sustainability

MDPI

Article

Short-Term Load Forecasting for Regional Smart Energy Systems Based on Two-Stage Feature Extraction and Hybrid Inverted Transformer

Zhewei Huang † and Yawen Yi *,†

School of Electrical and Electronic Engineering, Huazhong University of Science and Technology, Wuhan 430074, China; whztever@hust.edu.cn

* Correspondence: yawenyi@hust.edu.cn

† These authors contributed to the work equally and should be regarded as co-first authors.

Abstract: Accurate short-term load forecasting is critical for enhancing the reliability and stability of regional smart energy systems. However, the inherent challenges posed by the substantial fluctuations and volatility in electricity load patterns necessitate the development of advanced forecasting techniques. In this study, a novel short-term load forecasting approach based on a two-stage feature extraction process and a hybrid inverted Transformer model is proposed. Initially, the Prophet method is employed to extract essential features such as trends, seasonality and holiday patterns from the original load dataset. Subsequently, variational mode decomposition (VMD) optimized by the IVY algorithm is utilized to extract significant periodic features from the residual component obtained by Prophet. The extracted features from both stages are then integrated to construct a comprehensive data matrix. This matrix is then inputted into a hybrid deep learning model that combines an inverted Transformer (iTransformer), temporal convolutional networks (TCNs) and a multilayer perceptron (MLP) for accurate short-term load forecasting. A thorough evaluation of the proposed method is conducted through four sets of comparative experiments using data collected from the Elia grid in Belgium. Experimental results illustrate the superior performance of the proposed approach, demonstrating high forecasting accuracy and robustness, highlighting its potential in ensuring the stable operation of regional smart energy systems.

Keywords: load forecasting; Prophet; variational mode decomposition; IVY algorithm; inverted transformer; temporal convolutional network; multilayer perceptron

check for updates

Citation: Huang, Z.; Yi, Y. Short-Term Load Forecasting for Regional Smart Energy Systems Based on Two-Stage Feature Extraction and Hybrid Inverted Transformer. *Sustainability* **2024**, *16*, 7613. https://doi.org/10.3390/su16177613

Academic Editor: Behnam Mohammadi-Ivatloo

Received: 14 July 2024
Revised: 19 August 2024
Accepted: 30 August 2024
Published: 2 September 2024

Copyright: © 2024 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) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, global decarbonization efforts have intensified to mitigate climate change, with the transition toward renewable and smart energy systems becoming a key initiative in various regions. Regional smart energy systems require reliability and flexibility, integrating electricity, heating and transportation sectors to manage demand–supply fluctuations across various time scales [1]. Short-term load forecasting (STLF) is a critical component in the effective management and coordination of regional smart energy systems [2]. Precise STLF is crucial for maintaining an immediate equilibrium between electricity supply and consumption [3] and enhancing the dependability and robustness of the smart energy system. Furthermore, the emergence of regional smart energy systems and sophisticated metering technology has significantly increased the granularity and volume of load data. This presents new opportunities and challenges for regional STLF [4]. Electricity often exhibits nonlinear fluctuations. These fluctuations in load can be significantly influenced by unpredictable external factors such as weather conditions (temperature, humidity), economic activities and social events. Existing models may not incorporate these variables effectively, thus failing to account for sudden spikes or drops in demand. Therefore, developing robust and accurate regional STLF models

is paramount for optimizing power system operations, reducing energy wastage and facilitating the incorporation of renewable energy options [5].

Short-term load power forecasting techniques can be generally categorized into statistical fitting approaches and machine learning approaches [6]. Statistical fitting approaches such as regression analysis [7], exponential smoothing [8] and the autoregressive integrated moving average (ARIMA) [9] depend exclusively on past data and employ statistical frameworks to identify linear correlations within the data [10]. However, electricity demand often exhibits nonlinear and dynamic patterns influenced by numerous factors such as time of day, day of the week and seasonality. Traditional methods may not adequately capture these complexities, leading to suboptimal forecasts. In addition, these methods are limited to capturing only linear relationships. As the complexity of the data and prediction step size increase, these methods often stabilize around a consistent mean value, resulting in reduced prediction precision. Conversely, machine learning approaches have gained significant popularity for analyzing time-series data and are extensively applied in regional STLF [11]. Traditional machine learning techniques, including support vector machines [12] and random forests [13], along with deep learning techniques, leverage their powerful self-learning capabilities and robust nonlinear modeling abilities. Consequently, these machine learning approaches typically yield more accurate short-term load forecasts compared to statistical fitting methods [14].

As a prominent method in machine learning, deep learning has attracted considerable interest for STLF. Commonly employed models encompass Long Short-Term Memory (LSTM) networks [15], Convolutional Neural Networks (CNNs) [16], etc. For time-series forecasting, feature extraction from historical data plays a pivotal role. CNNs are extensively employed to derive features from time-series data. Alhussein et al. utilized CNNs for feature extraction from historical load data, combined with LSTM for future load forecasting [17]. Wang et al. integrated CNNs with the Informer, improving prediction precision and addressing the challenge of restricted time-series data on a singular temporal scale [18]. However, the feature extraction ability of CNNs is still limited, resulting in limited prediction accuracy. To further improve their ability, temporal neural networks (TCNs) were introduced [19]. TCNs overcome long-term dependency problems and performance declines by incorporating dilated causal convolutions and residual blocks, thereby exceeding CNNs in time-series feature extraction [20]. Gong et al. applied TCNs for hidden temporal feature extraction and the Informer for accurate short-term forecasting [20]. Zhou et al. proposed a load forecasting model incorporating Prophet and improved TCNs, using TCNs to extract the internal correlation in the input matrix obtained by Prophet. This helps to achieve high efficiency and prediction accuracy [21].

To enhance internal correlations and focus on crucial information in historical data, multi-level attention networks are being increasingly utilized and refined. In 2017, Vaswani et al. proposed a Transformer model that excels in capturing long-range dependencies through self-attention mechanisms, surpassing traditional recurrent neural networks in capturing complex dependencies [22]. Between the years of 2017 and 2024, many improvements have been made to enhance the capabilities of the Transformer. Zhou et al. proposed the Informer, which improved the Transformer by incorporating a hybrid architecture with temporal convolutional layers, enhancing its ability to model sequential data [23]. However, its increased complexity due to hybrid architecture may lead to higher computational costs and training time. Liu et al. introduced the iTransformer, an innovative model that applies attention and feed-forward networks to inverted dimensions [24]. This approach enhances the performance and generalization capabilities of the original Transformer, resulting in the best overall performance among all Transformer-based variants.

Electricity load sequences are intricate, typically comprising various elements like trends, seasonal variations and random fluctuations. Deep learning models have limitations in capturing these complexities effectively. Consequently, diverse data preprocessing techniques are utilized to dissect the multiple components within load data for robust feature extraction. Mounir et al. employed empirical mode decomposition (EMD) to segment

input series into separate elements for independent prediction. They then amalgamated the predictions of each component to derive the final forecast [25]. Li et al. utilized complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) to segment the original time series into several different components. Then, guided by sample entropy, they were restructured into a dual-component series for further processing [26]. Yuan et al. leveraged variational mode decomposition (VMD) to decompose the input load series, effectively minimizing modal confusion and improving feature extraction capabilities [27]. However, these decomposition methods often encounter high computational complexity and struggle to capture specific trends and seasonal changes within load data. Stefenon et al. aggregated Prophet and seasonal and trend decomposition utilizing LOESS (STL) for robust time-series analysis, achieving high prediction accuracy [28]. Lin et al. devised a two-stage feature extraction method combining STL and time-lag autocorrelation to extract features from time series [29]. Wang et al. employed Prophet to extract trends and seasonal variations within load data and further decomposed residual data to extract several stable data features. This method effectively captures features within the intricate original time series [30]. Nonetheless, the feature extraction methods proposed in the aforementioned studies still lack the ability to extract features comprehensively. This is due to their oversight of significant events like holidays and limited capacity to extract periodic features from residual components.

To improve the extraction ability of periodic features from time series using decomposition methods, optimization techniques have been implemented to refine the parameters of the decomposition algorithms [31]. Zhao et al. introduced an enhanced snake optimization algorithm (ISOA) to fine-tune VMD parameters, consequently improving the extraction of periodic features from load data [32]. However, the improved snake optimization algorithm may suffer from slow convergence rates and sensitivity to parameter settings. Su et al. fine-tuned the VMD parameters using the sparrow algorithm to extract components with significant features, avoiding redundant and complex components [33]. Nevertheless, the sparrow algorithm's dependence on centralized coordination can lead to scalability issues and potential single points of failure. Wang et al. proposed an improved beluga whale optimization (IBWO) method, resulting in enhanced prediction accuracy [34]. However, the IBWO method may exhibit limited exploration capabilities and susceptibility to premature convergence on suboptimal solutions. The aforementioned methods either overlook the significance of the fitness function or the optimization capability of the algorithm, leading to constrained optimization of the VMD parameters.

To tackle the challenges mentioned above, a short-term electricity load forecasting method for regional smart energy systems based on two-stage feature extraction and a hybrid inverted Transformer is proposed. This study's primary contributions and innovations include the following:

- A novel hybrid method is proposed in this study, integrating Prophet, IVYVMD, the iTransformer and a Multi-Perception Temporal Convolutional Network (MPTCN) to enhance the precision of short-term load forecasting in regional smart energy systems. Specifically, the hybrid method achieves forecasting accuracies exceeding 97% for a six-hour forecasting horizon.
- A two-stage feature extraction (TFE) approach, utilizing Prophet and IVYVMD, is employed to construct a feature matrix encompassing trends, seasonality, events and multiple periodic components within residuals. This method generates ample features for subsequent model predictions, effectively reducing data volatility and complexity. Notably, this approach surpasses other feature extraction methods, demonstrating superior model prediction accuracy.
- Leveraging the IVY algorithm to fine-tune VMD parameters can improve the feature extraction ability of VMD and optimize model performance in short-term load forecasting, thereby enhancing load forecasting precision. Specifically, the IVY algorithm surpasses other commonly used optimization algorithms in fine-tuning VMD parameters, resulting in superior prediction performance.

- The experimental analysis in this study employs the dataset of the Elia grid in Belgium, utilizing four evaluation metrics and comparing 21 related models. The findings demonstrate that the proposed approach surpasses other models in both accuracy and consistency for load predictions.
- In this paper, Section 1 offers a detailed summary of the academic background, shedding light on the latest advancements in the research concerning short-term load forecasting. Section 2 delineates the overarching framework of the TFE-iTransformer-MPTCN method proposed in this research, elucidating the six primary methods, namely Prophet, VMD, IVY algorithm, iTransformer, TCN and MLP, in intricate detail. Section 3 conducts a comparative analysis of various benchmark models, demonstrating the overall outstanding performance of the short-term load forecasting model proposed.

2. Methods

To improve the overall forecasting ability of short-term load forecasting for regional smart energy systems, a short-term electricity load forecasting method based on two-stage feature extraction and hybrid inverted Transformer is proposed. The overall architecture of the proposed method is shown in Figure 1.

