memory (LSTM) and Temporal Encoding. LSTM was chosen because it has an effective gating mechanism for capturing temporal dependencies in time series data, while Temporal Encoding explicitly represents time information to handle irregular time intervals without data imputation. The methods in this study include data collection via four electrical current sensors, followed by data aggregation every 10 minutes, and feature engineering using sinusoidal encoding and a time difference encoder. The features were normalized using min-max scaling, organized into a multivariate sequence using a sliding window, and divided using a holdout scheme. The model was trained using LSTM and evaluated using Mean Squared Error (MSE). The results show training MSE values of 9.892×10^{-4}, 7.349×10^{-4}, 9.535×10^{-4} and 1.906×10^{-3}, while the testing MSE values are 4.566×10^{-3}, 2.993×10^{-3}, 1.094×10^{-2} and 1.209×10^{-2} for each node. These findings indicate that temporal encoding performs well on the training data, but the model's generalization ability remains limited.

Corresponding Author:

Eko Verianto,
Information Systems,
Universitas Cendekia Mitra Indonesia,
Jl. Ngeksigondo No.60, Prenggan, Kotagede, Yogyakarta, D.I. Yogyakarta
Email: eko@unicimi.ac.id

https://doi.org/10.52465/joscex.v7i1.2

1. INTRODUCTION

Population growth has driven a massive increase in electricity consumption [1], [2], [3]. Based on data from the International Energy Agency (IEA), short-term projections from 2023 to 2030 show that the average annual growth in electricity demand will increase to 3.3% in the STEPS scenario, more than 3.5% in the APS scenario, and 4.5% in the NZE scenario [4]. This increase is due to the electrification of industrial processes and the growth in electricity demand for household appliances. This surge has resulted in more complex and irregular electricity consumption, making forecasting difficult. This condition is influenced by the absence of data-based consumption management and reliable forecasting models.

A reliable forecasting model can contribute positively to more responsive decision-making in power distribution systems and is key to supporting energy efficiency. However, one of the main challenges in building a reliable forecasting model is the quality of the data used to train the model. The characteristics of electricity consumption data obtained from sensors are often subject to technical disturbances, such as network communication disruptions and sensor firmware errors. In addition, data obtained from sensors is generally limited by certain thresholds. These conditions cause the data to become irregular in terms of time sequence (irregular time series) because data recording does not take place at consistent time intervals. Irregular time series are time series where the data points do not have the same time interval. Gaps in irregular time series data indicate inconsistent timing, which in some cases is intentional, while missing values are usually caused by sensor damage, data entry errors, or unavailability of information [5]. This irregularity in the data can affect the quality of the forecasting model produced through the training process, so an approach that can handle temporal irregularities in the data is needed.

Several approaches have been taken to handle irregular time series data in forecasting models. Qin (2023) proposed an integrated architecture for neural dynamic systems on irregular data samples, which proved to significantly improve model performance. However, this approach still faces obstacles in terms of high computational costs and ineffective utilization of temporal information in mini-batches [6]. Furthermore, research conducted by Niako and Che (2024) uses imputation techniques to handle missing data. Although imputation methods such as linear interpolation are often considered effective, the results of this study show that these techniques do not always guarantee better prediction quality and even tend to fail in reliably reconstructing time series [7], [8]. Another effort was demonstrated by Yalavarthi (2024), who proposed the Sparsity Structure Graph to improve forecasting accuracy, but this approach is still limited in effectively handling metadata [9]. Liu (2025) then developed a continuous-time neural network to evaluate the influence of timestamp variations. The results showed that these variations did not have a significant impact on the performance of either discrete or continuous time neural networks, thus limiting the effectiveness of this approach [10]. Additionally, Nam (2024) applied adversarial learning techniques, which are capable of generating variations in prediction results, but this actually caused inconsistencies in prediction performance [11]. Meanwhile, Susetyo et al. (2025) applied an Internet of Things (IoT)-based Long Short-Term Memory (LSTM) model to handle missing data, achieving more efficient results compared to commercial solutions. however, the solution of that study does not include an explicit mechanism for handling irregular intervals, thus its effectiveness in addressing inconsistent time dynamics remains limited [12].

Based on previous studies, the challenge of forecasting irregular time series lies in the limited ability of models to explicitly incorporate time interval information without incurring high computational costs or relying on potentially inaccurate imputation. Specifically, although the LSTM approach proposed by Susetyo et al. [12] is capable of handling missing data, it still assumes regular time intervals and depends on imputation techniques that may alter the original data and introduce bias. Therefore, this study adopts the LSTM architecture, as it is designed to capture temporal dependencies and can effectively model sequences with long-term dependencies [13]. The temporal encodings proposed in this study are Time Difference Encoder and Sinusoidal Encoding of cyclic features. The Time Difference Encoder is a temporal encoding of event-based signals that converts the time difference between two consecutive events into a group of output events, so that the number of output events and the time between events will encode temporal information more efficiently [14]. The Sinusoidal Encoding of cyclic features is an approach that maintains the continuous nature of time-based patterns while capturing the periodicity inherent in them [15]. This research contributes by integrating dual explicit temporal features that combine periodic patterns and time differences between observations into the LSTM model. Unlike previous approaches that assume a linear time sequence, the proposed method is capable of capturing the dynamics of irregular time series without requiring data imputation. Thus, the combination of LSTM with sinusoidal encoding of cyclic features and the time difference encoder enables more robust predictive model for handling irregular time series data. The LSTM leverages the temporal dependencies in the data, while the sinusoidal encoding of cyclic features and the time difference encoder enrich the temporal information provided to the model with more efficient and adaptive time representation.

2. METHOD

This study uses a quantitative approach with an experimental design as the research method. Quantitative research involves a deductive approach to prove or disprove developed hypotheses. This research includes descriptive, correlational, and experimental research [16]. Experimental design is a traditional approach in conducting quantitative research [17]. The methods in this research are structured in a systematic, planned, and clear manner.

