


Article

Short-Term Load Forecasting for Residential Buildings Based on Multivariate Variational Mode Decomposition and Temporal Fusion Transformer

Haoda Ye [†] , Qiuyu Zhu [†] and Xuefan Zhang ^{*}

College of Communication and Information Engineering, Shanghai University, Shanghai 200444, China; luminous@shu.edu.cn (H.Y.); zhuqiuyu@staff.shu.edu.cn (Q.Z.)

^{*} Correspondence: 10002461@shu.edu.cn

[†] These authors contributed equally to this work.

Abstract: Short-term load forecasting plays a crucial role in managing the energy consumption of buildings in cities. Accurate forecasting enables residents to reduce energy waste and facilitates timely decision-making for power companies' energy management. In this paper, we propose a novel hybrid forecasting model designed to predict load series in multiple households. Our proposed method integrates multivariate variational mode decomposition (MVMD), the whale optimization algorithm (WOA), and a temporal fusion transformer (TFT) to perform one-step forecasts. MVMD is utilized to decompose the load series into intrinsic mode functions (IMFs), extracting characteristics at distinct scales. We use sample entropy to determine the appropriate number of decomposition levels and the penalty factor of MVMD. The WOA is utilized to optimize the hyperparameters of MVMD-TFT to enhance its overall performance. We generate two distinct cases originating from BCHydro. Experimental results show that our method has achieved excellent performance in both cases.

Keywords: MVMD; energy consumption; residential buildings; load forecast; temporal fusion transformer



Citation: Ye, H.; Zhu, Q.; Zhang, X. Short-Term Load Forecasting for Residential Buildings Based on Multivariate Variational Mode Decomposition and Temporal Fusion Transformer. *Energies* **2024**, *17*, 3061. <https://doi.org/10.3390/en17133061>

Academic Editor: Álvaro Gutiérrez

Received: 19 May 2024

Revised: 13 June 2024

Accepted: 19 June 2024

Published: 21 June 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

As the global economy and population continue to expand, energy demand correspondingly escalates. A significant segment of this consumption is attributed to the building sector, which tops the list of energy consumers worldwide, followed by the industrial and transportation sectors. Notably, residential energy consumption comprises approximately 75% of the total energy utilization within the building sector [1]. Load forecasting (LF) plays a pivotal role in ensuring the highest level of efficiency and economic benefit within the power system. Accurate household load forecasting is crucial for the optimal planning, operation, and dispatch of the power system. It enables a reduction in energy wastage and maximizes benefits for both residents and power companies [2].

The advancement in information technology has paved the way for data-driven methods to become the mainstream approach in load forecasting. In order to achieve precise load prediction, there are two primary data-driven methodologies: statistics-based and machine learning-based methods.

Machine learning-based methods, such as XGBoost [3], MLR [4], and SVR [5], offer a more flexible approach capable of forecasting load series without the constraint of the series being of a specific type. By incorporating exogenous variables into their models, these methods yield more accurate predictions. AI-based methods, as an advancement within machine learning, are now extensively employed in individual load forecasts due to their enhanced capability for nonlinear processing. In addition to designing models for individual predictions, clustering and ensemble algorithms [6–8] have been applied to solve such problems. Recently, an algorithm based on a novel graph neural network has been used for prediction, achieving state-of-the-art performance [9]. These studies

showcase the ability of deep learning to handle complex relationships between input and output. However, load series characterized by high volatility and nonlinearity still pose challenges for these methods.

Statistics-based methods, including Holt–Winters and ARIMA, conceptualize load series as a composite of trend and seasonal components. The simplicity of their calculations enables swift and efficient forecasting. In contrast to deep learning methods, these approaches are ill-suited for forecasting nonlinear data [10,11].

Mode decomposition is a widely used methodology that facilitates the decomposition of signals into a series of oscillating components. These components, known as intrinsic mode functions (IMFs), are characterized as amplitude-modulated–frequency-modulated (AM–FM) signals. IMFs reflect distinct patterns inherent in the original signals, facilitating the model’s ability to capture its distinctive patterns. In order to further enhance the performance of AI-based methods, researchers have explored their amalgamation with mode decomposition, aiming to achieve accurate load forecasting. In [12], researchers integrated EMD with BiLSTM, utilizing EMD to extract intricate temporal and spectral characteristics from power load data. This process resulted in the generation of multiple IMFs spanning various frequency bands, ultimately enhancing the predictive accuracy of BiLSTM models. In the study detailed in [13], EEMD is employed to smooth the power load sequence and generate IMFs. The LSTM and ELM models are deployed to predict the high-frequency and low-frequency IMFs, respectively, which are subsequently integrated to yield the prediction results. In [14], EWT serves as an enhanced version of EMD and is utilized to extract IMFs and smooth the load data. LSTM is employed to predict the low-to-mid-frequency IMFs, whereas the high-frequency IMFs undergo enhanced DBSCAN clustering before being individually forecasted by another LSTM and LSSVM, capturing distinct samples within the outcomes. Similarly, researchers have forecasted electric load utilizing a hybrid VMD-LSTM network [15]. The IMFs generated by VMD are individually predicted by LSTM. When combined with error correction, the proposed method outperforms all other hybrid methods across all datasets in terms of various metrics. However, the studies on mode decomposition mentioned above can only process one sequence and require predictions for each IMF, making the entire forecasting process complex.

Load series in individual households display rapid fluctuations and are closely linked to residents’ behaviors. These dynamics pose a significant challenge when aiming for accurate prediction [16,17]. The temporal fusion transformer (TFT) is an advanced model developed by the Google team. It exhibits the ability to forecast multiple series simultaneously, showing excellent performance in predicting various types of time series, including load [18], PV power [19], wind speed [20], supply air temperature [21], and tourist demand [22] and volume [23].

In order to further enhance the performance of the TFT in individual forecasting problems, we incorporated MVMD. Our combined method avoids training models for each IMF component. Meanwhile, regarding the parameter selection issue of MVMD, we offer a new perspective on choosing its penalty factor and decomposition levels through sample entropy. Furthermore, the WOA is employed to optimize the hyperparameters of the MVMD-TFT to find reasonable hyperparameters. For performance comparisons, we consider separately training the CNN-LSTM, LSTM, BiGRU-CNN, MVMD-LSTM, and MVMD-CNN-LSTM models.

The main contributions of this article are the following:

- (1) We propose a hybrid MVMD-WOA-TFT model, which can forecast load for multiple houses accurately. MVMD is employed to decompose multi-load data into multiple IMFs, extracting the common features shared among different load sequences. The WOA is utilized to optimize the hyperparameter of the MVMD-TFT, enhancing its overall performance.
- (2) We select an appropriate decomposition level and penalty factor for MVMD from an entropy-based perspective.

- (3) We validate the performance of the proposed model by comparing it to the original TFT and multiple separate training models.

2. Methodology

2.1. Multivariate Variational Mode Decomposition

Multivariate variational mode decomposition (MVMD) serves as a comprehensive extension of VMD, designed to extract a specific number of multivariate modulated oscillations from input data that consist of multiple data channels. MVMD obtains IMFs by solving constraint optimization problems, as described in Equation (1), in which $u_+^{k,c}(t)$ denotes the analytic signal corresponding to $u_k(t)$, where $u_k(t)$ is defined as a vector comprising C channels, represented as $u_k(t) = [u_1(t), u_2(t), \dots, u_C(t)]$.

$$\min_{\{u_{k,c}\}, \{\omega_k\}} \left\{ \sum_k \sum_c \|\partial_t [u_+^{k,c}(t) e^{-j\omega_k t}]\|_2^2 \right\} \quad (1)$$

subject to $\sum_k u_{k,c}(t) = x_c(t) \quad c = 1, 2, 3, \dots, C$

The variable k represents the decomposition level (number of IMFs), and ω_k and ∂_t , respectively, denote the center frequency of the k th IMF and a partial derivative operation for time t .

$$\omega_k^{n+1} = \frac{\sum_c \int_0^\infty \omega |\hat{u}_{k,c}(\omega)|^2 d\omega}{\sum_c \int_0^\infty |\hat{u}_{k,c}(\omega)|^2 d\omega} \quad (2)$$

$$\hat{u}_{k,c}^{n+1}(\omega) = \frac{x_c(\omega) - \sum_{i \neq k} \hat{u}_{i,c}(\omega) + \frac{\hat{\lambda}_c(\omega)}{2}}{1 - 2\alpha(\omega - \omega_k)^2} \quad (3)$$

$$\hat{\lambda}_c^{n+1}(\omega) = \hat{\lambda}_c^n(\omega) + \tau(\hat{x}_c(\omega) - \sum_k \hat{u}_{k,c}^{n+1}(\omega)) \quad (4)$$

The ADMM algorithm is employed to convert the original expression into three iterative sub-optimization problems. These subproblems aim to optimize the center frequency ω_k , mode $u_{k,c}$, and Lagrangian multiplier. The solutions to these subproblems are presented in Equations (2)–(4). Upon completing one iteration using Equations (2)–(4), MVMD proceeds to check a convergence condition. The process continues until the convergence condition is met or until the specified number of iterations is reached. The implementation details of MVMD can be found in [24].

2.2. Parameter Setting for MVMD Based on Sample Entropy

The decomposition level, k , has a great influence on the performance of VMD. As indicated in [25], the residual between the sum of IMFs obtained by decomposition and the original signal can serve as a criterion for the decision of the decomposition level. If the input signal is effectively decomposed, the residuals should contain all the noise of the input signal; hence, the complexity of the residuals should be the largest.

$$r = \sum_k f(k) - x. \quad (5)$$

This principle is described in Equation (5), where $f(k)$ represents the IMF obtained by VMD, x denotes the original signal, and k is the number of IMFs. Sample entropy [25] is a widely used entropy-based method that quantifies the irregularity of signals by evaluating the repeatability of a template; it is an effective measurement of signal complexity. The larger the sample entropy of the residual, the higher the complexity of the residual.

term, and the more noise it contains. Hence, we can obtain a better decomposition effect of the signal.

$$r_s = \sum_i \sum_j f(i, j) - x(i) \quad (6)$$

In the case of multiple signals, the scenario can be extended as described in Equation (6), where $x(i)$ represents the i th signal, and $f(i, j)$ denotes the j th IMF of the i th signal. The concept of residuals is further extended to accommodate multiple signals. We assume there are multiple signals and that they are decomposed using the same decomposition level. The residual of each signal is calculated using Equation (5). An effectively decomposed signal is anticipated to exhibit a large sample entropy for its residual. Consequently, if all the signals are effectively decomposed, the sum of the sample entropies of their residuals should be maximized. From this point forward, we utilize the sum of sample entropies of the residuals from multiple signals to ascertain the appropriate decomposition level for MVMD. Additionally, the penalty factor is another crucial parameter that significantly impacts the decomposition effectiveness. Different penalty factors yield varying sample entropy values. Therefore, we calculate the sample entropy for different penalty factors to identify the one that maximizes the entropy. The final decomposition outcome is determined by both the chosen decomposition level and the optimal penalty factor.

