Synergistic preprocessing approaches for improved time series analysis

Andri Pranolo [a,1,*], Supriyanto [a,2], Ardi Pujiyanta [a,3]

[a] Informatics Department, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[1] andri.pranolo@tif.uad.ac.id; [2] supriyanto@tif.uad.ac.id; [3] ardipujiyanta@tif.uad.ac.id
* corresponding author

ARTICLE INFO

Article history
Received November 17, 2025
Revised December 14, 2025
Accepted December 15, 2025
Available online February 28, 2026

Keywords
Data preprocessing
Hyperparameter tuning
Forecasting
Time series analysis

ABSTRACT

This paper systematically evaluates the performance of an LSTM baseline model, along with four smoothing augmentation methods (Kalman, Laplace, Moving Average, Savitzky-Golay), under different normalization strategies (Min-Max and Z-Score) for multivariate time-series forecasting. Experiments were conducted on six publicly available datasets (electricity consumption, energy consumption, sensor data, household energy, Indian electricity, and Brazilian temperature), and model performance was comprehensively compared using three metrics: MAPE, RMSE, and R^2. Results indicate that Laplace smoothing achieved the best performance across five datasets, effectively reducing errors while maintaining high fit quality, demonstrating its advantage in handling highly volatile and noisy time-series data. However, in some instances, Laplace smoothing, along with MA and SG methods, may produce an "over-smoothing" effect, causing forecasts to lose sensitivity to spike fluctuations. The choice of normalization strategy is equally critical: Min-Max is more suitable for data with stable distributions, while Z-Score demonstrates greater advantages for data with large numerical ranges and significant volatility. Notably, in temperature datasets with small sample sizes and high volatility, complex smoothing methods actually degraded performance, making the baseline LSTM + Z-Score the optimal choice. However, the LSTM-Laplace model with Min-Max normalization achieves the best performance among the models. Overall, the study concludes that improving prediction performance relies not only on model architecture but also on optimizing data scale, distribution characteristics, and preprocessing strategies.

© 2026 The Author(s).
This is an open access article under the CC–BY-SA license.

1. Introduction

Time series forecasting, as a vital branch of data analysis, plays a crucial role across numerous fields, including energy management, transportation, financial markets, weather forecasting, and healthcare. By modelling patterns in historical data and capturing underlying trends, researchers and decision-makers gain advanced insights into future developments, thereby providing scientific foundations for resource allocation, risk control, and strategic planning. With the proliferation of IoT and sensor technologies, diverse time-series data now exhibit massive scales, multidimensional complexity, intricate noise patterns, and uneven distributions. Traditional statistical methods increasingly struggle to address such intricate environments. For instance, classical approaches like ARIMA and SARIMA excel with stationary and linear data, but suffer significant degradation in prediction when encountering nonlinear relationships or multidimensional external influences [1], [2].

In recent years, deep learning techniques have emerged as the dominant approach in time series forecasting research due to their powerful nonlinear modelling and feature-extraction capabilities. Deep learning models, represented by Long Short-Term Memory (LSTM) [3]-[8] Gated Recurrent Units (GRU) [9]-[12], Convolutional Neural Networks (CNN) [13], and the rapidly evolving. Transformer architecture demonstrates significant advantages in capturing long-term dependencies, extracting complex patterns, and modelling high-dimensional data. However, the predictive performance of deep learning models depends not only on the network architecture itself but also closely on data preprocessing methods and hyperparameter tuning. Without effective data processing and reasonable hyperparameter optimization, even state-of-the-art deep learning models may suffer from overfitting, underfitting, or slow convergence, resulting in suboptimal prediction accuracy [14]-[16].

In real-world time series data, noise is ubiquitous and may stem from sensor measurement errors [17], [18] external environmental disturbances [19], or uneven data sampling. Suppose such data is fed directly into deep learning models. In such cases, the models often become disrupted by high-frequency noise during training, making it difficult to identify underlying long-term trends and periodic patterns [20]. Consequently, data smoothing techniques emerge as a crucial preprocessing method, enhancing the structural features of time series by attenuating random fluctuations [21]-[25]. Among these, Kalman smoothing is widely applied for modelling noisy data due to its recursive computation advantages and adaptability to dynamic systems [26]-[29]. However, smoothing does not always enhance performance. In data with pronounced trends or moderate volatility, excessive smoothing may instead cause critical signals to be lost, thereby degrading forecasting accuracy [30], [31].

Beyond smoothing techniques, data normalization also plays an indispensable role in deep learning-based time series forecasting. Given the significant variations in dimensionality and numerical ranges across different time series, failure to normalize data may cause gradient instability during backpropagation [32], [33] thereby affecting model convergence speed and prediction accuracy. Standard normalization techniques include Min-Max normalization and Z-score standardization, each offering distinct advantages in different scenarios. Min-Max normalization scales data to the [0,1] range, enhancing the model's sensitivity to relative changes when handling highly volatile data [34], [35]. Conversely, Z-score normalization transforms data by shifting the mean and variance, thereby approximating a standard normal distribution [36], [37]. This makes it more suitable for data exhibiting clear long-term trends and stable distributions. Selecting the appropriate normalization method for different data types remains a crucial factor in determining model performance.

On the other hand, hyperparameter tuning is widely recognized as a crucial factor in determining the performance of deep learning models. Hyperparameters, including the number of hidden layers [38], number of neurons [39], learning rate [40], batch size [41], number of training epochs, and optimizer type, directly dictate a model's capacity, convergence speed, and generalization ability. Improper hyperparameter settings can lead to underfitting, overfitting, or inefficient training. Standard hyperparameter search methods include grid search [42], [43] random search [44] and Bayesian optimization [43], [45]. Among these, grid search is frequently employed in small-scale experiments to identify optimal configurations due to its systematic and comprehensive nature [46], [47]. While hyperparameter tuning can significantly enhance model performance, it often depends on dataset characteristics, with optimal configurations varying considerably across different datasets [39], [48]. This implies that no universal solution exists applicable to all scenarios.

Existing research predominantly focuses on isolated improvements, such as examining the role of Kalman smoothing in model denoising, comparing Min-Max versus Z-score normalization on specific datasets, or employing genetic algorithms and particle swarm optimization for hyperparameter optimization. However, systematic comparisons and analyses of the interactive effects of smoothing methods, normalization techniques, and hyperparameter tuning within a unified experimental framework remain insufficient. In other words, existing research has yet to address a critical question: When these three methods are applied simultaneously to deep learning prediction models, do they complement one another, add to one another, or cancel one another out? Answering this question not only deepens theoretical understanding of time-series prediction mechanisms but also has significant implications for the design and optimization of practical systems.

Based on this, this study selected six datasets with varying temporal granularities and characteristics, including second-level household energy consumption data, minute-level electricity and energy consumption data, hourly sensor data, daily electricity usage data, and annual temperature data. The aim is to comprehensively evaluate the effectiveness and applicability of smoothing methods, normalization methods, and hyperparameter tuning through systematic experiments, by comparing LSTM model variants: Min-Max LSTM and Z-score LSTM combined with Kalman smoothing, Laplace, Moving Average, and Savitzky-Golay. This study not only reveals the relative strengths and weaknesses of different methods across various datasets but also explores their interactive effects. This provides a scientific reference for applying deep learning to time-series forecasting.

Time series forecasting research has shifted from single-model optimisation toward multidimensional, comprehensive evaluation. Data preprocessing and hyperparameter tuning, as the most influential components, require systematic investigation within a unified framework [49], [50]. The significance of this study lies not only in comparing the independent effects of multiple methods but also in analysing their synergistic interactions and differences in applicability. This advances time series forecasting from *experience-driven* to *scientifically optimised* approaches, laying the foundation for constructing high-precision prediction systems in critical fields such as energy, meteorology, and transportation. The main contributions of this paper are reflected in:

- Evaluating the performance differences between LSTM and multiple smoothing methods (Kalman, Laplace, Moving Average, Savitzky-Golay) across six distinct types of time series datasets.
- Examining the suitability and advantages/disadvantages of Min-Max and Z-Score normalization methods across varying data distributions and volatility characteristics.

The remainder of this paper is organized as follows: Section 2 introduces the research methodology and experimental design; Section 3 presents experimental results and comparative analysis, discusses findings, and elucidates their theoretical and practical implications; Section 4 concludes the paper and proposes future research directions. Using this research framework, this paper aims to deepen the understanding of the roles of preprocessing and optimization techniques in deep learning-based time-series forecasting, building on existing literature.

2. Method

2.1. Dataset

Fig. 1 illustrates the time-series characteristics of six datasets for the target variable. Dataset 1 (electricity consumption) exhibits high overall volatility, with values predominantly between 20,000 and 50,000 and pronounced peaks and troughs. Dataset 2 (energy consumption time series) features target variables primarily concentrated between 0 and 10, yet exhibits occasional sharp spikes, indicating a sparse, highly sporadic pattern. Dataset 3 (hourly sensor data) shows lower initial recorded values that gradually increase over time, with amplified fluctuations in later periods, indicating a cumulative effect on sensor counts. Dataset 4 (Household Energy Usage Data) spans a long period, with more than 70,000 data points, and exhibits significant fluctuations, including substantial energy consumption surges during specific periods, reflecting typical uneven household energy usage patterns. Dataset 5 (India's 2019–2020 electricity consumption) remains generally stable, with most values between 200 and 500 and only minor fluctuations, indicating steady power consumption. Dataset 6 (Brazilian Urban Temperature Time Series) exhibits periodic patterns with sine-like fluctuations, reflecting seasonal temperature variations across different time points. Collectively, these datasets exhibit distinct distributional characteristics, encompassing high-frequency fluctuations, sparse bursts, upward trends, relatively stable patterns, and cyclical patterns. This diversity provides a versatile experimental environment for comparing and optimizing subsequent time series forecasting models.