Data Collection

Data collection is a technique or method that researchers can use to gather data [18]. The data collection process in this study was carried out through direct recording using four electric current sensors installed in industrial and household electrical systems. These sensors were designed to continuously record electric current measurements at one-second intervals for thirty days. The measurement data was then automatically sent to a cloud storage service so that it could be accessed and managed centrally. In practice, the data recording and transmission process did not always run consistently. Data irregularities could occur due to various factors, such as power outages that caused the sensors to stop temporarily, internet disruptions that hampered the data transmission process, limitations or disruptions to the cloud server, and problems with the sensor peripheral devices. These conditions result in irregular time intervals between observations, causing the data to be irregular. Therefore, the data obtained reflects the actual conditions of the electrical energy measurement system and is relevant for use in irregular time series forecasting research. Figure 1 below shows a visualization of irregular electricity consumption data.

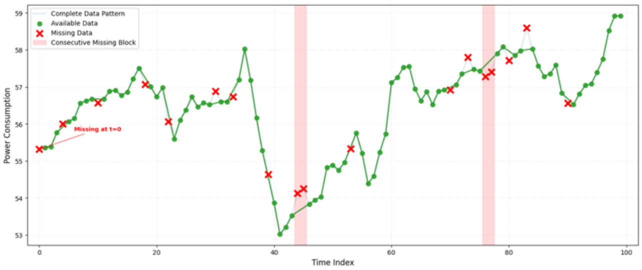


Figure 1. Visualization of electric current data

Data Management

Data preprocessing

Data preprocessing is an important stage in machine learning that plays a significant role in improving model performance [19]. During this stage, data aggregation is performed via query mechanisms on cloud storage to reduce the massive data volume while minimizing extreme fluctuations that could affect model stability. Each data point represents a ten-minute interval, calculated as the average value within each interval. In this process, irregular timestamps are aligned by rounding to the nearest 10-minute interval to achieve a more consistent data structure while preserving time gaps caused by data transmission disruptions.

The data aggregation process does not eliminate the irregular characteristics of the studied time series, as no imputation is performed for intervals lacking power consumption values. Based on an analysis of the four sensor nodes after aggregation, it was found that approximately 20.1% to 23.4% of the time intervals lacked power consumption values due to technical disruptions during data acquisition, leaving only timestamp information. The presence of these missing timestamps is retained as part of the important temporal information. Thus, although the data have been transformed into a more structured interval format, the characteristics of irregular time series are preserved in the form of time intervals without observations due to disruptions causing data loss.

At this stage, other important steps are also taken to ensure that the data is ready for use in modeling. First, the time attributes of the sensor recordings are transformed into a datetime format. This transformation aims to enable the systematic application of time data for calculating time differences in the implementation of temporal encoding techniques. In this section, adjustments are also made to the columns representing electricity consumption values to make them more general and easier to understand.

Feature engineering

Feature engineering aims to adjust, transform, and aggregate predictors to improve the integration of domain-specific knowledge into features, thereby increasing the chances of obtaining a good relationship between predictors and target variables [20]. Feature engineering in this study was carried out by applying two simple temporal encoding approaches, namely sinusoidal encoding of cyclic features that utilizes sine and cosine for time features and time difference encoder that utilizes the time distance between samples. The first approach was used to represent periodic time information, both in daily and weekly patterns. This technique works by transforming 10-minute time slots in a day and days in a week into sine and cosine coordinates. This transformation provides a continuous and cyclic representation of time, allowing the model to capture recurring patterns more effectively. The mathematical formulation for the 10-minute time slot and daily sine-cosine encoding is shown in Equations (1) to (3).

$$slot_{10minutes} = \left\lfloor \frac{hour \times 60 + minutes}{10} \right\rfloor \tag{1}$$

$$day_{sin} = sin\left(2\pi \frac{slot_{10minutes}}{144}\right) \tag{2}$$

$$day_{cos} = cos\left(2\pi \frac{slot_{10minutes}}{144}\right) \tag{3}$$

Meanwhile, weekly patterns are represented using sine and cosine functions based on the day index in a week, as shown in Equations (4) to (5).

$$week_{sin} = sin\left(2\pi \frac{day}{7}\right) \tag{4}$$

$$week_{cos} = cos\left(2\pi \frac{day}{7}\right) \tag{5}$$

The second approach applies a time difference encoder designed to represent the irregularity of the time interval between samples. This representation is done by calculating the time difference between observations in multiples of 600 seconds (10 minutes), as shown in Equation (6).

$$\Delta t_i = \begin{cases} 0, i = 1 \\ \frac{t_i - t_{i-1}}{600}, i > 1 \end{cases} \tag{6}$$

The results of these two temporal encoding techniques are then combined, producing a unified time feature vector as shown in Equation 7 below.

$$TE_{time} = [power\_consumption, day_{sin}, day_{cos}, week_{sin}, week_{cos}, \Delta t_i] \tag{7}$$

The feature vector produces new features that are used as part of the attributes to predict the target. This representation is expected to enrich the temporal information in the data, both in terms of periodicity and time irregularity, thereby improving the model's performance in predicting electricity consumption in irregular time series data. The visualization of the feature engineering results is shown in Figure 2.

