

Article

PEL: An Integrated Algorithm for Power Time Series Anomaly Detection

Lei Wang *, Yu Gao *  and Xiaoyong Zhao 

School of Management Science and Engineering, Beijing Information Science and Technology University, Beijing 102206, China; zhaoxiaoyong@bistu.edu.cn

* Correspondence: wanglei@bistu.edu.cn (L.W.); popcat12138@gmail.com (Y.G.)

Abstract

Power systems continuously generate large-scale load time series data for forecasting, consumption analysis, and equipment health monitoring. However, real-world load measurements are often contaminated by anomalies caused by sensor faults, communication errors, and abnormal consumption behaviors, which may degrade data quality and affect operational decision-making. To address this issue, this paper proposes an integrated anomaly detection framework named PEL, which combines Prophet-based seasonal-trend decomposition, ensemble empirical mode decomposition (EEMD), and a multilayer long short-term memory (LSTM) network. Prophet is first employed to decompose the original series into trend, seasonal, holiday, and residual components. Sample entropy analysis and white noise tests are then adopted to evaluate whether the residual component still contains complex structured information requiring secondary decomposition. Next, EEMD is applied to the residual component to extract multi-scale intrinsic mode functions. Finally, all decomposed components are normalized and fed into a multilayer LSTM model for anomaly detection. Experiments on a real-world power load dataset demonstrate that the proposed PEL framework achieves an accuracy of 99.92%, a precision of 97.33%, a recall of 100%, an F1-score of 98.65%, and an AUC of 0.9996, outperforming or matching several baseline and hybrid models.

Keywords: ensemble empirical mode decomposition; long short-term memory neural network; power load; prophet seasonal trend decomposition; time series anomaly detection

1. Introduction

The power industry is a fundamental infrastructure sector supporting national economic development and modern societal operations [1,2]. With the rapid integration of intelligent sensing, communication networks, and artificial intelligence into power systems, massive volumes of power load time series data are continuously generated for load forecasting, consumption analysis, and equipment health monitoring [3,4]. Such load time series often exhibit complex characteristics, including long-term trends, strong periodicity, temporal dependence, nonlinearity, and large-scale sampling [5]. However, real-world power load measurements are frequently contaminated by anomalies arising from measurement faults (e.g., sensor failures, communication interruptions, and data corruption) or abnormal consumption behaviors (e.g., electricity theft and equipment malfunctions) [6–10]. These anomalies may significantly reduce data reliability and further threaten the safe and stable operation of power systems [11].



Academic Editor: Phivos Mylonas

Received: 19 May 2026

Revised: 16 June 2026

Accepted: 17 June 2026

Published: 20 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

In smart-grid and advanced metering infrastructure scenarios, power load anomaly detection is not only a data-quality task but also an important means of supporting electricity theft prevention and operational risk control. Existing non-MDPI studies have shown that feature-engineering strategies and forecasting-based detection frameworks can exploit consumption-pattern deviations to improve theft and anomaly detection in smart-meter data [12,13].

Anomaly detection (also referred to as outlier detection) aims to identify observations that deviate from expected patterns or statistical distributions [14]. Traditional anomaly detection in power systems relied heavily on manual inspection and expert experience, which is labor-intensive and difficult to scale for large-scale monitoring scenarios [15]. In recent years, with the advancement of big data analytics and artificial intelligence, machine learning and deep learning methods have been widely explored for power load anomaly detection [16–18]. For example, Karkhaneh and Ozgoli [19] proposed a two-stage detection framework based on wavelet transform and robust regression. Despite these efforts, several challenges remain. First, power load series typically contain multi-scale seasonal patterns and strong nonstationarity, which makes effective feature extraction difficult. Second, anomalous events are usually sparse and diverse, leading to severe class imbalance and limited representation of abnormal behaviors. Third, directly learning from raw sequences may result in insufficient modeling of hidden structures embedded in residual fluctuations, thereby reducing detection accuracy [20].

To address these challenges, this paper proposes an integrated power load time series anomaly detection framework named PEL, which combines Prophet-based seasonal-trend decomposition, ensemble empirical mode decomposition (EEMD), and a multilayer long short-term memory (LSTM) network. Prophet is first applied to decompose the original sequence into trend, seasonal, holiday, and residual components. The residual component is then analyzed using sample entropy and white noise tests to evaluate whether it still contains complex structured temporal dependence and therefore requires secondary decomposition. Next, EEMD is employed to perform secondary decomposition on the residual term and extract multi-scale intrinsic mode functions, enabling a more complete representation of hidden temporal patterns. Finally, all decomposition components are normalized and fed into a multilayer LSTM model for anomaly detection. Experimental results on real-world power load data demonstrate that the proposed framework achieves strong and competitive performance compared with several baseline and hybrid models, indicating its effectiveness for power load time series anomaly detection.

The main contributions of this paper are summarized as follows:

- A novel integrated anomaly detection framework (PEL) is proposed by combining Prophet decomposition, EEMD-based residual secondary decomposition, and a multilayer LSTM network for complex power load time series.
- A residual complexity evaluation strategy based on sample entropy and white noise testing is introduced to guide whether further decomposition is required, improving the completeness of feature extraction.
- Experiments are conducted on a real-world power load dataset, including main-setting comparison, component ablation, robustness checks, and runtime analysis, to evaluate detection accuracy and computational cost comprehensively.

The remainder of this paper is organized as follows. Section 2 introduces the theoretical foundations of Prophet, EEMD, and LSTM. Section 3 describes the proposed PEL framework, including two-stage decomposition, normalization, and LSTM-based anomaly detection. Section 4 presents the dataset, anomaly-injection strategy, comparison experiments, ablation study, robustness checks, and computational-cost analysis. Section 5 concludes the paper and discusses future work.

2. Theoretical Foundation

2.1. Prophet

Prophet is an open-source time series forecasting framework developed by Facebook, which is widely used for modeling time series with long-term trends and multiple seasonal patterns [21]. It adopts an additive decomposition formulation and represents an observed time series as the combination of a trend component, seasonal components, holiday effects, and residual noise [22]. The decomposition form can be expressed as

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t, \quad (1)$$

where $y(t)$ denotes the observed time series, $g(t)$ represents the long-term trend, $s(t)$ captures periodic seasonal variations (e.g., weekly or yearly seasonality), and $h(t)$ models the effects of holidays or special events. The term ϵ_t denotes the residual component, which is typically assumed to follow a zero-mean noise distribution.

In Prophet, the trend term $g(t)$ is usually modeled as a piecewise linear or logistic growth function with automatic changepoint detection, enabling robust modeling of nonstationary temporal dynamics.

2.2. EEMD

Ensemble Empirical Mode Decomposition (EEMD) is an improved version of Empirical Mode Decomposition (EMD), which mitigates the mode mixing problem by introducing white noise into the original signal and performing ensemble averaging over multiple decompositions [23,24]. By leveraging the uniformly distributed spectrum of white noise, EEMD enables signals at different time scales to be separated into corresponding intrinsic mode functions (IMFs), thus achieving adaptive multi-scale decomposition [25].

Compared with conventional EMD, EEMD improves the stability and robustness of decomposition results, especially for nonlinear and nonstationary time series [26]. The overall procedure of EEMD is illustrated in Figure 1 and can be summarized as follows.

Step 1: Add a Gaussian white noise sequence $w_k(t)$ to the original signal $x(t)$ to generate a noise-assisted signal:

$$x_k(t) = x(t) + w_k(t), \quad (2)$$

where k denotes the index of the ensemble trial.

Step 2: Perform EMD on $x_k(t)$ to obtain a set of intrinsic mode functions and a residual term:

$$x_k(t) = \sum_{i=1}^n c_{k,i}(t) + r_k(t), \quad (3)$$

where n is the number of extracted IMFs, $c_{k,i}(t)$ denotes the i -th IMF in the k -th decomposition, and $r_k(t)$ represents the corresponding residual component.

Step 3: Repeat Steps 1–2 for K trials with different noise realizations to obtain multiple sets of IMFs and residual components.

Step 4: Compute the ensemble average of each IMF and the residual term to produce the final decomposition:

$$c_i(t) = \frac{1}{K} \sum_{k=1}^K c_{k,i}(t), \quad (4)$$

$$r(t) = \frac{1}{K} \sum_{k=1}^K r_k(t), \quad (5)$$

where K is the number of ensemble trials.