In order to calculate the sample entropy, a specific range was defined for both the decomposition level and the penalty factor with an enumeration of intervals. Considering the significant time cost associated with this process, we set the range of the penalty factor to (10, 5000) with an interval of 20, and the decomposition level was set to (1, 25) with an interval of 1. For each parameter combination defined by the decomposition levels and penalty factor, denoted as $[k, \alpha]$, we obtain IMFs through the MVMD process. Subsequently, based on Equation (8), we aggregate the acquired IMFs and compute their difference from the original signal to derive the residual term. We then calculate the sample entropy of this residual term. Figure 1 illustrates the iterative process, where the yellow boxes represent a series of sample entropies calculated for an individual k value with multiple α values, and the red box indicates the maximum sample entropy for the current k value.

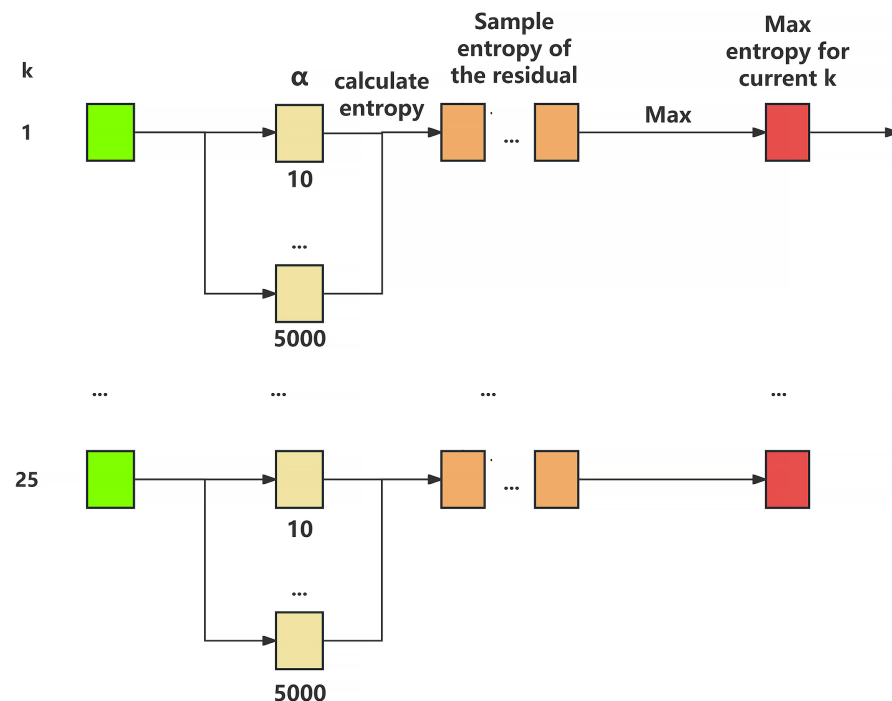


Figure 1. Search for the optimal decomposition level and the corresponding penalty factor.

2.3. Combination of MVMD and TFT

The TFT partitions the inputs into three distinct parts: the past input, the future input, and the static input, as illustrated in Figure 2. The past input can be represented as $X(t) = [X_{t-w}, \dots, X_t]$, where the size of the look-back window is denoted by w . The future input, denoted as $x(t) = [x_{t+1}, \dots, x_{t+\tau}]$, serves as a prior variable that resides within the same temporal range as the predicted target. Here, τ represents the forecast step, indicating the number of time units into the future that we aim to predict. The static input comprises variables that are independent of time.

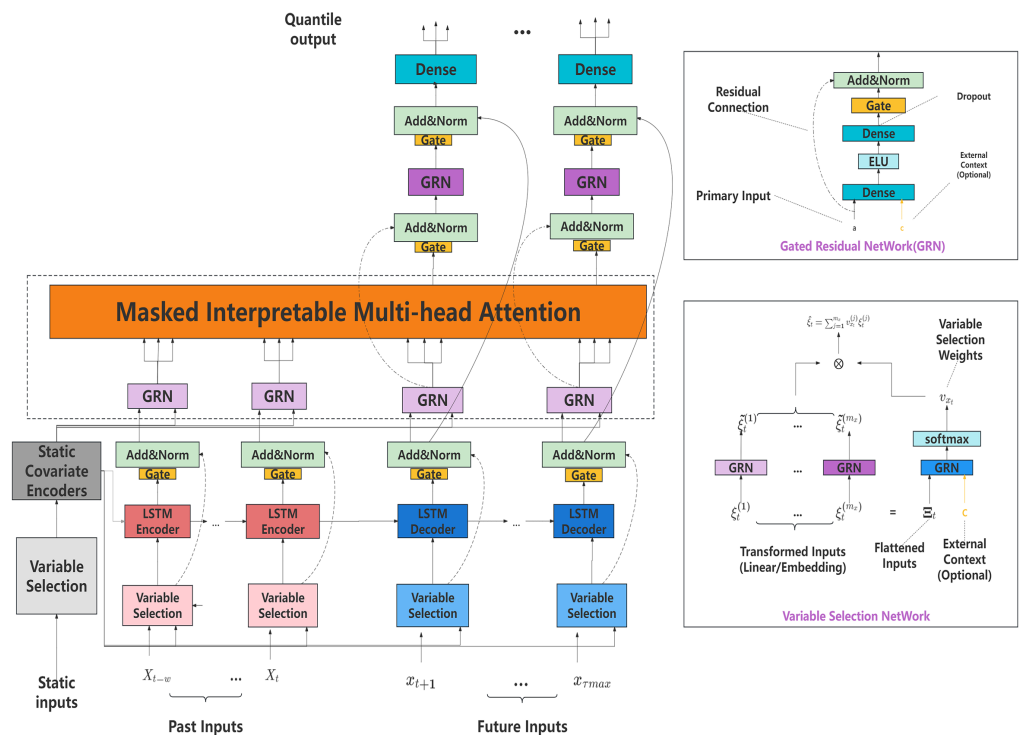


Figure 2. Architecture of the TFT.

The principle of the TFT is mainly composed of the following components:

- (1) Gated residual network: The GRN is designed to control the flexibility of nonlinear mapping in the model.
- (2) Variable selection network: The VSN is designed to provide instance-wise variable selection. It can learn the most salient input variable, which contributes to the prediction problem. It provides access to static information that enhances the weight-generation process.
- (3) Attention mechanism: The TFT applies an average attention mechanism that prevents the model from attending to different input features at different times and facilitates the evaluation of the importance of instance-wise attention weights.
- (4) Quantile loss: The TFT provides a distribution of possible future outcomes along with point estimates through quantile output, and it is trained using quantile loss. In our research, the quantile is set to $\{0.1, 0.5, 0.9\}$.

In order to enhance the ability of the TFT to simultaneously learn patterns from multiple load series, we introduced a novel approach by substituting the load series in past inputs with the results obtained from MVMD.

Figure 3 illustrates the integration of MVMD with the TFT. Consider a scenario with C load series and s exogenous variables, each having a length of L . The decomposition level of MVMD is set to k . After decomposition on the load series, a three-order matrix of dimensions $[C, k, L]$ is obtained. In order to incorporate the past known input using LSTM,

the decomposition matrix is reshaped into $[C \times L, k]$. By incorporating the exogenous variables, the final input matrix size becomes $[C \times L, k + s]$.

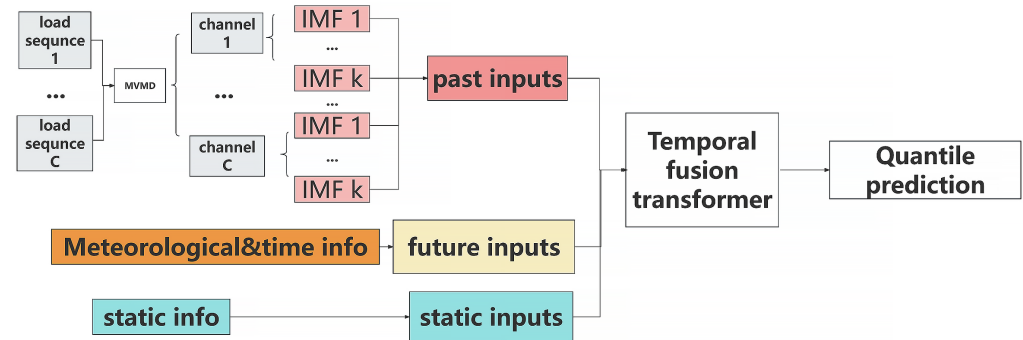


Figure 3. Architecture of MVMD-TFT.

2.4. The Process of Optimizing MVMD-TFT Using WOA

The whale optimization algorithm (WOA) [26] is a nature-inspired optimization algorithm that mimics the foraging behavior of whales to solve complex optimization problems. The WOA introduces two strategies for position updates, which are intended to occur with equal probability. It ensures the exploration and exploitation of the search space.

Strategy 1: Encircling prey. Suboptimal candidates update their positions based on a pair of cross-correlation vectors. The mathematical expression describing this update process is outlined in Equations (7) and (8):

$$\vec{X}(n+1) = \vec{X}^*(n) - \vec{A} \cdot \vec{D} \quad (7)$$

$$\vec{X}(n+1) = \overrightarrow{X_{rand}}(n) - \vec{A} \cdot \vec{D} \quad (8)$$

where n represents the current iterations, \vec{X} represents the position of suboptimal candidates, and \vec{X}^* denotes the position of the current optimal candidate. Vector \vec{A}, \vec{D} denotes the coefficient vectors. The algorithm incorporates a random foraging strategy, which is related to the norm of \vec{A} . If $|\vec{A}| > 1$, the position of the current optimal candidate is replaced with a randomly generated vector, and the remaining candidates are updated using Equation (8), with $\overrightarrow{X_{rand}}$ denoting a randomly generated vector. Otherwise, Equation (7) is employed for updating.

Strategy 2: Bubble-net attacking method:

$$\vec{X}(n+1) = |\vec{X}^*(n) - \vec{X}(n)| \cdot e^{\beta l} \cdot \cos(2\pi l) + \vec{X}^*(n). \quad (9)$$

The suboptimal candidates update their position by calculating the distance between themselves and the optimal candidate. The mathematical expression is presented in Equation (9), where l represents a random number between $[-1, 1]$ and β denotes a constant related to helicity.

Figure 4 depicts the process of optimizing MVMD-TFT hyperparameters with the WOA over one iteration. Initially, we establish the population size and boundary conditions, from which we derive the initial values of solution candidates. Subsequently, we feed the hyperparameter sets represented by each candidate into the TFT model for training, thereby acquiring the quantile loss specific to each individual. Based on the results of quantile loss, we determine the current optimal individual and randomly select the strategy to update the candidates.

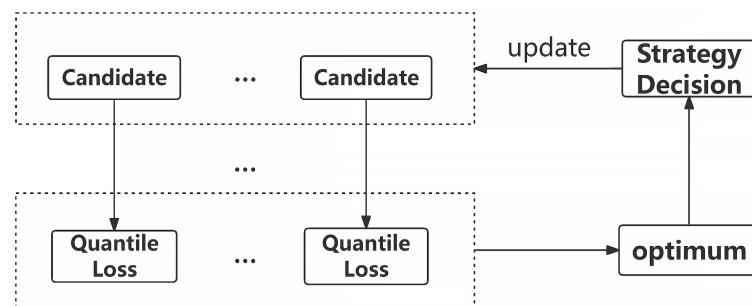


Figure 4. The procedure for MVMD-TFT optimization through the WOA.