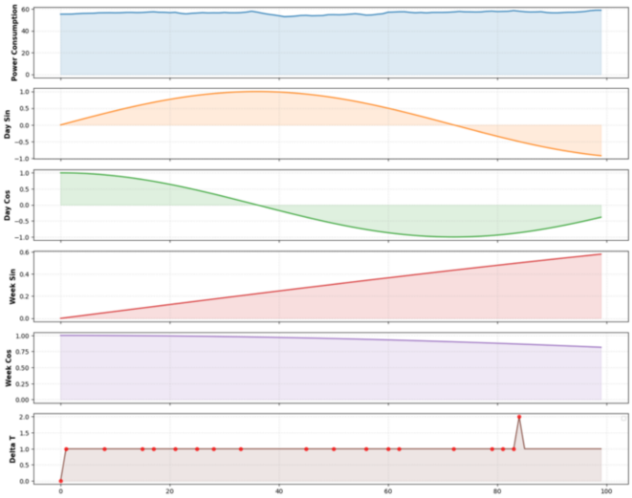


Figure 2. Visualization of feature engineering results

Data scaling

Scaling is a technique used to adjust data so that each feature varies within the same range [21]. This study uses min-max scaling to transform the values of the power consumption and delta t features to a specific range using Equation (8).

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{8}$$

Through the min-max scaling process, the values of each feature are adjusted to common range, typically 0 to 1 [22]. This technique also helps speed up the convergence process during model training because the gradient becomes more stable and avoids extreme fluctuations. Visualization of the data scaling results is shown in Figure 3.

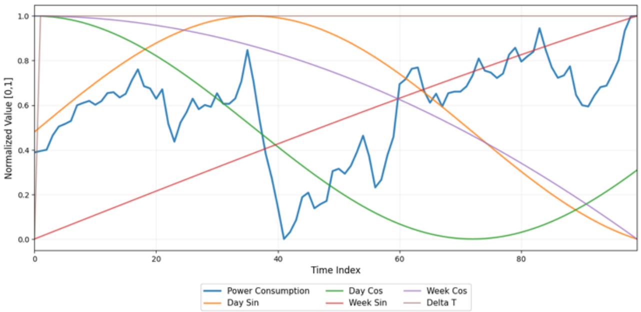
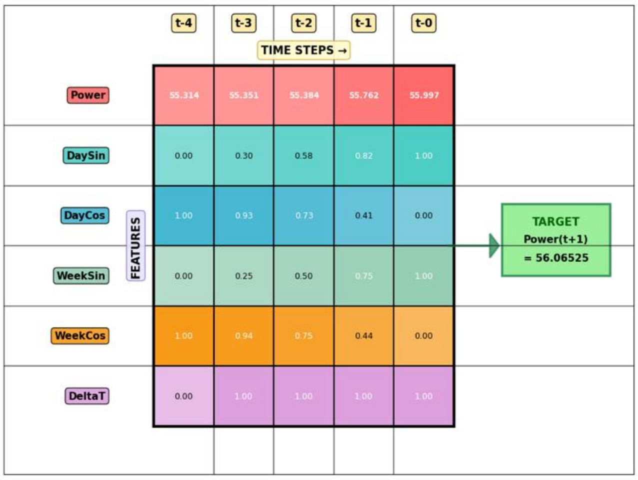


Figure 3. Visualization of data scaling results

Sliding window

A sliding window is used to convert time series data into a supervised learning structure [23]. At this stage, the dataset is divided into sequence segments with lengths adjusted to the model experiment requirements. Each sequence represents multivariate information consisting of six features, namely power consumption, day sin, day cos, week sin, week cos, and delta t. The power consumption value at the next time point (t+1) is set as the prediction target. With this mechanism, the model is trained to learn temporal patterns based on a number of historical observations. The concept of sequence formation using a sliding window and the relationship between input data and prediction targets is shown in Figure 4.



	t-4	t-3	t-2	t-1	t-0
Power	55.314	55.351	55.384	55.762	55.997
DaySin	0.00	0.30	0.58	0.82	1.00
DayCos	1.00	0.93	0.73	0.41	0.00
WeekSin	0.00	0.25	0.50	0.75	1.00
WeekCos	1.00	0.94	0.75	0.44	0.00
DeltaT	0.00	1.00	1.00	1.00	1.00

Figure 4. Sliding window representation

Data splitting

.

Data splitting is a commonly used approach in model validation, where a dataset is divided into two separate sets, namely training data and testing data [24]. In this study, The data was divided into training and testing sets using a holdout validation technique [25], [26]. This study divide the data with a proportion of 80% of the data used for model training and the other 20% used as test data to assess the model's performance objectively. Data separation is carried out sequentially in accordance with the characteristics of time series data, so that the temporal sequence is maintained and information leakage between training data and testing data is avoided. Visualization of the training and testing data division scheme using the holdout method is shown in Figure 5 below.

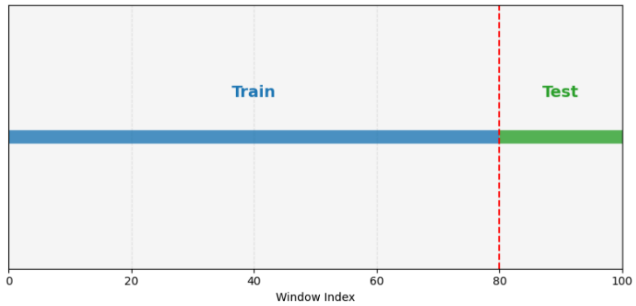


Figure 5. Visualization of data splitting scheme

Modelling

Machine learning builds models by learning and improving their performance based on data, especially when the explicit model structure is not yet clear and algorithms with good performance are difficult to design directly [27]. At this modeling stage, the model is developed by integrating LSTM and Temporal Encoding architectures. LSTM is used to overcome the vanishing gradient problem by utilizing a gating mechanism to manage long-term dependencies [28]. Meanwhile, Temporal Encoding is used to enrich the time representation by incorporating information on periodicity and irregularity of time intervals between observations into feature vectors. These two approaches are expected to improve the model's ability to understand the temporal dynamics of data. Figure 6 below shows the LSTM architecture with Temporal Encoding.