Finally, the EEMD reconstruction of the original signal can be expressed as

$$x(t) = \sum_{i=1}^n c_i(t) + r(t). \quad (6)$$

By introducing noise-assisted decomposition and ensemble averaging, EEMD effectively reduces mode mixing and improves the reliability of multi-scale feature extraction for complex time series.

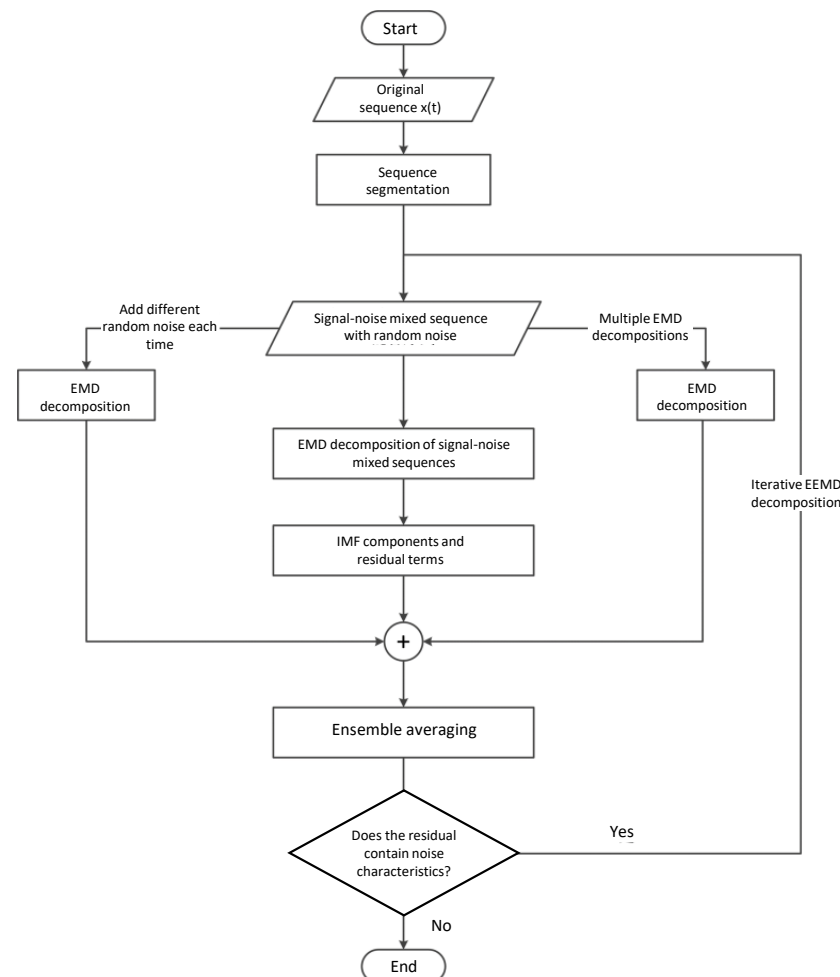


Figure 1. Flowchart of the ensemble empirical mode decomposition (EEMD) algorithm. The plus sign denotes ensemble averaging of the decomposed components.

2.3. LSTM

Long short-term memory (LSTM) is a variant of recurrent neural networks (RNNs) designed to address the vanishing gradient and exploding gradient problems in long-sequence learning [27]. An LSTM unit consists of a memory cell, a hidden state, and three gating mechanisms, including the input gate, forget gate, and output gate. These gates regulate the information flow and enable the network to preserve long-term dependencies, making LSTM particularly effective for modeling temporal dynamics in time series data. The structure of an LSTM cell is illustrated in Figure 2.

Given the input x_t and the previous hidden state h_{t-1} , the LSTM updates are defined as

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \quad (7)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \quad (8)$$

$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c), \quad (9)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \quad (10)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \quad (11)$$

$$h_t = o_t \odot \tanh(c_t), \quad (12)$$

where $\sigma(\cdot)$ denotes the sigmoid activation function and $\odot$ denotes element-wise multiplication.

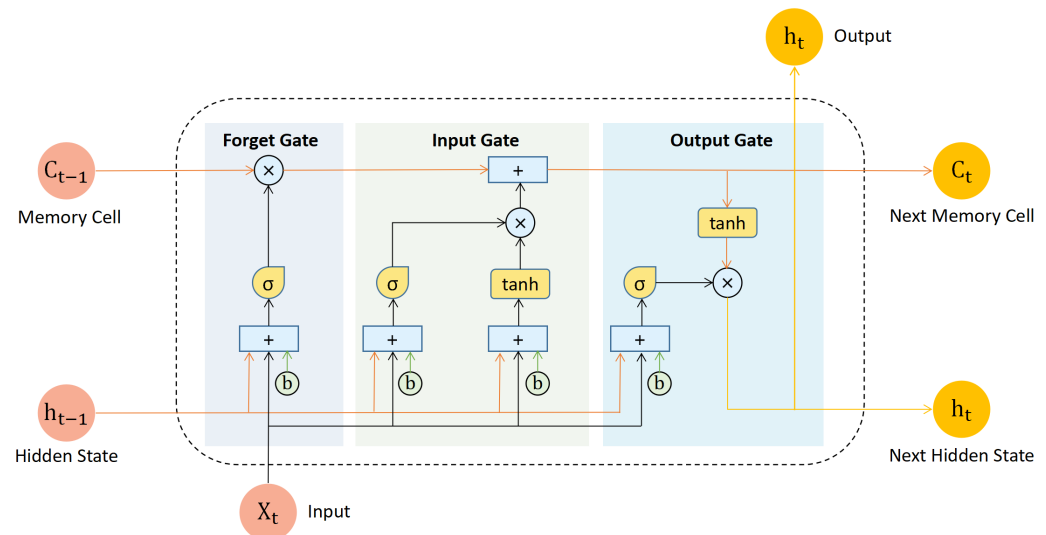


Figure 2. Structure of an LSTM cell.

3. Methodology

3.1. Time Series Decomposition Model Based on Prophet–EEMD

Power load time series typically exhibit complex characteristics, such as long-term trends, multiple seasonal patterns, and strong temporal dependencies. Due to the nonstationary and nonlinear nature of real-world power data, directly modeling the raw sequence often leads to insufficient feature extraction and degraded anomaly detection performance. Therefore, this paper proposes a two-stage time series decomposition framework based on Prophet and ensemble empirical mode decomposition (EEMD), aiming to extract multi-scale temporal components and enhance the representation of hidden patterns.

Prophet is first employed to decompose the original load series into interpretable components, including trend, seasonality, holiday effects, and residuals. Since the residual component may still contain complex temporal structures that are not fully captured by Prophet, EEMD is further applied to perform secondary decomposition and extract intrinsic mode functions (IMFs) at different time scales. The overall Prophet–EEMD decomposition framework is illustrated in Figure 3, and the procedure can be summarized as follows:

- Apply Prophet to the power load time series to obtain the trend component, seasonal components (e.g., daily and weekly seasonality), holiday component, and residual component.
- Evaluate the complexity and statistical dependence of the residual component using sample entropy and white noise testing to determine whether additional decomposition is required.
- Perform EEMD on the residual component to obtain a set of intrinsic mode functions, denoted as $\text{eIMF}_1, \text{eIMF}_2, \dots, \text{eIMF}_n$.
- Concatenate the Prophet decomposition components and the extracted EEMD IMFs to form the final multi-scale feature set for subsequent anomaly detection.