3. Results and Discussion

3.1. Experimental Setup

The environment used in this experiment is TensorFlow 2.5, Python 3.8, and a single RTX2080Ti. We implemented the Python version of MVMD by referring to vmdPy and MVMD source code.

3.2. Data Preprocessing

The datasets utilized in this study were obtained from BCHydro [27], encompassing the hourly electricity consumption of 28 residential customers. The consumption data span 3 years. The meteorological data from neighboring weather stations was also incorporated. The temporal boundaries for energy consumption data vary among the 28 buildings in the original dataset, making it impractical to use the entire dataset for training purposes. In order to ensure the selection of appropriate buildings and time frames from the original dataset, we applied the following criteria:

- (1) In order to minimize the missing values within the selected time range, we have stipulated that the proportion of missing values for all variables employed in model training within the specified time range be less than 0.5%. Variables exceeding this missing value threshold were excluded from consideration. In order to facilitate a comparison between LSTM and CNN-LSTM, we opted for two non-overlapping time ranges. One time range spans approximately 3 months (1 November 2017 to 29 January 2018) (Case A), similar to [7], while the second time range encompasses roughly 16 months (26 June 2016 to 30 October 2017) (Case B).
- (2) The meteorological data for the selected buildings all originate from the same weather station. Additionally, each building is accompanied by its corresponding descriptive information. Buildings that have incomplete descriptions are excluded from the analysis. Following the aforementioned criteria, Case A comprises 14 buildings, while Case B includes 10 buildings.

For cases A and B, we applied a consistent data processing methodology, which can be summarized as follows. For Case A, we first aggregated the data from different buildings. Next, we added the “building_id” feature to distinguish the load of different buildings. For the data from each building, we constructed training, validation, and test sets in an 8:1:1 ratio. Then, we performed MVMD decomposition on the load part of these three parts of the data separately. After that, we processed the training set of each building based on “building_id” using Equation (10) and obtained the corresponding maximum and minimum values. We then normalized the test set and validation set using the maximum and minimum values obtained from the training set.

Table 1 presents the variables that were selected as inputs for our model, taking into consideration the criteria based on the number of missing values.

$$\text{Min} - \text{Max Scaler}(x) = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (10)$$

Table 1. Variables input for the TFT.

Variable	Datatype	Description
building_id	Category	Identification of buildings
RUs	Category	The number of rental suites in the house
facing	Category	What direction the house is facing
housetype	Category	House types
weather	Category	A textual description of the type of weather
day	Category	Day of the week
weekend	Category	Boolean value to indicate weekend
hour	Category	Hour of the recording, from 1 to 24
temperature	Continuous	Outside ambient temperature in degrees Celsius (°C)
humidity	Continuous	Outside humidity in percentage (%)
pressure	Continuous	Atmospheric pressure in kilopascals (kPa)
energy_Kwh	Continuous	Hourly consumption (kWh)

3.3. Entropy Computation Results of MVMD

Figure 5 displays the results of residual sample entropy obtained by selecting the optimal penalty factor α for two training set cases (red boxes in Figure 1). We find that the residual sample entropy of Case B is lower than that of Case A, implying that the signal patterns in Case B are more intricate, resulting in lower residual complexity.

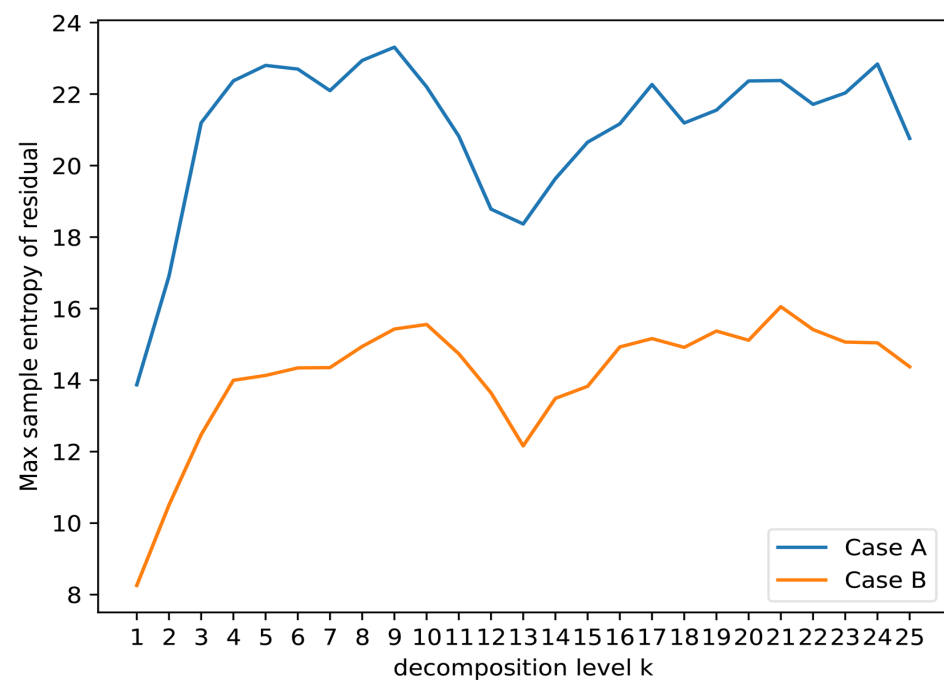
**Figure 5.** Maximum residual entropy in different decomposition levels.

Figure 6 illustrates the impact of different penalty factors on residual sample entropy (yellow boxes in Figure 1) for the same decomposition level. We selected five decomposition levels for analysis based on their highest residual sample entropy. The introduction of the penalty factor leads to complexity in the pattern of residual sample entropy. Specifically, we observe pronounced fluctuations in residual sample entropy for higher decomposition levels under varying penalty factors, for instance, in Case A, when the decomposition level was set to 24, and in Case B, when the decomposition level was set to 21, 22, and 19. These fluctuations gradually diminish as the penalty factor increases, eventually reaching a relatively stable state. Based on these findings, for Case A, the decomposition level was set to 9, with a corresponding penalty factor of 170. For Case B, the decomposition level

was set to 21, with a corresponding penalty factor of 4230. The initial center frequency of MVMD is initialized using a uniform distribution, and the tolerance level is set to 1×10^{-6} .

For the test set and validation set, we similarly obtained their optimal penalty factors based on Figure 1 while maintaining the same number of decomposition levels as the training set. The penalty factors for the validation set and test set in Case A were set as 1230 and 1240, respectively. For Case B, the penalty factors for the validation set and test set were set as 3410 and 4030, respectively.

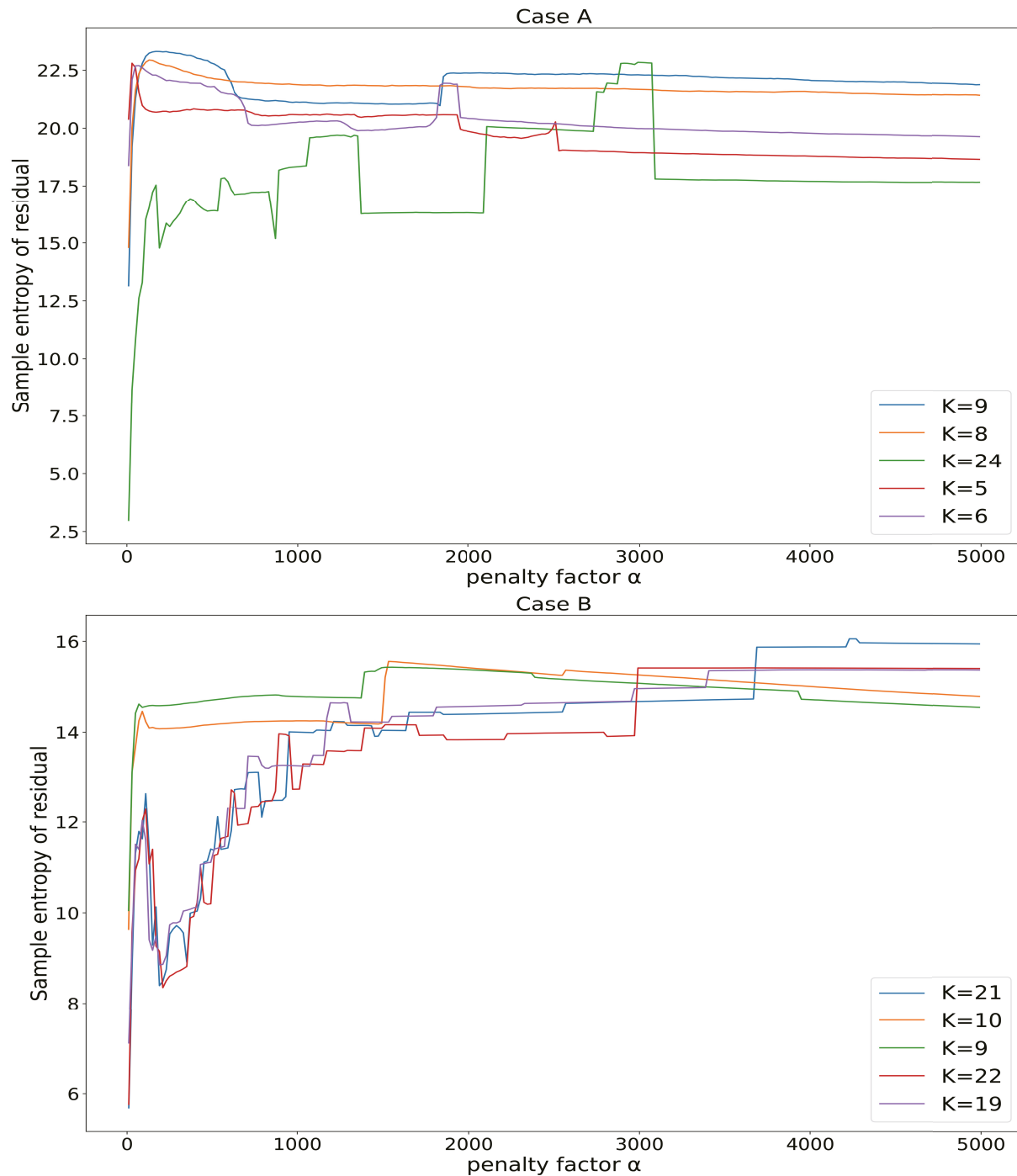


Figure 6. Relationship between the penalty factor and sample entropy of the residual.

3.4. Optimization Results Using WOA

In order to enhance the performance of the MVMD-TFT model, we employed the WOA to optimize its hyperparameters. In this paper, we set the population size and number

of iterations for the WOA to 10, taking into account the significant computational cost associated with the TFT training.

Figure 7 shows the optimal value obtained in each iteration from the WOA. In case A, during 10 iterations, the quantile loss was only optimized once, and the difference between the optimized loss and the previous loss was relatively small. In Case B, the quantile loss was optimized five times, with the resulting loss being lower than that of Case A. The hyperparameters of the optimized MVMD-TFT and corresponding search range are shown in Table 2.