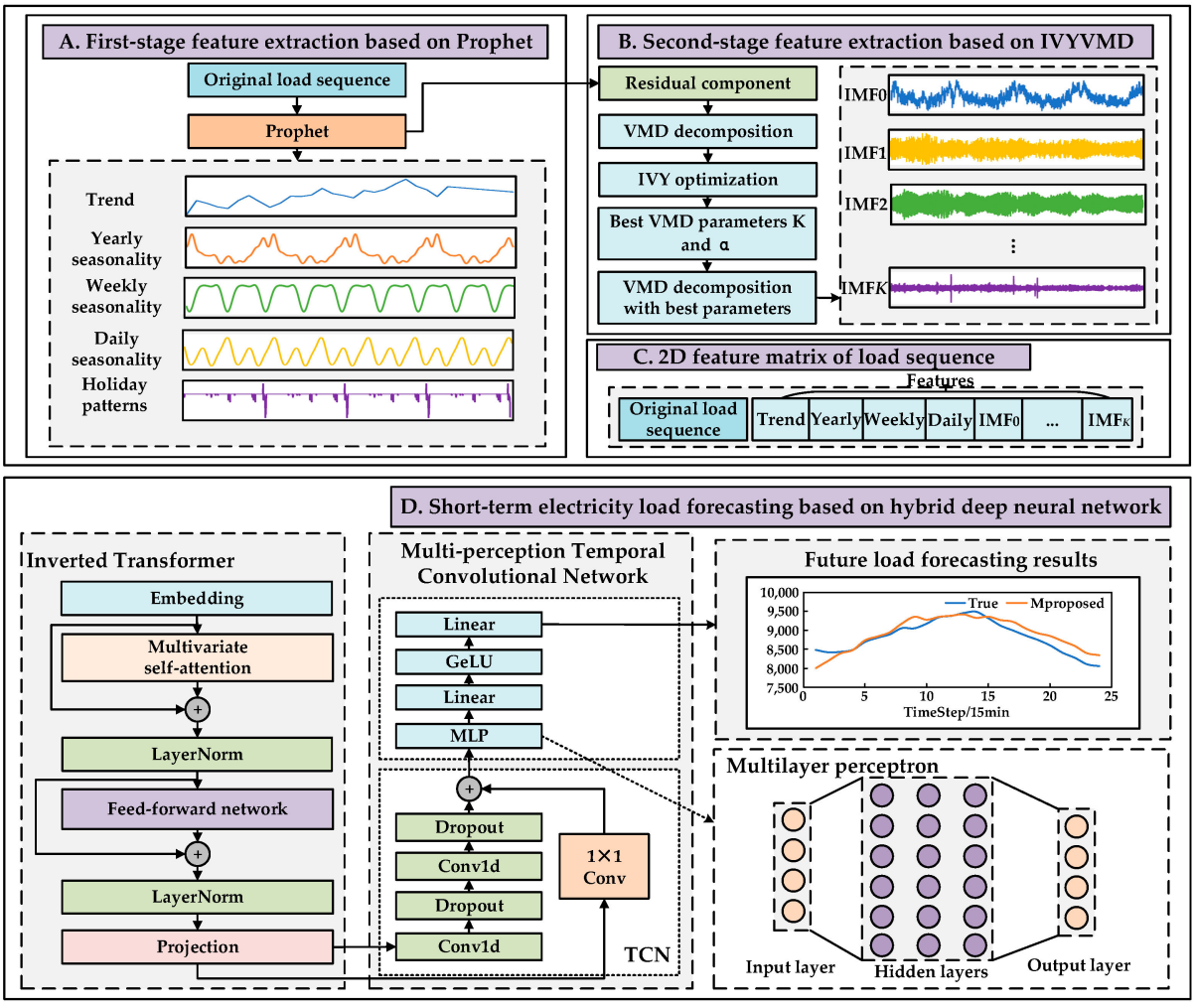


Figure 1. Overall architecture of proposed load forecasting method based on two-stage feature extraction and hybrid inverted Transformer.

As depicted in Figure 1, the proposed model comprises four distinct stages. Initially, the first-stage feature extraction using Prophet is applied to the original load series, isolating

components such as periodic fluctuations influenced by trends, yearly seasonality, weekly seasonality, daily seasonality and holidays. The residual component is subsequently derived by calculating the difference between these extracted components and the original load data. In the second stage, feature extraction leveraging the IVYVMD method is utilized to discern periodic patterns from the residual component produced by Prophet. This stage involves decomposing the residual into multiple periodic intrinsic mode functions (IMFs) with VMD parameters optimized by the IVY algorithm. This enhances the precision of periodic feature extraction, thereby facilitating improved accuracy in subsequent load forecasting. Subsequently, the features extracted from both the first and second stages are concatenated to form a comprehensive two-dimensional feature matrix. This matrix is then input into a hybrid iTransformer model designed for forecasting future electricity load. The hybrid iTransformer model integrates two principal components: the inverted Transformer (iTransformer) and the Multi-Perception Temporal Convolutional Network (MPTCN). The iTransformer is employed to extract internal correlations within the feature matrix, generating a feature map that highlights critical information. Subsequently, the MPTCN is utilized to further refine feature extraction and enhance the recognition of long-term dependencies from the iTransformer's output feature map.

2.1. First-Stage Feature Extraction Based on Prophet

The Prophet algorithm, introduced by Taylor et al. in 2017, is specifically designed for analyzing and forecasting intricate time series through decomposition to obtain the trend, seasonality and holiday components [35]. Traditional trend-cycle time-series decomposition methods, such as STL, exhibit limited flexibility in modeling seasonal effects and lack precision in feature extraction, which hinders prediction accuracy. In contrast, signal processing methods like EMD, CEEMDAN and VMD demand substantial computational resources for initial feature extraction. Prophet, however, enables rapid iterative optimization, requires fewer computational resources and effectively captures various periodicities and holiday effects in time-series data. This enhances a comprehensive understanding of complex load data, demonstrating exceptional flexibility and robustness in short-term load forecasting.

Load data are a complex sequence exhibiting diverse periodic patterns across different time scales. It can be particularly susceptible to fluctuations during special circumstances such as holidays, which may disrupt trends and periodic patterns. Consequently, this study employs the Prophet algorithm for the first-stage feature extraction from original load data. This process extracts stable and regular features including trend, yearly, weekly, daily and holiday seasonal components. For a specific load sequence, it can be defined as follows:

$$L(t) = g(t) + p(t) + h(t) + r(t) \tag{1}$$

where $g(t)$ represents the trend function in load data; $p(t)$ represents yearly, weekly and daily periodic variations; $h(t)$ represents the holiday seasonal pattern; $r(t)$ is the residual component representing idiosyncratic variations in load sequence.

2.1.1. Trend Extraction

In Prophet, two trend models are implemented: a saturating growth model and a piecewise linear model [34]. In this study, the piecewise linear model is chosen for precise outlining of the trend shifts within electricity load data, due to the lack of features of saturated growth. The piecewise linear model allows for changes in the trend's slope at change points, accommodating situations where the growth rate changes over time. This flexibility helps to better capture sudden shifts in trends compared to a saturating growth model that assumes a gradual leveling off. In addition, the piecewise linear model, by focusing on specific change points, can help mitigate the risk of overfitting and produce

more generalized forecasts compared with the saturating growth model. The trend function employed is defined in Equations (2) and (3):

$$g(t) = \left(p + \alpha^T(t)\delta\right)t + \left(q + \alpha^T(t)\gamma\right) \tag{2}$$

$$\lambda_j(t) = \begin{cases} 1, \ t \geq s_j \\ 0, \ otherwise \end{cases} \tag{3}$$

where p is the basic growth rate of the electricity load; δ is the load growth rate adjustment vector; $\alpha(t)$ is the indicator function; $\lambda_j(t)$ is the value of $\alpha(t)$ at the jth mutation point; s_j (j = 1, 2, $\cdots$, S) is the time when the jth mutation point in the load sequence is located; q is the offset parameter; γ is the smoothing offset at the mutation point; $p + \alpha^T(t)\delta$ is the growth rate of the substation load at time t.

2.1.2. Seasonality Extraction

The electricity load sequence exhibits multiple seasonal patterns, encompassing yearly, weekly and daily variations. Prophet utilizes the Fourier series to effectively capture the periodicity, providing a flexible approach to extract seasonal variations within the load data. This methodology is articulated in Equations (4) and (5):

$$p(t) = \sum_{m}^{M}\left(a_m \cos\left(\frac{2\pi mt}{P}\right) + b_m \sin\left(\frac{2\pi mt}{P}\right)\right) \tag{4}$$

$$\beta = [a_1, b_1, a_2, b_2, \cdots, a_M, b_M] \tag{5}$$

where P represents the consistent cycles in the input load sequence; β is the set of smoothing coefficients a_m and b_m, which satisfies the normal distribution; M is the number of smoothing coefficients a_m or b_m; m is the serial number of the smoothing coefficient.

2.1.3. Holiday Extraction

The electricity load data exhibit fluctuations influenced by holiday factors. Consequently, Prophet is employed to extract these holiday factors from the load sequence, facilitating the derivation of a holiday periodic feature for subsequent analysis. Specifically, the Prophet method necessitates manual holiday configuration. Users can either select an embedded holiday dataset for a specific country or create a customized holiday dataset. The Prophet model incorporates holidays as regression factors, enabling adjustments to forecasts for significant fluctuations during these periods. This incorporation enhances forecasting accuracy by allowing the model to capture anomalous patterns and trends associated with holidays and special events. The holiday component is delineated in Equations (6) through (8):

$$h(t) = U(t)\kappa \tag{6}$$

$$U(t) = [1(t \in Z_1), 1(t \in Z_2), \cdots, 1(t \in Z_D)] \tag{7}$$

$$\kappa \in N\left(0, \theta^2\right) \tag{8}$$

where $U(t)$ denotes the regression matrix; Z_d represents the set of past and future dates of the d-th holiday, where d = 1, 2, $\cdots$, D, emcompassing a total of D days; κ is the prior variation parameter corresponding to the holiday, which conforms to a normal distribution; θ quantifies the influence of the holiday, with larger θ indicating a greater imapct.

2.2. Second-Stage Feature Extraction Based on IVYVMD

2.2.1. Variational Mode Decomposition

Signal decomposition methods such as EMD, EEMD and CEEMDAN may lead to significant mode mixing or demand excessive computational resources. Unlike these methods, variational mode decomposition (VMD) is a robust, non-iterative method for signal

decomposition that effectively avoids aliasing by controlling bandwidth, thereby mitigating the modal aliasing phenomenon [36]. Therefore, VMD is chosen as the decomposition method in the second-stage feature extraction. VMD is capable of decomposing a signal into intrinsic mode functions (IMFs) across multiple center frequencies, which simplifies the complexity of the original signal and highlights its regularity.

In this study, VMD is applied to extract periodic features from the residual component resulting from the initial stage of feature extraction. This enriches the dataset with sufficient periodic features for subsequent short-term load forecasting. Assume the original load sequence is denoted by $L(t)$, and the constraint model for VMD is defined in Equation (9):

$$\begin{aligned} &\min_{\{u_k,\omega_k\}} \left\{ \sum_{k=1}^{K} \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \right\} \\ &s.t. \sum_{k=1}^{K} u_k = L(t) \end{aligned} \tag{9}$$

where $L(t)$ represents the load series; $\delta(t)$ denotes the Dirac distribution; $*$ denotes the convolution operator; j is the imaginary unit; $\{u_k\} = \{u_1, \ldots, u_K\}$ is the set of modes obtained by decomposition; $\{\omega_k\} = \{\omega_1, \ldots, \omega_K\}$ is the set of center frequencies associated with each mode; K is the modal number.

The VMD technique incorporates a quadratic penalty factor and a Lagrangian multiplication operator to maintain signal reconstruction fidelity and enforce constraint conditions within the objective function. The extended Lagrangian expression is presented in Equation (10):

$$Lg(\{u_k\}, \{\omega_k\}, \lambda) = \alpha \sum_{k=1}^{K} \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 + \left\| L(t) - \sum_{k=1}^{K} u_k(t) \right\|_2^2 + \left\langle \lambda(t), L(t) - \sum_{k=1}^{K} u_k(t) \right\rangle \tag{10}$$

where α is the quadratic penalty factor; λ is the Lagrangian multiplication operator.