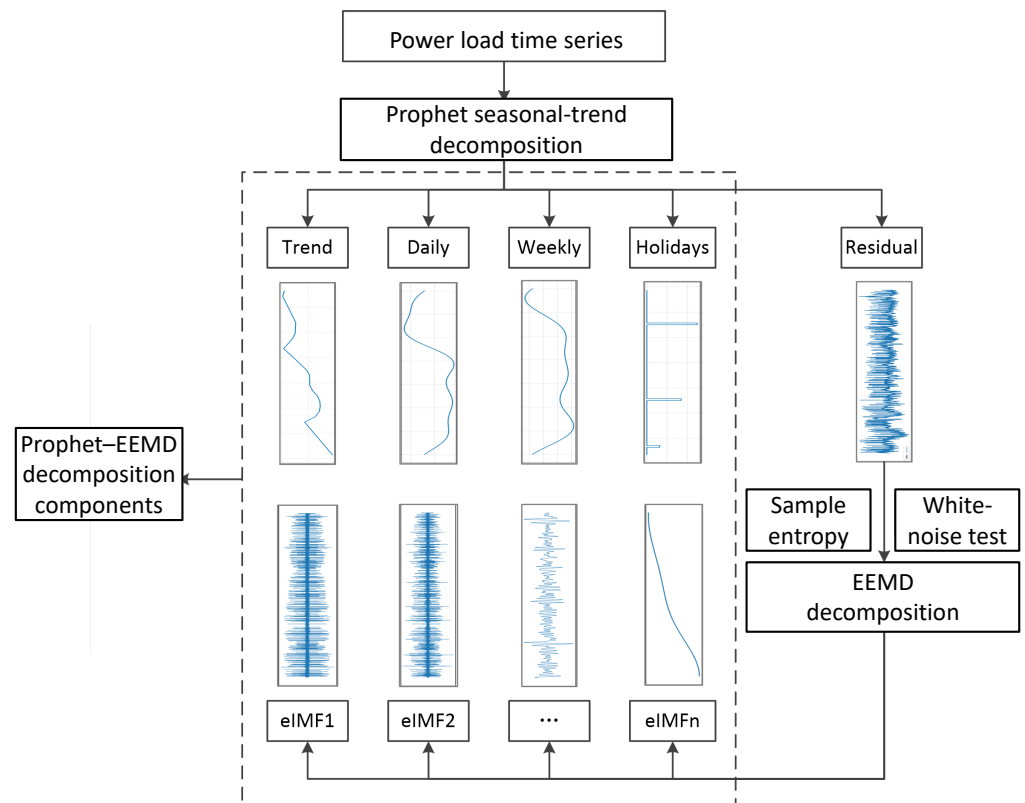


Figure 3. Prophet-EEMD time series decomposition modeling framework. Dashed lines and ellipses indicate process flow and repeated components.

3.2. Z-Score Normalization Module

To eliminate the influence of different feature magnitudes and improve model convergence, Z-score normalization is applied to all decomposed components. This standardization transforms each feature into a distribution with zero mean and unit variance, which is particularly suitable for time series components with unbounded ranges.

The Z-score normalization is defined as

$$X_{\text{norm}} = \frac{X - \mu}{\sigma}, \quad (13)$$

where μ and σ denote the mean and standard deviation of feature X , respectively, and X_{norm} denotes the normalized value.

3.3. LSTM-Based Anomaly Detection Module

After decomposition and normalization, all Prophet components and EEMD-derived IMFs are used as input features for anomaly detection. To effectively capture temporal dependencies and complex feature interactions, a multilayer LSTM network is designed in this work. As shown in Figure 4, the proposed network consists of an input layer, two stacked LSTM layers, a Dropout layer, and a fully connected output layer.

The first LSTM layer outputs hidden states at all time steps, enabling the model to learn sequential representations. A Dropout layer with a dropout rate of 0.5 is then applied to reduce overfitting. The second LSTM layer further extracts higher-level temporal features and outputs the hidden state of the last time step. Finally, a fully connected layer with a sigmoid activation function outputs the anomaly probability for binary classification.

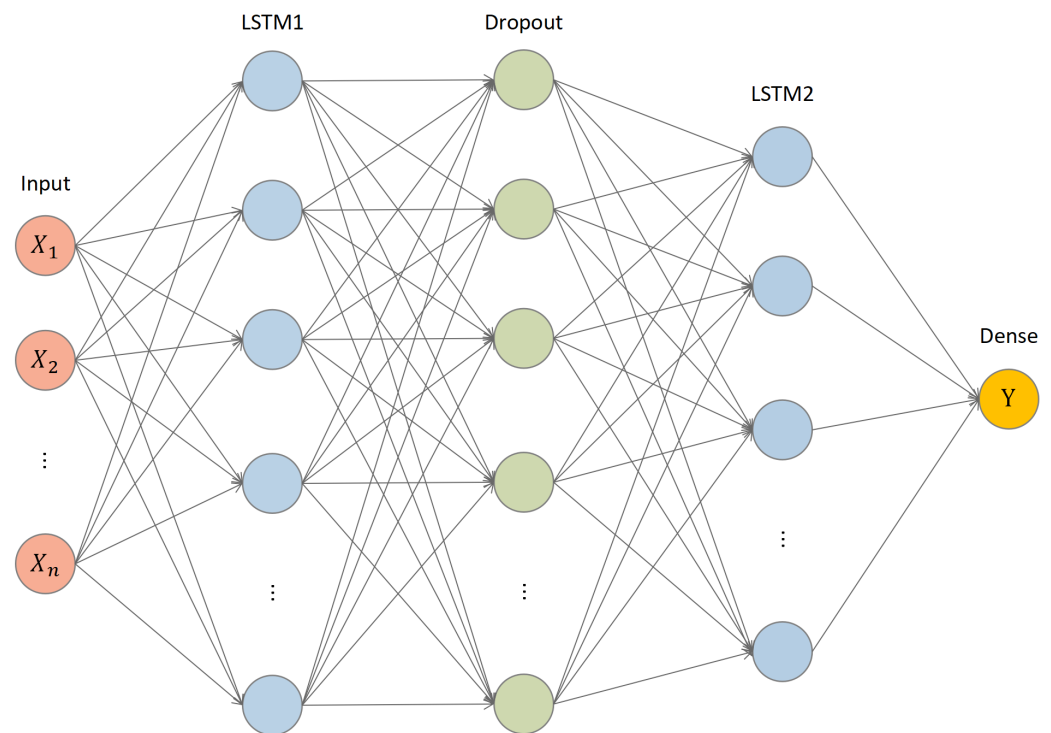


Figure 4. Multilayer LSTM neural network model. Here, Y denotes the model output, and ellipses indicate repeated hidden units.

To effectively exploit the multi-scale decomposition results for anomaly detection, all Prophet–EEMD components are used as input features of a multilayer LSTM network. The proposed network architecture is composed of the following modules:

(1) **Input layer:** The input layer receives all decomposition components produced by Prophet and EEMD. In this work, the input feature dimension is 17, including four Prophet components (trend, daily seasonality, weekly seasonality, and holiday effects) and 13 EEMD-derived intrinsic mode functions (IMFs) obtained from residual decomposition.

(2) **First LSTM layer:** The first LSTM layer contains 60 hidden units and outputs the hidden states at all time steps, enabling the model to capture long-term temporal dependencies.

(3) **Dropout layer:** A dropout layer with a dropout rate of 0.5 is applied after the first LSTM layer to mitigate overfitting and improve generalization performance.

(4) **Second LSTM layer:** The second LSTM layer contains 30 hidden units and outputs only the hidden state of the last time step, which serves as a compact representation for classification.

(5) **Output layer:** A fully connected layer with a sigmoid activation function is used to output the anomaly probability in the range of $[0, 1]$ for binary classification.

3.4. The PEL (Prophet–EEMD–LSTM) Model

As shown in Figure 5, the proposed PEL framework follows a three-stage pipeline. First, the raw power load time series is cleaned and the corresponding labels are preprocessed. Prophet decomposition is then applied to extract interpretable components, and the residual term is further decomposed by EEMD to obtain multi-scale IMFs. Second, residual complexity analysis is conducted, and all decomposition components are normalized using Z-score standardization. Finally, the normalized multi-component features are fed into the multilayer LSTM network for supervised training and anomaly detection. The network outputs the anomaly probability of each sample, and a sample is classified as anomalous if the probability exceeds a predefined threshold.