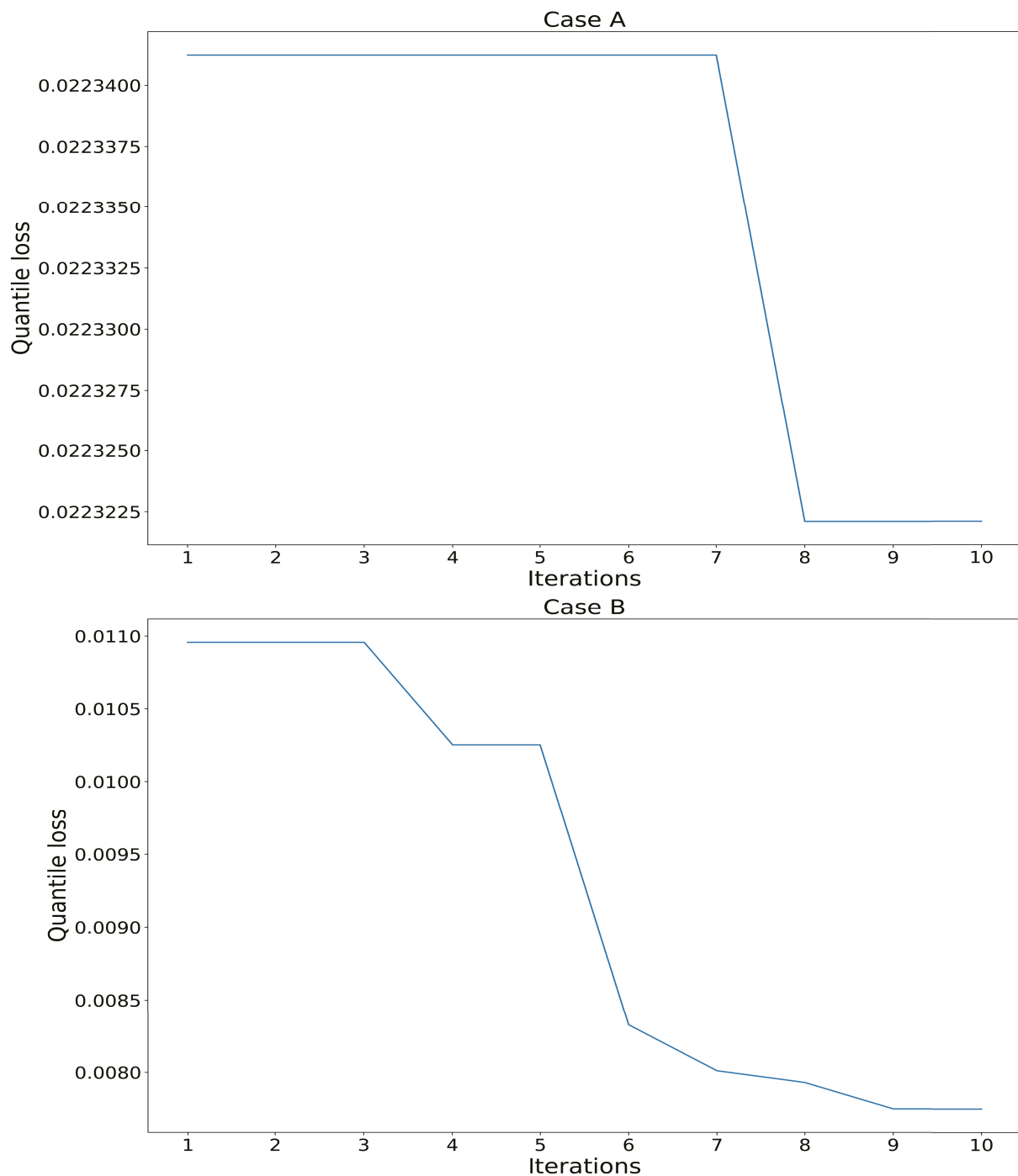


Figure 7. WOA optimization for the MVMD-TFT .

Table 2. Optimized hyperparameters for the MVMD-TFT.

Hyperparameter	Case A	Case B	Search Range
batch size	108	16	[4, 128]
hidden layer size	27	19	[5, 100]
number of heads	1	1	[1, 4]
learning rate	0.002	0.001	[0.0001, 0.01]
dropout rate	0.209	0.134	[0.1, 0.9]
max gradient norm	0.093	0.883	[0.1, 1]

3.5. Interpretability of MVMD-TFT

The TFT gives the interpretability between the input and output variables through the calculation in VSN. The results are presented in Figures 8–10.

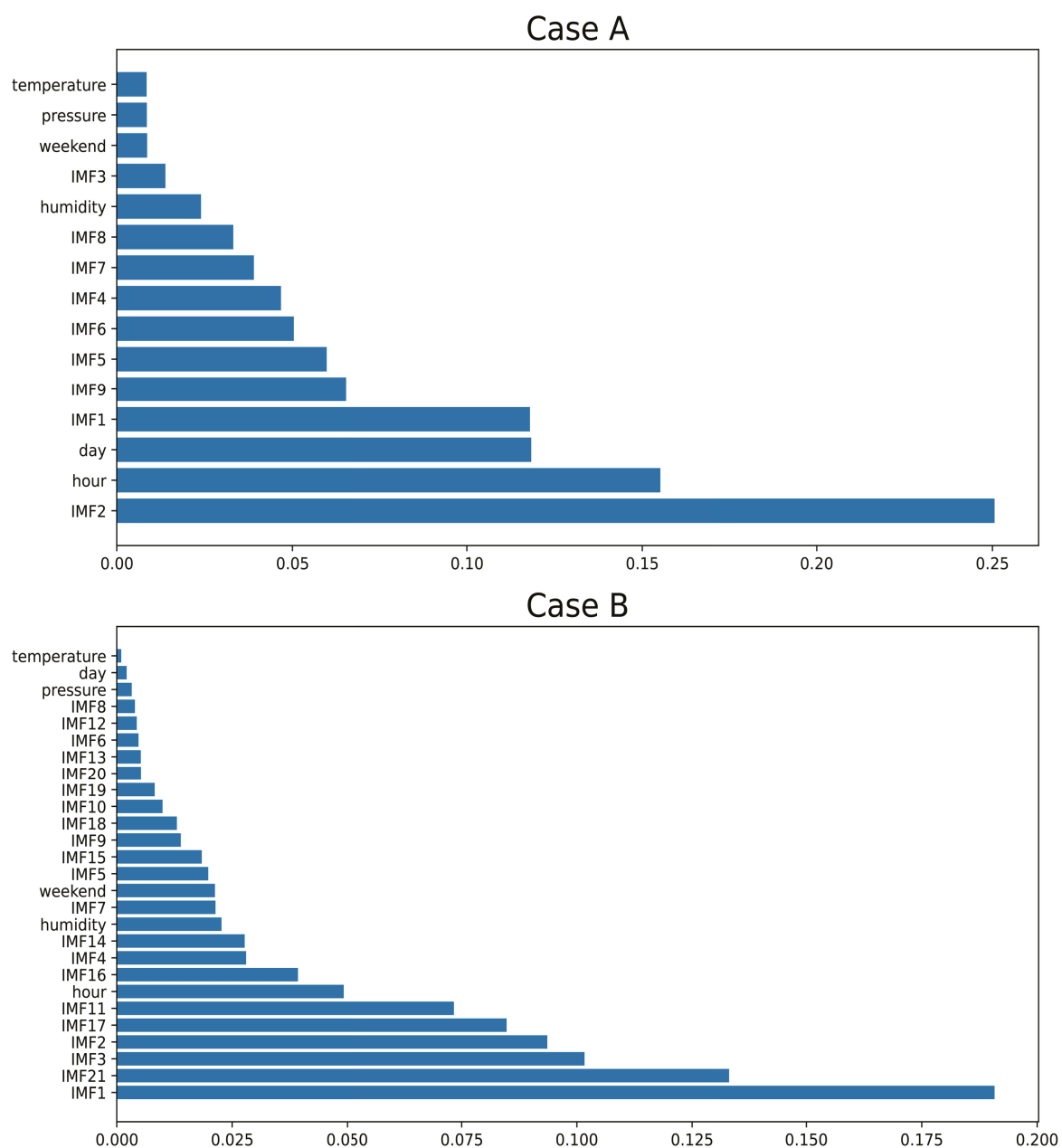
**Figure 8.** Variable importance of past inputs.

Figure 8 illustrates the weight distribution of VSN in past inputs, indicating that the IMFs obtained from MVMD emerge as a crucial factor, demonstrating a notable level of significance. Figure 9 illustrates the weight distribution of the VSN in future inputs, highlighting the significant roles played by the time indicators 'day' and 'hour'. Figure 10 presents the weight distribution of the VSN in static inputs, emphasizing 'building_id' as the most prominent feature.

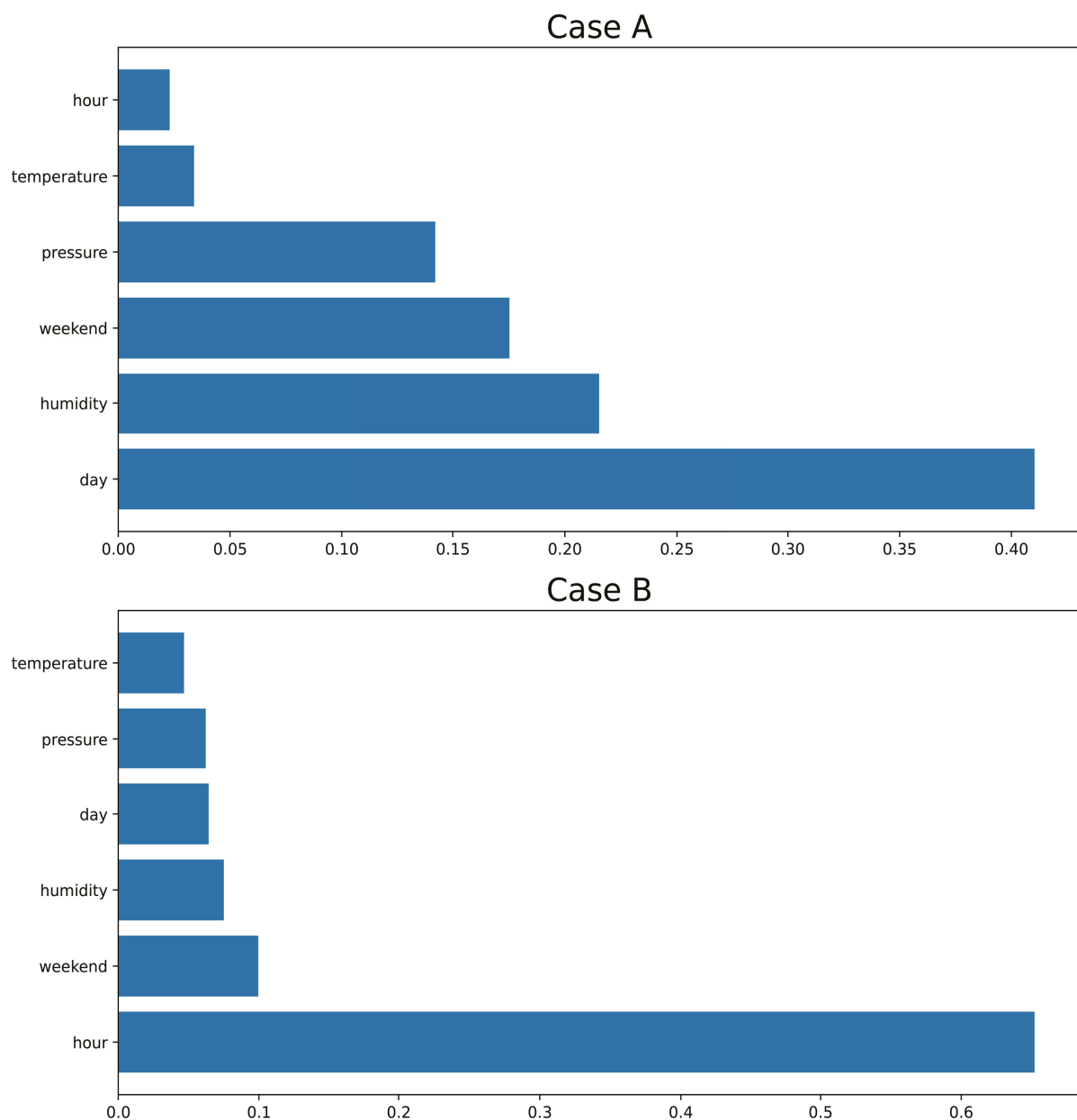


Figure 9. Variable importance of future variables.