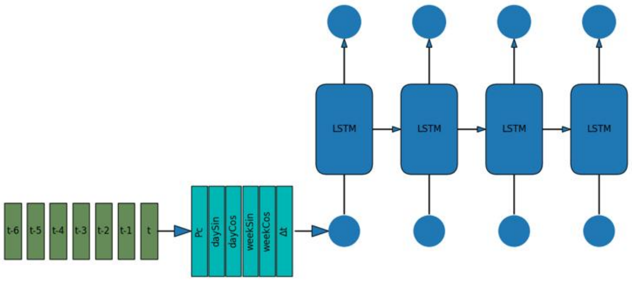


Figure 6. LSTM architecture with temporal encoding

The LSTM architecture works by receiving input in the form of a multivariate sequence resulting from a sliding window process that has gone through a data management stage. Each time step in the sequence contains a feature vector consisting of power consumption values, daily and weekly periodic features resulting from sinusoidal encoding, and time irregularity features. This information is processed sequentially by the LSTM layer through the forget gate, input gate, and output gate mechanisms to regulate the flow of information within the cell state. First, the forget gate determines the extent to which information in the previous cell state c_{t-1} needs to be retained or forgotten, which is formulated through Equation (9).

$$\boldsymbol{f_t} = \boldsymbol{\sigma}(\boldsymbol{W_f} \cdot [\boldsymbol{h_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b_f}) \tag{9}$$

Next, the input gate functions to determine the new information to be stored in the cell state. This process involves calculating the input gate and the new cell state candidate as shown in Equations (10) and (11) below.

$$\boldsymbol{i_t} = \boldsymbol{\sigma}(\boldsymbol{W_i} \cdot [\boldsymbol{h_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b_i}) \tag{10}$$

$$\tilde{c}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{11}$$

The cell state value at time t is then updated by combining the information retained from the previous cell state and the new information generated by the input gate, as shown in Equation (12) below.

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{12}$$

Next, the output gate determines the information to be output as the hidden state based on the updated cell state, as shown in Equations (13) and (14) below.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{13}$$

$$h_t = o_t \odot \tanh(c_t) \tag{14}$$

The hidden state output h_t is then passed to the fully connected layer to generate the predicted electricity consumption value for the next time using Equation (15) below.

$$\hat{y}_{t+1} = W_y h_t + b_y \tag{15}$$

With this mechanism, the LSTM architecture can retain important information in the long term and reduce the vanishing gradient problem, making it effective in modeling temporal dynamics in irregular time series data.

Model Evaluation

The model evaluation stage is used to measure the accuracy and performance of the model in predicting electricity consumption. In this study, model evaluation was performed using the Mean Squared Error (MSE) metric. MSE is the average variance between the predicted values and the actual values [29]. Mathematically, MSE is formulated in Equation (16) below.

$$\boldsymbol{MSE} = \frac{\boldsymbol{1}}{\boldsymbol{n}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (\boldsymbol{y_i} - \hat{\boldsymbol{y}}_{\boldsymbol{i}})^{\boldsymbol{2}} \tag{16}$$

Where y_i is the actual power consumption value, $\hat{y}_i$ is the model's prediction value, and *n* represents the number of test data. A lower MSE value indicates higher prediction accuracy [30]. This indicates that the model's prediction results are closer to the actual value, thus reflecting better model performance.

3. RESULTS AND DISCUSSIONS

Before modeling, it is necessary to determine the model hyperparameters, which consist of the sliding window length representing the number of historical observations in each sequence and LSTM architecture hyperparameters such as the number of units in the LSTM layer, the number of layers, the learning rate, the batch size, and the number of epochs. These hyperparameters were determined empirically through a series of experiments to obtain a configuration that provides forecasting performance on irregular time series data. Table 1 below shows the hyperparameter configuration used in the formation of the LSTM model with Temporal Encoding.

Table 1. Model hyperparameters

Components	Hyperparameter	Value
LSTM Architecture	Number of LSTM Layers	2
	Number of LSTM Units	Layer 1 and Layer 2: 50 Units
Layer Fully Connected	Number of Dense Layer Units	Dense 1: 25 Units Dense Output: 1 Unit
	Activation Function	Dense 1: ReLu Dense Output: Linear
Others	Optimizer	Adam
	Loss Function	MSE
	Epoch	100
	Batch Size	1
	Window Size (Sliding Window)	10

Next, the model was trained using the *hyperparameter* configuration shown in Table 1. The model training scenario was carried out on four models consisting of an LSTM model with Temporal Encoding, a Standard LSTM Model, an LSTM Model with Linear Interpolation Imputation, and an LSTM Model with Autoencoder Imputation. Training was conducted five times on each model to obtain results that could describe the performance of each model through evaluation results using MSE. The data used in this scenario was electricity consumption data in the first industry (node 1), second industry (node 2), first household (node 3), and second household (node 4). The difference in data characteristics between the industrial and household sectors had a clear impact on model performance. At Node 1 and Node 2, consumption patterns showed high fluctuations with sporadic spikes of large amplitude and extreme changes in value from one time to the next. In contrast, Node 3 and Node 4, which represent household data, show more stable and homogeneous consumption patterns. Consumption values oscillate within a narrower range, with more frequent but controlled spikes, and variations between times that tend to be smooth and repetitive. Table 2 below shows a comparison of the average training and testing performance between models at each node.

