Buletin Ilmiah Sarjana Teknik Elektro
Vol. 8, No. 4, August 2026, pp. 1259-1298
ISSN: 2685-9572, DOI: 10.12928/biste.v8i4.17107

Dynamic Multi-Scale Ensemble iTransformer for Day-Ahead Electric Load Forecasting

Tuan Anh Nguyen, Trung Dung Nguyen
FEET, Industrial University of Ho Chi Minh City (IUH), Ho Chi Minh City, Viet Nam

ARTICLE INFORMATION

Article History:

Received 15 June 2026
Revised 05 August 2026
Accepted 09 September 2026

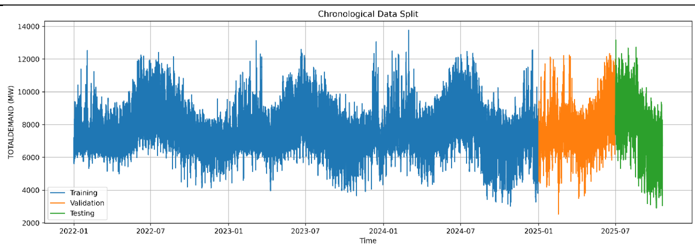
Keywords:

Electric Load Forecasting;
iTransformer;
Multi-Scale Temporal Learning;
Dynamic Ensemble;
Day-Ahead Forecasting

Corresponding Author:

Trung Dung Nguyen,
FEET, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Viet Nam.
Email: nguyentrungdung@iuh.edu.vn.

ABSTRACT



Accurate day-ahead electric load forecasting is essential for reliable power-system operation, yet high-resolution demand exhibits nonlinear short-term variations alongside strong daily and weekly recurrence. This study addresses the difficulty of representing these complementary temporal patterns within a single forecasting model. The research contribution is a validation-driven Dynamic Multi-Scale Ensemble iTransformer framework that integrates a tuned iTransformer with explicit daily, weekly, multi-week, and load-profile references, removes highly redundant candidates, and adaptively combines the retained forecasts across load states and forecast-horizon blocks. The framework uses only historical load observations and follows a chronological, leakage-free protocol for model tuning, candidate selection, weight optimization, strategy selection, and final testing. It was evaluated on New South Wales electricity demand data sampled at 5-minute intervals, using the previous 288 observations to forecast the next 288 observations. The proposed model achieved the best overall performance among the six evaluated methods. It yielded an MAE of 478.485 MW, an RMSE of 690.116 MW, a MAPE of 7.031%, an sMAPE of 6.718%, and an R^2 of 0.8484. Its MAPE was lower than those of Seasonal Naive (7.979%) and the standalone iTransformer (8.691%), corresponding to relative reductions of 11.88% and 19.10%, respectively. CNN, Persistence Naive, and LSTM achieved MAPEs of 12.419%, 17.596%, and 19.830%, respectively. These results show that explicit multi-scale temporal references and validation-optimized adaptive fusion complement the learned iTransformer representation, thereby improving deterministic day-ahead load forecasting accuracy.

Document Citation:

T. A. Nguyen and T. D. Nguyen, "Dynamic Multi-Scale Ensemble iTransformer for Day-Ahead Electric Load Forecasting," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 8, no. 4, pp. 1259-1298, 2026, DOI: 10.12928/biste.v8i4.17107.

1. INTRODUCTION

Accurate short-term electric load forecasting (STLF) is essential for the secure, economical, and efficient operation of modern power systems. Reliable load forecasts support generation scheduling, reserve allocation, electricity market operations, demand-side management, and other time-sensitive operational decisions. Forecasting errors can directly affect power-system performance. Underestimation may result in insufficient reserve preparation and increased operational risk, whereas overestimation can lead to unnecessary generation commitment and higher operating costs. These requirements are particularly important for day-ahead forecasting, in which system operators require a complete future demand trajectory rather than a single predicted value. For the electricity demand recorded at 5-min intervals, this task requires predicting 288 consecutive observations over the following 24 h.

Despite its practical importance, day-ahead STLF remains challenging because electricity demand exhibits nonlinear, non-stationary, and multi-scale temporal behavior. Short-term fluctuations, abrupt ramps, intraday patterns, daily and weekly recurrence, and gradual changes in the overall demand level coexist within the same time series. Furthermore, the relative importance of these patterns may vary across load conditions and forecast lead times. A temporal pattern that is informative during a stable operating period may become less useful during rapid demand changes or high-load conditions. Therefore, a single forecasting mechanism may not consistently capture all relevant temporal structures over the entire day-ahead horizon.