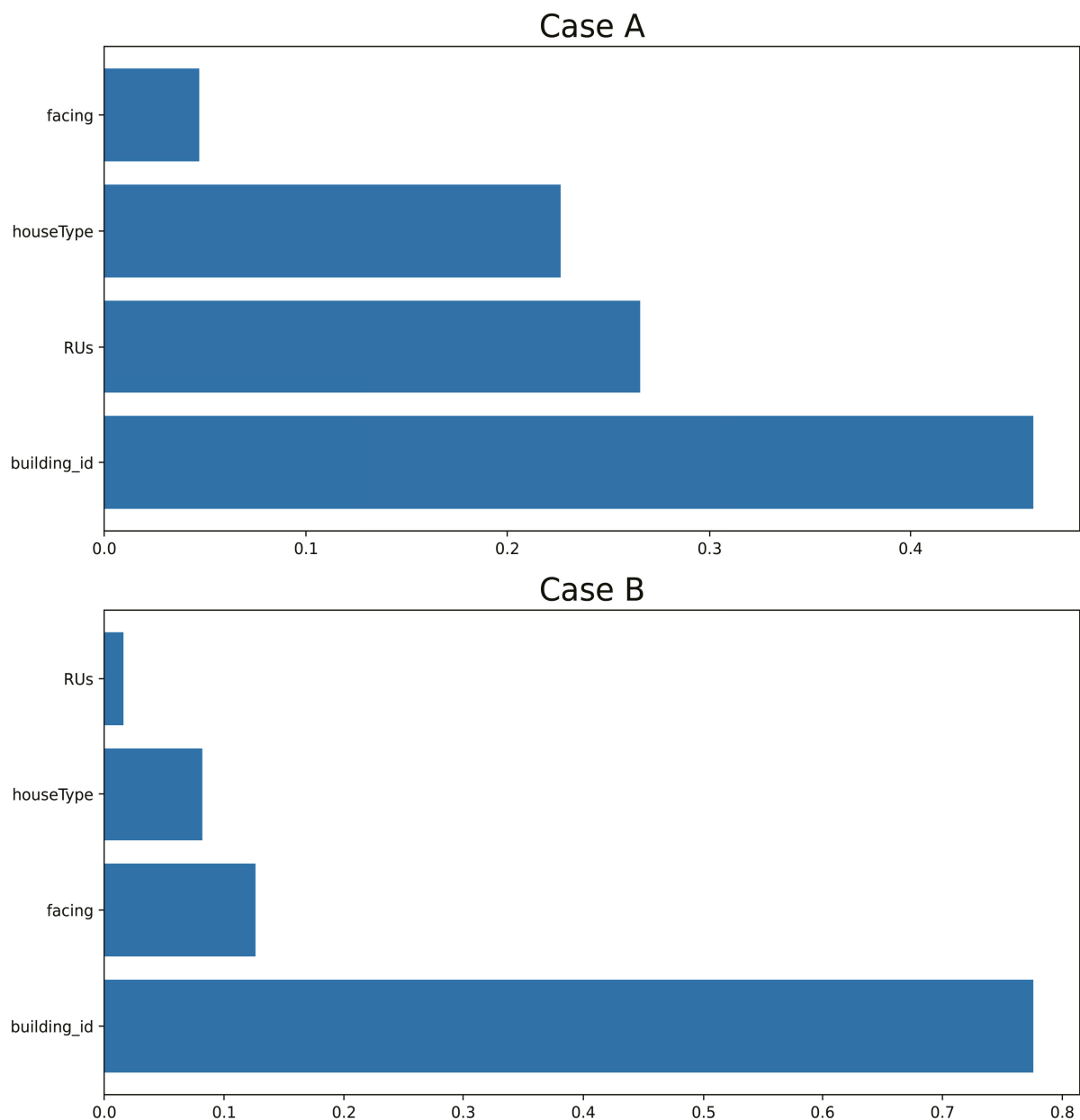


Figure 10. Variable importance of static variables.

3.6. Model Evaluation

For performance comparisons, we employed LSTM [6], CNN-LSTM, [7] and BiGRU-CNN [28]. A distinct model was trained for each building using these methods. In order to ensure consistency with the reference, we adopted the original preprocessing method, which involved representing the time indicator using one-hot encoding. The input for these models consisted of both time indicators and load series, enabling a comprehensive analysis of their predictive capabilities. Additionally, we conducted a comparative analysis with separately trained MVMD-LSTM and MVMD-CNN-LSTM models to further substantiate the efficacy of our method. The implementation of these standalone models followed the methodology outlined in [29], employing LSTM and CNN-LSTM architectures for the prediction of the IMFs and the subsequent reconstruction of the predicted signals. Concerning the CNN-LSTM-based, LSTM-based, and BiGRU-CNN models, we incorporated one of the lag inputs specified in [7]. Specifically, we configured it to 12. As a preliminary trial

for the TFT and MVMD-TFT models, we utilized 24 lag inputs. Table 3 displays other configurations of the models.

Table 3. Model configuration.

Model	Description
MVMD-TFT	(epochs = 50, patience = 5, loss = ‘quantile’)
TFT	(epochs = 50, patience = 5, loss = ‘quantile’, unit = 20, batch size = 54, number of heads = 4)
LSTM	same as [6] (epochs = 300, patience = 30, loss = ‘MAE’)
CNN-LSTM	same as [7] (epochs = 300, patience = 30, loss = ‘MAE’)
BiGRU-CNN	same as [28] (epochs = 300, patience = 30, loss = ‘MAE’, units = 20, filter = 20)
MVMD-LSTM	same as [6] (epochs = 300, patience = 30, loss = ‘MAE’)
MVMD-CNN-LSTM	same as [7] (epochs = 300, patience = 30, loss = ‘MAE’)

As our data contain both outliers and stable points, we utilized Equations (11)–(14) to evaluate the performance of the models, where y_i represents the actual value, and \hat{y}_i denotes the predicted value. MAE and MSE are the two most commonly used metrics in predictive problems, with MSE being more sensitive to outliers relative to MAE. MedAE serves as a robust measure of the variability of deviation of the observed values from the predict values. Additionally, we introduce WAPE to measure the percentage difference between actual and predicted values, as our data contain zeros, making this metric an alternative to MAPE. These four metrics provide different perspectives on the model’s performance. All methods exhibit a prediction horizon of 1, meaning that the predicted results correspond to the consumption data for the next hour. We made the original building name more concise by simplifying ‘Residential_’ to ‘R’.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (11)$$

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (12)$$

$$WAPE = \frac{\sum_{i=1}^N |y_i - \hat{y}_i|}{\sum_{i=1}^N |y_i|} \quad (13)$$

$$MedAE = median(|y_1 - \hat{y}_1|, \dots, |y_i - \hat{y}_i|) \quad (14)$$

Table 4 illustrates the prediction errors associated with Case A on the test set. The results show that the introduction of MVMD yielded substantial improvements in the performance of the CNN-LSTM, LSTM, and TFT models. A consistent decrease in all metrics shows this enhancement. Figure 11 depicts the corresponding prediction outcomes of ‘R11’ and ‘R24’ in Table 4, where the TFT and MVMD-TFT employ median forecasting. Table 5 illustrates the four loss function outcomes for Case B on the test set. We can derive similar conclusions to those of Case A when the dataset expands. It is worth noting that certain buildings, such as R13 and R5, demonstrate elevated MSE values that surpass the corresponding MAE values. After undergoing MVMD and subsequent prediction, it is observed that the MSE decreases to a level smaller than that of the MAE. This reveals that the implementation of MVMD preprocessing effectively attenuates the deleterious impact of outliers on predictive outcomes. Figure 12 aligns with the predictions of ‘R15’ and ‘R22’ outlined in Table 4.

Table 4. Evaluation metrics of different methods in Case A.