The Alternating Direction Method of Multipliers (ADMM) is implemented to solve Equations (9) and (10), yielding updated formulas for u_k, ω_k and λ. These variables are iteratively refined to derive the final IMFs and their center frequencies. The updated formulas are specified in Equations (11)–(13), and the convergence criterion for the iteration process is outlined in Equation (14):

$$\hat{u}_k^{n+1}(\omega) = \frac{\hat{L}(\omega) - \sum_{i \neq k} \hat{u}_i(\omega) + \frac{\hat{\lambda}(\omega)}{2}}{1 + 2\alpha(\omega - \omega_k)^2} \tag{11}$$

$$\omega_k^{n+1} = \frac{\int_0^\infty \omega |\hat{u}_k(\omega)|^2 d\omega}{\int_0^\infty |\hat{u}_k(\omega)|^2 d\omega} \tag{12}$$

$$\hat{\lambda}^{n+1}(\omega) = \hat{\lambda}^n(\omega) + \tau \left(\hat{f}(\omega) - \sum_{k=1}^{K} \hat{u}_k^{n+1}(\omega) \right) \tag{13}$$

$$\sum_{k=1}^{K} \left|\left| u_k^{n+1} - u_k^n \right|\right|_2^2 / ||u_k^n||_2^2 < \varepsilon \tag{14}$$

where $\hat{}$ represents the Fourier transformation function; n denotes the number of iterations; τ is the update factor; ε represents the minimum error defined during the iteration process.

2.2.2. Parameter Optimization Based on IVY Algorithm

For VMD, the modal number K and the quadratic penalty term α play a vital role in influencing the decomposition outcomes of the input time series. Inappropriate values for K and α values can result in either under- or overdecomposition. This leads to modal confusion or the creation of redundant modes, which in turn increase computational de-

mands. Such scenarios compromise the extraction of periodic and stable features from the residual component derived from the initial stage of feature extraction, ultimately affecting the forecasting precision. To enhance the effect of feature extraction from the residual component produced by Prophet, a novel optimization algorithm, the IVY algorithm (IVY), is introduced for optimizing VMD parameters K and α. Compared with other optimization methods such as the improved snake optimization algorithm [32] and improved sparrow search algorithm [33], the IVY algorithm demonstrates superior convergence rates and robust optimization capability, making it well-suited for optimizing VMD parameters. Additionally, IVY's distinctive ability to maintain population diversity, combined with its simplicity and flexibility, facilitates easy modifications and extensions. This allows researchers to implement the algorithm to improve the overall ability of specific tasks. The Pearson correlation coefficient is selected as the fitness function for the parameter optimization.

- IVY algorithm

The IVY algorithm is a bio-inspired metaheuristic based on the growth patterns of ivy plants [37]. It emulates ivy plant behavior by selecting the nearest and strongest neighbor for self-enhancement. It excels over many conventional metaheuristics in terms of optimization speed and computational complexity. Initially, the position of the IVY population in the search space is established with Equation (15):

$$A_i = A_{\min} + rand(1, M) \odot (A_{\max} - A_{\min}), i = 1, \cdots, Npop \tag{15}$$

where $Npop$ denotes the size of the population group; M represents decision variable size in the VMD parameter optimization problem; $rand(1, M)$ is a vector of M dimensions filled with random numbers uniformly distributed between 0 and 1; A_i represents the ith population member; $A_{\max}$ and $A_{\min}$ represent the maximum and minimum limits of the search area, respectively; $\odot$ indicates the Hadamard product.

The growth rate Gr of an ivy plant is modeled by a discrete function shown in Equation (16):

$$\Delta Gr_i(t+1) = rand^2 \odot (N(1, M) \odot \Delta Gr_i(t)) \tag{16}$$

where $\Delta Gr_i(t+1)$ and $\Delta Gr_i(t)$ denote the growth rates at consecutive time instants in a discrete-time system; $rand$ represents a number randomly generated with uniform distribution between 0 and 1; $rand^2$ is a value sampled from a random variable whose probability density function (PDF) equals $1/\left(2\sqrt{x}\right)$; $N(1, M)$ is a random vector of dimension M with components from the standard normal distribution.

In the simulation, each population member A_i improves by moving toward its nearest and strongest neighbor A_{ii} based on the fitness function, as illustrated in Equations (17) and (18):

$$A_i^{new} = A_i + |N(1, M)| \odot (A_{ii} - A_i) + N(1, M) \odot \Delta Gr_i, i = 1, 2, \cdots, Npop \tag{17}$$

$$\Delta Gr_i = \begin{cases} Hdiv(A_i, (A_{\max} - A_{\min})), & Iter = 1 \\ rand^2 \odot (N(1, M) \odot \Delta Gr_i), & Iter > 1 \end{cases} \tag{18}$$

where A_i^{new} is the improved version of the i-th member; $|N(1, M)|$ is a vector of absolute values of $N(1, M)$; $Hdiv()$ denotes Hadamard division.

The search process involves moving toward the best member A_{best} directly, simulating the behavior of finding optimal solutions around the best member. This phase is defined in Equations (19) and (20):

$$A_i^{new} = A_{best} \odot (rand(1, M) + N(1, M) \odot \Delta Gr_i) \tag{19}$$

$$\Delta Gr_i^{new} = Hdiv(A_i^{new}, (A_{\max} - A_{\min})) \tag{20}$$

where ΔGr_i^{new} is the updated growth rate of the current member A_i^{new}.

- Pearson correlation coefficient

The Pearson correlation coefficient (PCC) serves as the fitness function for the IVY algorithm in optimizing VMD parameters. It quantifies the direct correlation between two variables, facilitating the assessment of the relationship between the IMFs obtained by VMD and the original load sequence. The PCC varies between −1 to 1, with larger values suggesting a stronger retention of original information by the IMF, leading to superior decomposition outcomes. The calculation process is detailed in Equation (21):

$$\mathrm{PCC} = \frac{N\left(\sum_{i=1}^{N} X_i Y_i\right) - \left(\sum_{i=1}^{N} X_i\right)\left(\sum_{i=1}^{N} Y_i\right)}{\sqrt{\left[N\sum_{i=1}^{N} X_i{}^2 - \left(\sum_{i=1}^{N} X_i\right)^2\right]\left[N\sum_{i=1}^{N} Y_i{}^2 - \left(\sum_{i=1}^{N} Y_i\right)^2\right]}} \quad (21)$$

where PCC is the Pearson correlation coefficient; X_i represent the i-th data points of an IMF; Y_i represent the i-th data point of the original load sequence; N is the total number of data points.

- VMD parameter optimization based on IVY algorithm

By fine-tuning the VMD settings to improve the Pearson correlation coefficient of the signal, the efficacy of the second-stage feature extraction is significantly improved. The VMD parameter is optimized by the IVY algorithm. This optimization reduces noise and randomness in the residual component derived from Prophet, thereby enabling more precise extraction of periodic information pertinent to the original load sequence. This allows for more precise decomposition of the residual component, with more accurate center frequencies and less modal confusion. Such refinement aids subsequent deep learning models in more effectively learning precise features associated with historical load data, ultimately enhancing short-term load forecasting precision. The comprehensive process of parameter optimization using the IVY algorithm is depicted in Figure 2.

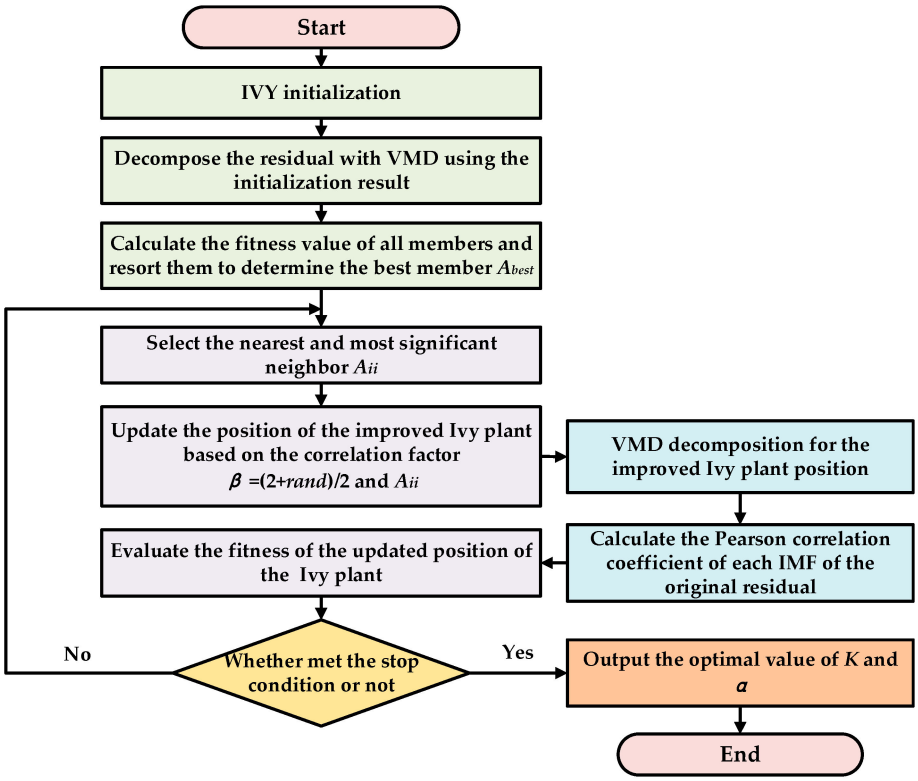


Figure 2. The flowchart of VMD parameter optimization based on IVY algorithm.

As illustrated in Figure 2, firstly, the IVY algorithm is initialized, and the residual is decomposed by VMD into multiple components with the initialized parameters. Secondly, calculate the fitness of all the members in the initialized population and resort them to determine the best member. Then, select the nearest and most significant neighbor and update the position of the improved ivy plant based on the neighbor and the correlation factor. With the improved parameters associated with the improved ivy plant position, VMD is applied to the residual again. And the Pearson correlation coefficient of each IMF obtained by VMD is calculated. With this coefficient, the fitness of the updated position of the ivy plant can be evaluated. This evaluation decides whether the current position meets the stop condition or not. If yes, output the optimal value of K and α. If not, keep updating the position of ivy plants until the current position meets the stop condition.

2.3. Short-Term Load Forecasting Based on Hybrid Transformer Model

In this study, a novel hybrid deep learning model for short-term load forecasting, the hybrid inverted Transformer model with Multi-Perception Temporal Convolutional Network (iTransformer-MPTCN) is introduced. This model integrates an enhanced inverted Transformer (iTransformer) with a Multi-Perception Temporal Convolutional Network (MPTCN). The iTransformer can efficiently capture long-range dependencies and complex patterns within sequential data. Its self-attention mechanism allows for adaptive focus on relevant features, improving predictive accuracy and robustness in dynamic environments compared to traditional models. Integrating MPTCN with iTransformer combines the strengths of both architectures, effectively capturing local temporal patterns through convolutions while leveraging self-attention for global dependencies, achieving better prediction ability compared with other traditional models. Initially, the iTransformer extracts correlations from the feature matrix generated by the two-stage feature extraction module and models global dependencies through its attention mechanism. Subsequently, the feature map output from the iTransformer, which focuses on significant information, is processed by the MPTCN to capture long-term dependencies. This dual approach enhances the comprehension of the feature map, leading to more accurate predictions based on identified patterns.

2.3.1. Inverted Transformer (iTransformer)

The inverted Transformer, a variant within the Transformer family, applies the attention and feed-forward network mechanisms across inverted dimensions [24]. This design enhances performance, generalization across different variates and utilization of arbitrary lookback windows, making it an excellent choice for short-term load forecasting tasks [24]. In this model, the iTransformer serves as the primary stage, focusing on important correlations within the load feature matrix derived from two-stage feature extraction. This strategy improves the model's overall predictive capability. The architecture of the iTransformer is detailed in Figure 3.