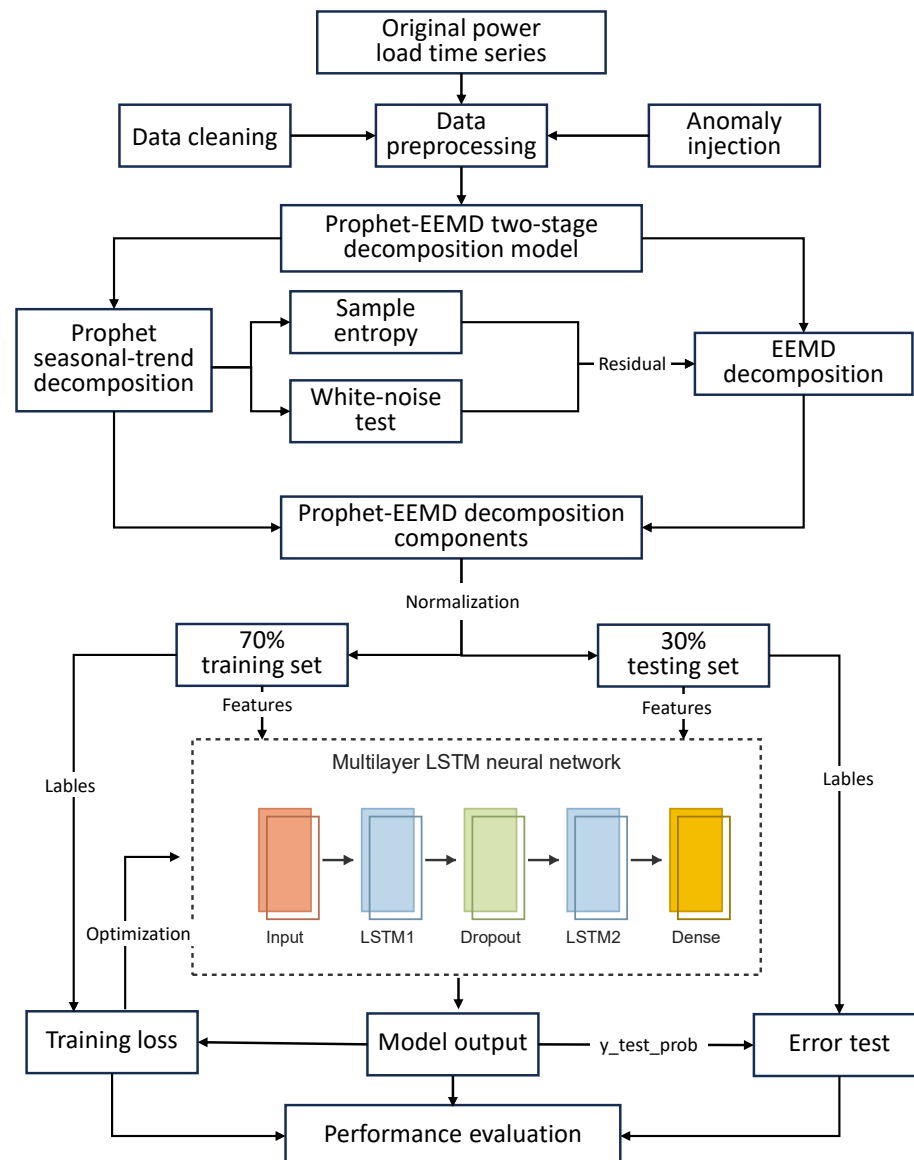


Figure 5. Overall framework of the proposed Prophet-EEMD-LSTM (PEL) anomaly detection model.

4. Experimental Results and Analysis

4.1. Dataset Description

The experimental data are obtained from the Area2_Load dataset released in the 2016 Electrician's Cup Mathematical Modeling Competition (Problem A). The dataset records real power load measurements of Area 2 with a sampling interval of 15 min (96 data points per day), and the unit is MW. In this study, a total of 8736 load records collected from 1 April 2013 to 30 June 2013 are used for evaluation.

The Area2_Load dataset was originally released for load forecasting rather than supervised anomaly detection, and it does not provide verified event-level anomaly labels. Therefore, the original sequence cannot be directly used to distinguish all real abnormal events from normal samples. Before artificial anomaly injection, the raw load series is first inspected and cleaned. The descriptive statistics show that the selected data contain no missing values, but include physically unreasonable negative load values, with the minimum value equal to -123 MW. Since negative power load values violate the normal business rule of load measurement, they are treated as erroneous measurements rather than

target anomalies. These erroneous values are replaced using an adjacent-value averaging strategy: if an erroneous value occurs at the first time point, it is replaced by the next load value; if it occurs at the last time point, it is replaced by the previous load value; otherwise, it is replaced by the average of the preceding and following load values. The cleaned sequence is then used as the baseline sequence for controlled anomaly injection. The load series before and after data cleaning is shown in Figure 6.

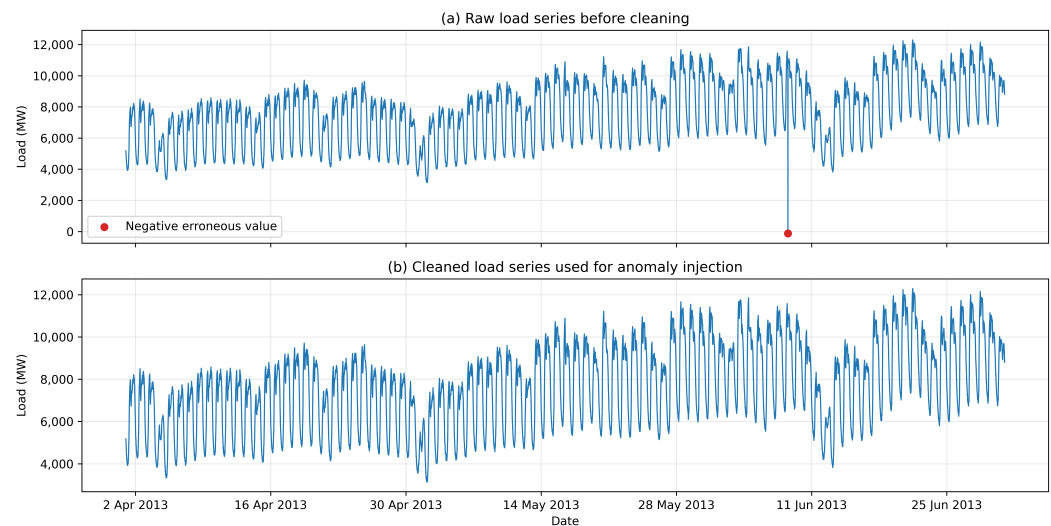


Figure 6. Power load series before and after data cleaning.

Because verified real anomaly labels are unavailable, anomaly injection is adopted to construct controlled labels for quantitative evaluation. The reported confusion matrices and classification metrics are computed with respect to these injected labels, not with respect to unknown real-world anomaly labels. Possible unverified abnormal events in the original sequence are therefore treated as unlabeled background data in the current evaluation setting, which is further discussed as a limitation in Section 5.

In smart-meter and advanced metering infrastructure studies, synthetic abnormal consumption data are commonly generated by modifying normal load profiles according to false-data injection or electricity-theft attack patterns [9,10,28]. Typical injection patterns include proportional scaling, threshold clipping, fixed-offset reduction, interval interruption, mean-value substitution, and sample-wise random scaling, as summarized in Table 1. These rules are simulation strategies used for controlled evaluation rather than industrial standards for deciding whether a load value is abnormal.

Table 1. Common synthetic anomaly-injection patterns used in smart-meter anomaly and electricity-theft studies.

Pattern	Representative Form	Anomaly Type	Simulated Behavior
Proportional scaling	$h(x_t) = \alpha x_t$	Point drop	Under-reported consumption with a fixed reduction ratio
Threshold clipping	$h(x_t) = \min(x_t, \gamma)$	Point drop	Meter readings capped by an upper threshold
Offset reduction	$h(x_t) = \max(x_t - \gamma, 0)$	Point drop	Consumption reduced by a fixed amount
Interval interruption	$h(x_t) = 0$ for $t_i < t \leq t_j$	Persistent drop	Consecutive abnormal readings caused by outage, tampering, or communication failure
Mean substitution	$h(x_t) = \text{mean}(\mathbf{x})$	Pattern smoothing	Abnormal flattening that removes normal temporal fluctuation
Random scaling	$h(x_t) = \alpha_t x_t$	Point drop	Sample-wise random under-reporting with varying attack intensity

Following the common assumption that anomalous events are rare in practical monitoring scenarios, the main anomaly ratio is set to 3%. Among the above injection patterns, the main experiment uses a scaling-based anomaly injection strategy to simulate abnormal load reduction:

$$h(x_t) = \alpha x_t, \quad (14)$$

where x_t denotes the original load value at time t , and $\alpha = 0.7$ in the main experiment. After injection, the dataset contains 8474 normal samples labeled as 0 and 262 anomalous samples labeled as 1. The detailed sample distribution is summarized in Table 2.