Vol. 12, No. 1, February 2026, pp. 16-37

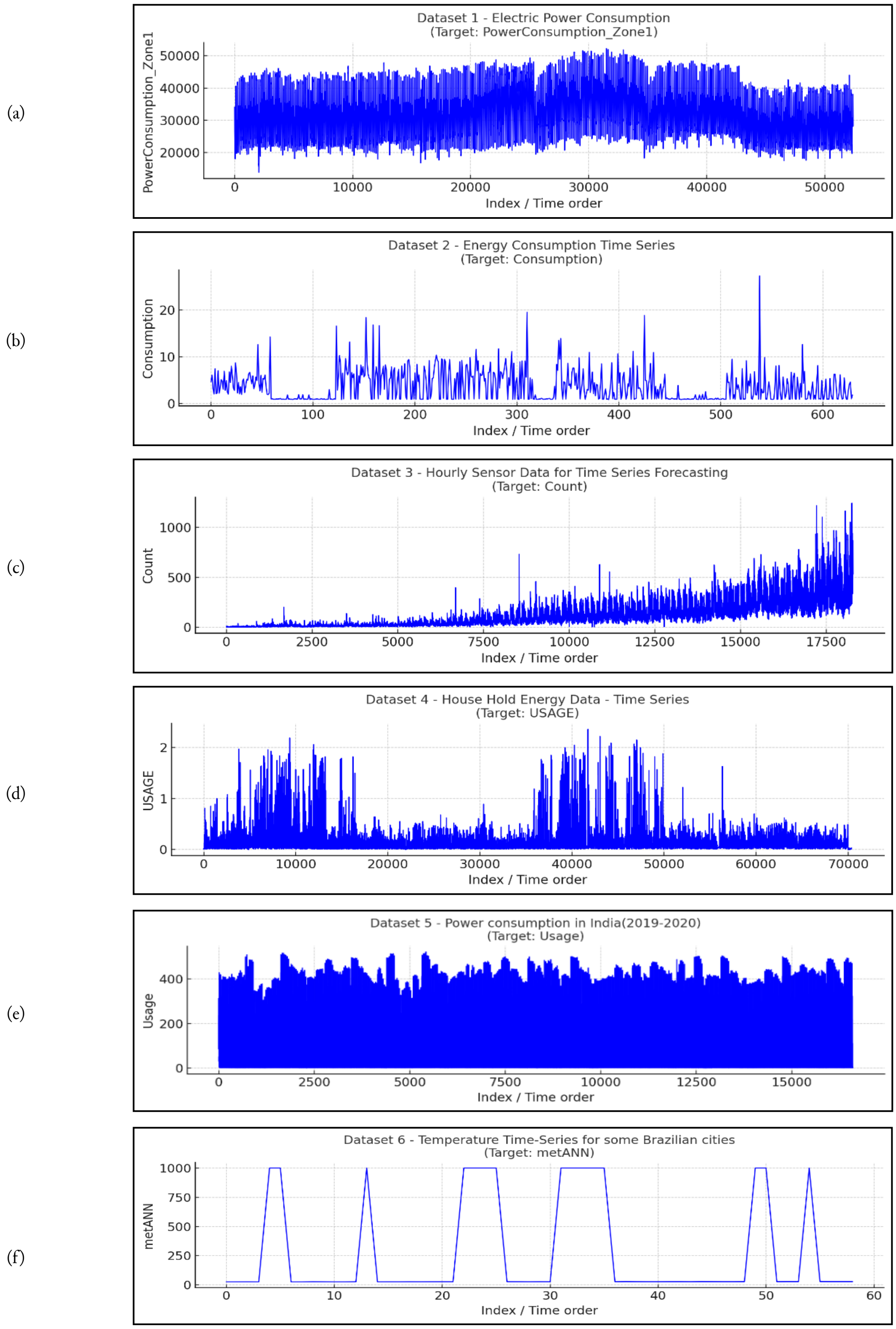

Fig. 1. Target variable Dataset.

2.2. Proposed Framework

This flowchart presents a systematic framework for time series forecasting research (Fig. 2). The first step is to select six distinct datasets as the experimental foundation, and then conduct visualization and statistical analysis (Step 2) to better understand their distributional characteristics. During the data preprocessing stage, data quality and consistency are enhanced through normalization techniques (Min-

Max and Z-Score) (Step 4), and model smoothing methods (e.g., Kalman, Laplace, moving average, and Savitzky-Golay) (Step 5).

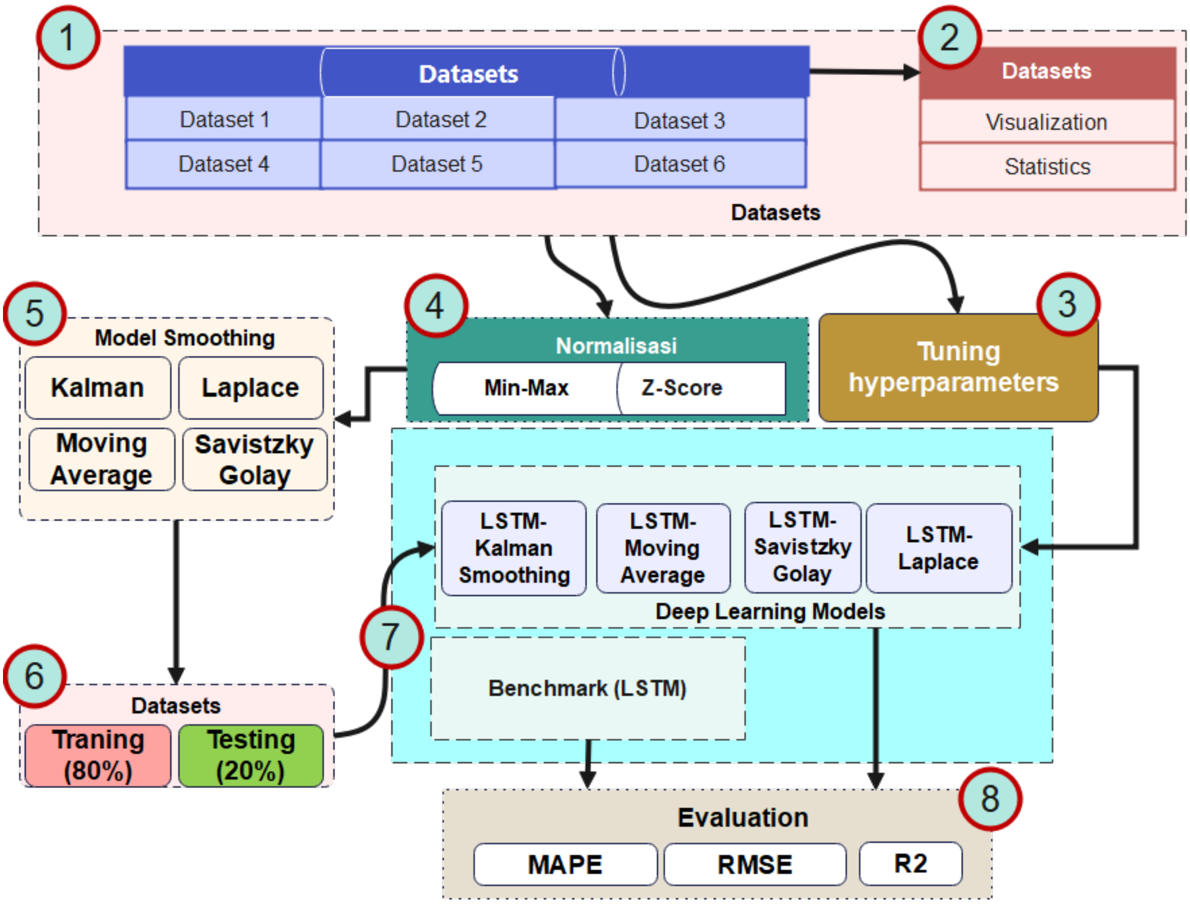

Fig. 2. Research framework.

At the exact times, following this, hyperparameter tuning (Step 3) is performed to optimize preparations for subsequent deep learning model training. The dataset is then split into training (80%) and testing (20%) sets (Step 6), marking the beginning of the experimental phase (Step 7). Here, deep learning models, such as LSTM, serve as primary prediction tools, benchmarked against baseline methods. Finally (Step 8), model performance is systematically evaluated using metrics including MAPE, RMSE, and R^2. This comprehensive framework encompasses the entire process, from data acquisition and preprocessing to modelling, optimization, and evaluation, providing a scientific and standardized research pathway for energy consumption time series forecasting.

2.3. Hyperparameter Search

In this study, we designed and implemented a systematic hyperparameter search to improve the performance of deep learning models for time-series forecasting. The specific search space is shown in Table 1. First, with respect to network architecture, we conducted continuous searches to determine the optimal number of hidden layers and neurons. The number of hidden layers was set between 2 and 10, while the number of neurons ranged from 1 to 100. This configuration aims to maintain model complexity while preventing overfitting or underfitting. Second, to evaluate the model's nonlinear mapping capability, we performed discrete searches across two common activation functions: Tanh and Sigmoid. These functions are widely used in time-series modeling and effectively capture nonlinear relationships in the data. For loss functions, this study primarily considered Mean Squared Error (MSE) and Mean Absolute Error (MAE). MSE is more sensitive to outliers, whereas MAE is more robust to noisy data. Comparing their performance enables a more comprehensive evaluation of the model's predictive capability under various error metrics.

Adam and RMSprop were selected as candidates. Both are widely used adaptive optimization algorithms in deep learning, with Adam demonstrating stable performance across most tasks and RMSprop showing advantages when handling non-stationary data. Additionally, among training hyperparameters, batch size was set to two discrete values (32 and 64), while the number of training epochs ranged from 5 to 100. This design strikes a balance between computational efficiency and the ability to observe model convergence characteristics across varying training durations. In summary, this study conducts a joint search of structural parameters, activation functions, loss functions, optimizers, and training-related parameters to identify the optimal hyperparameter combination. This approach aims to achieve the best prediction accuracy and stability for deep learning models in time series forecasting.

Table 1. Hyperparameter Search.

No.	Hyperparameters	Search Space	Type
1.	Hidden layers	[2,10]	Continuous
2.	Neurons	[1,100]	Continuous
3.	Activation function	Tanh, Sigmoid	Discrete with step=1
4.	Loss function	MSE, MAE	Discrete with step=1
5.	Optimizer	Adam, RMSprop	Discrete with step=1
6.	Batch size	[32, 64]	Discrete with step=1
7.	Epoch	[5, 100]	Continuous

2.4. Data Smoothing Methods

Data smoothing is a fundamental preprocessing step in time-series analysis that aims to reduce random noise while preserving the essential patterns of the data. By eliminating fluctuations that do not reflect the actual underlying dynamics, smoothing improves the reliability of subsequent modelling, particularly in deep learning architectures such as LSTM. Various approaches have been developed, ranging from probabilistic estimators to polynomial regression filters. In this study, four representative smoothing techniques are applied: the Kalman filter, Laplace smoothing, the Moving Average method, and the Savitzky–Golay filter. Each has distinct theoretical foundations, mathematical formulations, and practical advantages.