Early STLF studies mainly relied on conventional statistical techniques, including linear regression [1]-[3], exponential smoothing [4]-[6], autoregressive models [7]-[9], autoregressive integrated moving average models [10]-[12], and their seasonal extensions [13]-[15]. These methods remain attractive because of their simplicity, moderate computational requirements, and interpretability. They can provide reliable predictions when electricity demand follows a relatively stable, approximately linear relationship. However, their dependence on predefined assumptions may limit their ability to represent nonlinear demand behavior, interacting seasonal patterns, and rapidly changing operating conditions.

Machine-learning methods, such as support vector regression [16]-[20], decision trees [21]-[25], random forests [26]-[28], gradient boosting [29]-[34], XGBoost [35]-[38], and LightGBM [39]-[42], have been widely applied to improve nonlinear forecasting capability. These methods can capture complex relationships among historical load observations, lagged values, rolling statistics, calendar variables, and other engineered features. Nevertheless, their performance remains strongly dependent on feature design and predictor selection. For high-resolution multi-step forecasting, numerous lagged and seasonal predictors may be required, while horizon-specific prediction mechanisms can further increase model complexity.

Deep-learning methods provide an alternative by automatically learning nonlinear representations from sequential data. Recurrent neural networks [43]-[47], long short-term memory networks [48]-[52], gated recurrent units [53]-[56], convolutional neural networks [57]-[58], and hybrid architectures have all been applied to STLF. LSTM models can retain temporal information over extended sequences but are constrained by sequential computation and may struggle with long-horizon forecasting. CNNs efficiently extract local temporal patterns, although their ability to represent distant dependencies depends strongly on network depth, kernel size, and receptive-field design. More generally, an individual neural model must simultaneously learn recent fluctuations, recurring daily and weekly patterns, and changes in the overall demand level within a single representation.

Transformer-based models [59]-[64] have attracted increasing attention because attention mechanisms provide an effective alternative for long-sequence forecasting. Several Transformer variants have been developed to improve computational efficiency, long-range dependency learning, periodic representation, and multivariate forecasting. Among them, iTransformer is particularly relevant because it applies attention across inverted variate tokens and has demonstrated strong potential for time-series forecasting. However, a standalone iTransformer [65]-[68] remains dependent on its input representation and selected model configuration. Although it can learn complex nonlinear relationships, it does not naturally preserve deterministic temporal references embedded in electricity demand, such as the time of the previous day, the previous week, or recurring historical load profiles.

These information sources are complementary. Recent observations describe the current demand level; one-day lags represent immediate daily recurrence; longer lags capture weekly and multi-week periodicity; and historical profiles provide stable representations of recurring load behavior. The usefulness of these sources may change according to operating conditions. A one-day lag may be highly informative during a stable weekday, whereas weekly patterns may become more useful when the most recent day differs from the expected condition. Similarly, iTransformer can capture nonlinear demand dynamics that simple seasonal references cannot represent, whereas historical profiles may provide useful structural information when neural predictions deviate from a recurring demand trajectory.

Multi-scale and ensemble forecasting provide a natural mechanism for integrating these complementary information sources. Existing methods commonly combine forecasting branches through averaging, fixed weighting, concatenation, stacking, or learned fusion. However, equal or globally optimized weights assume that the importance of each source remains constant across time. Such strategies may therefore ignore differences among low-, normal-, and high-load conditions or among different positions within a day-ahead forecasting horizon. In addition, retaining many highly correlated candidates may unnecessarily increase ensemble complexity and encourage overfitting. Effective integration of iTransformer forecasts and explicit multi-scale temporal references, therefore, requires both careful candidate screening and adaptive fusion.

Fair comparison is another important methodological consideration. Forecasting performance is influenced by input length, forecasting horizon, rolling interval, evaluation period, target series, and the information supplied to each model. Changing these settings only for the proposed method would make it difficult to determine whether an improvement results from the forecasting framework or from unequal experimental conditions. Therefore, this study maintains the same principal forecasting configuration across all comparative models. Each model uses 288 historical observations to predict the following 288 observations, with a rolling step of 288 for a single regional load series. Both the input and output, therefore, correspond to complete 24-h periods.

The present study focuses exclusively on deterministic, load-only forecasting. No weather variables, electricity prices, holiday indicators, renewable-generation measurements, or other external variables are included in the model inputs. Temporal attributes such as month, day of week, hour, and minute are used only to organize historical observations and construct recurring load profiles. Therefore, this study does not claim probabilistic forecasting capability or performance improvements obtained from exogenous information. Instead, it examines the forecasting value of historical load dynamics, explicit temporal recurrence, and validation-optimized ensemble fusion.