Building Name	Metric	MVMD-TFT	TFT	LSTM	CNN-LSTM	MVMD-LSTM	MVMD-CNN-LSTM	BiGRU-CNN
R3	MAE (kWh)	0.095	0.294	0.245	0.268	0.109	0.128	0.319
	MSE (kWh) ²	0.016	0.272	0.211	0.243	0.023	0.034	0.372
	WAPE (%)	10.2	31.5	26.3	28.8	11.7	13.7	34.2
	MedAE (kWh)	0.073	0.138	0.111	0.116	0.101	0.085	0.115
R4	MAE (kWh)	0.128	0.370	0.358	0.389	0.137	0.157	0.365
	MSE (kWh) ²	0.030	0.303	0.294	0.312	0.037	0.047	0.295
	WAPE (%)	7.5	21.9	21.2	23.0	8.2	9.3	21.7
	MedAE (kWh)	0.108	0.246	0.228	0.280	0.101	0.112	0.236
R5	MAE (kWh)	0.127	0.376	0.395	0.417	0.140	0.178	0.399
	MSE (kWh) ²	0.031	0.433	0.469	0.544	0.045	0.070	0.532
	WAPE (%)	13.3	39.2	41.3	43.5	14.6	18.6	41.7
	MedAE (kWh)	0.090	0.166	0.174	0.196	0.101	0.115	0.177
R6	MAE (kWh)	0.044	0.131	0.129	0.139	0.049	0.083	0.131
	MSE (kWh) ²	0.003	0.042	0.043	0.046	0.004	0.012	0.046
	WAPE (%)	9.1	26.9	26.5	28.7	10.1	17.2	26.9
	MedAE (kWh)	0.036	0.081	0.072	0.086	0.038	0.065	0.073
R9	MAE (kWh)	0.089	0.182	0.159	0.187	0.109	0.122	0.176
	MSE (kWh) ²	0.015	0.088	0.080	0.109	0.023	0.023	0.090
	WAPE (%)	13.2	27.0	23.5	27.7	16.1	18.1	26.0
	MedAE (kWh)	0.065	0.093	0.072	0.091	0.080	0.115	0.083
R10	MAE (kWh)	0.111	0.273	0.293	0.315	0.133	0.147	0.384
	MSE (kWh) ²	0.033	0.308	0.331	0.337	0.043	0.056	0.486
	WAPE (%)	16.7	41.1	44.0	47.3	20.0	22.0	57.7
	MedAE (kWh)	0.075	0.115	0.110	0.139	0.092	0.089	0.184
R11	MAE (kWh)	0.077	0.209	0.195	0.217	0.081	0.091	0.218
	MSE (kWh) ²	0.011	0.111	0.112	0.126	0.012	0.015	0.134
	WAPE (%)	13.7	36.9	34.5	38.3	14.4	16.2	38.5
	MedAE (kWh)	0.056	0.123	0.102	0.087	0.069	0.073	0.097
R13	MAE (kWh)	0.121	0.363	0.325	0.332	0.113	0.152	0.383
	MSE (kWh) ²	0.027	0.371	0.330	0.360	0.025	0.043	0.409
	WAPE (%)	10.1	30.3	27.1	27.8	9.4	12.7	32.0
	MedAE (kWh)	0.101	0.182	0.159	0.104	0.093	0.130	0.170
R14	MAE (kWh)	0.107	0.374	0.395	0.446	0.136	0.159	0.402
	MSE (kWh) ²	0.020	0.322	0.337	0.382	0.035	0.041	0.360
	WAPE (%)	6.7	23.6	24.8	28.1	8.6	10.0	25.3
	MedAE (kWh)	0.085	0.232	0.264	0.316	0.096	0.122	0.250
R19	MAE (kWh)	0.123	0.386	0.346	0.342	0.120	0.139	0.367
	MSE (kWh) ²	0.025	0.264	0.252	0.241	0.025	0.033	0.272
	WAPE (%)	6.1	18.9	17.0	16.8	5.9	6.8	18.0
	MedAE (kWh)	0.095	0.299	0.250	0.231	0.103	0.106	0.258
R20	MAE (kWh)	0.115	0.333	0.322	0.294	0.119	0.135	0.343
	MSE (kWh) ²	0.021	0.273	0.247	0.191	0.028	0.036	0.283
	WAPE (%)	9.1	26.2	24.7	23.1	9.3	10.6	27.0
	MedAE (kWh)	0.100	0.181	0.185	0.179	0.077	0.099	0.205
R21	MAE (kWh)	0.049	0.124	0.127	0.139	0.053	0.078	0.124
	MSE (kWh) ²	0.005	0.070	0.074	0.079	0.008	0.015	0.074
	WAPE (%)	16.9	42.8	44.6	48.2	18.5	26.9	42.9
	MedAE (kWh)	0.033	0.059	0.067	0.053	0.034	0.054	0.056

Table 4. Cont.

Building Name	Metric	MVMD-TFT	TFT	LSTM	CNN-LSTM	MVMD-LSTM	MVMD-CNN-LSTM	BiGRU-CNN
R24	MAE (kWh)	0.060	0.185	0.174	0.196	0.094	0.116	0.187
	MSE (kWh) ²	0.009	0.115	0.116	0.126	0.026	0.033	0.118
	WAPE (%)	11.6	35.7	33.6	37.7	18.2	22.3	36.1
	MedAE (kWh)	0.037	0.094	0.079	0.102	0.062	0.080	0.108
R25	MAE (kWh)	0.089	0.246	0.252	0.303	0.119	0.155	0.236
	MSE (kWh) ²	0.014	0.231	0.253	0.271	0.031	0.052	0.240
	WAPE (%)	14.2	39.5	40.4	48.5	19.1	24.8	37.9
	MedAE (kWh)	0.071	0.118	0.122	0.161	0.083	0.103	0.093

The bold represent the best performance.

Table 5. Evaluation metrics of different methods in Case B.

Building Name	Metric	MVMD-TFT	TFT	LSTM	CNN-LSTM	MVMD-LSTM	MVMD-CNN-LSTM	BiGRU-CNN
R4	MAE (kWh)	0.070	0.304	0.314	0.329	0.079	0.150	0.323
	MSE (kWh) ²	0.008	0.206	0.212	0.229	0.010	0.038	0.227
	WAPE (%)	5.6	24.4	25.1	26.4	6.3	12.0	25.9
	MedAE (kWh)	0.058	0.199	0.196	0.214	0.066	0.114	0.197
R5	MAE(kWh)	0.084	0.335	0.365	0.385	0.098	0.168	0.377
	MSE (kWh) ²	0.012	0.403	0.457	0.518	0.018	0.058	0.482
	WAPE (%)	10.6	42.2	46.0	48.5	12.3	21.1	47.6
	MedAE (kWh)	0.062	0.127	0.135	0.127	0.075	0.120	0.137
R6	MAE (kWh)	0.028	0.115	0.103	0.103	0.028	0.048	0.112
	MSE (kWh) ²	0.002	0.041	0.040	0.038	0.001	0.005	0.047
	WAPE (%)	8.6	34.9	31.4	31.9	8.4	14.6	34.0
	MedAE (kWh)	0.022	0.062	0.039	0.043	0.021	0.035	0.041
R9	MAE (kWh)	0.048	0.186	0.186	0.187	0.060	0.100	0.193
	MSE (kWh) ²	0.004	0.119	0.121	0.133	0.007	0.024	0.136
	WAPE (%)	7.9	30.5	30.5	30.7	9.8	16.5	31.7
	MedAE (kWh)	0.036	0.078	0.075	0.069	0.045	0.066	0.078
R10	MAE (kWh)	0.062	0.228	0.228	0.243	0.066	0.106	0.237
	MSE (kWh) ²	0.007	0.207	0.210	0.251	0.008	0.028	0.233
	WAPE (%)	10.4	38.5	38.6	41.0	11.1	17.9	40.0
	MedAE (kWh)	0.043	0.098	0.097	0.084	0.051	0.068	0.085
R13	MAE (kWh)	0.083	0.295	0.333	0.364	0.098	0.160	0.337
	MSE (kWh) ²	0.011	0.315	0.358	0.431	0.017	0.045	0.403
	WAPE (%)	8.3	29.3	33.3	36.2	9.7	16.0	33.5
	MedAE (kWh)	0.069	0.109	0.145	0.147	0.076	0.131	0.128
R14	MAE (kWh)	0.084	0.372	0.371	0.385	0.085	0.162	0.382
	MSE (kWh) ²	0.014	0.390	0.387	0.417	0.016	0.058	0.412
	WAPE (%)	5.2	23.1	23.0	23.9	5.3	10.1	23.7
	MedAE (kWh)	0.065	0.194	0.186	0.209	0.062	0.130	0.191
R15	MAE (kWh)	0.161	0.643	0.512	0.552	0.332	0.321	0.611
	MSE (kWh) ²	0.056	1.700	1.418	1.428	0.230	0.231	1.723
	WAPE (%)	13.6	54.4	44.3	46.7	28.1	27.2	51.7
	MedAE (kWh)	0.094	0.216	0.166	0.196	0.200	0.204	0.219

Table 5. Cont.

Building Name	Metric	MVMD-TFT	TFT	LSTM	CNN-LSTM	MVMD-LSTM	MVMD-CNN-LSTM	BiGRU-CNN
R20	MAE (kWh)	0.055	0.258	0.257	0.272	0.062	0.101	0.268
	MSE (kWh) ²	0.005	0.181	0.178	0.205	0.007	0.022	0.200
	WAPE (%)	6.2	29.1	29.0	30.7	7.0	11.4	30.2
	MedAE (kWh)	0.044	0.148	0.146	0.148	0.046	0.073	0.140
R22	MAE (kWh)	0.060	0.245	0.210	0.205	0.084	0.105	0.213
	MSE (kWh) ²	0.008	0.210	0.235	0.235	0.015	0.024	0.245
	WAPE (%)	17.2	69.9	59.9	58.5	24.2	30.1	60.7
	MedAE (kWh)	0.041	0.135	0.066	0.059	0.062	0.079	0.062

The bold represent the best performance.

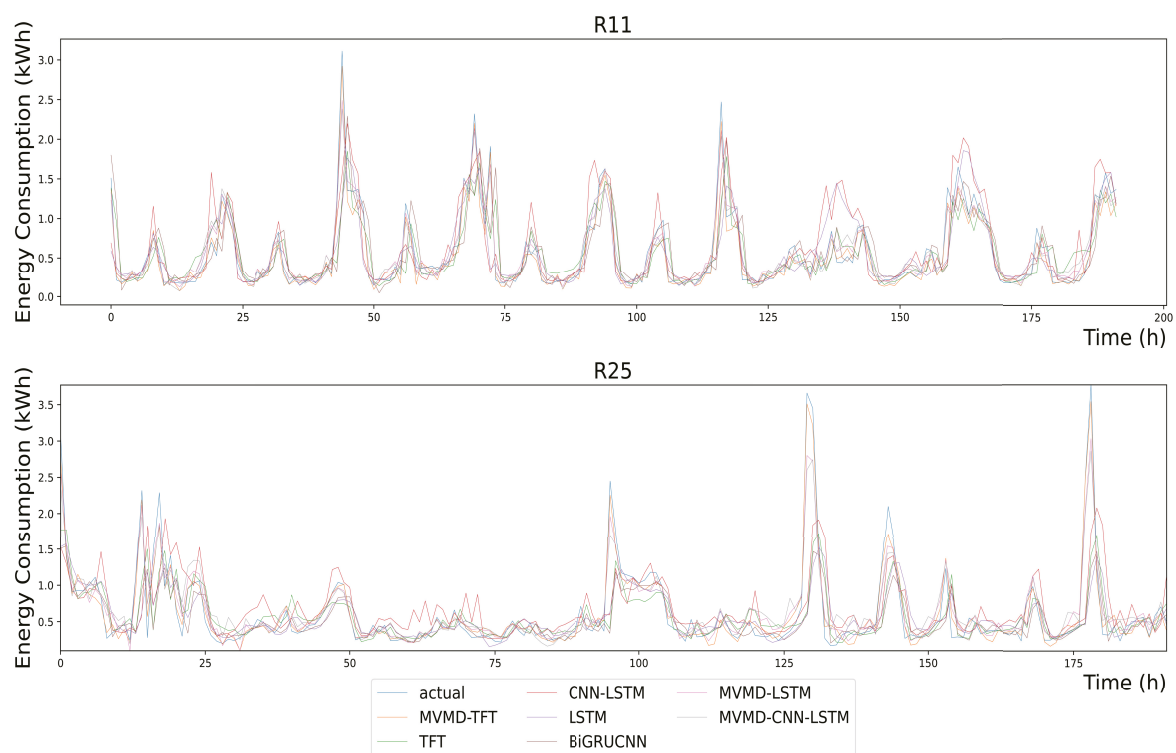


Figure 11. Forecasting results for Case A (22 January 2018 00:00:00 to 29 January 2018 23:00:00).

The results in Tables 4 and 5 reveal a similarity in the experimental outcomes between the MVMD-based Models. In order to establish the statistical significance of the results displayed in Tables 4 and 5, we introduce the Friedman and post-hoc Nemenyi tests [30] to assess the differences among various models for evaluation. The null hypothesis of the Friedman test is that there is no difference among all comparison methods in the 24 datasets of Case A and Case B. We set the p -value to be 0.05. If the calculation result of the Friedman test is less than 0.05, the null hypothesis is rejected; otherwise, it indicates a difference between the methods. Furthermore, to assess the performance between pairwise models, we introduced the Nemenyi post-hoc test. This test launches a comparison between a threshold (critical difference) and the difference in average rankings of the performance. If the ranking difference is lower than the threshold, it is considered that there is no significant difference in performance between the pairwise models. On the contrary, there is a significant performance difference between them. The evaluation metric of the Nemenyi test is MAE.