The iTransformer employs an encoder-only structure comprising three main components: embedding, projection and Transformer blocks. The input time series is tokenized to capture the properties of the variate, followed by self-attention for mutual interactions and individual processing through feed-forward networks. The forecasting process utilizing the iTransformer is outlined in Equations (22) to (24):

$$\mathbf{h}_n^0 = \text{Embedding}(\mathbf{C}_{:,n}) \tag{22}$$

$$\mathbf{H}^{d+1} = \text{TrfmBlock}\left(\mathbf{H}^d\right),\ d = 0, \cdots, D-1 \tag{23}$$

$$\hat{\mathbf{O}}_{:,n} = \text{Projection}\left(\mathbf{h}_n^D\right) \tag{24}$$

where $\mathbf{C}_{:,n}$ is the lookback series; Embedding() represents the embedding operation; $\mathbf{H} = \{\mathbf{h}_1, \cdots, \mathbf{h}_N\} \in R^{N \times Dim}$ contains N embedded tokens of dimension Dim; D represents the number of layers in the Transformer block; $\mathbf{H}^d$ and $\mathbf{H}^{d+1}$ are the d-th and $(d+1)$-th layer

of the Transformer block; TrfmBlock() represents the Transformer block operation; $\hat{\mathbf{O}}_{:,n}$ is the prediction variate in future series; Projection() represents the projection operation.

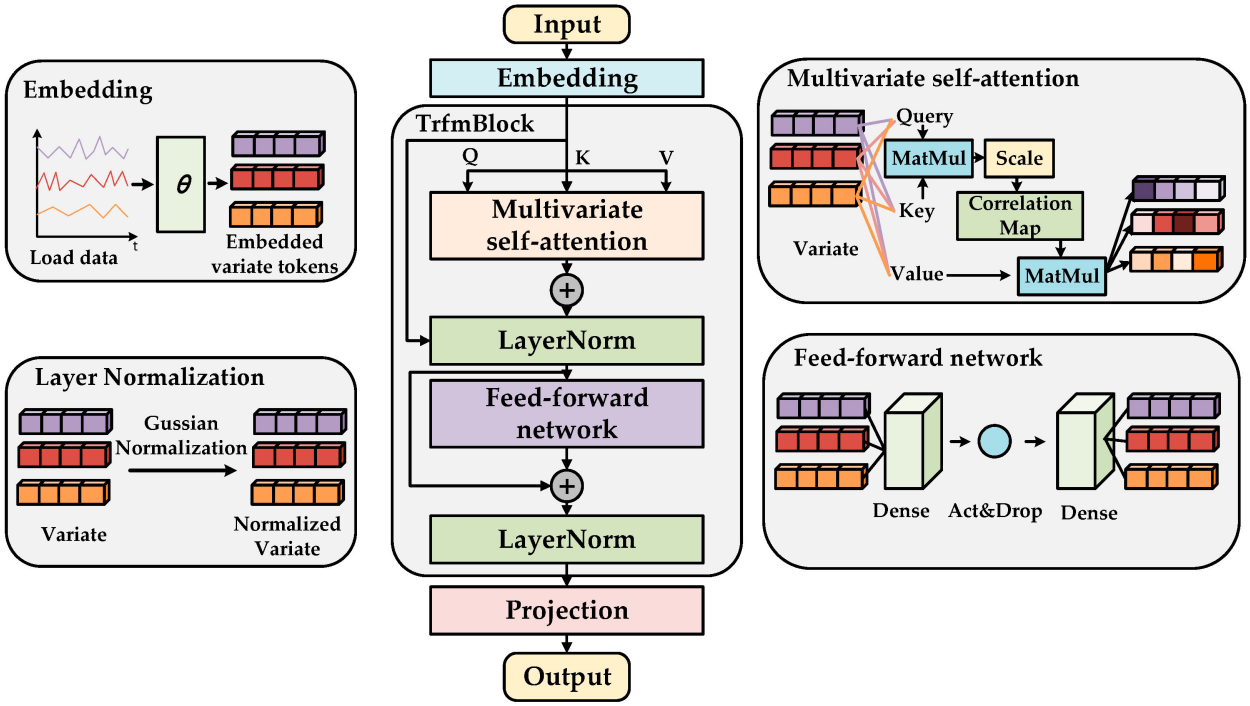


Figure 3. The architecture of the inverted Transformer.

The Transformer block includes three key components: layer normalization, a feed-forward network and a self-attention module. Layer normalization, particularly beneficial for non-stationary issues like short-term load forecasting, normalizes series into a standard distribution, reducing discrepancies caused by inconsistent measurements. The normalization process is detailed in Equation (25):

$$\text{LayerNorm}(\mathbf{H}) = \left\{ \frac{\mathbf{h}_n - \text{Mean}(\mathbf{h}_n)}{\sqrt{\text{Var}(\mathbf{h}_n)}} \middle| n = 1, \cdots, N \right\} \tag{25}$$

where LayerNorm() represents the layer normalization function; Mean() represent the calculation of the mean value; Var() represent variance calculation.

The iTransformer utilizes feed-forward networks for each variate token, thereby enriching the complexity of time-series representations. Through the strategic stacking of densely connected inverted blocks, it not only encodes and decodes time series more effectively but also outstrips the capabilities of contemporary MLP-based approaches. This structure promotes collaborative learning of temporal features, leading to enhanced predictive accuracy and demonstrating strong generalization across different scenarios.

Additionally, the self-attention mechanism in the iTransformer processes each variate token independently, using linear projections to generate queries, keys and values. This normalization across feature dimensions allows the model to effectively identify and leverage correlations specific to each variate, enhancing the interactions among representations. This approach enhances the interpretability and intuitiveness of the framework, significantly improving the forecasting of multivariate time series and enhancing the model's predictive precision.

2.3.2. Multi-Perception Temporal Convolutional Network

The Multi-Perception Temporal Convolutional Network (MPTCN) is designed to augment the traditional TCN's ability to perceive important patterns and capture long-term

dependencies from the feature map output of the iTransformer. Initially employing TCN for extracting long-term dependencies, the integration of a multi-layer perceptron (MLP) into the TCN framework enhances feature learning through nonlinear transformations, thereby increasing model flexibility and performance in capturing complex temporal patterns.

- Temporal convolutional network (TCN)

TCN excels at detecting long-term relationships within time-series data, utilizing dilated convolutions that facilitate efficient parallel processing. With its expansive receptive fields, the TCN adeptly models temporal sequences, providing interpretable feature extraction and delivering competitive performance across a variety of tasks. This capability has established TCNs as a favored approach for time-series forecasting. The architecture of the TCN comprises two primary modules: the dilated causal convolutions and the residual block, both of which are depicted in Figure 4.

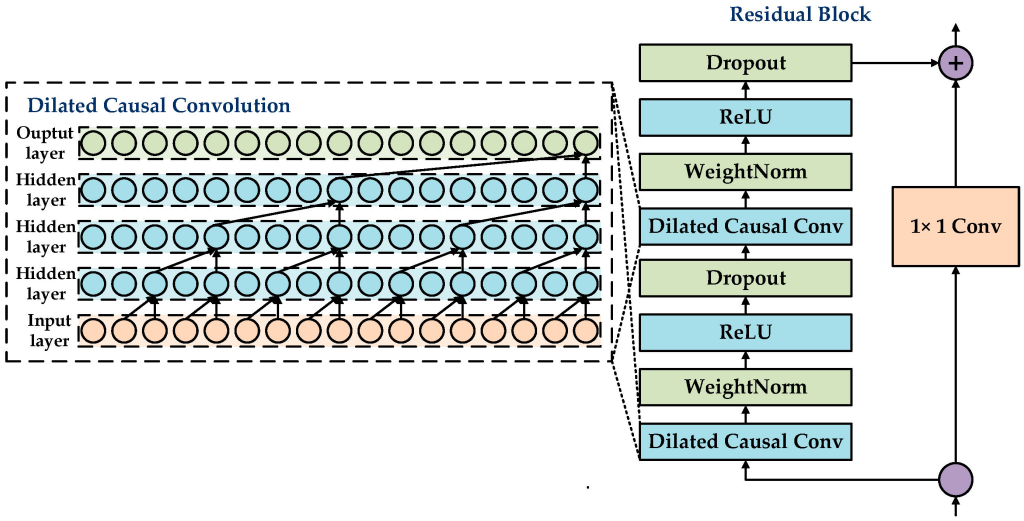


Figure 4. The structure of dilated causal convolution and residual block.

Dilated causal convolutions allow the TCN to maintain an extensive receptive field while avoiding a significant increase in parameter count. This characteristic is particularly beneficial for detecting long-term relationships within the input data. The result of applying dilated convolutions to the sequence element p is detailed in Equation (26):

$$y(t) = \sum_{k=1}^{K} x(t + r \cdot k) \cdot w(k) \tag{26}$$

where $y(t)$ is the dilated causal convolution operation result at time instant t; k is the index of convolution kernel; K is the total of convolution kernels; r is the dilation rate, which follows a sequence of r = 1, 2, 4, 8, 16 as the number of network layers increases; $x(t + r \cdot k)$ represents the input at time instant $(t + r \cdot k)$; $w(k)$ denotes the filter weight of the k-th convolution kernel.

The inclusion of a residual block within the TCN framework addresses potential issues of diminishing or escalating gradients encountered during the training of deep neural networks. This module consists of two sections: the first section merges two levels of dilated convolutions with weight normalization, ReLU activation and Dropout layers. The subsequent layer accepts the initial layer's output. Simultaneously, the second section employs a 1×1 convolution for input identity transformation, enhancing stability and learning efficiency. The operation of the residual block is captured in Equation (27):

$$x^{i+1} = \mathrm{ReLU}\left(F\left(x^{i}\right) + x^{i}\right) \tag{27}$$

where x^i and x^{i+1} denote the output of the preceding and current residual block, respectively; ReLU() represents the activation function; $F()$ represents the transformations of the first branch applied to x^i.

- Multilayer perceptron (MLP)

The integration of an MLP into the TCN architecture augments its feature learning capabilities, thereby enhancing the model's forecasting accuracy. The MLP, a feedforward neural network, incorporates several layers of nodes, each fully connected to the next. Through nonlinear activation functions, the network transforms input data across hidden layers, facilitating the extraction of complex features and learning detailed patterns. The output computation for a single neuron in a hidden layer is outlined in Equation (28):

$$h_j = \text{Activation}\left(\sum_{i=1}^{I} w_{ij} \cdot x_i + b_j\right) \tag{28}$$

where h_j is the output of the j-th neuron; Activation() represents the activation function; w_{ij} are the weights from the i-th input to the j-th neuron; x_i are the inputs; b_j is the bias associated with the j-th neuron.

3. Experiment and Analysis

3.1. Data Preparation

3.1.1. Data

To verify the effectiveness of the proposed short-term load forecasting method, a dataset collected from the regional smart energy system Elia grid in Belgium is used. The dataset contains 140,256 sampling points with a resolution of 15 min, covering the period from 1 January 2020 to 31 December 2023. This dataset comprises the most recent load data from Belgium, effectively capturing the patterns and fluctuations characteristic of rapidly evolving regional smart energy systems. Spanning four years, it provides ample information for models to identify annual periodicity, thereby enhancing short-term load forecasting accuracy. Furthermore, as the Belgian energy market increasingly integrates renewable sources, this dataset supports the development of models that account for their intermittent nature, improving load forecasting during fluctuations in renewable production. The fluctuation of the Belgium load dataset collected from the Elia grid is demonstrated in Figure 5.