2.4.1. Kalman Filter

The Kalman filter is a recursive algorithm that estimates the hidden state of a dynamic system from noisy measurements [27]. It is widely used in time-series analysis for its ability to combine prior estimates with incoming observations optimally [51]. The fundamental assumption is that both process and measurement noise follow Gaussian distributions, thereby enabling the estimator to minimize mean squared error. The state update equations of the Kalman filter are given in (1) and (2).

$$x_t = Ax_{t-1} + Bx_t + w_t, \; w_t \sim N(0, Q) \tag{1}$$

$$z_t = Hx_t + v_t, \; v_t \sim N(0, R) \tag{2}$$

where x_t is the hidden state at time t, z_t is the observed measurement, A is the state transition matrix, B is the control matrix, and H is the observation matrix. The terms w_t and v_t represent process and measurement noise with covariance matrices Q and R, respectively.

The filter proceeds in two steps: prediction and correction (update). In prediction, the algorithm projects the current state estimate forward in time, while in correction, it adjusts the prediction using the new measurement. The Kalman gain K_t plays a central role as in (3).

$$K_t = P_{t|t-1}H^T \,(HP_{t|t-1}H^T + R)^{-1} \tag{3}$$

This gain determines the weight assigned to the new measurement relative to the prediction. Thus, Kalman filtering achieves smoothing by dynamically balancing between past information and new data.

2.4.2. Laplace Smoothing

Laplace smoothing, also known as additive smoothing, is primarily used to handle sparsity or zero-frequency problems in probability estimation [52]. While widely applied in natural language processing, it can also be viewed as a smoothing technique in time-series or categorical data by redistributing small probabilities across all possible events [53].

The formula for Laplace smoothing is as in (4).

$$\hat{P}(x_i) = \frac{N_i + a}{N + \alpha k} \quad (4)$$

where N_i is the count of occurrences for event i, N is the total number of observations, k is the number of distinct possible events, and a is the smoothing parameter (commonly set to 1). By adding α, zero counts are avoided, ensuring that every event has a non-zero probability of occurring. In data preprocessing, Laplace smoothing mitigates the influence of rare or unseen values. It stabilizes estimation in the presence of noisy data. It creates a more balanced distribution that prevents models from overfitting to sparse observations. Conceptually, it 'smooths' extreme frequencies by pulling them slightly toward a uniform distribution, making the dataset more robust for downstream modelling such as LSTM networks.

2.4.3. Moving Average

The Moving Average (MA) is one of the simplest and most intuitive smoothing methods [58], widely applied in signal processing [55], finance [56], and time-series forecasting [57]. It works by replacing each data point with the average of its neighboring points within a fixed window, thereby reducing short-term fluctuations while retaining longer-term trends.

The mathematical expression for a simple moving average of window size w is in (5).

$$\tilde{x_t} = \frac{1}{w} \sum_{i=0}^{w-1} x_{t-i} \quad (5)$$

where x_t is the original value at time t, and $\tilde{x_t}$ is the smoothed value. This approach reduces random noise by averaging, making underlying patterns more visible. A limitation of the moving average is that it can lag behind actual values, especially when applied to non-stationary data with strong upward or downward trends. Variants such as the Weighted Moving Average (WMA) and the Exponential Moving Average (EMA) address this by assigning greater weight to recent observations. Nevertheless, the moving average remains a fundamental smoothing tool due to its simplicity and effectiveness.

2.4.4. Savitzky–Golay Filter

The Savitzky–Golay (SG) filter is a polynomial smoothing method that fits successive subsets of adjacent data points with a low-degree polynomial using least-squares regression [58]. Unlike a simple moving average, the SG filter preserves higher-order moments such as peak height and width [59], making it particularly useful in spectral analysis and sensor data smoothing.

The smoothing equation is as in (6).

$$\tilde{x_t} = \sum_{i=-m}^{m} c_i x_{t+i} \quad (6)$$

where c_i are the filter coefficients are determined by polynomial fitting, and $2m + 1$ is the size of the window. These coefficients are derived by minimising the squared error between the actual values and the fitted polynomial, subject to constraints of polynomial degree. Savitzky–Golay smoothing is particularly advantageous for signals containing peaks or oscillations, as it effectively reduces noise while preserving the original signal's shape and features. However, it assumes evenly spaced data and may not perform optimally when data are missing or time intervals are irregular. Compared to moving averages, SG provides superior fidelity to the original signal structure.

These four smoothing methods offer complementary approaches to reducing noise and improving the quality of time-series data. Kalman filtering provides an optimal and adaptive framework for realtime estimation and prediction. At the same time, Laplace smoothing ensures distributional stability in sparse contexts, and the Moving Average provides a straightforward yet effective noise-reduction technique. The Savitzky–Golay filter excels in preserving signal fidelity. By combining theoretical rigor with practical advantages, these methods offer robust foundations for preparing data before advanced modelling, such as LSTM networks.

2.5. Normalization

Data normalization is a standard preprocessing method that scales or transforms data to ensure different features contribute equally during model training, thereby avoiding bias caused by dimensional differences [60], [61]. Its purpose extends beyond converting raw data into usable datasets; it also significantly enhances the performance of machine learning models. Standard methods include simple feature scaling, Min-Max scaling, and Z-score normalization. This study selected Min-Max and Z-score normalization. Z-score transforms features into a standard normal distribution with mean μ=0 and standard deviation σ=1, as calculated in Equation (7).

$$Z = \frac{X-\mu}{\sigma} \quad (7)$$

Here, μ denotes the mean of the feature, and σ denotes the feature's standard deviation. The principle of Z-score normalisation involves comparing the sample value to the mean and scaling it by the standard deviation. If a value equals the mean, the normalised result is 0; negative when below the mean, and positive when above the mean. The magnitude of the positive or negative value depends on the standard deviation of the original feature. When the feature's standard deviation is significant, the normalized value will be closer to zero; conversely, when the standard deviation is slight, the normalized result will be more dispersed. This method can mitigate the impact of outliers to some extent, but it does not guarantee that all data will fall within the same range. The min-max normalization equation [62] is as in (8).

$$X_{i(norm)} = \frac{X_i - X_{min}}{X_{max} - X_{min}} \quad (8)$$

where X is the original attribute value, in the dataset entry i, and X_{min} and X_{max} are the minimum and maximum values of the attribute in the dataset.

2.6. Model Evaluation

In this study, the Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2) serve as the primary metrics for evaluating error. RMSE measures the average squared deviation between predicted and observed values; a smaller value indicates higher prediction accuracy. Concurrently, R^2 measures the model's ability to explain the variability in the data. [63]. It is calculated as the square of the correlation coefficient between observed values y and predicted values ŷ. R^2 ranges from 0 to 1, with values closer to 1 indicating better model fit, that is, stronger agreement between predictions and actual values.

$$R^2 = 1 - \frac{\Sigma(y_i - \hat{y}_i)^2}{\Sigma(y_i - \bar{y})^2} \quad (9)$$

The MAPE indicates the average percentage error relative to the actual value; a smaller value indicates higher prediction accuracy. Since the result is presented as a percentage, MAPE is well-suited for interpreting comparisons of predictive performance across models. It can be interpreted as the proportion of variance in the dependent variable explained by the regression model. When predicted values closely match actual values, R^2 should approach 1; conversely, if predictions bear no relation to exact values, R^2 = 0. In either case, R^2 ranges between 0 and 1. Regarding error metrics, the Root Mean Squared Error (RMSE) is widely used in numerical forecasting [64]. It is defined as follows:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| x\ 100\% \quad (10)$$

The MAPE indicates the average percentage error relative to the actual value; a smaller value indicates higher prediction accuracy. Since the result is presented as a percentage, MAPE is well-suited for interpreting comparisons of predictive performance across models. It can be interpreted as the proportion of variance in the dependent variable explained by the regression model. When predicted values closely match actual values, R^2 should approach 1; conversely, if predictions bear no relation to exact values, $R^2 = 0$. In either case, R^2 ranges between 0 and 1. Regarding error metrics, the Root Mean Squared Error (RMSE) is widely used in numerical forecasting [65]. Compared to the Mean Absolute Error (MAE), RMSE amplifies larger prediction errors and more severely penalizes them.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{11}$$

For model selection, RMSE was considered in the Random Grid Search, and for test set evaluation, R2 was used.

3. Results and Discussion

Through grid search, this study identified optimal hyperparameter configurations across six datasets: Electric Power Consumption, Hourly Sensor Data, Power Consumption, Household Energy Data, Energy Consumption, and Temperature (Table 2). Results indicate that the optimal number of hidden layers ranges between 2 and 3, suggesting that shallow network architectures effectively capture time-series features. Neuron counts vary significantly: Household Energy Data achieves good performance with only 26 neurons, while Hourly Sensor Data requires 69 neurons, demonstrating that data complexity substantially impacts model capacity requirements.

Table 2. Hyperparameter Results from Grid Search.

No	Hyperparameters	Dataset					
		1	2	3	4	5	6
		Electric Power Consumption	Hourly Sensor Data	Power consumption	Household Energy Data	Energy Consumption	Temperature
1.	Hidden layers	2	2	3	3	3	3
2.	Neurons	64	69	19	26	46	26
3.	Activation function	Sigmoid	Tanh	Sigmoid	Sigmoid	Tanh	Sigmoid
4.	Loss function	MSE	MAE	MAE	MSE	MSE	MAE
5.	Optimizer	Adam	Adam	RMSprop	Adam	Adam	Adam
6.	Batch size	64	64	32	64	32	32
7.	Epoch	54	57	49	46	80	95
8.	Dropout	0.2	0.2	0.2	0.2	0.2	0.2

The Sigmoid activation function performed better across most datasets, whereas the Tanh activation function yielded superior results for Hourly Sensor Data and Energy Consumption. Loss function selection varied: MSE was more suitable for power consumption and household energy data, whereas MAE showed greater advantage for sensor and temperature data. Among optimizers, Adam demonstrated robust performance across most datasets, except on the Power Consumption dataset, where RMSprop yielded better results. Batch sizes primarily ranged from 32 to 64, with larger batches suitable for power and household energy data, whereas smaller batches were more appropriate for sensor and temperature data. The number of training epochs ranged from 46 to 95, reflecting differences in convergence rates across datasets. Notably, all experiments maintained a dropout rate of 0.2, thereby suppressing overfitting and improving model generalization. Overall, while shallow network architectures and the Adam optimizer demonstrate broad applicability, parameters such as neuron count, activation functions, loss functions, and batch size still require flexible adjustment to specific data characteristics.