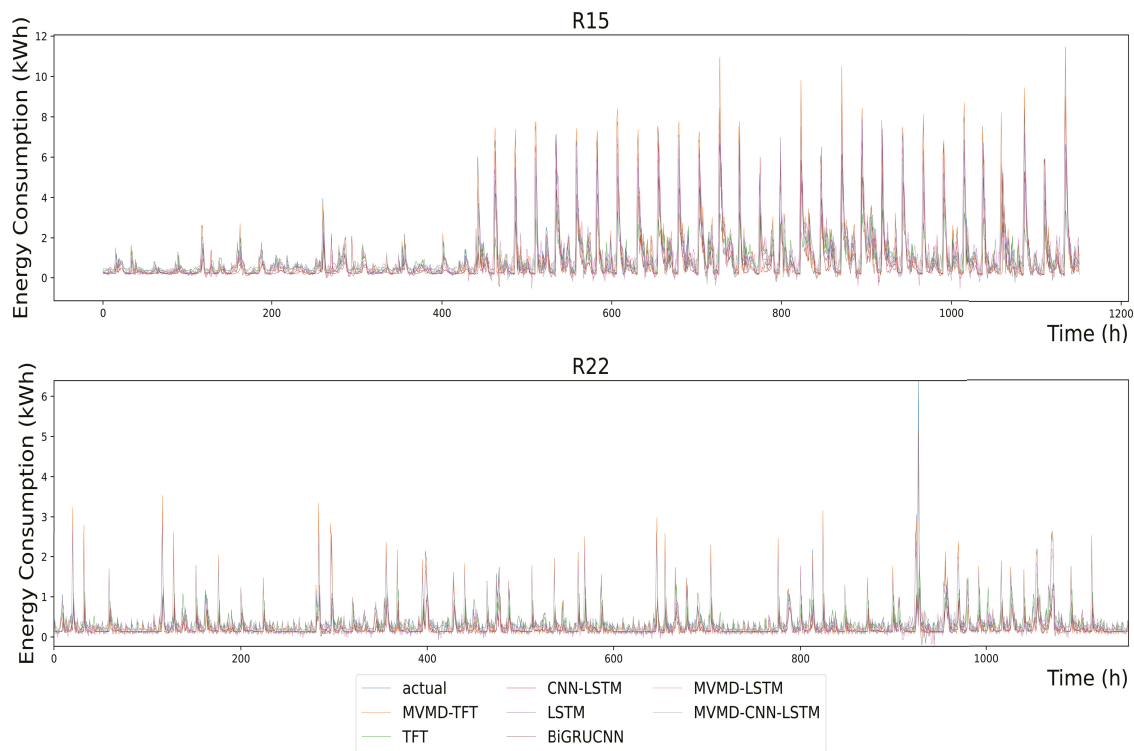


Figure 12. Forecasting results for Case B (14 September 2017 00:00:00 to 31 October 2017 23:00:00).

After performing the calculations on 24 datasets, we obtained a p -value of 5.3×10^{-19} for the Friedman test. This is significantly lower than the preset threshold of 0.05, indicating a significant difference in performance among the seven methods.

Figure 13 displays the results of the Nemenyi post-hoc tests, with a calculated CD value of 1.84. The results of the Nemenyi tests indicate that there are no significant differences in the performances of the CNN-LSTM, TFT, LSTM, and BiGRU-CNN. In the case of the MVMD-TFT, it shows differences with all non-MVMD-based methods. Based on the results from Tables 4 and 5, our method achieved an average reduction of 69.9% in MAE for an individually trained CNN-LSTM and BiGRU-CNN, as well as a 67.7% reduction in MAE for an individually trained LSTM. Although no significant performance differences were detected for the MVMD-based methods, the proposed method demonstrated the best performance in the current experiment.

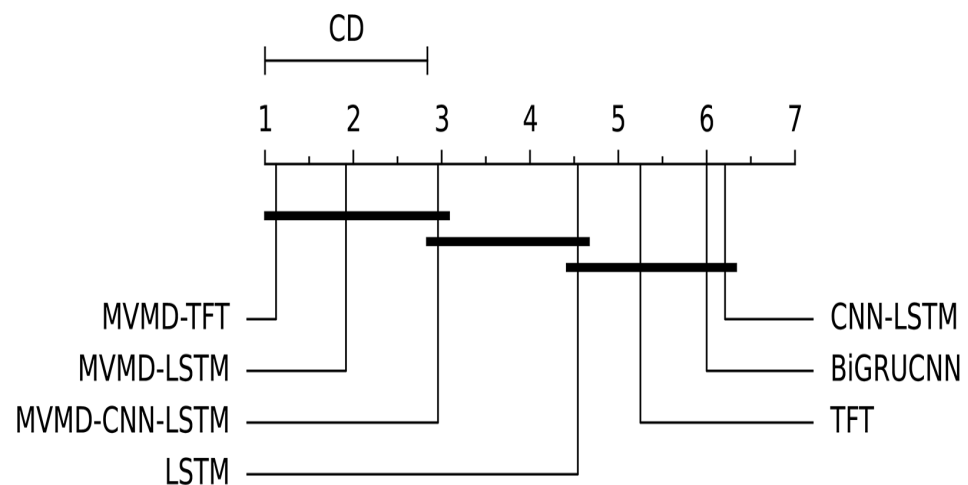


Figure 13. Nemenyi post-hoc test (p -value (calculated by Friedman test) = 5.3×10^{-19}).

One of the important features of the TFT is that it provides a distribution of possible future outcomes along with point estimates, which is valuable for understanding the uncertainty associated with each prediction. Given that our model utilizes a quantile set of (0.1, 0.5, 0.9), the TFT produces a prediction interval of 80%. Figure 14 presents both the median prediction results and their associated 80% prediction interval in two chosen buildings where peak values are observed. The results reflect the last 8 days from the entire test set. The proposed model attains a narrower prediction interval than the original method and is more likely to encompass peak values.

$$q\text{-Risk} = \frac{2\sum_{i,t} QL(y_i, \hat{y}_i, q)}{\sum_{i,t} |y_i|} \quad (15)$$

In order to perform a thorough assessment of the quantile forecast, we utilized the P50 loss and P90 loss, as specified in [31] and described in Equation (15). In the context of the 8-day evaluation, the test set comprises a total of 192 data points. Equation (15) is applied to compute the q -Risk value for each data point in the 1-hour-ahead forecast. Consequently, the average quantile loss for the 192 points was derived.

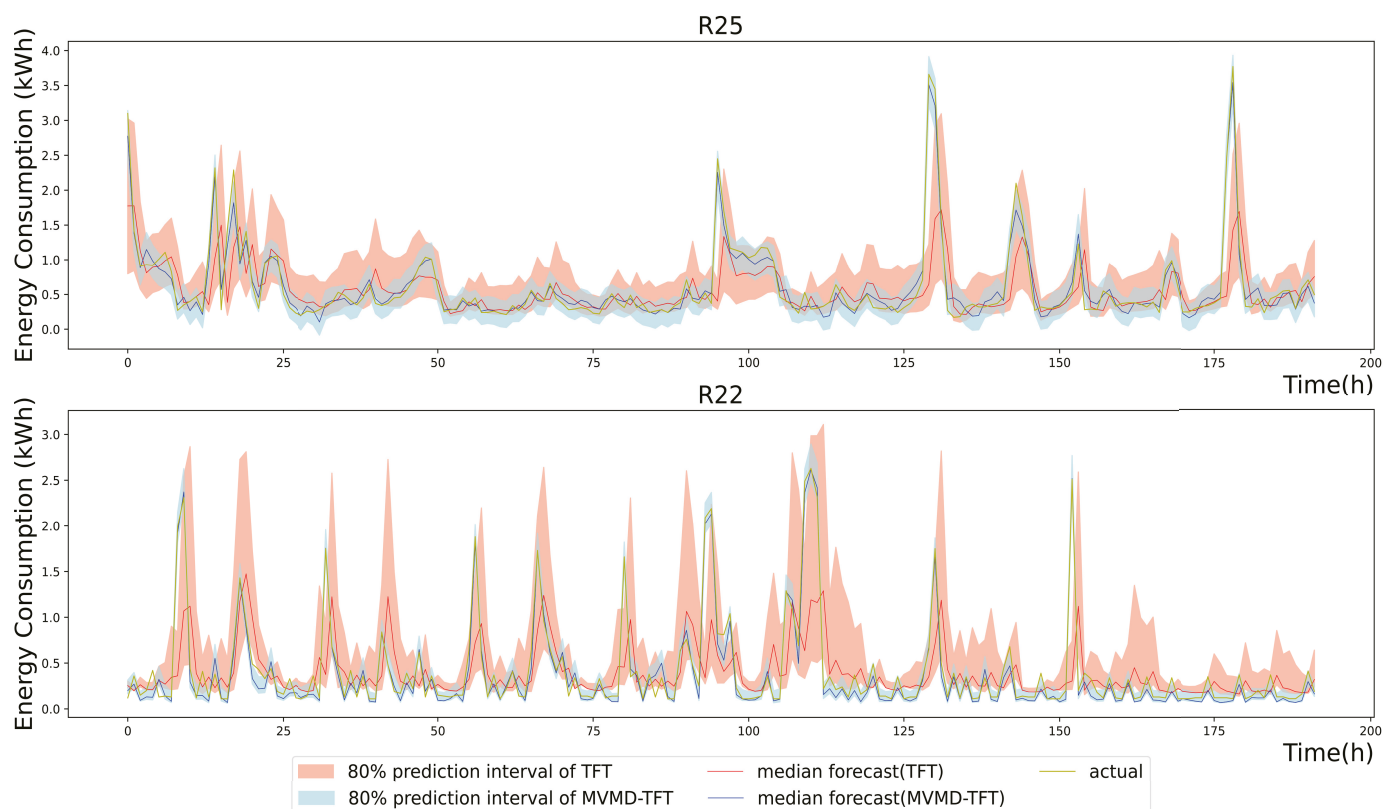


Figure 14. Quantile prediction results of the MVMD-TFT and TFT. Case A (R25) (22 January 2018 00:00:00 to 29 January 2018 23:00:00). Case B (R22) (24 October 2017 00:00:00 to 31 October 2017 23:00:00).

Table 6 showcases the average q -Risk value for the forecast of the MVMD-TFT and TFT across all points in both cases. The results show that our method yields 75.9% lower P90 loss and 65.8% lower P50 loss on average, providing additional evidence for the effectiveness of our method.

Although our method has successfully produced accurate predictions in two instances, it is burdened by substantial time-related limitations that cannot be ignored. This deficiency is particularly evident in the WOA, where the computational demands of the TFT impose limitations on the selection of optimal initial parameters for the WOA. Consequently, the convergence of the WOA is insufficient, with the magnitude of this flaw becoming

increasingly apparent as the dataset size and model complexity grow. Moreover, our method incorporates MVMD, which requires substantial time in data preprocessing to ascertain the most appropriate parameters. The determination of the parameters of the two cases together is approximately seven days. These limitations underscore the need for further improvements in our approach to address the substantial time constraints associated with the optimization process for the TFT and MVMD.