This strategy is adopted because it can generate abnormal load reductions while preserving the original temporal trend and periodic structure. The scaling factor is an experimental anomaly-intensity setting rather than an industrial threshold. To avoid drawing conclusions from a single artificial setting, additional robustness checks are designed with different anomaly intensities, positive anomalies, and persistent anomalies, as summarized in Table 3.

To avoid temporal information leakage, the dataset is split chronologically into training and testing sets with a ratio of 7:3, where the first 70% samples are used for training and the remaining 30% samples are used for testing.

Table 2. Sample distribution of the training and testing sets.

Subset	Normal	Anomalous	Total
Training set (70%)	5926	189	6115
Testing set (30%)	2548	73	2621
Total	8474	262	8736

Table 3. Anomaly-injection settings for robustness analysis.

Scenario	Injection Rule	Anomaly Pattern
Point drop 0.7	$h(x_t) = 0.7x_t$	Negative point
Point drop 0.8	$h(x_t) = 0.8x_t$	Negative point
Point increase 1.2	$h(x_t) = 1.2x_t$	Positive point
Persistent drop 0.7	$h(x_t) = 0.7x_t$ for 8 consecutive samples	Persistent

4.2. Evaluation Metrics

To evaluate anomaly detection performance, five widely used metrics are adopted, including Accuracy, Precision, Recall, F1-score, and the area under the ROC curve (AUC). These metrics are computed based on the confusion matrix shown in Table 4, where label 0 denotes normal samples (negative class) and label 1 denotes anomalous samples (positive class).

Table 4. Confusion matrix for anomaly detection.

	Predicted Normal	Predicted Anomaly
Actual Normal	TN	FP
Actual Anomaly	FN	TP

Accuracy, Precision, Recall, and F1-score are defined as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (15)$$

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (16)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (17)$$

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (18)$$

In addition, AUC is reported to evaluate the discriminative capability under imbalanced data settings, since anomalous samples account for only 3% of the dataset.

4.3. Prophet Decomposition Results

In the proposed framework, Prophet is first applied to decompose the power load series into trend, seasonal, holiday, and residual components. The corresponding decomposition results are shown in Figures 7 and 8.

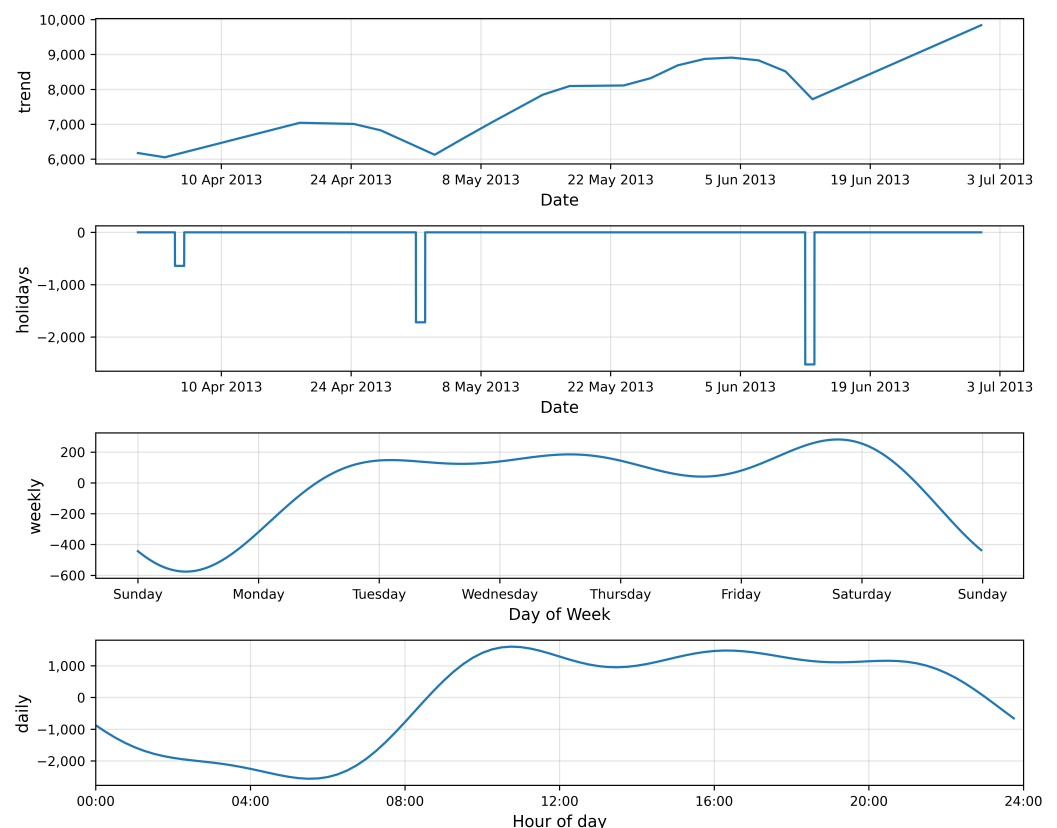


Figure 7. Seasonal-trend decomposition components obtained by Prophet.

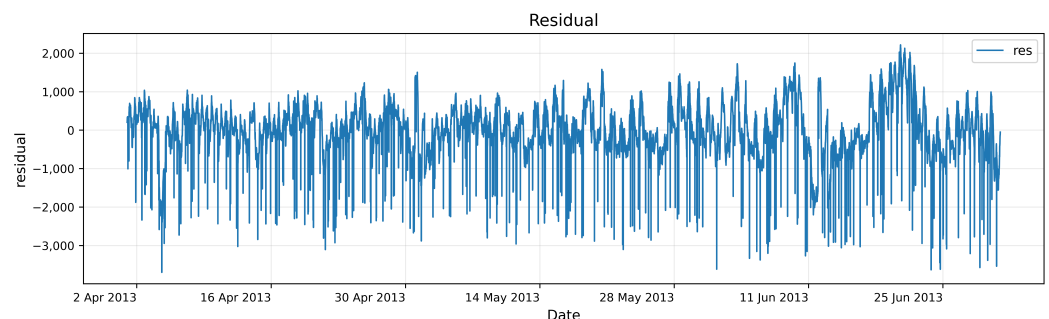


Figure 8. Residual component after Prophet decomposition.

As shown in Figure 7, the trend component indicates an overall increasing pattern of electricity load during the studied period. The holiday component shows notable decreases around Ching Ming Festival, May Day, and Dragon Boat Festival, suggesting that holidays

may lead to reduced electricity consumption. The weekly and daily seasonal components capture periodic load variations at different temporal scales.

From the residual sequence in Figure 8, it can be observed that the residual still exhibits structured fluctuations rather than purely random noise. Therefore, secondary decomposition is further conducted on the residual component using EEMD.

4.4. Residual Complexity Analysis

To quantitatively evaluate whether the residual component obtained from Prophet decomposition still contains meaningful information, sample entropy analysis and statistical white-noise hypothesis testing are conducted. Sample entropy measures the irregularity and complexity of a sequence; a higher value indicates stronger complexity and less regularity. The residual component shows higher sample entropy than the trend and seasonal components, suggesting that Prophet does not fully extract all informative temporal structures from the original load series. To further verify this observation, the Box–Pierce test and the Ljung–Box test are used to examine whether the autocorrelations of the residual series are significantly different from zero.

Table 5 reports the test results under multiple lag orders. The p -values are consistently far below the significance level of 0.05, indicating that the null hypothesis of white noise can be rejected. Therefore, the Prophet residual component still contains structured temporal dependence and latent multi-scale patterns, which motivates the use of EEMD for secondary decomposition and further feature extraction.

Table 5. White noise test results of Prophet residual series.

Lag	Box–Pierce Test		Ljung–Box Test	
	bp_stat	bp_pvalue	lb_stat	lb_pvalue
1	24.0077	9.595×10^{-7}	24.3696	7.951×10^{-7}
2	30.3169	2.611×10^{-7}	30.8063	2.044×10^{-7}
6	46.8931	1.965×10^{-8}	47.8313	1.277×10^{-8}
12	50.6715	1.064×10^{-6}	51.8623	6.558×10^{-7}

4.5. EEMD Secondary Decomposition Results

EEMD is applied to further decompose the Prophet residual component into multiple intrinsic mode functions (IMFs) at different time scales. The decomposition results are shown in Figure 9.