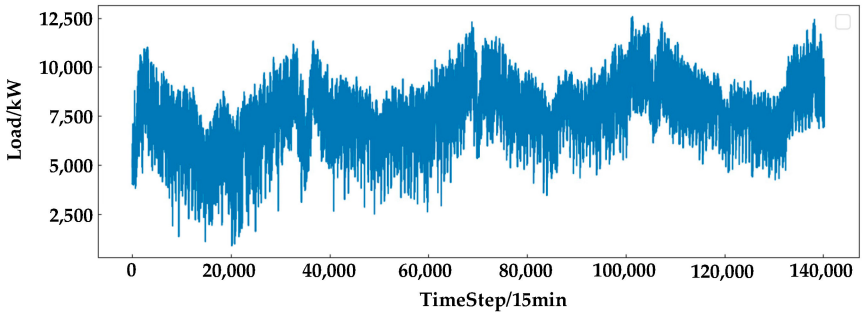


Figure 5. Fluctuation of Belgium load dataset collected from Elia grid.

The fluctuations in Belgium's load curve exhibit distinct periodic patterns attributed to seasonal variations, economic activities, holidays and more. Winter sees heightened demand for heating, elevating the load, while summer witnesses a dip due to reduced heating needs in warmer weather. Working hours drive higher loads, contrasting with lower evening demands during non-working periods, showcasing daily periodicity. Random fluctuations emerge from integrating renewables like solar and wind. Furthermore, load fluctuations reflect a rising trend over time, mirroring the region's increasing economic

activities. Understanding these patterns is crucial for effective short-term load forecasting and sustainable grid operation in smart energy systems in Belgium.

3.1.2. Data Preprocessing

Data preprocessing contains three stages: data splitting, data normalization and data denormalization. In this study, the Belgium load dataset is divided into training, testing and validation datasets in a 7:2:1 ratio, allocating 70% for training, 20% for testing and 10% for validation. Secondly, data normalization is performed on the three training datasets to obtain three valid datasets with a value ranging from 0 to 1. This process is defined in Equation (29).

$$X_{\text{normalized}} = \frac{X - X_{min}}{X_{\max} - X_{\min}} \tag{29}$$

where $X_{\text{normalized}}$ is the normalized outcome; X is the initial value prior to normalization; $X_{\min}$ is the lowest value in the data sequence, and $X_{\max}$ is the highest value in the data sequence.

Finally, data denormalization is performed on forecasted results to transform the initial forecasted results from the range of 0 to 1 to the original wind power range. The calculation method is defined in Equation (30):

$$\hat{z} = (z_{\max} - z_{\min}) \times \hat{z}' + z_{\min} \tag{30}$$

where $\hat{z}$ is the denormalized forecasting results; $z_{\max}$ is the maximum value in the normalized forecasting results; $z_{\min}$ is the maximum value in the normalized forecasting results; $\hat{z}'$ is the normalized forecasting results.

3.2. Evaluation Metrics

Four prediction effect evaluation indicators are used in this study to assess the effectiveness of the approaches, namely root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE) and explained variance score (EVS). RMSE, MAE and MAPE are used for prediction error evaluation. The lower these three metrics, the greater the accuracy of the predictions. EVS is employed to assess the load forecasting model fitting ability. The EVS ranges from 0 to 1. The closer to 1, the more accurate the model's fitting ability. The computation methods for these metrics are as follows:

$$\text{RMSE} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(y_i - \hat{y}_i)^2} \tag{31}$$

$$\text{MAE} = \frac{1}{M}\sum_{i=1}^{M}|y_i - \hat{y}_i| \tag{32}$$

$$\text{MAPE} = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{y_i - \hat{y}_i}{y_i}\right) \times 100\% \tag{33}$$

$$\text{EVS} = 1 - \frac{var(y_i - \hat{y}_i)}{var(y_i)} \tag{34}$$

where M is the sample total; y_i is the true value; $\hat{y}_i$ is the predicted value; $var()$ is the variance function.

In addition, in some experiments conducted in our study, to better evaluate the computational cost of the proposed model and comparison models, the run time of each model is also evaluated.

3.3. Two-Stage Feature Extraction

3.3.1. First-Stage Feature Extraction

The initial load data obtained from the Elia grid in Belgium undergoes preprocessing using the Prophet method to delineate trends, seasonal variations and holiday patterns. The regular parameters adopted in this study are the default parameters of the Prophet method. The holiday configuration utilizes a tailored dataset derived from the real holidays observed in Belgium. This meticulous approach takes into account the specific duration and impact of each holiday on a yearly basis, enhancing the precision in capturing distinctive holiday patterns in the load data. The Prophet method is implemented with Pytorch 2.0.1 under the software environment of Python 3.10.8, and the specific model parameters of the Prophet method are demonstrated in Table 1.

Table 1. Parameters of the Prophet method.

Parameter	Value	Definition
growth	Linear	Linear progression
n_changepoints	25	Count of trend shifts
Changepoint_range	0.8	Scope of change points covers initial 80% data
Change_point_prior_scale	0.05	Model adaptability to trend variations
Yearly_seasonality	True	Activate yearly seasonality
Weekly_seasonality	True	Activate weekly seasonality
Daily_seasonality	True	Activate daily seasonality
Seasonality_mode	Multiplicative	Seasonal elements are handled multiplicatively
Holidays_mode	Multiplicative	Holiday elements are handled multiplicatively

In this study, linear progression is chosen as the growth of the Prophet model since the load data grow without bounds. Twenty-five is chosen as the number of the trend shifts within the original load sequence, which is a moderate number for time series. The scope of change points is set to cover the initial 80% of data. The change point prior scale is set as 0.05; a higher value allows the model to fit more changepoints. The yearly, weekly and daily seasonality is activated for feature extraction. Seasonality and holiday modes are set as multiplicative since the seasonal effects increase with the level of the series. With these parameters established in the algorithm, the Prophet can be employed to decompose the input load sequence into annual, weekly, daily and holiday periodic components. These components are selected as the initial features for subsequent forecasting.

To enhance the analysis of the primary stage features derived from the load data, the extraction outcomes of the initial load data are visually depicted in Figure 6. To effectively showcase the periodic attributes related to weekly and daily seasonality, the initial 8760 sample points, spanning two months, are selected for demonstrating weekly seasonality. Meanwhile, the first 672 sample points, covering a week, are utilized for illustrating daily seasonality. Additionally, other periodic features are visualized using a total of 140,256 sample points for comprehensive analysis.

The trend curve depicted in Figure 6 illustrates a progressive incline from the year 2020 to 2023, attributed to the burgeoning economic landscape in Belgium. The yearly seasonality reveals robust cyclic patterns, signaling consistent load fluctuations annually within the country. Within the yearly seasonality, it can be inferred that the peak load occurs in winter owing to the heightened usage of heating systems. Meanwhile, the load bottom is observed in summer due to reduced heating demands and minimal air conditioning usage, facilitated by the temperate summer climate. The weekly seasonality showcases distinct cyclic patterns throughout the week, with elevated loads during weekdays and diminished loads over the weekends. Daily seasonality exhibits pronounced cyclic variations over the course of a day, with peak loads in the morning, a relative dip at noon, a secondary peak in the afternoon and the lowest point at midnight. This pattern is influenced by the electricity demands of industrial processes and economic activities during working hours in the morning and afternoon, contrasted with reduced consumption during non-working hours

at noon and midnight. The holiday seasonality displays a consistent pattern each year, reflecting fluctuations in the annual load data due to the occurrence of various holidays.

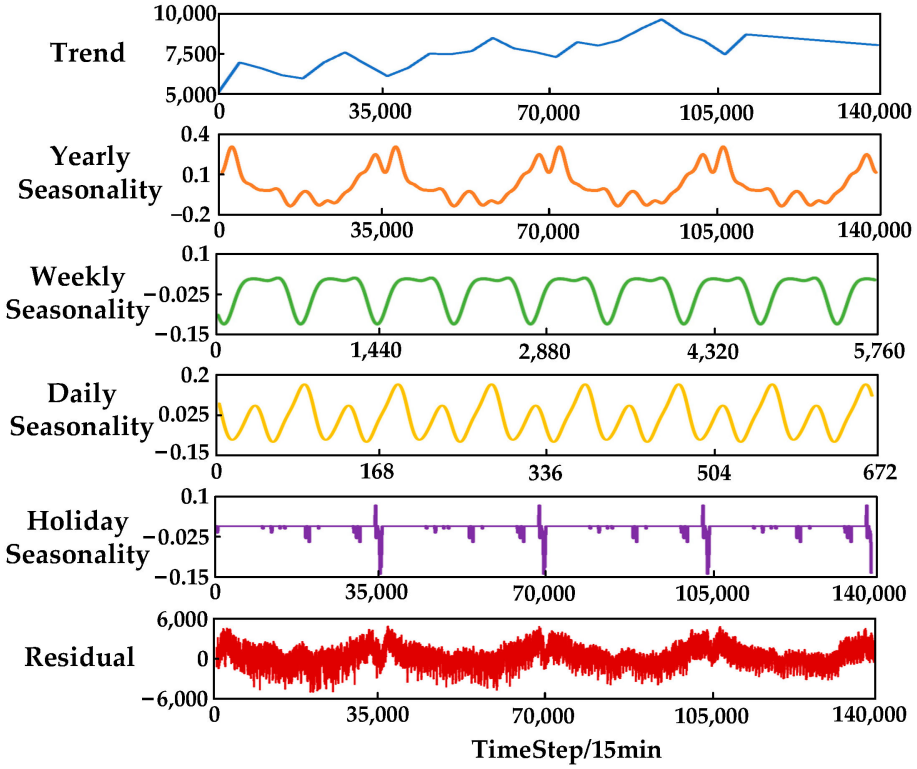


Figure 6. First-stage feature extraction results from the Elia grid data in Belgium.

3.3.2. Second-Stage Feature Extraction

Following the initial feature extraction process, the residual component obtained from the first-stage feature extraction undergoes a secondary feature extraction phase utilizing IVYVMD. This method aims to capture periodic patterns with multiple frequencies, effectively reducing the randomness within the residual component. This approach significantly bolsters the feature extraction capabilities of the proposed methodology. The VMD parameters are optimized with the IVY algorithm. The IVY algorithm is employed to determine the optimal parameters for the VMD method. Implemented using Python 3.10.8, the initialized parameters for the IVY algorithm are detailed in Table 2. The iteration number for the IVY algorithm is set to 20, which is deemed sufficient to ensure convergence. Additionally, a population size of 20 is selected as the initial setting; while a larger population can enhance exploration, it also increases computational complexity. The ranges for parameters K and α are established based on the specific characteristics of short-term load forecasting. Utilizing these parameters, the IVY algorithm is applied to the VMD to achieve an optimal solution. Specifically, the parameters K and α are selected as the targets for optimization within the VMD framework. The detailed implementation process of the IVY-VMD algorithm is illustrated in Figure 2.

Table 2. Parameter initialization of the IVY algorithm.

Parameter	Value	Definition
$iter_{max}$	20	Number of iterations
Sizepop	20	Size of the initial population
K	(2–10)	The initialized range of K
α	(1000–3000)	The initialized range of α

After applying the IVY algorithm, the optimized VMD parameters can be obtained. The optimized parameters of VMD are demonstrated in Table 3.

With these optimal parameters, the VMD method is then applied to the residual component obtained from the first stage to extract periodical features. The extracted features are visually presented in Figure 7.

Table 3. Parameters of the VMD method.