Motivated by these considerations, this study proposes a Dynamic Multi-Scale Ensemble iTransformer (DME-iTransformer) framework for high-resolution day-ahead STLF. The framework combines learned iTransformer predictions with deterministic temporal candidates derived exclusively from historical electricity demand. These candidates represent recent daily behavior, weekly and multi-week recurrence, static and dynamically adjusted load profiles, and complementary lag–profile combinations. Rather than treating neural and seasonal forecasting methods as competing alternatives, the proposed framework integrates them into a unified forecasting system.

The proposed framework consists of four main modules. First, complementary lag- and profile-based candidates are constructed from historical observations at different temporal offsets. Static profiles describe recurring demand patterns, whereas dynamically adjusted profiles reflect recent changes in the overall load level. Additional blended candidates combine short- and long-range temporal information, thereby explicitly representing multiple recurring structures rather than requiring a single neural model to implicitly recover all periodic relationships.

Second, iTransformer is employed as the learned nonlinear forecasting component. To ensure a fair comparison with the standalone iTransformer, the principal forecasting configuration remains fixed at a 288-step input, a 288-step forecast horizon, a 288-step rolling interval, and a single-series setting. Model selection is restricted to internal hyperparameters. The selected iTransformer prediction is then combined with lag- and profile-based references to form hybrid candidates that incorporate both learned nonlinear dynamics and explicit temporal recurrence.

Third, candidate ranking and screening are performed using an independent validation period. Candidate forecasts are initially ranked according to validation MAPE. Correlation-based pruning is then applied to remove forecasts that provide highly redundant information. This procedure aims to retain a compact candidate set containing both individually accurate and mutually complementary predictions. The tuned iTransformer is explicitly retained to ensure that the learned forecasting component remains represented in the final ensemble.

Fourth, the retained candidates are combined through validation-optimized ensemble learning. Three weighting levels are considered: global weighting, load-state-dependent weighting, and state–horizon-dependent weighting. This hierarchical structure allows the contributions of iTransformer forecasts, recent lags, weekly patterns, and historical profiles to vary with both the predicted demand regime and the position within the 24-h forecasting horizon. Hierarchical fallback rules are used when a local group contains insufficient validation observations.

A separate validation-based strategy-selection procedure, referred to as SAFE-SELECT, is used to avoid unnecessary ensemble complexity. Rather than automatically adopting the most complex dynamic weighting strategy, the framework compares individual and ensemble forecasting alternatives on an independent validation period. The strategy with the lowest validation MAPE is then applied to the untouched test period. Test labels are not used for iTransformer tuning, candidate evaluation, ensemble-weight estimation, strategy selection, or residual-bias calibration. This chronological separation ensures that the reported results reflect out-of-sample forecasting performance.

The research contribution is a validation-driven Dynamic Multi-Scale Ensemble iTransformer framework that integrates learned nonlinear forecasting with explicit daily, weekly, multi-week, and load-profile references, removes highly redundant candidates, and adaptively combines the retained predictions across load states and forecast-horizon segments.

The principal contributions of this study are summarized as follows:

- A multi-scale candidate-generation mechanism is developed to represent recent daily behavior, weekly and multi-week recurrence, static historical profiles, dynamically adjusted profiles, and complementary lag–profile combinations.
- A tuned iTransformer is incorporated under a fixed 288-step input, 288-step forecast horizon, 288-step rolling interval, and single-series configuration, ensuring that performance improvements are not caused by unequal forecasting settings.
- A diverse hybrid candidate set is constructed from iTransformer, lag-based, profile-based, and combined forecasts. Validation-based ranking and correlation pruning are then employed to reduce unnecessary redundancy while preserving complementary forecasting information.
- A hierarchical dynamic fusion mechanism is introduced to compare global, load-state-dependent, and state–horizon-dependent weighting, allowing candidate contributions to adapt to operating conditions and forecast lead times.

Hyperparameter tuning, candidate evaluation, ensemble-weight optimization, strategy selection, residual-bias calibration, and final testing are chronologically separated to prevent test-label leakage.

The proposed framework is evaluated against naive, deep-learning, and Transformer-based baseline models under an identical high-resolution day-ahead forecasting protocol.