As shown in Figure 9, the residual component is decomposed into 13 intrinsic mode functions, denoted as $\text{eIMF}_1, \text{eIMF}_2, \dots, \text{eIMF}_{13}$. The first several IMFs mainly capture high-frequency fluctuations, while the frequency and amplitude gradually decrease in higher-order IMFs, indicating that multi-scale temporal patterns are effectively extracted.

4.6. Main Detection and Ablation Results

The main detection experiment is conducted under the original point-drop setting, namely $h(x_t) = 0.7x_t$. This setting is used as the primary controlled scenario because it follows the original experimental design and provides a common basis for comparing the proposed Prophet–EEMD–LSTM (PEL) model with baseline and ablation models. Figure 10 shows the confusion matrix of the proposed PEL model, and Figure 11 presents its ROC curve.

As shown in Figure 10, the PEL model correctly detects all anomalous samples in the testing set while keeping the number of false alarms low. The ROC curve in Figure 11 further indicates strong discrimination under the imbalanced anomaly-detection setting.

To avoid changing the original comparison basis, the revised manuscript keeps the original baseline results and adds EEMD–LSTM as an additional ablation model. The original comparison includes GRU, SVM, Prophet, LSTM, EMD–LSTM, Prophet–LSTM, Prophet–EMD–LSTM, and the full PEL model. The newly added EEMD–LSTM model applies EEMD directly to the original load sequence without the preceding Prophet decomposition, thereby directly answering the reviewer question of what happens when EEMD is used without Prophet. All models are evaluated under the same data split, anomaly-injection setting, preprocessing strategy, and training configuration.

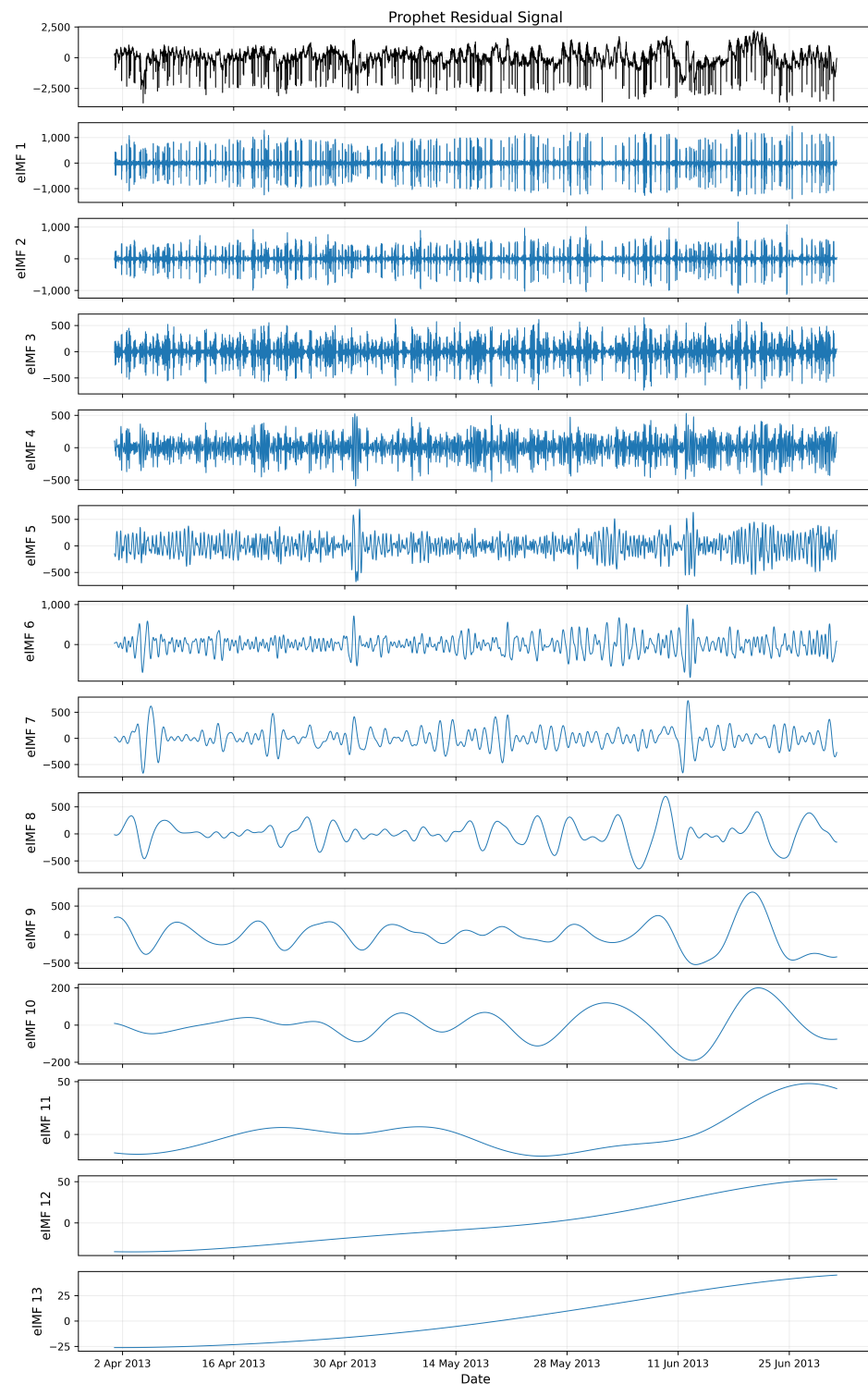


Figure 9. EEMD-based secondary decomposition of the residual series.

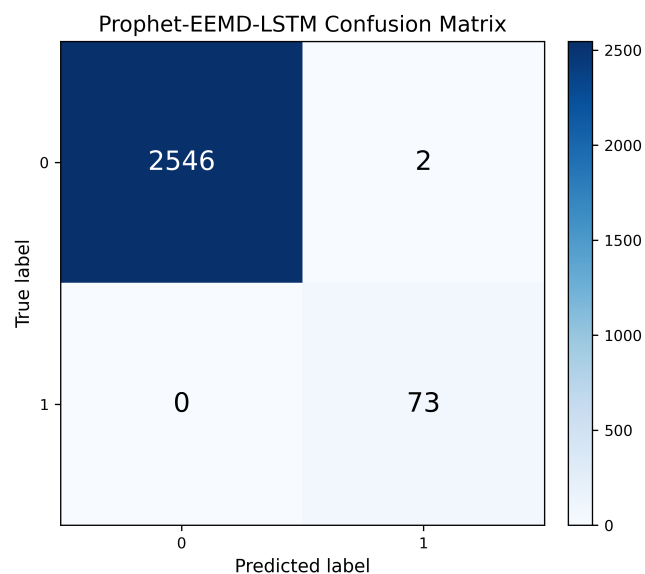


Figure 10. Confusion matrix of the proposed PEL model.

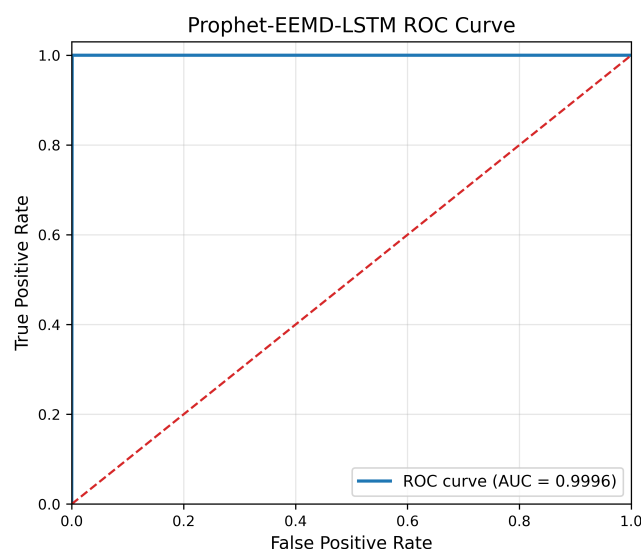


Figure 11. ROC curve of the proposed PEL model.