Table 2. Comparison of model performance at each node

Models		Node 1	Node 2	Node 3	Node 4
LSTM with Temporal Encoding	**MSE Train**	$9{,}892 \times 10^{-4}$	$7{,}349 \times 10^{-4}$	$9{,}535 \times 10^{-4}$	$1{,}906 \times 10^{-3}$
	MSE Test	$4{,}566 \times 10^{-3}$	$2{,}993 \times 10^{-3}$	$1{,}094 \times 10^{-2}$	$1{,}209 \times 10^{-2}$
Standard LSTM	**MSE Train**	$2{,}017 \times 10^{-3}$	$1{,}664 \times 10^{-3}$	$2{,}862 \times 10^{-3}$	$6{,}334 \times 10^{-3}$
	MSE Test	$3{,}926 \times 10^{-3}$	$1{,}666 \times 10^{-3}$	$1{,}145 \times 10^{-2}$	$8{,}605 \times 10^{-3}$
LSTM with Linear Interpolation	**MSE Train**	$8{,}866 \times 10^{-4}$	$9{,}588 \times 10^{-4}$	$2{,}459 \times 10^{-3}$	$4{,}340 \times 10^{-3}$
	MSE Test	$1{,}464 \times 10^{-3}$	$1{,}155 \times 10^{-3}$	$8{,}062 \times 10^{-3}$	$6{,}467 \times 10^{-3}$
LSTM with Autoencoder	**MSE Train**	$3{,}602 \times 10^{-3}$	$3{,}740 \times 10^{-3}$	$5{,}031 \times 10^{-3}$	$5{,}981 \times 10^{-3}$
	MSE Test	$9{,}697 \times 10^{-3}$	$5{,}403 \times 10^{-3}$	$9{,}912 \times 10^{-3}$	$7{,}962 \times 10^{-3}$

Based on the results of training and testing on four nodes consisting of two industrial nodes (node 1 and node 2) and two household nodes (node 3 and node 4), it was found that the characteristics of irregular time series data had a significant effect on the learning behavior and performance of each modeling approach. Differences in consumption dynamics, fluctuation levels, and electricity surge patterns caused varying model responses. The LSTM model with Temporal Encoding consistently produced the lowest training MSE values or values close to the lowest values in almost all nodes, namely nodes 2, 3, and 4, and had a slightly higher difference than the LSTM with Linear Interpolation Imputation in node 1. This shows that the addition of explicit temporal representation through periodicity encoding and time differences between observations allows the model to build a latent representation that is more adaptive to irregular time intervals. The LSTM model with Linear Interpolation Imputation consistently produces the lowest test MSE values across all nodes, including highly fluctuating industrial data and more stable household data. The standard LSTM showed unstable and inconsistent performance across nodes, while the LSTM model with Autoencoder-based imputation consistently showed the lowest performance across all nodes. In industrial data, the autoencoder reconstruction process was unable to accurately maintain sharp spikes and extreme changes, resulting in distorted consumption patterns. Meanwhile, in household data, even though the fluctuations are relatively smoother, the autoencoder still tends to smooth out local dynamics that are important for short-term forecasting. These findings show that the choice of approach in handling irregular time series data has different implications for the learning process and the generalization ability of the model. Approaches that utilize explicit time representations and those that rely on data reconstruction through imputation provide different performance characteristics when applied to data with varying levels of volatility. Therefore, further analysis of the performance of each model at each node is needed to understand how these approaches adapt to different characteristics. Figure 7 below shows a comparison of the average performance of each model at node 1.

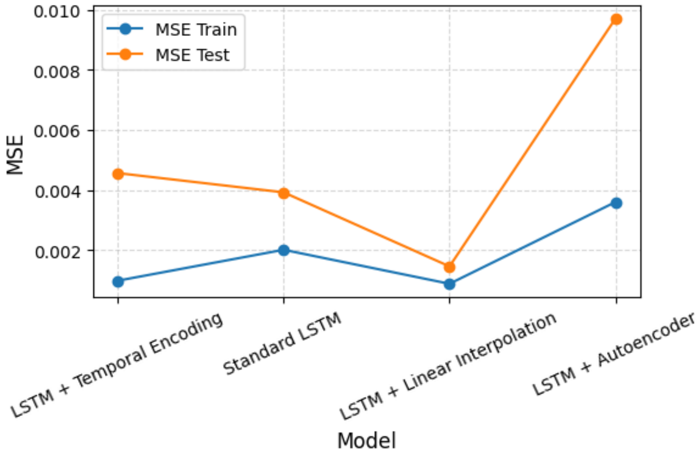


Figure 7. Comparison of training and testing MSE for each model on node 1

Based on the comparison of the average training and testing MSE of node 1 as shown in Figure 7, it can be concluded that each approach has different performance characteristics. The standard LSTM model tends to be less adaptive to data irregularities because it does not explicitly utilize time information. The LSTM approach with Linear Interpolation Imputation provides the best performance on node 1 data, but it depends on the assumption of continuity, which is not always valid in real conditions. Meanwhile, LSTM with Autoencoder-based imputation shows limitations in reconstructing complex and dynamic power consumption data.

The next step is to train each model on node 2 data. Model training is performed using training data and then evaluated using test data. This step aims to evaluate each model's ability to learn historical power consumption patterns and test the model's generalization performance on node 2 data. Figure 8 below shows a comparison of the average performance between models on node 2.

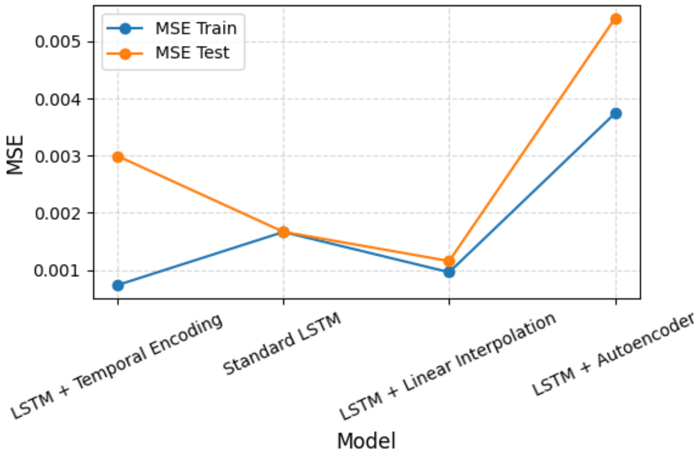


Figure 8. Comparison of training and testing MSE for each model on node 2

Figure 8 shows that the Linear Interpolation imputation approach provides the best performance on node 2 data in terms of forecasting accuracy. However, this approach still relies on the assumption of linear continuity between observations, which does not always reflect actual power consumption conditions. The standard LSTM model shows good stability but is less capable of explicitly utilizing temporal information, while LSTM with Temporal Encoding offers a balance between training stability and adaptability to temporal irregularities without requiring data imputation. On the other hand, LSTM with autoencoder-based imputation still cannot provide good results when compared to the other three models.