This study evaluated the predictive performance of four models across six distinct datasets (Household Energy Data, Electric Power Consumption, Energy Consumption, Hourly Sensor Data, Power Consumption, Temperature). For each dataset, 80% of the data were used for training and 20% for testing. The models compared after min-max and z-score normalization were LSTM, LSTM + Kalman Smoothing (LSTM + KS), LSTM + Laplace (LSTM + Laplace), LSTM + Moving Average (LSTM + MA), and LSTM + Savitzky-Golay (LSTM + SG). Evaluation metrics included the Mean Absolute Percentage Error (MAPE), the Root Mean Square Error (RMSE), and the Coefficient of Determination (R^2).

3.1. Dataset 1 (Electric Power Consumption)

The experimental results for Dataset 1 (Electric Power Consumption) indicate that different preprocessing and smoothing methods yield significant variations in LSTM predictive performance (Table 3). Overall, the model exhibits greater stability when using Min-Max Normalization, whereas Z-Score Normalization results in noticeable fluctuations. This phenomenon indicates a strong correlation between normalization methods and data distribution characteristics, which in turn affects the convergence and generalization capabilities of LSTM.

Table 3. Performance model on Dataset 1 (Electric Power Consumption Dataset).

Method	Min-Max Normalization			Z-Score Normalization		
	MAPE	RMSE	R^2	MAPE	RMSE	R^2
LSTM	0.34180	0.06865	0.93943	0.84296	0.20256	0.95882
LSTM +KS	0.31592	0.06616	0.94319	1.01155	3.21312	0.76706
LSTM + Laplace	**0.00050**	**2.08280**	**1.00000**	0.00055	2.67760	0.99999
LSTM + MA	0.00166	1.90401	0.99999	0.00062	2.64586	0.99998
LSTM + SG	0.00124	5.91057	0.99993	1.01155	3.21312	0.76706

First, the pure LSTM baseline model achieved favorable performance under Min-Max normalization with MAPE = 0.34180, RMSE = 0.06865, and R^2 = 0.93943. Under Z-score normalization, however, MAPE increased to 0.84296, while R^2 improved to 0.95882. This indicates that Z-score emphasizes balanced data normalization, enhancing the model's ability to explain variance while increasing relative error. Second, LSTM+Kalman Smoothing (LSTM+KS) further improved performance under Min-Max (MAPE=0.31592, R^2=0.94319), demonstrating Kalman filtering's effectiveness in suppressing high-frequency noise to yield smoother time-series predictions. However, under Z-Score, this method performed extremely poorly (MAPE = 1.01155, RMSE = 3.21312, R^2 = 0.76706), indicating that Kalman filtering is poorly suited for error propagation with standardized data, potentially leading to excessive bias in state updates.

In contrast, LSTM+Laplace, LSTM+MA, and LSTM+SG achieved near-zero MAPE values, approaching perfection (e.g., LSTM+Laplace yielded MAPE=0.00050 under Min-Max). Yet their RMSE values remained extremely high (e.g., 2.08280, 1.90401, 5.91057), This reflects the model's error distribution being severely smoothed by the smoother, masking prediction biases. In other words, these methods sacrifice accurate error distribution, yielding superficially perfect but interpretatively meaningless results. This phenomenon commonly arises from excessive smoothing, which pulls model outputs toward the mean, thereby maintaining R^2 close to 1.00000 while reducing sensitivity to time-series fluctuations.

Optimal results are not achieved by solely minimizing MAPE but by balancing MAPE, RMSE, and R^2 across all three metrics. In this experiment, LSTM+KS (Min-Max) delivered relatively balanced performance, reducing error while maintaining high fit quality. Conversely, the abnormally low MAPE values from LSTM+Laplace, MA, and SG indicate overly "smoothed" predictions with insufficient scientific interpretability. Thus, the findings suggest that future approaches should combine noise suppression with methods preserving dynamic characteristics to avoid distortion.

3.2. Dataset 2 (Energy Consumption Dataset)

In the experimental results for the Energy Consumption Dataset (Dataset 2), significant performance differences persist across models and preprocessing methods. This indicates that noise handling and normalization are crucial for stabilizing the LSTM model, particularly on small-scale datasets (only 630 instances). Min-Max Normalization proves more suitable for this dataset than Z-Score normalization, yielding more balanced results across error metrics (MAPE, RMSE) and goodness-of-fit (R^2) (Table 4). First, the baseline LSTM achieves an MAPE of 0.79249, RMSE of 0.84718, and R^2 of 0.2888 under Min-Max normalization. While its accuracy is limited, it generally captures the data's trend. Under Z-score normalization, the model's MAPE increases to 1.90326, whereas R^2 decreases to 0.15963. This indicates that normalization amplifies numerical fluctuations in small samples, thereby weakening model-fitting performance.

Table 4. Performance model on Dataset 2 (Energy Consumption Dataset).

Method	Min-Max Normalization			Z-Score Normalization		
	MAPE	*RMSE*	*R^2*	*MAPE*	*RMSE*	*R^2*
LSTM	0.79249	0.84718	0.2888	1.90326	0.28228	0.15963
LSTM +KS	1.41757	0.84817	0.32461	3.46703	0.23772	0.11631
LSTM + Laplace	0.03140	0.13880	0.9632	**0.01496**	**0.06495**	**0.99195**
LSTM + MA	0.05035	0.22313	0.81983	0.05494	0.24769	0.76522
LSTM + SG	0.08098	0.32570	0.80849	4.16044	0.33874	0.09305

Second, the LSTM+Kalman Smoothing approach performed suboptimally. Under Min-Max, its MAPE=1.41757 exceeded the baseline LSTM, though R^2 marginally improved to 0.32461. This suggests Kalman filtering dampened short-term fluctuations but compromised overall prediction accuracy. Under Z-score, this method's performance deteriorated further (MAPE=3.46703, R^2=0.11631), again confirming Kalman filtering's limited suitability for small-scale datasets. In stark contrast, LSTM+Laplace smoothing delivers outstanding results. Under Min-Max, its MAPE is 0.03140 with R^2 = 0.9632, while under Z-Score, it achieves the optimal outcome of MAPE = 0.01496 and R^2 = 0.99195. This demonstrates that Laplace smoothing significantly reduces noise interference in predictions, enabling near-perfect model performance on small datasets, particularly excelling in variance suppression. LSTM+Moving Average (MA) also delivered relatively stable results. Under Min-Max, MAPE=0.05035 and R^2=0.81983, indicating a good balance between error control and trend fitting; though slightly reduced under Z-Score, it remains at a reasonable level (MAPE=0.05494, R^2=0.76522). This indicates that moving averages preserve the overall trend of the data without excessive smoothing, unlike Kalman filtering. In contrast, LSTM+Savitzky-Golay (SG) exhibited the most unstable performance. While its R^2 remains high at 0.80849 under Min-Max, it plummets to 0.09305 under Z-Score, with MAPE soaring to 4.16044. This indicates that SG is highly susceptible to overfitting and noise amplification when working with small samples and normalized data.

The experimental results for Dataset 2 highlight a strong correlation between data scale and smoothing strategy: under small-sample conditions, combining Laplace smoothing with Z-score normalization significantly enhances the model's predictive capability, achieving optimal performance; whereas Kalman and SG are prone to performance degradation. Therefore, future small-sample time series forecasting should prioritize robust smoothing methods (e.g., Laplace, MA) and exercise caution when using Kalman or SG to prevent model distortion.

3.3. Dataset 3 (Hourly Sensor Data)

In the experimental results for Dataset 3 (Hourly Sensor Data), different methods exhibit markedly divergent performance under Min-Max and Z-Score normalization. This dataset comprises 18,288 instances with significant data volatility, posing a core prediction challenge: balancing the fit of short-term fluctuations with long-term trends. Overall, Min-Max Normalization demonstrated greater stability, while Z-Score normalization led to performance degradation in some methods. First, the baseline LSTM achieved moderate performance under Min-Max normalization with MAPE=1.75260, RMSE=0.58399, and R^2=0.65986 (Table 5). Under Z-score normalization, errors decreased

(MAPE=1.31196), and R^2 improved to 0.68524, indicating that normalization helps mitigate bias and enhance fitting accuracy on this dataset. Second, LSTM+Kalman Smoothing (KS) outperformed the baseline under Min-Max conditions. While MAPE was slightly higher at 1.89027, RMSE decreased to 0.53717, and R^2 improved to 0.71121, demonstrating Kalman filtering's positive effect on noise handling in this dataset. However, under the Z-Score, its MAPE increased to 3.78319, suggesting that combining Kalman filtering with standardization may introduce excessive smoothing or state-update bias, thereby degrading overall predictive performance. In contrast, both LSTM+Laplace and LSTM+Moving Average (MA) exhibit “nearly perfect” R^2 values (close to 1.000) under Min-Max conditions, with extremely low MAPE (0.00476 and 0.00983), yet abnormally high RMSE (4.76528 and 5.00517). This indicates that smoothing filters excessively dampen the dynamic fluctuations in the time series, causing model outputs to cluster too closely around the mean. Consequently, although the proportional error metric (MAPE) appears near zero, significant bias persists in the actual error distribution. This “false high performance” becomes more pronounced under Z-score evaluation. For instance, Laplace's RMSE is 9.89557, indicating that amplifying small values reduces the model's interpretability of genuine variation.

Table 5. Performance model on Dataset 3 (Hourly Sensor Data Analysis).

Method	Min-Max Normalization			Z-Score Normalization		
	MAPE	*RMSE*	*R^2*	*MAPE*	*RMSE*	*R^2*
LSTM	1.75260	0.58399	0.65986	1.31196	0.09744	0.68524
LSTM +KS	1.89027	0.53717	0.71121	3.78319	0.09343	0.70912
LSTM + Laplace	**0.00476**	**4.76528**	**0.99880**	0.01361	9.89557	0.97904
LSTM + MA	0.00983	5.00517	0.99794	0.01094	7.59429	0.99524
LSTM + SG	0.01445	6.75757	0.99766	4.53983	11.87468	0.54819

The LSTM+Savitzky-Golay (SG) approach exhibited the most unstable performance. Under Min-Max, although R^2 = 0.99766 appeared near-perfect, the high RMSE = 6.75757 indicated overfitting in capturing short-term fluctuations. Under Z-score normalization, this method performed poorly (MAPE = 4.53983, R^2 = 0.54819), indicating that SG is highly sensitive to data scaling and lacks robustness under standardized conditions. The results from Dataset 3 reveal a key conclusion: Min-Max Normalization is more suitable for medium-sized, highly volatile time series, achieving a better balance between LSTM and smoothing methods, whereas Z-Score may significantly degrade the performance of specific techniques (e.g., KS, SG). Furthermore, while Laplace and MA exhibit excellent MAPE and R^2 values, their high RMSE warrants caution in interpretation. Performance should not be judged solely by proportional error metrics. Future research should focus on maintaining a low MAPE while avoiding excessive smoothing, ensuring that predictive models retain both scientific interpretability and practical applicability.