The red dashed diagonal line in the ROC curve denotes the random-classifier baseline. The main results are summarized in Table 6.

Table 6. Main anomaly-detection results and representative references under the point-drop 0.7 setting.

Model	Ref.	Accuracy	Precision	Recall	F1-Score	AUC
GRU	[8]	0.9596	0.2105	0.1644	0.1846	0.5734
SVM	[1]	0.9630	0.2272	0.1370	0.1709	0.5618
Prophet	[22]	0.9405	0.2872	0.7671	0.4179	0.8563
LSTM	[27]	0.9683	0.2075	0.1507	0.1746	0.5671
EMD-LSTM	[3]	0.9969	0.9333	0.9589	0.9459	0.9900
EEMD-LSTM	[23]	0.9939	0.8608	0.9315	0.8947	0.9636
Prophet-LSTM	[11]	0.9870	0.7746	0.7534	0.7639	0.8736
Prophet-EMD-LSTM	[3,11]	0.9989	0.9605	1.0000	0.9799	0.9994
Prophet-EEMD-LSTM	This study	0.9992	0.9733	1.0000	0.9865	0.9996

Table 6 keeps the original baseline results and adds the reviewer-requested EEMD-LSTM ablation. The “Ref.” column lists representative studies related to the corresponding

baseline or hybrid modeling strategy, while the full Prophet–EEMD–LSTM model is the proposed method of this study. The comparison between LSTM and the decomposition-based variants evaluates whether adaptive decomposition is useful. The comparison between EEMD–LSTM and PEL evaluates whether applying EEMD directly to the original sequence is sufficient, or whether Prophet-based seasonal-trend decomposition before residual decomposition provides additional value. The comparison between Prophet–LSTM and the Prophet-based residual-decomposition models further evaluates the contribution of secondary decomposition.

4.7. Robustness and Computational Cost Analysis

Because the main experiment uses injected anomalies, robustness checks are further conducted under different injection settings. The tested settings include the main point-drop setting, a weaker negative anomaly, a positive anomaly, and a persistent anomaly. The corresponding results are reported in Table 7.

Table 7. Robustness results under different anomaly-injection settings.

Scenario	EMD–LSTM	EEMD–LSTM	Prophet–LSTM	Prophet–EMD–LSTM	PEL
Point drop 0.7	0.9459	0.8947	0.9185	0.9726	0.9865
Point drop 0.8	0.9252	0.8844	0.9054	0.9467	0.9669
Point increase 1.2	0.9116	0.8707	0.8980	0.9396	0.9589
Persistent drop 0.7, 8 samples	0.7816	0.8114	0.8427	0.8778	0.9492
Mean F1	0.8911	0.8653	0.8911	0.9342	0.9654

Table 7 reports F1-score because the dataset is highly imbalanced and F1-score jointly reflects precision and recall for the minority anomalous class. The robustness settings are intended to avoid drawing conclusions from a single random point-drop case. In particular, the point increase setting tests whether the model can detect positive deviations, while the persistent-drop setting tests whether the method remains effective when consecutive measurements are affected.

Several observations can be made from Table 7. First, under point-type anomalies, EMD–LSTM performs better than EEMD–LSTM. For example, the F1-scores of EMD–LSTM and EEMD–LSTM are 0.9459 and 0.8947 under the point-drop 0.7 setting, 0.9252 and 0.8844 under the point-drop 0.8 setting, and 0.9116 and 0.8707 under the point-increase 1.2 setting, respectively. This phenomenon suggests that, when the abnormal pattern is an isolated abrupt change, the noise-assisted ensemble averaging in EEMD may smooth part of the local fluctuation that is useful for point-anomaly discrimination. In contrast, under the persistent-drop setting, EEMD–LSTM obtains a higher F1-score than EMD–LSTM (0.8114 vs. 0.7816), indicating that EEMD can be more helpful when the anomaly affects consecutive samples and has a sustained temporal structure. Second, the Prophet-based residual-decomposition models remain stronger than the direct decomposition baselines, and the full PEL model achieves the highest mean F1-score of 0.9654. This result indicates that separating trend and seasonal components before residual decomposition improves robustness across different injected-anomaly settings.

Finally, the computational cost of each model is recorded to show the trade-off between detection performance and runtime. The runtime experiment is conducted on an NVIDIA GeForce RTX 3080 Ti GPU with PyTorch 2.9.1+cu130. The results are summarized in Table 8. The proposed PEL framework introduces additional decomposition cost because Prophet and EEMD add preprocessing steps before LSTM training; therefore, the runtime table is used to evaluate whether the improved detection performance justifies the additional computational cost.

Table 8. Computational cost comparison under the main point-drop setting.

Model	Decomp. (s)	Train (s)	Infer. (s)	Total (s)
LSTM	0.00	35.12	0.02	35.13
EMD–LSTM	4.80	36.60	0.00	41.40
EEMD–LSTM	4.26	34.08	0.00	38.34
Prophet–LSTM	7.25	35.28	0.00	42.53
Prophet–EMD–LSTM	11.93	35.40	0.00	47.34
Prophet–EEMD–LSTM	15.64	35.94	0.00	51.58

The runtime experiment records decomposition time, LSTM training time, inference time, and total time separately. This design makes it possible to discuss the accuracy–cost trade-off explicitly: direct LSTM has the lowest decomposition cost, EMD/EEMD-based methods add adaptive decomposition cost, and Prophet-based hybrid models introduce additional seasonal-trend decomposition cost before LSTM training. Therefore, the runtime table is interpreted together with the main and robustness results rather than as an isolated speed comparison.

5. Conclusions and Future Work

This paper investigates the problem of anomaly detection in power load time series. To address the challenges brought by non-stationarity, multi-seasonality, temporal dependence, and complex residual fluctuations, a Prophet–EEMD–LSTM (PEL) anomaly detection framework is proposed. In the proposed method, Prophet is first applied to extract interpretable trend, seasonal, and holiday components, while the residual component is further decomposed using EEMD to obtain multi-scale intrinsic mode functions. These decomposition components are then normalized and fed into a multilayer LSTM network for anomaly identification. Experimental results on a real-world power load dataset demonstrate that the proposed PEL model achieves strong and competitive performance compared with several baseline methods, obtaining an accuracy of 0.9992, a precision of 0.9733, a recall of 1.0000, an F1-score of 0.9865, and an AUC of 0.9996 under the main injected-anomaly setting. In response to the limitations of relying on a single artificial anomaly setting, the revised experimental design further incorporates EEMD–LSTM ablation, multiple anomaly-injection scenarios, and computational-cost analysis to evaluate component contributions, robustness, and the accuracy–cost trade-off more comprehensively.

Nevertheless, this study still has several limitations. First, because reliable real anomaly labels are unavailable in the selected public load dataset, the quantitative evaluation is based on controlled anomaly injection. Although robustness checks are added with different injection settings, further validation on datasets with verified real events is still needed. Second, the current framework is mainly designed for offline detection and has not been extended to streaming scenarios with real-time constraints. Third, the model performance may be affected by dataset characteristics and hyperparameter settings, and further improvements are needed to enhance robustness and reduce manual tuning. Fourth, the proposed method focuses on detection accuracy but does not explicitly provide interpretability for explaining the detected anomalies.

Future work will be conducted in the following directions. (1) Online anomaly detection: extending the PEL framework to streaming power load data by integrating incremental learning or online updating mechanisms to support real-time anomaly monitoring. (2) Adaptive optimization: exploring automated hyperparameter optimization and adaptive decomposition strategies to improve model generalization under distribution shifts. (3) Model explainability: incorporating explainable artificial intelligence (XAI) techniques to provide interpretable anomaly explanations, such as feature contribution analysis and

visualization of temporal patterns, thereby improving model transparency and trustworthiness in practical power system applications.

Author Contributions: Conceptualization, L.W. and Y.G.; methodology, Y.G.; software, Y.G.; validation, Y.G. and X.Z.; formal analysis, Y.G.; investigation, Y.G.; resources, L.W.; data curation, Y.G.; writing—original draft preparation, Y.G.; writing—review and editing, L.W. and X.Z.; visualization, Y.G.; supervision, L.W. and X.Z.; project administration, L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this study are from the Area2_Load dataset released in the 2016 Electrician’s Cup Mathematical Modeling Competition (Problem A). Additional experimental materials are available from the corresponding author upon reasonable request.