Next, we trained and tested each model on node 3. This process was also used to evaluate the performance of each model on node 3 data. Figure 9 below shows a comparison of the average performance between models on node 3.

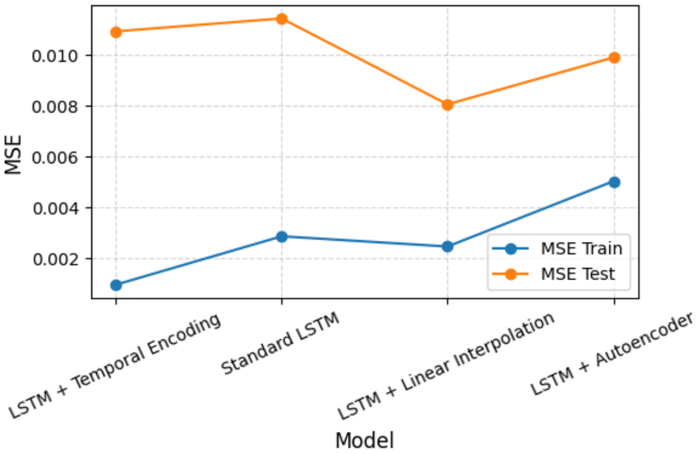


Figure 9. Comparison of training and testing MSE for each model on node 3

Figure 9 shows a clear difference in performance between models. The LSTM model with Temporal Encoding produces the lowest training MSE value, which reflects the model's ability to effectively learn temporal patterns without requiring an imputation process. However, the testing MSE value of this model is relatively comparable to the standard LSTM, indicating that the temporal complexity of node 3 data poses a challenge in the generalization process.

The imputation approach using Linear Interpolation showed the best performance in the testing phase, as reflected in the lowest average testing MSE value among the four other models. These results indicate that the linear missing data imputation process can reduce disturbances caused by time irregularities at node 3. Conversely, the LSTM model with Autoencoder imputation again showed suboptimal performance, both in the training and testing stages, indicating that additional complexity in the imputation stage is not always directly proportional to an increase in forecasting accuracy.

Next, we trained and tested each model on node 4. This process was carried out to evaluate the performance of each model on node 4 data. Figure 10 below shows a comparison of the average performance between models on node 4.

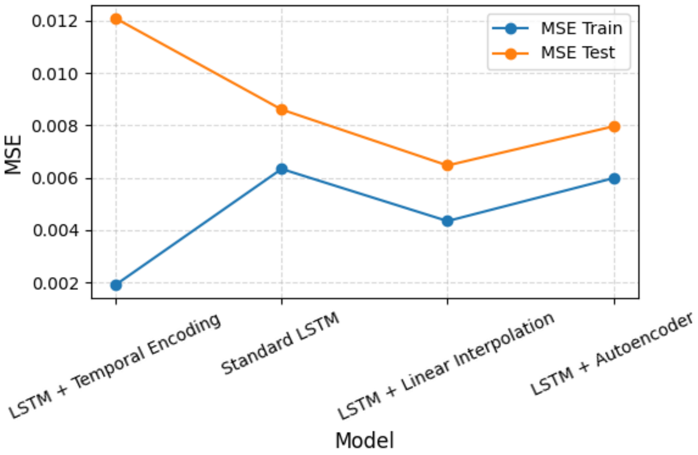


Figure 10. Comparison of training and testing MSE for each model on node 4

Figure 10 shows the average training and testing MSE on node 4 data, revealing clear differences in performance characteristics between approaches. LSTM with Temporal Encoding shows the lowest training MSE value, reflecting the effectiveness of the time encoding mechanism in capturing historical dynamics and irregularities in observation intervals, even though this model's testing MSE value is not the highest. Stability in the training phase indicates that the model is capable of generalizing well on complex data.

LSTM with the Linear Interpolation imputation approach produced the lowest testing MSE value on node 4, indicating that linear data filling can reduce the impact of missing data on forecasting accuracy. However, this approach still relies on the assumption of linear changes between observations, which do not always reflect the nonlinear fluctuations in electricity consumption. The standard LSTM model and LSTM with Autoencoder imputation show relatively similar performance, with higher error values in the training and testing phases.

Overall, LSTM with Temporal Encoding can model industrial nodes with extreme fluctuations and spikes quite well and household nodes very well, without requiring an imputation process, so that the original temporal structure is retained. Despite performing well in the learning process, the experimental results show that LSTM with Temporal Encoding does not excel in terms of testing accuracy, especially when compared to the linear imputation approach that explicitly smooths the data. The relatively higher testing MSE values on some nodes indicate that although Temporal Encoding is effective in representing time information, this approach still faces limitations in capturing extreme dynamics and sharp spikes with precision.

This research contributes to the development of forecasting methods for irregular time series through the integration of LSTM architecture with an explicit temporal encoding approach. Unlike previous approaches that rely on imputation techniques which alter the original data and thus potentially introduce bias. Although test results show that the linear imputation approach yields lower error rates in some cases, this approach relies on the assumption of data continuity as well as the availability of preceding and subsequent values-which are not yet available in streaming or real-time systems. The method proposed in this study has stronger practical implications, particularly for implementation in real-time energy monitoring systems, as it provides adaptive predictions without requiring data reconstruction while preserving the original temporal characteristics of the data.