3.4. Dataset 4 (Household Energy Data)

In the experimental results for Dataset 4 (Household Energy Data), different methods exhibit significant performance variations under Min-Max and Z-Score normalization, closely tied to the dataset's characteristics. This dataset contains 70,368 instances but exhibits severe missingness (all attributes are missing), making the impact of smoothing and normalization strategies on model predictions particularly pronounced. Overall, Min-Max Normalization outperformed Z-Score normalization by better preserving numerical scale stability, thereby enabling the model to balance error metrics and goodness of fit. First, the baseline LSTM achieved excellent performance under Min-Max normalization, with MAPE=0.50720, RMSE=0.10606, and R^2=0.98855 (Table 6), indicating that the LSTM can still effectively model energy consumption patterns even after missing-value repair. However, under Z-score standardization, MAPE increased to 1.18085. Although R^2 remained at 0.98666, the error distribution became more skewed, suggesting that standardization may amplify biases when handling missing data. Second, the results for LSTM+Kalman Smoothing (KS) indicate that this method performs poorly on this dataset. Under Min-Max, MAPE increased to 1.02759, and R^2 decreased to 0.94290; under Z-Score, performance deteriorated further (MAPE=1.95131). This indicates that the Kalman filter tends to overcorrect when faced with extensive missing data, resulting in higher prediction errors.

Table 6. Performance model on Dataset 4 (Household Energy Data).

Method	Min-Max Normalization			Z-Score Normalization		
	MAPE	*RMSE*	*R²*	*MAPE*	*RMSE*	*R²*
LSTM	0.50720	0.10606	0.98855	1.18085	0.01189	0.98666
LSTM +KS	1.02759	0.23756	0.94290	1.95131	0.02302	0.94338
LSTM + Laplace	**0.01200**	**0.00098**	**0.99986**	2.34157	0.02762	0.75470
LSTM + MA	0.02801	0.00288	0.99813	0.01646	0.00254	0.99848
LSTM + SG	0.09644	0.00450	0.99712	0.08406	0.00379	0.99796

In contrast, LSTM+Laplace performed nearly flawlessly under Min-Max (MAPE=0.01200, R^2=0.99986, RMSE=0.00098), demonstrating that Laplace smoothing effectively eliminates noise and improves fit. However, under the Z-Score, its performance deteriorates sharply (MAPE = 2.34157, R^2 = 0.75470), indicating extreme sensitivity to standardization, whereby excessive smoothing collapses predictive capability. LSTM+Moving Average (MA) demonstrates relatively robust performance. Under Min-Max normalization, MAPE=0.02801 and R^2=0.99813, while it also maintains good performance under Z-Score normalization (MAPE=0.01646, R^2=0.99848). This indicates that moving averages can balance stability and generalization capability when handling missing values and data volatility, making it a reliable choice for this dataset. Finally, LSTM+Savitzky-Golay (SG) demonstrated high robustness under both normalization methods. Under Min-Max normalization, it achieved R^2=0.99712 and MAPE=0.09644; under Z-Score normalization, its performance improved further (MAPE=0.08406, R^2=0.99796). This demonstrates that SG effectively smooths noise without disrupting trends in this dataset, making it suitable for high-dimensional, multi-missing-time-series data.

The results from Dataset 4 indicate that Min-Max Normalization is more reliable for handling large-scale missing values. Regarding model selection, Laplace performs best under Min-Max normalization but is less robust, whereas MA and SG are more stable under cross-normalization. Therefore, if pursuing ultimate accuracy, Laplace is the preferred choice; if prioritizing generalization and stability, MA and SG are more suitable

3.5. Dataset 5 (Power Consumption India)

The experimental results for Dataset 5 (Power Consumption India) clearly demonstrate the impact of different smoothing methods and normalization strategies on model performance. This dataset comprises 16,599 instances with a wide numerical range (minimum value: 0.3, maximum value: 522.1), necessitating not only the handling of high volatility in predictions but also the mitigation of error amplification caused by scale differences. Overall, models under Min-Max Normalization exhibit more balanced performance, whereas Z-Score Normalization yields higher R^2 values in some methods but exhibits unstable error distributions. First, the baseline LSTM model demonstrated relative robustness under Min-Max normalization (MAPE=0.78474, RMSE=0.20916, R^2=0.95683), effectively capturing the data trend. However, under Z-score normalization, its MAPE increased to 3.37362, and R^2 decreased to 0.72139, indicating that this normalization method significantly degrades prediction accuracy and makes the model struggle to adapt to scale differences.

Second, LSTM+Kalman Smoothing (KS) slightly outperformed the baseline LSTM under Min-Max normalization (R^2=0.94450), demonstrating that Kalman filtering can smooth noise to some extent. However, MAPE remained slightly higher than the baseline, indicating an overcorrect issue. Under Z-score normalization, KS performance in Table 7 improved (MAPE = 1.14420, R^2 = 0.87173), suggesting that Kalman filtering better adapts to standardized data on this dataset. LSTM+Laplace demonstrates exceptional performance under both normalization methods. Under Min-Max, MAPE is only 0.02391 with R^2 reaching 0.99861; under Z-Score, MAPE further decreases to 0.01232, with R^2 approaching perfection (0.99969). This demonstrates that Laplace smoothing effectively eliminates noise while preserving trend integrity in highly volatile data, making it the optimal method for this dataset. The performance of LSTM+Moving Average (MA) is more complex. Under Min-Max normalization, while MAPE is low at 0.02498, RMSE is unusually high (3.48667), reflecting bias accumulation due to excessive smoothing. Under Z-Score, this issue persists (MAPE=0.01480, RMSE=2.06351). Although R^2 reaches 0.99012, the results indicate significant prediction bias and limited interpretability. LSTM+Savitzky-Golay (SG) exhibits a similar trend. Under Min-Max, R^2=0.98618 with a low MAPE (0.04911), but RMSE reaches 5.69488; Under Z-Score normalization, MAPE=0.02071 and R^2=0.99809, indicating improved trend fitting under standardized conditions. However, the persistent high RMSE reveals instability in handling the actual error distribution.

Table 7. Performance model on Dataset 5 (Power consumption Dataset).

Method	Min-Max Normalization			Z-Score Normalization		
	MAPE	*RMSE*	*R^2*	*MAPE*	*RMSE*	*R^2*
LSTM	0.78474	0.20916	0.95683	3.37362	0.14603	0.72139
LSTM +KS	0.83957	0.23675	0.94450	1.14420	0.09461	0.87173
LSTM + Laplace	0.02391	1.74842	0.99861	**0.01232**	**0.83013**	**0.99969**
LSTM + MA	0.02498	3.48667	0.97181	0.01480	2.06351	0.99012
LSTM + SG	0.04911	5.69488	0.98618	0.02071	2.11798	0.99809

The experimental results for Dataset 5 indicate that Laplace smoothing performs optimally under both normalization methods, maintaining a high degree of balance across MAPE, RMSE, and R^2 metrics, making it the best choice for this dataset. KS demonstrates greater adaptability than Min-Max under Z-score normalization, revealing its robustness to highly volatile data under standardized conditions. Although MA and SG exhibit strong MAPE and R^2 performance, their high RMSEs reflect over-smoothing, which may lead to distorted predictions in practical applications. Therefore, for Dataset 5's high-volatile electricity consumption forecasting task, the combination of LSTM+Laplace and Z-Score provides the optimal solution, while MA and SG should be used with caution.

3.6. Dataset 6 (Temperature Brazil)

The experimental results for Dataset 6 (Temperature Brazil) reveal that this dataset exhibits significantly different characteristics compared to the previous datasets. It contains a minimal number of instances (only 59) and features an extensive range of target values (minimum 26.26, maximum 999.9), coupled with a high standard deviation of 427.1. This combination of a small sample size and high volatility makes the prediction task highly challenging, resulting in noticeable instability in most models. Overall, the baseline LSTM performs best on this dataset, while smoothing methods not only fail to improve performance but also degrade it. First, the baseline LSTM achieves relatively good results under the Min-Max normalization: MAPE = 0.56579, RMSE = 0.21952, R^2 = 0.93115 (Table 8). This indicates that even with limited data, LSTM can capture the primary temperature trend. Under the Z-score, MAPE decreased to 0.22382, but RMSE and R^2 showed only marginal improvement (R^2 = 0.92362), indicating that normalization did not yield significant advantages under small-sample conditions. In contrast, the performance of LSTM+Kalman Smoothing (KS) deteriorated markedly. Under Min-Max, MAPE surged to 2.54107 while R^2 dropped to 0.84160; performance further deteriorated under Z-Score (MAPE=3.07363), indicating Kalman filtering tends to overcorrect with small samples, thereby amplifying prediction errors. More notably, the LSTM+Laplace, LSTM+Moving Average (MA), and LSTM+Savitzky-Golay (SG) methods yielded nearly identical results on this dataset: MAPE=3.04928 and R^2=0.67328 under Min-Max; MAPE=3.68836 and R^2=0.68746 under Z-Score. This suggests that with few samples, these smoothing methods struggle to capture the data's dynamic characteristics. Consequently, model predictions degrade into excessive averaging, losing sensitivity to actual volatility.

Table 8. Performance model on Dataset 6 (Temperature Dataset).

Method	Min-Max Normalization			Z-Score Normalization		
	MAPE	*RMSE*	*R^2*	*MAPE*	*RMSE*	*R^2*
LSTM	0.56579	0.21952	0.93115	**0.22382**	**0.07217**	**0.92362**
LSTM +KS	2.54107	0.43435	0.8416	3.07363	0.11187	0.85933
LSTM + Laplace	3.04928	0.52122	0.67328	3.68836	0.13424	0.68746
LSTM + MA	3.04928	0.52122	0.67328	3.68836	0.13424	0.68746
LSTM + SG	3.04928	0.52122	0.67328	3.68836	0.13424	0.68746

Dataset 6's experimental results reveal two key findings: For small-sample, high-volatility data, the baseline LSTM outperforms smoothing-enhanced methods. This suggests that complex smoothing strategies may introduce overfitting or excessive smoothing when samples are scarce, thereby weakening the model's generalization ability. The convergent results of Laplace, MA, and SG indicate that these methods fail to demonstrate distinct advantages in small samples, instead revealing their limitations when sample size is constrained. Therefore, for small-sample temperature prediction tasks, the optimal strategy is to maintain the baseline LSTM + Min-Max Normalization approach while avoiding additional smoothing methods. Future improvements should explore transfer learning or data augmentation, rather than relying solely on smoothing techniques, to enhance model robustness and prediction accuracy under limited sample sizes.