Acknowledgments: The authors would like to thank the Party Building and Ideological-Political Work Research Project of Beijing Information Science and Technology University for supporting this study (Project title: Research on the Application of Artificial Intelligence and Big Data in University Party Building Work). We are grateful for their academic and policy support, which made this research possible.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AUC	Area Under the Receiver Operating Characteristic Curve
EEMD	Ensemble Empirical Mode Decomposition
EMD	Empirical Mode Decomposition
IMF	Intrinsic Mode Function
LSTM	Long Short-Term Memory
PEL	Prophet–EEMD–LSTM
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine

References

1. Chen, Y.; Xiao, C.; Yang, S.; Yang, Y.; Wang, W. Research on long term power load grey combination forecasting based on fuzzy support vector machine. *Comput. Electr. Eng.* **2024**, *116*, 109205. [\[CrossRef\]](#)
2. Li, Z.; Liu, S.; Zhang, Y.; Wang, Z.; Wang, L.; Huang, L. Short and Medium-Term Power Load Anomaly Detection Method Based on Convolutional Neural Network and EL-DCC. *IEEE Access* **2025**, *13*, 136656–136670. [\[CrossRef\]](#)
3. Mounir, N.; Ouadi, H.; Jrhilifa, I. Short-term electric load forecasting using an EMD-BI-LSTM approach for smart grid energy management system. *Energy Build.* **2023**, *288*, 113022. [\[CrossRef\]](#)
4. Lin, R.; Chen, S.; He, Z.; Wu, B.; Zou, H.; Zhao, X.; Li, Q. Electricity Behavior Modeling and Anomaly Detection Services Based on a Deep Variational Autoencoder Network. *Energies* **2024**, *17*, 3904. [\[CrossRef\]](#)
5. Tang, L.; Wang, C.; Wang, S. Energy time series data analysis based on a novel integrated data characteristic testing approach. *Procedia Comput. Sci.* **2013**, *17*, 759–769. [\[CrossRef\]](#)
6. Ghasemi, A.A.; Gitizadeh, M. Detection of illegal consumers using pattern classification approach combined with Levenberg-Marquardt method in smart grid. *Int. J. Electr. Power Energy Syst.* **2018**, *99*, 363–375. [\[CrossRef\]](#)
7. Tariq, M.; Poor, H.V. Electricity theft detection and localization in grid-tied microgrids. *IEEE Trans. Smart Grid* **2016**, *9*, 1920–1929. [\[CrossRef\]](#)
8. Aseeri, A.O. Effective RNN-based forecasting methodology design for improving short-term power load forecasts: Application to large-scale power-grid time series. *J. Comput. Sci.* **2023**, *68*, 101984. [\[CrossRef\]](#)

9. Park, C.H.; Kim, T. Energy Theft Detection in Advanced Metering Infrastructure Based on Anomaly Pattern Detection. *Energies* **2020**, *13*, 3832. [[CrossRef](#)]
10. Hasan, M.N.; Toma, R.N.; Nahid, A.A.; Islam, M.M.M.; Kim, J.M. Electricity Theft Detection in Smart Grid Systems: A CNN-LSTM Based Approach. *Energies* **2019**, *12*, 3310. [[CrossRef](#)]
11. Bashir, T.; Haoyong, C.; Tahir, M.F.; Liqiang, Z. Short term electricity load forecasting using hybrid prophet-LSTM model optimized by BPNN. *Energy Rep.* **2022**, *8*, 1678–1686. [[CrossRef](#)]
12. Razavi, R.; Gharipour, A.; Fleury, M.; Akpan, I.J. A Practical Feature-Engineering Framework for Electricity Theft Detection in Smart Grids. *Appl. Energy* **2019**, *238*, 481–494. [[CrossRef](#)]
13. Aoufi, S.; Derhab, A.; Guerroumi, M.; Guemmouma, H.; Lazali, H. LITE-FORT: Lightweight Three-Stage Energy Theft Detection Based on Time Series Forecasting of Consumption Patterns. *Electr. Power Syst. Res.* **2023**, *225*, 109840. [[CrossRef](#)]
14. Wang, X.; Yao, Z.; Papaefthymiou, M. A real-time electrical load forecasting and unsupervised anomaly detection framework. *Appl. Energy* **2023**, *330*, 120279. [[CrossRef](#)]
15. Li, G.; Jung, J.J. Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges. *Inf. Fusion* **2023**, *91*, 93–102. [[CrossRef](#)]
16. Ganjkhani, M.; Abbaspour, A.; Fattaheian-Dehkordi, S.; Gholami, M.; Lehtonen, M. Application of machine learning in determining and resolving state estimation anomalies in power systems. *Sustain. Energy Grids Netw.* **2024**, *38*, 101335. [[CrossRef](#)]
17. Wang, F.; Zhang, Z.; Wu, K.; Jian, D.; Chen, Q.; Zhang, C.; Dong, Y.; He, X.; Dong, L. Artificial intelligence techniques for ground fault line selection in power systems: State-of-the-art and research challenges. *Math. Biosci. Eng.* **2023**, *20*, 14518–14549. [[CrossRef](#)] [[PubMed](#)]
18. Sun, Y.; Sun, X.; Hu, T.; Zhu, L. Smart Grid Theft Detection Based on Hybrid Multi-Time Scale Neural Network. *Appl. Sci.* **2023**, *13*, 5710. [[CrossRef](#)]
19. Karkhaneh, M.; Ozgoli, S. Anomalous load profile detection in power systems using wavelet transform and robust regression. *Adv. Eng. Inform.* **2022**, *53*, 101639. [[CrossRef](#)]
20. Li, C.; Li, X.; Liu, X.; Zhao, L.; Xiao, D.; Hao, X.; Xie, J.; Xu, X. Research and application of power system data anomaly identification based on time series and deep learning. *J. Phys. Conf. Ser.* **2021**, *2095*, 012021.
21. Priyadarshini, I.; Alkhayyat, A.; Gehlot, A.; Kumar, R. Time series analysis and anomaly detection for trustworthy smart homes. *Comput. Electr. Eng.* **2022**, *102*, 108193. [[CrossRef](#)]
22. Satrio, C.B.A.; Darmawan, W.; Nadia, B.U.; Hanafiah, N. Time series analysis and forecasting of coronavirus disease in Indonesia using ARIMA model and PROPHET. *Procedia Comput. Sci.* **2021**, *179*, 524–532. [[CrossRef](#)]
23. Li, D.; Jiang, M.R.; Li, M.W.; Hong, W.C.; Xu, R.Z. A floating offshore platform motion forecasting approach based on EEMD hybrid ConvLSTM and chaotic quantum ALO. *Appl. Soft Comput.* **2023**, *144*, 110487. [[CrossRef](#)]
24. Taheri, S.; Talebjedi, B.; Laukkanen, T. Electricity demand time series forecasting based on empirical mode decomposition and long short-term memory. *Energy Eng. J. Assoc. Energy Eng.* **2021**, *118*, 1577–1594. [[CrossRef](#)]
25. Zhang, Y.; Li, C.; Jiang, Y.; Sun, L.; Zhao, R.; Yan, K.; Wang, W. Accurate prediction of water quality in urban drainage network with integrated EMD-LSTM model. *J. Clean. Prod.* **2022**, *354*, 131724. [[CrossRef](#)]
26. Liu, Z.; Wang, X.; Zhang, Q.; Huang, C. Empirical mode decomposition based hybrid ensemble model for electrical energy consumption forecasting of the cement grinding process. *Measurement* **2019**, *138*, 314–324. [[CrossRef](#)]
27. Sahu, S.K.; Mohapatra, D.P.; Rout, J.K.; Sahoo, K.S.; Pham, Q.V.; Dao, N.N. A LSTM-FCNN based multi-class intrusion detection using scalable framework. *Comput. Electr. Eng.* **2022**, *99*, 107720. [[CrossRef](#)]
28. Jokar, P.; Arianpoo, N.; Leung, V.C.M. Electricity Theft Detection in AMI Using Customers' Consumption Patterns. *IEEE Trans. Smart Grid* **2016**, *7*, 216–226. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.