These findings suggest that temporal feature enrichment alone is not sufficient to fully address the complexity of irregular time series data, especially when the data has high volatility and non-periodic change patterns. Therefore, future model development may consider approaches that explicitly integrate time mechanisms into the model's memory structure, such as time-aware recurrent networks or continuous-time models. Such approaches have the potential to enable model memory to adapt to the length of time intervals between observations, allowing for a more adaptive learning of temporal dynamics compared to simply adding time features as additional inputs.

4. CONCLUSION

This study successfully developed a model for forecasting electricity consumption in irregular time series data by integrating Long Short-Term Memory (LSTM) architecture and Temporal Encoding. The proposed approach is capable of explicitly representing temporal information through a combination of sinusoidal encoding of cyclic features used for periodic patterns and a time-difference- -encoder to capture the irregularity of time intervals. This integration allows the model to obtain a richer temporal representation without data reconstruction, thereby preserving the original temporal structure.

The results of experiments conducted on four electricity consumption datasets originating from two industrial nodes and two household nodes, as well as four models consisting of an LSTM model with Temporal Encoding, a Standard LSTM Model, an LSTM Model with Linear Interpolation Imputation, and an LSTM Model with Autoencoder Imputation, show that data characteristics have a significant effect on model performance. The LSTM model with Temporal Encoding consistently produced relatively low training mean square error (MSE) values for almost all nodes, namely 9.892×10-4 at node 1, 7.349×10-4 at node 2, 9.535×10-4 at node 3 and 1.906×10-3 at node 4. These results indicate that the addition of explicit temporal representation helps the model learn temporal dynamics and observation interval irregularities more effectively than the standard LSTM model.

During the testing phase, the LSTM model with Temporal Encoding showed a mean squared error (MSE) of 4.566×10-3 at node 1, 2.993×10-3 at node 2, 1.094×10-2 at node 3 and 1.209×10-2 at node 4. Although the linear imputation approach yields lower test values at some nodes, it relies on the assumption of linear continuity between observations, which may alter the original temporal structure of the data. Meanwhile, the LSTM approach with Temporal Encoding is capable of preserving the characteristics of irregular time series without relying on data reconstruction and provides a more realistic alternative for implementation in real-time energy monitoring systems. However, these results also indicate that temporal feature enrichment alone is not yet fully capable of capturing extreme dynamics and complex time interval variations. Therefore, future research is recommended to develop models that integrate more adaptive time mechanisms into the network's memory structure, such as time-aware recurrent networks or continuous-time neural models, so that the representation of temporal dynamics can be studied more comprehensively.

ACKNOWLEDGEMENTS

We would like to express our deepest gratitude to the Ministry of Higher Education, Science, and Technology of the Republic of Indonesia for providing funding support for this research through the Penelitian Dosen Pemula Program. We also extend our thanks to the research team for their dedication and cooperation. Furthermore, we are grateful to the entire academic community of Universitas Cendekia Mitra Indonesia for their support, which enabled this research to be completed.

REFERENCES

[1] N. Alnavis, R. Wirawan, K. Solihah, and V. Nugroho, “Energi listrik berkelanjutan: Potensi dan tantangan penyediaan energi listrik di Indonesia,” *J. Innov. Mater. Energy, Sustain. Eng.*, vol. 1, Jan. 2024, doi: 10.61511/jimese.v1i2.2024.544.

[2] H. Nurhaeni, A. Delhi, O. P. M. Daeli, S. A. Anjani, and N. A. Yusuf, “Optimizing Electrical Energy Use through AI: An Integrated Approach for Efficiency and Sustainability,” *Int. Trans. Artif. Intell.*, vol. 2, no. 2, pp. 106–113, 2024, doi: https://doi.org/10.33050/italic.v2i2.533.

[3] D. Amrina and M. Kuncoro, “KAUSALITAS ANTARA KONSUMSI LISTRIK DAN FAKTOR-FAKTOR YANG MEMPENGARUHINYA: ASEAN-5,” *J. Econ. Bus. Eng.*, vol. 4, pp. 271–286, Apr. 2023, doi: 10.32500/jebe.v4i2.4468.

[4] IEA, “World Energy Outlook 2024,” IEA. [Online]. Available: https://www.iea.org/reports/world-energy-outlook-2024

[5] F. A. Shaqra, “Modelling Heterogeneous Time Series from Irregular Data Streams,” Aug. 07, 2024. doi: 10.25439/rmt.26506969.

[6] L. Qin, T. Sun, S. Zhang, and X.-M. Sun, “Tensorized Timeline Alignment for Neural Dynamical Systems in Irregularly Sampled Time Series Prediction,” in *2023 42nd Chinese Control Conference (CCC)*, 2023, pp. 8599–8604. doi: 10.23919/CCC58697.2023.10240296.

[7] N. Niako, J. D. Melgarejo, G. E. Maestre, and K. P. Vatcheva, “Effects of missing data imputation methods on univariate blood pressure time series data analysis and forecasting with ARIMA and LSTM,” *BMC Med. Res. Methodol.*, vol. 24, no. 1, p. 320, 2024, doi: 10.1186/s12874-024-02448-3.

[8] X. Che *et al.*, “Linearly interpolating missing values in time series helps little for land cover classification using recurrent or attention networks,” *ISPRS J. Photogramm. Remote Sens.*, vol. 212, pp. 73–95, 2024, doi: https://doi.org/10.1016/j.isprsjprs.2024.04.021.

[9] V. K. Yalavarthi *et al.*, “GraFITi: Graphs for Forecasting Irregularly Sampled Time Series,” *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 15 SE-AAAI Technical Track on Machine Learning VI, pp. 16255–16263, Mar. 2024, doi: 10.1609/aaai.v38i15.29560.