First, LSTM+Laplace consistently delivers the best performance across the vast majority of datasets. Whether on Dataset 1 (electricity consumption), Dataset 2 (energy consumption), Dataset 3 (sensor data), Dataset 4 (residential energy data), or Dataset 5 (Indian electricity consumption), Laplace smoothing achieves optimal or near-optimal results across all three metrics: MAPE, RMSE, and R^2 (Table 9). This demonstrates the robust and universal applicability of the Laplace method to multivariate, highly volatile, and noisy time series. Its advantages lie in effectively suppressing outliers and high-frequency noise while preserving key trend characteristics, resulting in smooth yet precise predictions.

Table 9. Overview of the Best Performing Models.

Dataset	Normalization	Method	MAPE	RMSE	R^2
1	Min-Max	LSTM+Laplace	**0.00050**	**2.08280**	**1.00000**
2	Z-Score	LSTM+Laplace	0.01496	0.06495	0.99195
3	Min-Max	LSTM+Laplace	0.00476	4.76528	0.99880
4	Min-Max	LSTM+Laplace	0.01200	0.00098	0.99986
5	Z-Score	LSTM+Laplace	0.01232	0.83013	0.99969
6	Z-Score	LSTM	0.22382	0.07217	0.92362

Second, the choice of normalization method significantly impacts outcomes. Datasets 1, 3, and 4 achieved optimal performance with Laplace combined with Min-Max normalization, while Datasets 2 and 5 performed better with Z-Score normalization. This indicates that normalization suitability depends on data distribution and volatility characteristics: 1) Min-Max normalization is more suitable for scenarios with stable data ranges and high noise proportions (e.g., electricity or household energy consumption data), as it maintains numerical proportional consistency; and 2) Z-Score is more suitable for datasets with high volatility and significant numerical range (e.g., energy consumption and Indian power data), where standardization reduces the impact of extreme values and facilitates model convergence. Third, the sole exception is Dataset 6 (Brazilian temperature data). This dataset features extremely sparse samples and intense volatility. Neither Laplace nor other smoothing methods improved performance; instead, they weakened predictive capability. Conversely, the baseline LSTM + Z-Score achieved optimal results (MAPE=0.22382, R^2=0.92362). This suggests that under small-sample conditions, excessive smoothing may lead to overfitting or distortion, whereas retaining the original LSTM structure better captures the true variations in the trend.

Fig. 3 provides an intuitive comparison of predicted values with actual values across the six datasets, corroborating the numerical findings in the preceding tables. For Datasets 1, 3, and 5 (electricity and energy consumption data), the prediction curves closely align with the actual curves, indicating the model captures the primary trend of periodic fluctuations. This aligns with the earlier conclusion that Laplace smoothing achieved near-perfect R^2 values on these datasets.

Dataset 2 exhibits significant overlap between peak predictions and actual values, further validating the superiority of Z-Score + Laplace for small-sample energy consumption data. In Dataset 4 (household energy data), the predicted values appear smoother than the actual values. While the overall trend aligns, some peaks are flattened, revealing the "over-smoothing" effect inherent in smoothing methods, excellent performance on numerical metrics, but insufficient sensitivity to high-frequency fluctuations.

Finally, Dataset 6 (Brazilian temperature data) shows that the prediction curve generally follows the actual curve, yet exhibits significant deviations at specific points. This reflects that complex smoothing methods struggle to perform effectively under conditions of minimal samples and high volatility, whereas the baseline LSTM proves more robust. Overall, these figures clearly demonstrate that the effectiveness of different normalization and smoothing methods depends on dataset size, volatility, and noise characteristics. This further reinforces the earlier conclusion: Laplace smoothing is optimal in most scenarios, but overreliance on smoothing should be avoided when sample sizes are small.

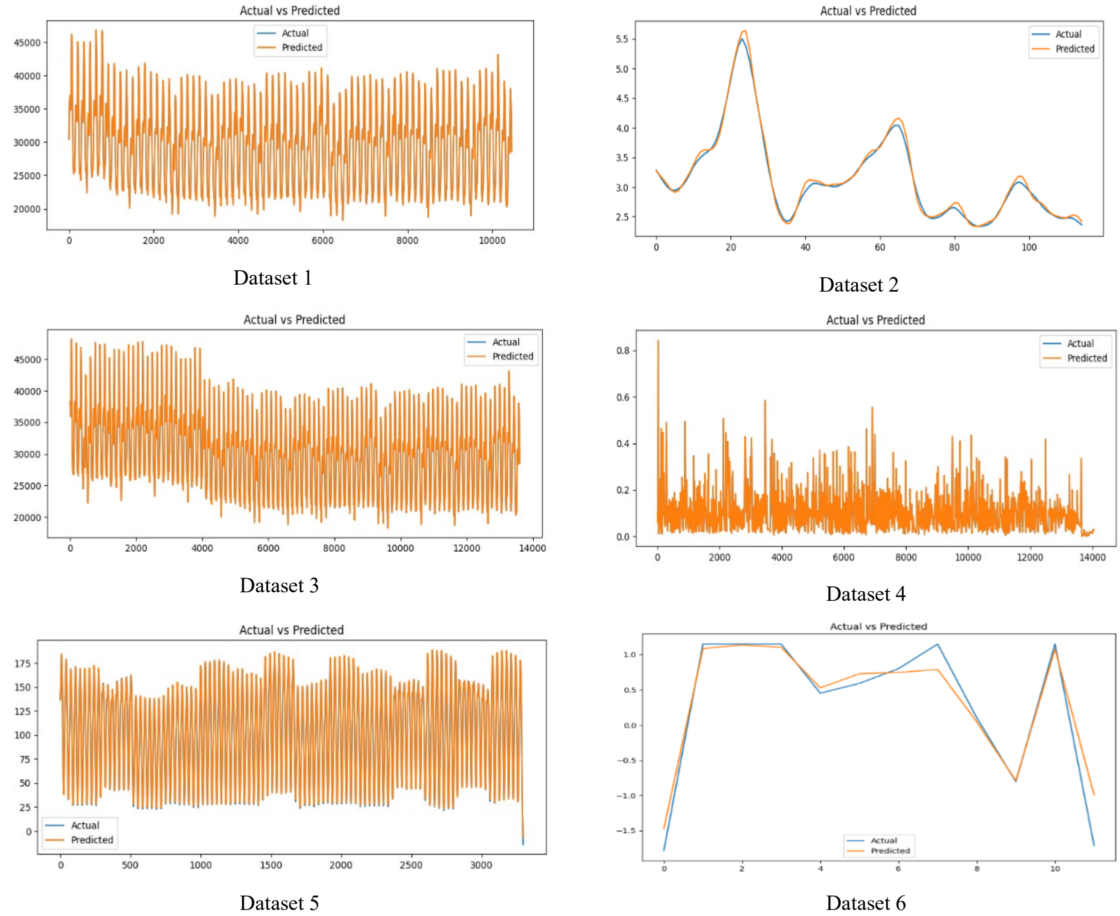

Fig. 3. Comparison of the prediction of the best performing model results versus actual values across the six datasets

Synthesizing results across six datasets yields three core conclusions: 1) Laplace smoothing is the optimal augmentation method across datasets, outperforming MA, SG, KS, and other methods in most scenarios; 2) Normalization methods require distribution-specific selection: Min-Max excels on stable data, while Z-Score holds greater advantage for highly volatile data; 3) Under extreme small sample conditions, the baseline LSTM proves more reliable, while smoothing methods may actually degrade forecasting performance. However, overall, LSTM+Laplace with Min-Max normalization achieves the best performance among the models. These findings underscore a crucial insight: forecasting performance improvements depend not only on complex model structures but also on the alignment between preprocessing methods and data characteristics. Future research should therefore explore adaptive normalization and dynamic smoothing strategies to achieve robust cross-domain forecasting performance.

4. Conclusion

This study systematically compared the predictive performance of the LSTM baseline model against various smoothing augmentation methods (Kalman, Laplace, Moving Average, Savitzky-Golay) under different normalization approaches across six distinct time series datasets. Experimental results demonstrate that Laplace smoothing consistently yields the best performance across nearly all large-scale datasets, significantly enhancing R^2 while maintaining a low MAPE. This demonstrates its robust capability to handle multivariate, highly volatile time-series data. However, a drawback is the tendency toward "over-smoothing" in certain scenarios, which averages predictions and may obscure extreme volatility features in practice. Further analysis reveals that normalization methods critically influence model performance. Min-Max normalization is more suitable for scenarios with relatively stable data ranges and high noise levels (e.g., electricity and household energy consumption data). In contrast, Z-score normalization performs better for highly volatile data and wide numerical ranges (e.g., energy consumption and Indian electricity data). This finding suggests that the choice of normalization method should align with the distributional characteristics of the data to fully leverage the model's potential. The sole exception is Dataset 6 (temperature data). Due to its small sample size and high volatility, complex smoothing methods not only failed to improve predictive performance but also caused deterioration. In this case, the baseline LSTM combined with Z-score proved more robust. Therefore, the study concludes that smoothing methods are not inherently better simply because they are more complex. The optimal strategy should be selected by considering dataset size, noise characteristics, and normalization methods. Future work could explore adaptive smoothing and transfer learning further to enhance the model's generalization in cross-domain time-series forecasting.

Acknowledgement

The authors would like to thank the Directorate of Research and Community Service (DPPM), Ministry of Higher Education, Science, and Technology (Kemdiktisaintek) of the Republic of Indonesia for funding support in the implementation of this research through Fundamental Research Schema with Contract Number 126/C3/DT.05.00/PL/2025 dated May 28, 2025; 0498.12/LL5-INT/AL.04/2025 dated Juni 4, 2025; and 011/PFR/LPPM.UAD/V/2025 dated June 5, 2025. The authors also thank Universitas Ahmad Dahlan for facilities support and Professor Aji Prasetya Wibawa and Agung Bella Putra Utama for in-depth discussions during the conduct of this research.