The proposed framework is evaluated using New South Wales electricity-demand data recorded at 5-min intervals. The previous 288 observations are used to predict the subsequent 288 observations, corresponding to one complete day of historical information and one complete day of future demand. DME-iTransformer is compared with five representative baselines: Persistence Naive, Seasonal Naive, CNN, LSTM, and standalone iTransformer.

The experimental results show that DME-iTransformer achieves the best overall forecasting accuracy among the six evaluated methods. It obtains a MAPE of 7.031%, compared with 7.979% for Seasonal Naive and 8.691% for standalone iTransformer. CNN obtains a MAPE of 12.419%, whereas Persistence Naive and LSTM produce MAPEs of 17.596% and 19.830%, respectively. Relative to Seasonal Naive and standalone iTransformer, the proposed method reduces MAPE by 11.88% and 19.10%, respectively. These results indicate that explicit multi-scale temporal references complement the nonlinear representation learned by iTransformer and that validation-optimized adaptive fusion improves deterministic day-ahead load forecasting accuracy.

The remainder of this paper is organized as follows. Section 2 reviews statistical, machine-learning, deep-learning, Transformer-based, multi-scale, and ensemble forecasting approaches. Section 3 presents the proposed DME-iTransformer framework. Section 4 describes the dataset, experimental configuration, comparative models, implementation settings, and evaluation criteria. Section 5 presents and discusses the forecasting results, performance across forecast horizons and load regimes, internal component-wise behavior, adaptive fusion characteristics, and the limitations of the present experimental evaluation. Section 6 summarizes the principal findings, contributions, limitations, and future research directions.

2. RELATED WORK

Short-term load forecasting has been investigated using statistical, machine-learning, deep-learning, Transformer-based, multi-scale, and hybrid forecasting approaches. These method families differ in their assumptions, representation capabilities, computational requirements, and dependence on manually designed features.

Traditional methods, including regression, exponential smoothing, autoregressive models, and ARIMA-based techniques, remain attractive because of their simplicity, relatively low computational requirements, and interpretability. However, their reliance on predefined statistical assumptions may limit their ability to represent nonlinear and non-stationary electricity-demand behavior [69][70].

Machine-learning methods, such as support vector regression, random forests, gradient boosting, XGBoost, and LightGBM, provide greater flexibility by learning nonlinear relationships from lagged demand, rolling statistics, calendar information, and other engineered temporal features. Although these methods can provide accurate forecasts, their performance remains dependent on feature selection and feature engineering.

High-resolution multi-step forecasting may require a large number of lagged and horizon-dependent predictors, which can increase model complexity and introduce redundant information [71]-[73].

Deep-learning models reduce part of this dependence by learning temporal representations directly from sequential observations. LSTM, GRU, CNN, and related hybrid architectures have therefore been widely applied to load forecasting. Recurrent models can represent long-term temporal dependencies but rely on sequential computation, whereas CNN-based approaches provide efficient local feature extraction but require appropriately designed receptive fields to capture long-range relationships [74]-[76]. In addition, a single deep-learning model must recover short-term variations, daily recurrence, weekly periodicity, and changing load levels from one learned representation.

Transformer-based forecasting models have attracted considerable attention because of their ability to represent long-range dependencies and support parallel computation. Recent studies have explored different Transformer-based architectures for electric-load and general time-series forecasting [77]-[79]. Among these approaches, iTransformer is particularly relevant to the present study because it provides the learned forecasting backbone used in the proposed framework. However, a standalone iTransformer does not explicitly expose previous-day, previous-week, multi-week, and historical-profile forecasts as separate information sources. These temporal references must instead be inferred implicitly from the available input sequence.

Multi-scale forecasting provides a complementary approach by exploiting historical information at different temporal offsets. Daily lags can describe immediate recurrence, weekly lags can represent repeated weekday and weekend behavior, and multi-week lags can provide more stable seasonal references. Historical load profiles conditioned on temporal positions can additionally represent typical intraday demand patterns. However, no single historical reference is consistently optimal under all operating conditions. Recent lags may be more informative during short-term changes, whereas weekly patterns or historical profiles may become more useful during periods with stable recurring behavior.

Ensemble learning can combine these complementary forecasting sources through averaging, weighted fusion, stacking, or other combination strategies. However, many existing ensemble methods use equal weights or one globally optimized weight vector. These approaches assume that the importance of each candidate remains constant across load conditions and forecast lead times. Such an assumption may be restrictive for day-ahead forecasting because candidate performance can differ between low-load valleys, normal operating periods, high-load peaks, and different sections of the 288-step horizon.