[10] M. Liu, D. Geißler, S. Bian, B. Zhou, and P. Lukowicz, *Assessing the Impact of Sampling Irregularity in Time Series Data: Human Activity Recognition As A Case Study*. 2025. doi: 10.48550/arXiv.2501.15330.

[11] H. Nam, J. Kim, and J. Yeom, *An Adversarial Learning Approach to Irregular Time-Series Forecasting*. 2024. doi: 10.48550/arXiv.2411.19341.

[12] Rusdiyanto, Susetyorini, and U. Elan, *Good Corporate Governance: Teori dan Implementasinya di Indonesia*. Bandung: Refika Aditama, 2019.

[13] A. Isah *et al.*, “Digital Twins Temporal Dependencies-Based on Time Series Using Multivariate Long Short-Term Memory,” 2023. doi: 10.3390/electronics12194187.

[14] D. Gutierrez-Galan, T. Schoepe, J. P. Dominguez-Morales, A. Jimenez-Fernandez, E. Chicca, and A. Linares-Barranco, “An Event-Based Digital Time Difference Encoder Model Implementation for Neuromorphic Systems,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 33, no. 5, pp. 1959–1973, 2022, doi: 10.1109/TNNLS.2021.3108047.

[15] A. Bansal, K. Balaji, and Z. Lalani, *Temporal Encoding Strategies for Energy Time Series Prediction*. 2025. doi: 10.48550/arXiv.2503.15456.

[16] E. Barroga *et al.*, “Conducting and Writing Quantitative and Qualitative Research.,” *J. Korean Med. Sci.*, vol. 38, no. 37, p. e291, Sep. 2023, doi: 10.3346/jkms.2023.38.e291.

[17] M. A. Ramadhana and D. K. Allo, “Experimental Research in English Language Teaching: A Peek from Undergraduate Students’ Theses,” *J. Stud. Guru dan Pembelajaran*, vol. 4, no. 1 SE-Regular Articles, pp. 32–38, doi: 10.30605/jsgp.4.1.2021.474.

[18] Sukmawati, Sudarmin, and Salmia, “DEVELOPMENT OF QUALITY INSTRUMENTS AND DATA COLLECTION TECHNIQUES,” *J. Pendidik. dan Pengajaran Guru Sekol. Dasar*, vol. 6, no. 1, 2023, doi: jppguseda.v6i1.7527.

[19] S. Strasser and M. Klettke, “Transparent Data Preprocessing for Machine Learning,” in *Proceedings of the 2024 Workshop on Human-In-the-Loop Data Analytics*, in HILDA 24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 1–6. doi: 10.1145/3665939.3665960.

[20] A. Gal-Tzur, S. Bekhor, and Y. Barsky, “Feature engineering methodology for congestion forecasting,” *J. Traffic Transp. Eng. (English Ed.*, vol. 9, no. 6, pp. 1055–1068, 2022, doi: https://doi.org/10.1016/j.jtte.2021.03.007.

[21] L. B. V de Amorim, G. D. C. Cavalcanti, and R. M. O. Cruz, “The choice of scaling technique matters for classification performance,” *Appl. Soft Comput.*, vol. 133, p. 109924, 2023, doi: https://doi.org/10.1016/j.asoc.2022.109924.

[22] F. Rachmawati, J. Jaenudin, N. B. Ginting, and P. Laksono, “Machine Learning for the Model Prediction of Final Semester Assessment (FSA) using the Multiple Linear Regression Method,” *J. Tek. Inform.*, vol. 17, no. 1 SE-ARTICLES, pp. 1–9, doi: 10.15408/jti.v17i1.28652.

[23] R. Farsani, E. Pazouki, and J. Jecei, “A Transformer Self-Attention Model for Time Series Forecasting,” *J. Electr. Comput. Eng. Innov.*, vol. 9, pp. 1–10, Jan. 2021, doi: 10.22061/JECEI.2020.7426.391.

[24] V. R. Joseph, “Optimal ratio for data splitting,” *Stat. Anal. Data Min. ASA Data Sci. J.*, vol. 15, no. 4, pp. 531–538, Aug. 2022, doi: 10.1002/sam.11583.

[25] L. Lamrani, C. Bongiorno, and M. Potters, *Optimal Data Splitting for Holdout Cross-Validation in Large Covariance Matrix*

Estimation. 2025. doi: 10.48550/arXiv.2503.15186.

[26] L. Lamrani, B. Collins, and J.-P. Bouchaud, *Holdout cross-validation for large non-Gaussian covariance matrix estimation using Weingarten calculus*. 2025. doi: 10.48550/arXiv.2509.13923.

[27] X. Shu and Y. Ye, "Knowledge Discovery: Methods from data mining and machine learning," *Soc. Sci. Res.*, vol. 110, p. 102817, 2023, doi: https://doi.org/10.1016/j.ssresearch.2022.102817.

[28] I. D. Raharjo and E. R. Subhiyakto, "Implementing Long Short Term Memory (LSTM) in Chatbots for Multi Usaha Raya," *Adv. Sustain. Sci. Eng. Technol.*, 2024, [Online]. Available: https://api.semanticscholar.org/CorpusID:273434923

[29] S. Namasudra, S. Dhamodharavadhani, and R. Rathipriya, "Nonlinear Neural Network Based Forecasting Model for Predicting COVID-19 Cases.," *Neural Process. Lett.*, vol. 55, no. 1, pp. 171–191, 2023, doi: 10.1007/s11063-021-10495-w.

[30] N. Hasdyna, "MODEL PREDIKSI PREVALENSI STUNTING DI PROVINSI ACEH MENGGUNAKAN PENDEKATAN REGRESI DENGAN EVALUASI KINERJA BERBASIS MSE DAN RMSE," *Djtechno J. Teknol. Inf.*, vol. 6, pp. 920–930, Dec. 2025, doi: 10.46576/djtechno.v6i3.7688.

.