Declarations

Author contribution. All authors contributed equally to the primary contributor to this paper. All authors read and approved the final paper.

Funding statement. This research was funded by Directorate of Research and Community Service (DPPM), Directorate General of Research and Development, Ministry of Higher Education, Science, and Technology (Kemdiktisaintek) of the Republic of Indonesia for funding support in the implementation of this research through Fundamental Research Schema with Contract Number 126/C3/DT.05.00/PL/2025 dated May 28, 2025; 0498.12/LL5-INT/AL.04/2025 dated Juni 4, 2025; and 011/PFR/LPPM.UAD/V/2025 dated June 5, 2025.

Conflict of interest. The authors declare that they have no conflict of interest.

Additional information. No additional information is available for this paper.

Conflict of interest. The authors declare that they have no conflict of interest.

Additional information. No additional information is available for this paper.

Data and Software Availability Statements

- The Electric Power Consumption (https://www.kaggle.com/datasets/fedesoriano/electric-power-consumption).
- The Energy Consumption dataset (https://www.kaggle.com/datasets/vitthalmadane/energy-consumption-time-series-dataset).
- The Hourly Sensor Data dataset (https://www.kaggle.com/datasets/sudhanvahg/hourly-sensor-data-for-forecasting).
- Household Energy Data dataset (https://www.kaggle.com/datasets/jaganadhg/house-hold-energy-data).

- The Power Consumption dataset (https://www.kaggle.com/datasets/twinkle0705/state-wise-power-consumption-in-india).
- The Temperature dataset (https://www.kaggle.com/datasets/volpatto/temperature-timeseries-for-some-brazilian-cities)

References

[1] M. H. Alharbi, "Prediction of the Stock Market Using LSTM, ARIMA, and Hybrid of LSTM-ARIMA Models," *J. Knowl. Manag. Appl. Pract. An Int. J.*, vol. 7, no. 1, p. 15, 2025, doi: 10.18576/jkmap/070102.

[2] V. R. Thota, S. Engineering, and M. Candidate, "Comparative Study of Time Series Forecasting for Trucking Shipments : Evaluating Arima , Lstm , and Hybrid Arima-Lstm Models," 2025, p. 2025. [Online]. Available at: https://www.binghamton.edu/ssie/about/venkatesh_thesis_abstract.pdf.

[3] Y. Ding and Y. Zhai, "Intrusion Detection System for NSL-KDD Dataset Using Convolutional Neural Networks," in *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence*, New York, NY, USA: ACM, Dec. 2018, pp. 81–85. doi: 10.1145/3297156.3297230.

[4] W. Zhang, Z. Lin, and X. Liu, "Short-term offshore wind power forecasting - A hybrid model based on Discrete Wavelet Transform (DWT), Seasonal Autoregressive Integrated Moving Average (SARIMA), and deep-learning-based Long Short-Term Memory (LSTM)," *Renew. Energy*, vol. 185, no. February, pp. 611–628, Feb. 2022, doi: 10.1016/j.renene.2021.12.100.

[5] B. Wang *et al.*, "Long short-term memory deep learning model for predicting the dynamic performance of automotive PEMFC system," *Energy AI*, vol. 14, no. October, p. 100278, Oct. 2023, doi: 10.1016/j.egyai.2023.100278.

[6] I. Malashin, V. Tynchenko, A. Gantimurov, V. Nelyub, and A. Borodulin, "Applications of Long Short-Term Memory (LSTM) Networks in Polymeric Sciences: A Review," *Polym. 2024, Vol. 16, Page 2607*, vol. 16, no. 18, p. 2607, Sep. 2024, doi: 10.3390/POLYM16182607.

[7] H. Alizadegan, B. Rashidi Malki, A. Radmehr, H. Karimi, and M. A. Ilani, "Comparative study of long short-term memory (LSTM), bidirectional LSTM, and traditional machine learning approaches for energy consumption prediction," *Energy Explor. Exploit.*, vol. 43, no. 1, pp. 281–301, Jan. 2025, doi: 10.1177/01445987241269496/FORMAT/EPUB.

[8] K. E. ArunKumar, D. V. Kalaga, C. M. S. Kumar, M. Kawaji, and T. M. Brenza, "Forecasting of COVID-19 using deep layer Recurrent Neural Networks (RNNs) with Gated Recurrent Units (GRUs) and Long Short-Term Memory (LSTM) cells," *Chaos, Solitons & Fractals*, vol. 146, no. May, p. 110861, May 2021, doi: 10.1016/j.chaos.2021.110861.

[9] S. Nosouhian, F. Nosouhian, and A. Kazemi Khoshouei, "A Review of Recurrent Neural Network Architecture for Sequence Learning: Comparison between LSTM and GRU," Preprints, Jul. 2021, p. 7. doi: 10.20944/preprints202107.0252.v1.

[10] X. Qin, X. Hu, H. Liu, W. Shi, and J. Cui, "A Combined Gated Recurrent Unit and Multi-Layer Perception Neural Network Model for Predicting Shale Gas Production," *Processes*, vol. 11, no. 3, p. 806, Mar. 2023, doi: 10.3390/pr11030806.

[11] Z. Tarek *et al.*, "An Optimized Model Based on Deep Learning and Gated Recurrent Unit for COVID-19 Death Prediction," *Biomimetics 2023, Vol. 8,* vol. 8, no. 7, p. 552, Nov. 2023, doi: 10.3390/biomimetics8070552.

[12] A. Sahi *et al.*, "SGDM-GRU: Spectral graph deep learning based Gated Recurrent Unit model for accurate fake news detection," *Expert Syst. Appl.*, vol. 281, no. 2, p. 127572, Jul. 2025, doi: 10.1016/j.eswa.2025.127572.

[13] M. M. Taye, "Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions," *Computation*, vol. 11, no. 3, p. 52, Mar. 2023, doi: 10.3390/computation11030052.

[14] K. Bian and R. Priyadarshi, "Machine Learning Optimization Techniques: A Survey, Classification, Challenges, and Future Research Issues," *Arch. Comput. Methods Eng.*, vol. 31, no. 7, pp. 4209–4233, Mar. 2024, doi: 10.1007/s11831-024-10110-w.

[15] R. Moradi, R. Berangi, and B. Minaei, "A survey of regularization strategies for deep models," *Artif. Intell. Rev.*, vol. 53, no. 6, pp. 3947–3986, Aug. 2020, doi: 10.1007/s10462-019-09784-7.

[16] P. Anand Kumar and S. Sountharrajan, "Insurance claims estimation and fraud detection with optimized deep learning techniques," *Sci. Rep.*, vol. 15, no. 1, p. 27296, Jul. 2025, doi: 10.1038/s41598-025-12848-0.

[17] Y. Liu, T. Dillon, W. Yu, W. Rahayu, and F. Mostafa, "Noise Removal in the Presence of Significant Anomalies for Industrial IoT Sensor Data in Manufacturing," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7084–7096, Aug. 2020, doi: 10.1109/JIOT.2020.2981476.

[18] Y. Shi, X. Ying, and J. Yang, "Deep Unsupervised Domain Adaptation with Time Series Sensor Data: A Survey," *Sensors*, vol. 22, no. 15, p. 5507, Jul. 2022, doi: 10.3390/s22155507.

[19] F. Raza, D. Owaki, and M. Hayashibe, "Modeling and Control of a Hybrid Wheeled Legged Robot: Disturbance Analysis," in *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, IEEE, Jul. 2020, pp. 466–473. doi: 10.1109/AIM43001.2020.9158833.

[20] L. Zhang and L. Hua, "Major Issues in High-Frequency Financial Data Analysis: A Survey of Solutions," *Mathematics*, vol. 13, no. 3, p. 347, Jan. 2025, doi: 10.3390/math13030347.

[21] M. Usmani, Z. A. Memon, A. Zulfiqar, and R. Qureshi, "Preptimize: Automation of Time Series Data Preprocessing and Forecasting," *Algorithms*, vol. 17, no. 8, p. 332, Aug. 2024, doi: 10.3390/a17080332.

[22] S. F. Stefenon, L. O. Seman, V. C. Mariani, and L. dos S. Coelho, "Aggregating Prophet and Seasonal Trend Decomposition for Time Series Forecasting of Italian Electricity Spot Prices," *Energies*, vol. 16, no. 3, p. 1371, Jan. 2023, doi: 10.3390/en16031371.

[23] U. Zbezhkhovska and D. Chumachenko, "Smoothing Techniques for Improving COVID-19 Time Series Forecasting Across Countries," *Computation*, vol. 13, no. 6, p. 136, Jun. 2025, doi: 10.3390/computation13060136.

[24] Y. Dai, Y. Wang, M. Leng, X. Yang, and Q. Zhou, "LOWESS smoothing and Random Forest based GRU model: A short-term photovoltaic power generation forecasting method," *Energy*, vol. 256, no. October, p. 124661, Oct. 2022, doi: 10.1016/j.energy.2022.124661.

[25] I. E. Livieris, S. Stavroyiannis, L. Iliadis, and P. Pintelas, "Smoothing and stationarity enforcement framework for deep learning time-series forecasting," *Neural Comput. Appl.*, vol. 33, no. 20, pp. 14021–14035, Oct. 2021, doi: 10.1007/s00521-021-06043-1.

[26] M. Khodarahmi and V. Maihami, "A Review on Kalman Filter Models," *Arch. Comput. Methods Eng.*, vol. 30, no. 1, pp. 727–747, Jan. 2023, doi: 10.1007/s11831-022-09815-7.

[27] X. Yu and J. Li, "Adaptive Kalman Filtering for Recursive Both Additive Noise and Multiplicative Noise," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 3, pp. 1634–1649, Jun. 2022, doi: 10.1109/TAES.2021.3117896.

[28] A. Y. Alanis, "Exploring Kalman Filtering Applications for Enhancing Artificial Neural Network Learning," *Algorithms*, vol. 18, no. 9, p. 587, Sep. 2025, doi: 10.3390/a18090587.

[29] T. Kruse, T. Griebel, and K. Graichen, "Adaptive Kalman Filtering: Measurement and Process Noise Covariance Estimation Using Kalman Smoothing," *IEEE Access*, vol. 13, pp. 11863–11875, 2025, doi: 10.1109/ACCESS.2025.3528348.

[30] C. Lei, H. Zhang, Z. Wang, and Q. Miao, "Deep Learning for Demand Forecasting: A Framework Incorporating Variational Mode Decomposition and Attention Mechanism," *Processes*, vol. 13, no. 2, p. 594, Feb. 2025, doi: 10.3390/pr13020594.