Parameter	Value	Definition
α	3000	Quadratic penalty factor
K	5	Modal number
τ	0	Noise in load data is tolerated
DC	0	The initial mode is not required to capture the direct current
Init	1	All center frequencies start uniformly distributed
tol	1×10^{-6}	Convergence tolerance threshold

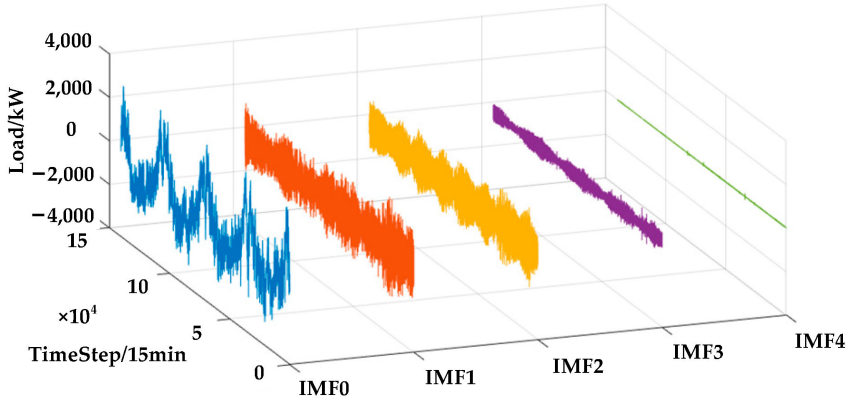


Figure 7. Second-stage feature extraction results from the Elia grid data in Belgium.

The intrinsic mode functions (IMFs) derived from the second-stage feature extraction aptly showcase a low-frequency component, IMF0, and three high-frequency periodic patterns, namely IMF1, IMF2 and IMF3, within the residual component extracted from Prophet. This analysis effectively unveils a rich set of features inherent in the original load data.

3.3.3. Data Matrix

Through the integration of the outcomes from the initial and secondary feature extraction stages, a data feature matrix is constructed, as illustrated in Figure 8. This matrix encapsulates the extracted trend, key seasonal patterns and event influences within the dataset. By amalgamating these essential components, the data feature matrix effectively captures the most critical and comprehensive features embedded in the load sequence. This comprehensive representation facilitates enhanced understanding and learning capabilities within the subsequent hybrid deep learning model, yielding more precise short-term load forecasting results.

As depicted in Figure 8, an original load sequence and nine feature sequences are amalgamated to form the input data matrix for the subsequent hybrid deep learning model. Among the nine features examined, the trend, as well as the yearly, weekly, daily and IMF0 components, plays a pivotal role in enhancing the performance of the subsequent deep learning model in accurately predicting future loads. In contrast, the features related to holidays and the IMF1, IMF2 and IMF3 components are of relatively lesser significance but still help to improve the prediction accuracy of the deep learning model.

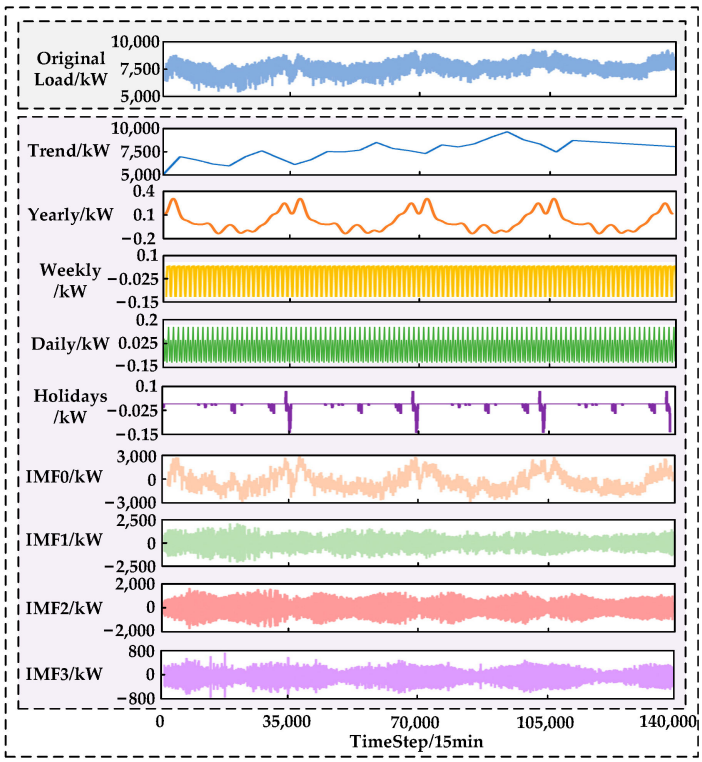


Figure 8. The data matrix obtained by the two-stage feature extraction module.

3.4. Experiments and Analysis

Four sets of experiments are meticulously executed to showcase the exceptional performance of the proposed model in short-term electricity load forecasting. In Section 3.4.1, an ablation experiment is conducted on the proposed model to assess the performance contribution of each module within the framework. Section 3.4.2 features a comparative experiment evaluating various feature extraction methods, emphasizing the superiority of the proposed two-stage feature extraction approach. Subsequently, in Section 3.4.3, a comparative analysis of different parameter optimization methods is conducted, underscoring the remarkable efficacy of the IVY optimization technique. Finally, in Section 3.4.4, a comprehensive comparison of diverse models related to the proposed model is presented to demonstrate the outstanding performance of the proposed model over commonly used models.

The model parameters across all experiments were meticulously fine-tuned using the common grid search method, with specific details outlined in Table 4. Notably, all experiments were conducted on a LINUX 20.04 system. The deep learning model was developed using Python 3.10.8 and Pytorch 2.0.1. The specific software libraries utilized include pandas 1.5.3, scikit-learn 1.2.2, numpy 1.23.5, matplotlib 3.7.0 and reformer-pytorch 1.4.4. The NVIDIA GEForce RTX 4090D GPU was used as the model training platform, and CUDA 12.4 was used to accelerate the GPU.

Table 4. Parameters of the hybrid deep learning model iTransformer-MPTCN.

Parameter	Value	Parameter	Value
Learning rate	0.0001	Sequence length	96
Epochs	15	Label length	72
Batch size	320	Prediction length	24
Dropout probability	0.05	Number of heads	8
Optimizer	Adam	Model dimension	64
Loss function	MSE	Forward dimension	64

3.4.1. Ablation Experiment of the Proposed Short-Term Load Forecasting Model

The proposed short-term load forecasting model is built upon the baseline model iTransformer, comprising four key modules: two-stage feature extraction (TFE), iTransformer, TCN and MLP. To elucidate the significance of each module within the proposed method (TFE-iTransformer-MPTCN), an ablation experiment is conducted. In this experiment, seven comparative models are developed, including the baseline model iTransformer (M1), TFE-iTransformer (M2), iTransformer-TCN (M3), iTransformer-MLP (M4), TFE-iTransformer-TCN (M5), TFE-iTransformer-MLP (M6) and iTransformer-TCN-MLP (M7). The model proposed in this study is denoted as Mproposed. Throughout the experiment, variables are rigorously controlled, and the primary model parameters remain consistent with those outlined in Table 4. The results of the ablation experiment are presented in Figure 9 and Table 5. And the prediction curves of the comparative models are illustrated in Figure 10.

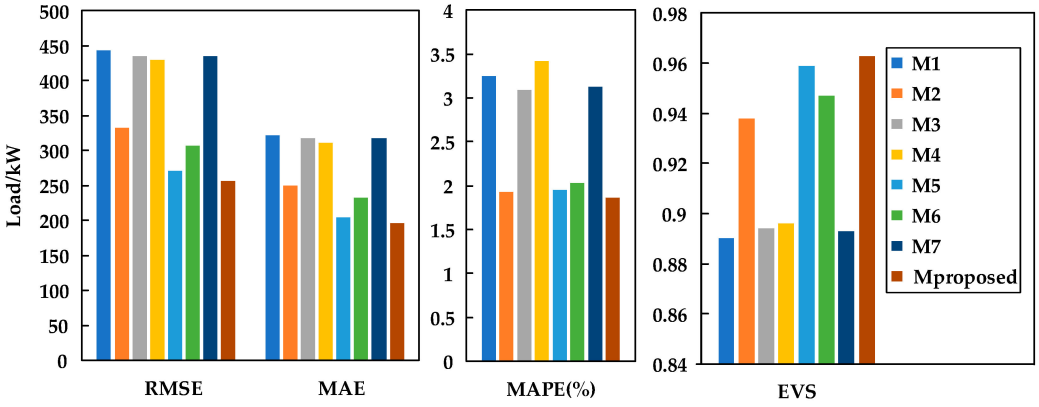


Figure 9. The evaluation results of the ablation experiment.

Table 5. Ablation experiment of the proposed short-term load forecasting model.

Model	Module				Evaluation Metrics				
	iTransformer	TFE	TCN	MLP	RMSE (kW)	MAE (kW)	MAPE (%)	EVS	Run Time (s)
M1	√				443.518	322.416	3.248	0.890	145.892
M2	√	√			331.852	250.069	1.925	0.938	205.044
M3	√		√		435.654	317.697	3.090	0.894	182.124
M4	√			√	429.363	311.406	3.423	0.896	159.658
M5	√	√	√		270.515	204.459	1.949	0.959	239.278
M6	√	√		√	306.688	232.768	2.026	0.947	215.336
M7	√		√	√	435.654	317.697	3.126	0.893	191.310
Mproposed	√	√	√	√	256.360	196.590	1.862	0.963	250.951

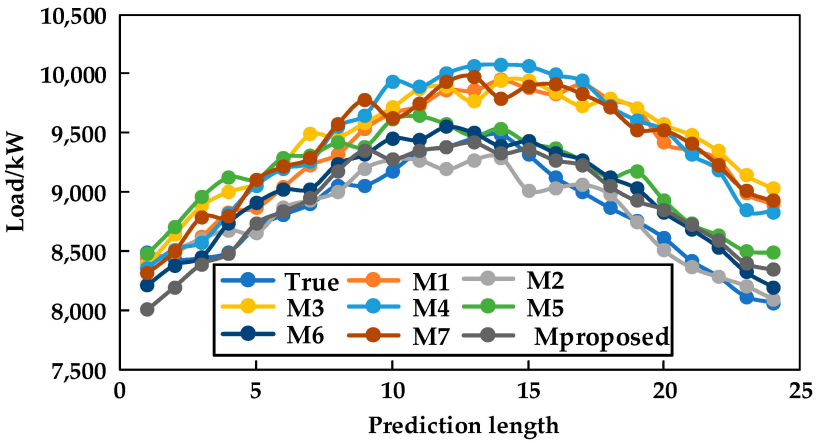


Figure 10. The prediction results of the ablation experiment.

As presented in Figures 8 and 9 and Table 5, although moderately increasing run time, the incorporation of the two-stage feature extraction module, based on the baseline model iTransformer, notably decreased the prediction errors RMSE, MAE and MAPE by 25.178%, 22.41% and 1.323%, respectively, achieving a MAPE of 1.925%. This enhancement resulted in significantly more accurate predictions and an improved EVS by 5.393%, signifying a superior model fitting effect. The TFE module's effectiveness is attributed to its ability to extract multiple periodic features from the original load data. These extracted periodic features serve as additional inputs for the deep learning model. By incorporating these supplementary features, the model can more effectively identify critical patterns, thereby improving its capacity to model electricity load dynamics. Furthermore, the integration of the TCN led to an increase in computational costs, while resulting in a reduction in prediction errors RMSE, MAE and MAPE by 1.773%, 1.463% and 0.158%, indicating enhanced accuracy in short-term load forecasting. The increase in EVS by 0.449% underscores the improved model-fitting effect attributed to the TCN's capability to capture long-term dependencies. Similarly, the integration of MLP resulted in reduced prediction errors RMSE and MAE by 3.192% and 3.415%, respectively, indicating more precise short-term load forecasting. This is because the architecture of one MLP includes one or more hidden layers, enabling it to learn hierarchical representations of the data. This allows the model to extract features at different levels of abstraction, enhancing its ability to identify patterns relevant to load forecasting. Additionally, the EVS increased by 0.674%, reflecting the robust ability of MLP to learn periodic patterns. By amalgamating the four key components to form our proposed method, the prediction errors RMSE, MAE and MAPE were reduced by 42.199%, 39.024% and 1.386 %, respectively, reaching a forecasting accuracy of 98.138% with a run time of 250.951 s. The EVS reached a commendable value of 0.963, enhancing the model's fitting effect by 8.202%.