Table 6. Average q -Risk value.

Metric	Case A		Case B	
	MVMD-TFT	TFT	MVMD-TFT	TFT
P50 loss	0.104	0.271	0.094	0.315
P90 loss	0.048	0.172	0.043	0.213

4. Conclusions

In this paper, we propose a novel MVMD-WOA-TFT hybrid model for accurate hourly load consumption forecasting across multiple houses. MVMD is leveraged to decompose the original load series into multiple IMFs. This enables the extraction of shared characteristics from various load series, thereby assisting the TFT in comprehending the underlying patterns and relationships among different load sequences. In order to select a suitable decomposition level and penalty factors for MVMD, we employed a method that maximizes the sum of the residual sample entropy, as a higher entropy signifies a better decomposition by capturing more noise in the residual. Subsequently, we partitioned the dataset into training, validation, and test sets and performed separate decompositions on each. The WOA was employed to determine the hyperparameters of the MVMD-TFT model, improving its overall performance. We used separately trained LSTM, CNN-LSTM, BiGRU-CNN, MVMD-LSTM, and MVMD-CNN-LSTM models for performance comparisons. Our approach exhibits competitive performance compared to MVMD-LSTM and MVMD-CNN-LSTM for the 24 datasets, but it achieved excellent performance with the non-MVMD method. Our method achieved an average reduction of 69.9% for CNN-LSTM and BiGRU-CNN, as well as a 67.7% reduction for LSTM in MAE. We conducted additional evaluations on quantile predictions for the TFT, achieving an average improvement of 65.8% at 0.5 risk and 75.9% at 0.9 risk. However, it is worth noting that the three processes (MVMD, WOA, and TFT) involved in our methodology entail a high computational cost. Additionally, our study only focused on 1-hour-ahead prediction and utilized fixed lag inputs to examine the prediction accuracy. Research is warranted to examine the influence of lag inputs and variations in the prediction range on the performance of the model.

Author Contributions: Conceptualization, Q.Z. and X.Z.; methodology, H.Y. and Q.Z.; software, H.Y.; validation, H.Y. and Q.Z.; formal analysis, H.Y.; resources, X.Z.; writing—original draft preparation, H.Y.; writing—review and editing, Q.Z. and X.Z.; supervision, X.Z. 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: Data are contained within the article.

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

Abbreviations

The following abbreviations are used in this manuscript:

AdaBoost	Adaptive boosting
ADMM	Alternating direction method of multipliers
ARIMA	Autoregressive integrated moving average
BiGRU	Bidirectional Gated recurrent unit network
BiLSTM	Bidirectional long short-term memory network
CD	Critical Difference
CNN	Convolutional neural network
DBSCAN	Density-based spatial clustering of applications with noise
EEMD	Ensemble empirical mode decomposition
ELM	Extreme learning machine
EMD	Empirical mode decomposition
EWT	Empirical wavelet transform
GRU	Gated recurrent unit network
IMF	Intrinsic mode function
LSSVM	Least squared support vector machine
LSTM	Long short-term memory network
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MedAE	Median absolute error
MLR	Multiple linear regression
MSE	Mean squared error
MVMD	Multivariate variational mode decomposition
RNN	Recurrent neural network
SVR	Support vector regression
TFT	Temporal fusion transformer
VMD	Variational mode decomposition
VSN	Variable selection network
WAPE	Weighted average percentage error
XGBoost	Extreme gradient boosting

References

1. González-Torres, M.; Pérez-Lombard, L.; Coronel, J.F.; Maestre, I.R.; Yan, D. A review on buildings energy information: Trends, end-uses, fuels and drivers. *Energy Rep.* **2022**, *8*, 626–637. [\[CrossRef\]](#)
2. Guo, X.; Gao, Y.; Li, Y.; Zheng, D.; Shan, D. Short-term household load forecasting based on Long- and Short-term Time-series network. *Energy Rep.* **2021**, *7*, 58–64. [\[CrossRef\]](#)
3. Al-Rakhami, M.; Gumaei, A.; Alsanad, A.; Alamri, A.; Hassan, M.M. An Ensemble Learning Approach for Accurate Energy Load Prediction in Residential Buildings. *IEEE Access* **2019**, *7*, 48328–48338. [\[CrossRef\]](#)
4. Chen, S.; Zhou, X.; Zhou, G.; Fan, C.; Ding, P.; Chen, Q. An online physical-based multiple linear regression model for building's hourly cooling load prediction. *Energy Build.* **2022**, *254*, 111574. [\[CrossRef\]](#)
5. Chen, Y.; Xu, P.; Chu, Y.; Li, W.; Wu, Y.; Ni, L.; Bao, Y.; Wang, K. Short-term electrical load forecasting using the Support Vector Regression (SVR) model to calculate the demand response baseline for office buildings. *Appl. Energy* **2017**, *195*, 659–670. [\[CrossRef\]](#)
6. Kong, W.; Dong, Z.Y.; 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. [\[CrossRef\]](#)
7. Alhussein, M.; Aurangzeb, K.; Haider, S.I. Hybrid CNN-LSTM Model for Short-Term Individual Household Load Forecasting. *IEEE Access* **2020**, *8*, 180544–180557. [\[CrossRef\]](#)
8. Peng, C.; Tao, Y.; Chen, Z.; Zhang, Y.; Sun, X. Multi-source transfer learning guided ensemble LSTM for building multi-load forecasting. *Expert Syst. Appl.* **2022**, *202*, 117194. [\[CrossRef\]](#)
9. Zhu, N.; Wang, Y.; Yuan, K.; Yan, J.; Li, Y.; Zhang, K. GGNet: A novel graph structure for power forecasting in renewable power plants considering temporal lead-lag correlations. *Appl. Energy* **2024**, *364*, 123194. [\[CrossRef\]](#)
10. Rumbe, G.; Hamasha, M.; Mashaqbeh, S.A. A comparison of Holts-Winter and Artificial Neural Network approach in forecasting: A case study for tent manufacturing industry. *Results Eng.* **2024**, *21*, 101899. [\[CrossRef\]](#)
11. Tarmanini, C.; Sarma, N.; Gezezin, C.; Ozgonenel, O. Short term load forecasting based on ARIMA and ANN approaches. *Energy Rep.* **2023**, *9*, 550–557. [\[CrossRef\]](#)

12. 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. [\[CrossRef\]](#)
13. Yuan, J.; Wang, L.; Qiu, Y.; Wang, J.; Zhang, H.; Liao, Y. Short-term electric load forecasting based on improved Extreme Learning Machine Mode. *Energy Rep.* **2021**, *7*, 1563–1573. [\[CrossRef\]](#)
14. Short-Term Load Forecasting Method Based on EWT and IDBSCAN. *J. Electr. Eng. Technol.* **2020**, *15*, 58–64.
15. Lv, L.; Wu, Z.; Zhang, J.; Zhang, L.; Tan, Z.; Tian, Z. A VMD and LSTM Based Hybrid Model of Load Forecasting for Power Grid Security. *IEEE Trans. Ind. Inform.* **2022**, *18*, 6474–6482. [\[CrossRef\]](#)
16. Lusi, P.; Khalilpour, K.R.; Andrew, L.; Liebman, A. Short-term residential load forecasting: Impact of calendar effects and forecast granularity. *Appl. Energy* **2017**, *205*, 654–669. [\[CrossRef\]](#)
17. Kong, W.; Dong, Z.Y.; Hill, D.J.; Luo, F.; Xu, Y. Short-Term Residential Load Forecasting Based on Resident Behaviour Learning. *IEEE Trans. Power Syst.* **2018**, *33*, 1087–1088. [\[CrossRef\]](#)
18. Huy, P.C.; Minh, N.Q.; Tien, N.D.; Anh, T.T.Q. Short-Term Electricity Load Forecasting Based on Temporal Fusion Transformer Model. *IEEE Access* **2022**, *10*, 106296–106304. [\[CrossRef\]](#)
19. López Santos, M.; García-Santiago, X.; Echevarría Camarero, F.; Blázquez Gil, G.; Carrasco Ortega, P. Application of Temporal Fusion Transformer for Day-Ahead PV Power Forecasting. *Energies* **2022**, *15*, 5232. [\[CrossRef\]](#)
20. Wu, B.; Wang, L.; Zeng, Y. Interpretable wind speed prediction with multivariate time series and temporal fusion transformers. *Energy* **2022**, *252*, 123990. [\[CrossRef\]](#)
21. Feng, G.; Zhang, L.; Ai, F.; Zhang, Y.; Hou, Y. An Improved Temporal Fusion Transformers Model for Predicting Supply Air Temperature in High-Speed Railway Carriages. *Entropy* **2022**, *24*, 1111. [\[CrossRef\]](#) [\[PubMed\]](#)
22. Wu, B.; Wang, L.; Zeng, Y.R. Interpretable tourism demand forecasting with temporal fusion transformers amid COVID-19. *Appl. Intell.* **2023**, *53*, 14493–14514. [\[CrossRef\]](#) [\[PubMed\]](#)
23. Wu, B.; Wang, L.; Tao, R.; Zeng, Y.R. Interpretable tourism volume forecasting with multivariate time series under the impact of COVID-19. *Neural Comput. Appl.* **2023**, *35*, 5437–5463. [\[CrossRef\]](#)
24. Rehman, N.U.; Aftab, H. Multivariate Variational Mode Decomposition. *IEEE Trans. Signal Process.* **2019**, *67*, 6039–6052. [\[CrossRef\]](#)
25. Wang, Y.; Sun, S.; Chen, X.; Zeng, X.; Kong, Y.; Chen, J.; Guo, Y.; Wang, T. Short-term load forecasting of industrial customers based on SVM and XGBoost. *Int. J. Electr. Power Energy Syst.* **2021**, *129*, 106830. [\[CrossRef\]](#)
26. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [\[CrossRef\]](#)
27. Makonin, S. HUE: The Hourly Usage of Energy Dataset for Buildings in British Columbia. *Data Brief* **2019**, *23*, 103744. [\[CrossRef\]](#)
28. Soares, L.D.; Franco, E.M.C. BiGRU-CNN neural network applied to short-term electric load forecasting. *Production* **2022**, *32*. [\[CrossRef\]](#)
29. Zhang, K.; Yang, X.; Wang, T.; Thé, J.; Tan, Z.; Yu, H. Multi-step carbon price forecasting using a hybrid model based on multivariate decomposition strategy and deep learning algorithms. *J. Clean. Prod.* **2023**, *405*, 136959. [\[CrossRef\]](#)
30. Cai, J.; Wang, C.; Hu, K. LCDFormer: Long-term correlations dual-graph transformer for traffic forecasting. *Expert Syst. Appl.* **2024**, *249*, 123721. [\[CrossRef\]](#)
31. Lim, B.; Arik, S.Ö.; Loeff, N.; Pfister, T. Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *Int. J. Forecast.* **2021**, *37*, 1748–1764. [\[CrossRef\]](#)

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.