[31] M. Taktak and F. Derbel, "Evaluating the Impact of Frequency Decomposition Techniques on LSTM-Based Household Energy Consumption Forecasting," *Energies*, vol. 18, no. 10, p. 2507, May 2025, doi: 10.3390/en18102507.

[32] J. Kim, H. Kim, H. Kim, D. Lee, and S. Yoon, "A comprehensive survey of deep learning for time series forecasting: architectural diversity and open challenges," *Artif. Intell. Rev.*, vol. 58, no. 7, p. 216, Apr. 2025, doi: 10.1007/s10462-025-11223-9.

[33] L. Boongasame, J. Muangprathub, and K. Thammarak, "Laor Initialization: A New Weight Initialization Method for the Backpropagation of Deep Learning," *Big Data Cogn. Comput.*, vol. 9, no. 7, p. 181, Jul. 2025, doi: 10.3390/bdcc9070181.

[34] M. Shantal, Z. Othman, and A. A. Bakar, "A Novel Approach for Data Feature Weighting Using Correlation Coefficients and Min–Max Normalization," *Symmetry (Basel).*, vol. 15, no. 12, p. 2185, Dec. 2023, doi: 10.3390/sym15122185.

[35] K. Cabello-Solorzano, I. Ortigosa de Araujo, M. Peña, L. Correia, and A. J. Tallón-Ballesteros, "The Impact of Data Normalization on the Accuracy of Machine Learning Algorithms: A Comparative Analysis," in *Lecture Notes in Networks and Systems*, Springer, Cham, 2023, pp. 344–353. doi: 10.1007/978-3-031-42536-3_33.

[36] A. Al-Mekhlafi, S. Klawitter, and F. Klawonn, "Standardization with zlog values improves exploratory data analysis and machine learning for laboratory data," *J. Lab. Med.*, vol. 48, no. 5, pp. 215–222, Oct. 2024, doi: 10.1515/labmed-2024-0051.

[37] R. Castaldo, K. Pane, E. Nicolai, M. Salvatore, and M. Franzese, "The Impact of Normalization Approaches to Automatically Detect Radiogenomic Phenotypes Characterizing Breast Cancer Receptors Status," *Cancers (Basel).*, vol. 12, no. 2, p. 518, Feb. 2020, doi: 10.3390/cancers12020518.

[38] S. Nikbakht, C. Anitescu, and T. Rabczuk, "Optimizing the neural network hyperparameters utilizing genetic algorithm," *J. Zhejiang Univ. A*, vol. 22, no. 6, pp. 407–426, Jun. 2021, doi: 10.1631/jzus.A2000384.

[39] L. Liao, H. Li, W. Shang, and L. Ma, "An Empirical Study of the Impact of Hyperparameter Tuning and Model Optimization on the Performance Properties of Deep Neural Networks," *ACM Trans. Softw. Eng. Methodol.*, vol. 31, no. 3, pp. 1–40, Jul. 2022, doi: 10.1145/3506695.

[40] Y. Li, X. Ren, F. Zhao, and S. Yang, "A Zeroth-Order Adaptive Learning Rate Method to Reduce Cost of Hyperparameter Tuning for Deep Learning," *Appl. Sci.*, vol. 11, no. 21, p. 10184, Oct. 2021, doi: 10.3390/app112110184.

[41] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT Express*, vol. 6, no. 4, pp. 312–315, Dec. 2020, doi: 10.1016/j.icte.2020.04.010.

[42] D. M. Belete and M. D. Huchaiah, "Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results," *Int. J. Comput. Appl.*, vol. 44, no. 9, pp. 875–886, Sep. 2022, doi: 10.1080/1206212X.2021.1974663.

[43] H. Alibrahim and S. A. Ludwig, "Hyperparameter Optimization: Comparing Genetic Algorithm against Grid Search and Bayesian Optimization," in *2021 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, Jun. 2021, pp. 1551–1559. doi: 10.1109/CEC45853.2021.9504761.

[44] A. Esmaeili, Z. Ghorrati, and E. T. Matson, "Agent-Based Collaborative Random Search for Hyperparameter Tuning and Global Function Optimization," *Systems*, vol. 11, no. 5, p. 228, May 2023, doi: 10.3390/systems11050228.

[45] Y. Ali, E. Awwad, M. Al-Razgan, and A. Maarouf, "Hyperparameter Search for Machine Learning Algorithms for Optimizing the Computational Complexity," *Processes*, vol. 11, no. 2, p. 349, Jan. 2023, doi: 10.3390/pr11020349.

[46] R. Sen and S. C. Bhattacharyya, "Off-grid electricity generation with renewable energy technologies in India: An application of HOMER," *Renew. Energy*, vol. 62, no. February, pp. 388–398, Feb. 2014, doi: 10.1016/j.renene.2013.07.028.

[47] P. Balaprakash, M. Salim, T. D. Uram, V. Vishwanath, and S. M. Wild, "DeepHyper: Asynchronous Hyperparameter Search for Deep Neural Networks," in *2018 IEEE 25th International Conference on High Performance Computing (HiPC)*, IEEE, Dec. 2018, pp. 42–51. doi: 10.1109/HiPC.2018.00014.

[48] B. Bischl *et al.*, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *WIREs Data Min. Knowl. Discov.*, vol. 13, no. 2, p. 1484, Mar. 2023, doi: 10.1002/widm.1484.

[49] M. M. Musthafa, I. Manimozhi, T. R. Mahesh, and S. Guluwadi, "Optimizing double-layered convolutional neural networks for efficient lung cancer classification through hyperparameter optimization and advanced image pre-processing techniques," *BMC Med. Inform. Decis. Mak.*, vol. 24, no. 1, p. 142, May 2024, doi: 10.1186/s12911-024-02553-9.

[50] M. S. Khan, T. Peng, H. Akhlaq, and M. Adeel Khan, "Comparative Analysis of Automated Machine Learning for Hyperparameter Optimization and Explainable Artificial Intelligence Models," *IEEE Access*, vol. 13, pp. 84966–84991, 2025, doi: 10.1109/ACCESS.2025.3566427.

[51] M. Ahmadi, H. Aly, and M. Khashei, "Enhancing power grid stability with a hybrid framework for wind power forecasting: Integrating Kalman Filtering, Deep Residual Learning, and Bidirectional LSTM," *Energy*, vol. 334, no. October, p. 137752, Oct. 2025, doi: 10.1016/j.energy.2025.137752.

[52] I. Aizenberg and Y. Tovt, "Intelligent Frequency Domain Image Filtering Based on a Multilayer Neural Network with Multi-Valued Neurons," *Algorithms*, vol. 18, no. 8, p. 461, Jul. 2025, doi: 10.3390/a18080461.

[53] S. Chakrabarty, R. Talwadker, and T. Mukherjee, "ScarceGAN," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, New York, NY, USA: ACM, Oct. 2021, pp. 140–150. doi: 10.1145/3459637.3482474.

[54] Y. Su, C. Cui, and H. Qu, "Self-Attentive Moving Average for Time Series Prediction," *Appl. Sci.*, vol. 12, no. 7, p. 3602, Apr. 2022, doi: 10.3390/app12073602.

[55] S. Zhang, X. Ma, Z. Fang, H. Pan, G. Yang, and G. R. Arce, "Financial time series forecasting based on momentum-driven graph signal processing," *Appl. Intell.*, vol. 53, no. 18, pp. 20950–20966, Sep. 2023, doi: 10.1007/s10489-023-04563-y.

[56] A. Lazcano, M. A. Jaramillo-Morán, and J. E. Sandubete, "Back to Basics: The Power of the Multilayer Perceptron in Financial Time Series Forecasting," *Mathematics*, vol. 12, no. 12, p. 1920, Jun. 2024, doi: 10.3390/math12121920.

[57] Z.-J. Peng, C. Zhang, and Y.-X. Tian, "Crude Oil Price Time Series Forecasting: A Novel Approach Based on Variational Mode Decomposition, Time-Series Imaging, and Deep Learning," *IEEE Access*, vol. 11, pp. 82216–82231, 2023, doi: 10.1109/ACCESS.2023.3301576.

[58] E. Kim *et al.*, "Innovative strategies for protein content determination in dried laver (Porphyra spp.): Evaluation of preprocessing methods and machine learning algorithms through short-wave infrared imaging," *Food Chem. X*, vol. 23, no. 1, p. 101763, Oct. 2024, doi: 10.1016/j.fochx.2024.101763.

[59] X. Wang, C. Qian, Z. Zhao, J. Li, and M. Jiao, "A Novel Gas Recognition Algorithm for Gas Sensor Array Combining Savitzky–Golay Smooth and Image Conversion Route," *Chemosensors*, vol. 11, no. 2, p. 96, Jan. 2023, doi: 10.3390/chemosensors11020096.

[60] R. H. F. Peshawa J. Muhammad Ali, "Data Normalization and Standardization: A Technical Report," The Machine Learning Lab. at Koya University Koya, Erbil, Iraq. Accessed: Mar. 01, 2026. [Online]. Available at: https://www.researchgate.net/publication/340579135_Data_Normalization_and_Standardization_A_Technical_Report.

[61] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Appl. Soft Comput.*, vol. 97, no. December, p. 105524, Dec. 2020, doi: 10.1016/j.asoc.2019.105524.

[62] H. W. Herwanto, A. N. Handayani, A. P. Wibawa, K. L. Chandrika, and K. Arai, "Comparison of Min-Max, Z-Score and Decimal Scaling Normalization for Zoning Feature Extraction on Javanese Character Recognition," in *2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, IEEE, Oct. 2021, pp. 1–3. doi: 10.1109/ICEEIE52663.2021.9616665.

[63] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Comput. Sci.*, vol. 7, p. e623, Jul. 2021, doi: 10.7717/peerj-cs.623.

[64] X. Shen, H. Liu, G. Qiu, Y. Liu, J. Liu, and S. Fan, "Interpretable Interval Prediction-Based Outlier-Adaptive Day-Ahead Electricity Price Forecasting Involving Cross-Market Features," *IEEE Trans. Ind. Informatics*, vol. 20, no. 5, pp. 7124–7137, May 2024, doi: 10.1109/TII.2024.3355105.

[65] O. Kisi, "Wavelet Regression Model as an Alternative to Neural Networks for River Stage Forecasting," *Water Resour. Manag.*, vol. 25, no. 2, pp. 579–600, Jan. 2011, doi: 10.1007/s11269-010-9715-8.