In summary, each component in our proposed method demonstrated improvements in the overall forecasting ability of the short-term load forecasting model. And the overall performance considering both forecasting accuracy and computational costs of the proposed model proves outstanding.

3.4.2. Comparative Experiment of Different Feature Extraction Methods

To demonstrate the superiority of the proposed two-stage feature extraction method, a comparative experiment involving different two-stage feature extraction methods is conducted. Five comparative models are developed, including two models with varying second-stage feature extraction methods: Prophet-EMD-iTransformer-MPTCN (M8) and Prophet-CEEMDAN-iTransformer-MPTCN (M9). Additionally, three models with varying first-stage feature extraction methods are included: STL-EMD-iTransformer-MPTCN (M10), STL-CEEMDAN-iTransformer-MPTCN (M11) and STL-VMD-iTransformer-MPTCN (M12). The experiment strictly controlled variables and maintained consistent model parameters as outlined in Table 4.

The results of the comparative experiment of different feature extraction methods are shown in Figure 11. The forecasting performance evaluation comparison is shown in Table 6.

As shown in Figure 10 and Table 6, for models with varying second-stage feature extraction methods, utilizing EMD and CEEMDAN as the second-stage feature extraction methods results in a noticeable decrease in prediction accuracy and model fitting, compared to the VMD method. Compared with M8 and M9, the RMSE, MAE and MAPE of the proposed method increased by 28.509%, 25.152% and 0.273%, respectively, on average, and EVS decreased by 3.660%. The run time of the proposed model also decreased by 6.035% on average. These results are due to EMD's limitations, such as mode mixing, boundary effects and noise sensitivity. This hinders its ability to accurately capture complex temporal patterns, leading to suboptimal decomposition outcomes. Similarly, CEEMDAN's susceptibility to mode mixing, end effects and parameter sensitivity impede its effectiveness in capturing and representing the dynamics of complex temporal data. This results in less

efficient decomposition of the residual component obtained from the first stage. And the complex computation process of CEEMDAN leads to higher computational costs compared with the proposed model.

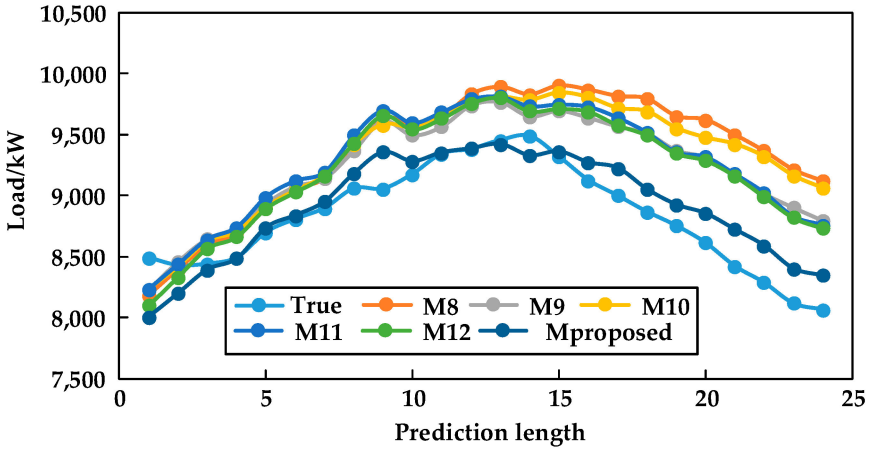


Figure 11. Comparison of models based on different feature extraction methods.

Table 6. Comparison of different feature extraction methods.

Model	RMSE (kW)	MAE (kW)	MAPE (%)	EVS	Run Time (s)
M8	368.026	268.942	2.049	0.927	228.365
M9	349.152	256.360	2.220	0.931	305.772
M10	361.735	268.942	2.354	0.923	245.612
M11	350.725	261.078	2.339	0.924	317.245
M12	377.462	275.233	2.304	0.934	266.193
Mproposed	256.360	196.590	1.862	0.963	250.951

For models with varying first-stage feature extraction methods, compared with M12, using the regular STL method for the first-stage feature extraction leads to an increase in prediction errors (RMSE, MAE and MAPE) by 32.083%, 28.57% and 0.442%, respectively. This indicates less accurate forecasting outcomes. Moreover, when the Prophet method is replaced by STL for the first-stage feature extraction, the MAPE of M10 and M11 increases by 0.308% and 0.477%, respectively, compared to M8 and M9. This highlights the superior feature extraction capability of the Prophet method over STL, as the latter's limited feature extraction ability results in inaccurate extraction of trend and seasonal patterns from the original load data.

In conclusion, utilizing Prophet for the first-stage feature extraction outperforms STL, while employing VMD for the second-stage feature extraction surpasses other signal decomposition methods like EMD and CEEMDAN. Therefore, our proposed two-stage feature extraction method exhibits the best capability in capturing data patterns, thereby enhancing forecasting accuracy.

3.4.3. Comparative Experiment of Different Parameter Optimization Methods

In order to highlight the superiority of the IVY algorithm for VMD parameter optimization proposed in this study, a comparative experiment of different parameter optimization methods is conducted. In this experiment, four comparative models are constructed based on the differential evolution algorithm (DE), gray wolf optimization (GWO), whale optimization algorithm (WOA) and sparrow search algorithm (SSA), respectively. Specific comparative models include Prophet-DE-VMD-iTransformer-MPTCN (M13), Prophet-GWO-VMD-iTransformer-MPTCN (M14), Prophet-WOA-VMD-iTransformer-MPTCN (M15) and Prophet-SSA-VMD-iTransformer-MPTCN (M16). The comparison of different parameter optimization method evaluation results is shown in Figure 12 and Table 7.

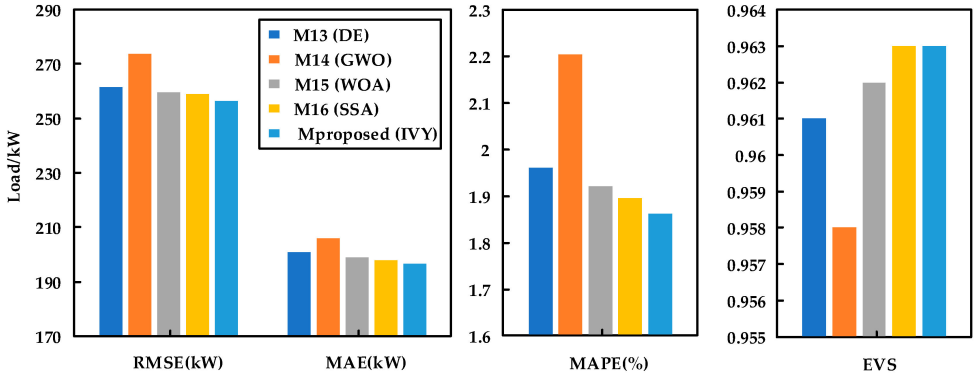


Figure 12. Comparison of different parameter optimization methods' evaluation results.

Table 7. Comparison of forecasting evaluation of different parameter optimization methods.

Model	RMSE (kW)	MAE (kW)	MAPE (%)	EVS	Run Time (s)
M13 (DE)	261.373	200.895	1.960	0.961	262.886
M14 (GWO)	273.660	206.031	2.203	0.958	258.335
M15 (WOA)	259.567	198.722	1.921	0.962	269.732
M16 (SSA)	258.878	197.992	1.895	0.963	260.124
Mproposed (IVY)	256.360	196.590	1.862	0.963	250.951

As illustrated in Figure 11 and Table 7, with less run time, the model employing the IVY algorithm significantly outperforms traditional optimization algorithms such as DE, achieving a reduction in MAPE of 0.098%. This improvement is primarily due to DE's slow convergence rate, which compromises the efficacy of VMD decomposition. In comparison with newer metaheuristic methods like GWO, WOA and SSA, the IVY algorithm demonstrates superior load modeling capabilities, reducing the MAPE by 0.341%, 0.059% and 0.033%, respectively. The limited effectiveness of the GWO algorithm in VMD parameter optimization can be attributed to its inadequate exploration and exploitation capabilities. Similarly, the suboptimal search mechanisms and insufficient parameter tuning of the WOA and SSA algorithms impede the effective identification of optimal parameters. The IVY algorithm's advanced optimization capabilities enhance VMD parameter selection, leading to improved decomposition efficiency and the extraction of more precise features for subsequent hybrid deep learning models.

In conclusion, the IVY algorithm surpasses other prevalent optimization techniques, notably enhancing feature extraction from the residual component and providing accurate inputs for advanced modeling. Therefore, its preeminence is established in enhancing predictive accuracy and model robustness.

3.4.4. Overall Comparison of Short-Term Load Forecasting Methods

In order to illustrate the overall outstanding performance of the multi-step short-term wind power forecasting methods proposed, an overall comparative experiment is conducted. In this experiment, the proposed model is compared with five related common short-term time-series forecasting models including the Transformer (M17), Autoformer (M18), Informer (M19), Reformer (M20) and TimesNet (M21). Variables are strictly controlled during the experiment, and the model parameters are consistent with Table 3. The comparison of the related models is shown in Figure 13 and Table 8.

As depicted in Figure 13 and Table 8, the short-term load forecasting method proposed in this study surpasses the Transformer and other comparative Transformer-based models, achieving a high forecasting accuracy of 98.138% and a model fitting effect of 0.963. Compared to the original Transformer, the proposed method demonstrates a reduction in prediction errors RMSE, MAE and MAPE of 47.419%, 45.177% and 1.729%, respectively. The EVS also improves from 0.867 to 0.963, enhancing the model fitting effect by 9.969%.

The run time also decreased by 24.89%, indicating lower computational cost and higher efficiency. These improvements can be attributed to the integration of a two-stage feature extraction process, enabling the deep learning model to effectively learn periodic patterns and features inherent in the original load data. In addition, the integration of MPTCN allows for the leveraged strengths of each architecture and model temporal features more effectively, ensuring that short-term and medium-term patterns are captured. In comparison to the Autoformer and Reformer, the proposed method significantly enhances prediction accuracy, outperforming them by 4.974% and 2.705%, respectively. Although the run time of the proposed method is longer than the Informer and Reformer, its prediction accuracy outperforms these models to a great extent, exhibiting better overall performance. This superiority stems from the structural limitations of both Autoformer and Reformer models, characterized by challenges in handling long-range dependencies and computational complexity. This impedes their ability to accurately capture intricate temporal patterns. The iTransformer model utilized in our proposed model emphasizes the extraction of local features through convolutions while preserving global context, thereby effectively capturing both short-term and long-term dependencies. Additionally, its straightforward design enhances model interpretability, which can lead to improved forecasting outcomes.

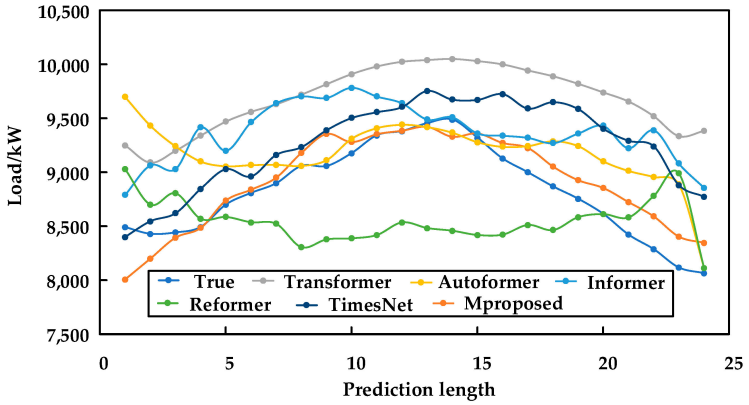


Figure 13. The iteration curve of different parameter optimization methods.

Table 8. Comparison of related common short-term load forecasting models.

Model	RMSE (kW)	MAE (kW)	MAPE (%)	EVS	Run Time (s)
Transformer (M17)	487.552	358.591	3.591	0.867	334.115
Autoformer (M18)	882.323	698.304	6.836	0.564	310.239
Informer (M19)	515.861	388.472	3.451	0.851	230.113
Reformer (M20)	762.795	592.929	4.567	0.677	177.143
TimesNet (M21)	462.392	336.572	2.970	0.880	374.060
MProposed	256.360	196.590	1.862	0.963	250.951

4. Conclusions and Discussion

This paper introduces a novel short-term load forecasting method for regional smart energy systems. The approach leverages a two-stage feature extraction process and a hybrid iTransformer model to enhance forecasting accuracy. The model incorporates Prophet for initial feature extraction, IVYVMD for subsequent feature extraction and iTransformer-MPTCN for modeling. Experimental results using load data from the Elia grid in Belgium validate the effectiveness of the proposed method, achieving a remarkable prediction accuracy of 98.138% and outperforming 21 comparative models.

The two-stage feature extraction module effectively captures essential patterns such as trends, seasonality and holidays, thereby improving the model's ability to learn significant features for accurate short-term load forecasting. The use of Prophet for the first-stage feature extraction demonstrates superiority over the commonly employed STL method, particularly in extracting holiday and specific periodic features at yearly, weekly and daily

intervals. In addition, the application of the VMD method for the second-stage feature extraction outperforms other signal decomposition techniques by avoiding modal mixing and accurately extracting periodic patterns with varying frequencies from the residual component. The IVY algorithm employed for VMD parameter optimization surpasses other optimization methods, leading to enhanced decomposition of the residual component from the first-stage feature extraction. Moreover, the iTransformer-MPTCN model, with its robust feature extraction and pattern learning capabilities, effectively captures both short- and long-term dependencies within the input data matrix. This characteristic results in improved forecasting performance compared to the baseline iTransformer model. In summary, the comprehensive approach combining two-stage feature extraction and the hybrid iTransformer model outperforms existing Transformer-based deep learning models, showcasing a high level of accuracy in short-term load forecasting.

Despite the impressive performance of the proposed model, its structural complexity may present significant challenges for practical implementation, requiring substantial computational resources. Additionally, the parameter tuning for both the Prophet and IVY algorithms necessitates careful consideration. Future work will aim to refine the model, enhancing its efficiency while preserving high forecasting accuracy. We will also explore more effective feature extraction methods that demand minimal parameter tuning and exhibit greater adaptability to varying input load data. The proposed model will also be tested in different regions of smart energy systems to fully represent general applicability. These improvements will be crucial for enhancing the model's practicality.

Author Contributions: Conceptualization, Z.H. and Y.Y.; methodology, Y.Y.; software, Z.H.; validation, Z.H. and Y.Y.; formal analysis, Z.H. and Y.Y.; investigation, Z.H. and Y.Y.; resources, Z.H. and Y.Y.; data curation, Z.H. and Y.Y.; writing—original draft preparation, Y.Y.; writing—review and editing, Z.H. and Y.Y.; visualization, Z.H. 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: Restrictions apply to the availability of these data. Data were obtained from the Elia grid in Belgium. The data can be obtained from https://www.elia.be/en/grid-data, accessed on 5 May 2024. The specific data utilized in our study is available at https://github.com/Whztever/Time-Series-Library_YAML accessed on 5 July 2024.

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

References

1. Xiao, B.; Zhang, B.; Wang, X.; Gao, N. Emerging information and communication technologies for smart energy systems and renewable transition. *Autom. Electr. Power Syst.* **2023**, *47*, 110–117.
2. Liu, S.; Yan, J.; Yan, Y.; Zhang, H.; Zhang, J.; Liu, Y.; Han, S. Joint operation of mobile battery, power system, and transportation system for improving the renewable energy penetration rate. *Appl. Energy* **2024**, *257*, 122455. [CrossRef]
3. Fan, G.; Han, Y.; Li, J.; Peng, L.; Yeh, Y.; Hong, W. A hybrid model for deep learning short-term power load forecasting based on feature extraction statistics techniques. *Expert Syst. Appl.* **2024**, *238*, 122012. [CrossRef]
4. Wei, N.; Yin, C.; Yin, L.; Tan, J.; Liu, J.; Wang, S.; Qiao, W.; Zeng, F. Short-term load forecasting based on WM algorithm and transfer learning model. *Appl. Energy* **2024**, *353*, 122087. [CrossRef]
5. Morais, L.B.S.; Aquila, G.; Faria, V.A.D.; Lima, L.M.M.; Lima, J.W.M.; Queiroz, A.R. Short-term load forecasting using neural networks and global climate models: An application to a large-scale electrical power system. *Appl. Energy* **2023**, *348*, 121439. [CrossRef]
6. Yang, D.; Guo, J.; Li, Y.; Sun, S.; Wang, S. Short-term load forecasting with an improved dynamic decomposition-reconstruction-ensemble approach. *Energy* **2023**, *263*, 125609. [CrossRef]
7. Hong, T.; Wilson, J.; Xie, J. Long term probabilistic load forecasting and normalization with hourly information. *IEEE Trans. Smart Grid* **2014**, *5*, 456–462. [CrossRef]
8. Dudek, G.; Pelka, P.; Smyl, S. A hybrid residual dilated LSTM and exponential smoothing model for midterm electric load forecasting. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 2879–2891. [CrossRef]

9. Grandon, T.; Schwenzer, J.; Steens, T.; Breuing, J. Electricity demand forecasting with hybrid classical statistical and machine learning algorithms: Case study of Ukraine. *Appl. Energy* **2024**, *355*, 122249. [CrossRef]
10. Wei, N.; Li, C.; Peng, X.; Zeng, F.; Lu, X. Conventional models and artificial intelligence based models for energy consumption forecasting: A review. *J. Pet. Sci. Eng.* **2019**, *181*, 106187. [CrossRef]
11. Wen, Y.; Pan, S.; Li, X.; Li, Z. Highly fluctuating short-term load forecasting based on improved secondary decomposition and optimized VMD. *Sustain. Energy Grids Netw.* **2024**, *37*, 101270. [CrossRef]
12. Zhao, Z.; Zhang, Y.; Yang, Y.; Yuan, S. Load forecasting via Grey Model-Least Squares Support Vector Machine model and spatial-temporal distribution of electric consumption intensity. *Energy* **2022**, *255*, 124468. [CrossRef]
13. Fan, G.; Zhang, L.; Yu, M.; Hong, W.; Dong, S. Applications of random forest in multivariable response surface for short-term load forecasting. *Int. J. Electr. Power Energy Syst.* **2022**, *139*, 108073. [CrossRef]
14. Xiao, W.; Li, M.; Xu, Z.; Liu, C.; Zhang, Y. A hybrid electric load forecasting model based on decomposition considering fisher information. *Appl. Energy* **2024**, *364*, 123149. [CrossRef]
15. Kong, W.; Dong, Z.; Jia, Y.; Hill, D.J.; Xu, Y.; Zhang, Y. Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Trans. Smart Grid* **2019**, *10*, 841–851. [CrossRef]
16. Li, K.; Mu, Y.; Yang, F.; Wang, H.; Yan, Y.; Zhang, C. A novel short-term multi-energy load forecasting method for integrated energy system based on feature separation-fusion technology and improved CNN. *Appl. Energy* **2023**, *351*, 121823. [CrossRef]
17. Alhussein, M.; Aurangzeb, K.; Haider, S.I. Hybrid CNN-LSTM model for short-term individual household load forecasting. *IEEE Access* **2020**, *8*, 180544. [CrossRef]
18. Wang, H.; Song, K.; Cheng, Y. A hybrid forecasting model based on CNN and informer for short-term wind power. *Front. Energy Res.* **2022**, *9*, 788320. [CrossRef]
19. Zhu, R.; Liao, W.; Wang, Y. Short-term prediction for wind power based on temporal convolutional network. *Energy Rep.* **2020**, *6*, 424–429. [CrossRef]
20. Gong, M.; Yan, C.; Xu, W.; Zhao, Z.; Li, W.; Liu, Y.; Li, S. Short-term wind power forecasting model based on temporal convolutional network and Informer. *Energy* **2023**, *283*, 129171. [CrossRef]
21. Zhou, S.; Li, Y.; Guo, Y.; Qiao, X.; Mei, Y.; Deng, W. Ultra-short-term load forecasting based on temporal convolutional network considering temporal feature extraction and dual attention fusion. *Autom. Electr. Power Syst.* **2023**, *47*, 193–205.
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Neural Information Processing Systems, Long Beach, CA, USA, 12 June 2017.
23. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 2 February 2021.
24. Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; Long, M. iTransformer: Inverted Transformers are effective for time series forecasting. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 7 May 2024.
25. 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]
26. Li, K.; Huang, W.; Hu, G.; Li, J. Ultra-short term power load forecasting based on CEEMDAN-SE and LSTM neural network. *Energy Build.* **2023**, *279*, 112666. [CrossRef]
27. Yuan, F.; Che, J. An ensemble multi-step M-RMLSSVR model based on VMD and two-group strategy for day-ahead short-term load forecasting. *Knowl. Based Syst.* **2022**, *252*, 109440. [CrossRef]
28. Stefenon, S.F.; Seman, L.O.; Mariani, V.C.; Coelho, L.S. Aggregating prophet and seasonal trend decomposition for time series forecasting of Italian electricity spot prices. *Energies* **2023**, *16*, 1371. [CrossRef]
29. Lin, Z.; Xie, L.; Zhang, S. A compound framework for short-term gas load forecasting combining time-enhanced perception transformer and two-stage feature extraction. *Energy* **2024**, *298*, 131365. [CrossRef]
30. Wang, C.; Zhao, H.; Liu, Y.; Fan, G. Minute-level ultra-short-term power load forecasting based on time series data features. *Appl. Energy* **2024**, *372*, 123801. [CrossRef]
31. Silva, B.N.; Khan, M.; Wijesinghe, R.E.; Wijenayake, U. Meta-heuristic optimization based cost efficient demand-side management for sustainable smart communities. *Energy Build.* **2024**, *303*, 113599. [CrossRef]
32. Zhao, W.; Fan, L. Short-term load forecasting method for industrial buildings based on signal decomposition and composite prediction model. *Sustainability* **2024**, *16*, 2522. [CrossRef]
33. Su, J.; Han, X.; Hong, Y. Short term power load forecasting based on PSVMD-CGA model. *Sustainability* **2023**, *15*, 2941. [CrossRef]
34. Wang, X.; Ma, W. A hybrid deep learning model with an optimal strategy based on improved VMD and transformer for short-term photovoltaic power forecasting. *Energy* **2024**, *295*, 131071. [CrossRef]
35. Taylor, S.J.; Letham, B. Forecasting at Scale. *Am. Stat.* **2018**, *72*, 37–45. [CrossRef]
36. Dragomiretskiyd, K.; Zosso, D. Variational Mode Decomposition. *IEEE Trans. Signal Process.* **2014**, *62*, 531–544. [CrossRef]
37. Ghasemi, M.; Zare, M.; Trojovsky, P.; Rao, R.V.; Trojovska, E.; Kandasamy, V. Optimization based on the smart behavior of plants with its engineering applications: Ivy algorithm. *Knowl. Based Syst.* **2022**, *295*, 111850. [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.