Adaptive ensemble methods address this limitation by allowing candidate weights to depend on recent performance, demand conditions, temporal context, or forecast lead time. Nevertheless, greater adaptation introduces additional parameters and may result in unstable weight estimates when local validation samples are insufficient. Moreover, combining many similar forecasts does not necessarily improve accuracy because strongly correlated candidates may contain largely redundant information. Therefore, an effective adaptive ensemble should include candidate screening, local-sample safeguards, and validation-based control of ensemble complexity.

Existing studies have demonstrated the individual value of Transformer forecasting, multi-scale temporal representation, and ensemble learning. However, these components are often examined separately. Transformer-based methods commonly rely on a single learned representation, whereas multi-scale ensemble approaches may not fully exploit a strong Transformer forecasting component. Furthermore, fixed-weight ensembles do not adapt candidate contributions to changes in demand regime or forecast position, while highly adaptive methods may be vulnerable to overfitting when their complexity is not independently validated.

The present study addresses this gap by proposing a Dynamic Multi-Scale Ensemble iTransformer framework. A tuned iTransformer is combined with multi-scale lag and load-profile candidates representing recent daily behavior, weekly and multi-week recurrence, and recurring historical demand structures. Hybrid candidates are generated by combining the iTransformer prediction with selected temporal references. Validation-based ranking and correlation pruning are subsequently used to reduce redundancy before ensemble optimization.

The retained forecasts are combined using global, load-state-dependent, and state–horizon-dependent weighting. Hierarchical fallback rules are applied when insufficient validation observations are available for local weight estimation. In addition, SAFE-SELECT compares alternative individual and ensemble strategies on an independent validation period instead of automatically adopting the most complex forecasting configuration.

The proposed framework is also distinguished by its chronological experimental design. Model training, iTransformer configuration selection, candidate evaluation, ensemble-weight estimation, strategy selection, residual-bias calibration, and final testing are assigned to separate time periods. Consequently, test labels are not used for any model-development decision.

Accordingly, the research gap addressed in this study is the absence of a unified load-only forecasting framework that simultaneously integrates: an iTransformer prediction; explicit daily, weekly, multi-week, and historical-profile references; validation-based candidate ranking; correlation-based redundancy removal; load-state- and horizon-dependent fusion; and independent validation-based selection of ensemble complexity.

The main research question is whether explicit multi-scale temporal references can complement a standalone iTransformer and whether validation-optimized adaptive fusion can improve forecasting accuracy beyond individual deep-learning, Transformer-based, and seasonal forecasting methods under an identical high-resolution day-ahead forecasting protocol. These considerations motivate the DME-iTransformer methodology presented in the following section.

3. PROPOSED METHOD

The proposed DME-iTransformer framework integrates explicit multi-scale temporal information with a tuned iTransformer and validation-optimized dynamic ensemble learning. The overall methodology consists of four main stages: multi-scale lag and load-profile construction, iTransformer forecasting, hybrid candidate generation and selection, and adaptive ensemble optimization. Historical demand patterns are first extracted from daily, weekly, and multi-week lags together with static and dynamic load profiles. These temporal candidates are then combined with the tuned iTransformer forecast to form a diverse forecasting pool. Finally, validation-based candidate screening and dynamic weighting are applied to determine the most suitable forecasting strategy without using test information during model development. The following subsections describe each component of the proposed framework in detail.

3.1. Multi-Scale Lag and Profile Construction

Electricity demand contains recurring patterns at multiple temporal scales. A single historical window may not explicitly preserve daily, weekly, and longer-term seasonal relationships. Therefore, the first stage of the proposed framework constructs a diverse set of deterministic forecasting candidates from historical lags and load profiles.

For a forecast time t, the d-day lag candidate is defined as

$$L_d(t) = y_{t-dD}, d \in \{1,2,3,7,14,21,28\} \tag{1}$$

where $D = 288$ is the number of 5-min observations per day. These lags represent recent daily behavior, weekly recurrence, and multi-week periodicity.

In addition to direct lags, three historical load profiles are constructed using the median demand at comparable temporal positions:

$$\begin{gathered} P_{\text{MHM}}(t) = \text{Med}\{y_\tau : m(\tau) = m(t), h(\tau) = h(t), u(\tau) = u(t)\}, \\ P_{\text{DHM}}(t) = \text{Med}\{y_\tau : d_w(\tau) = d_w(t), h(\tau) = h(t), u(\tau) = u(t)\}, \\ P_{\text{HM}}(t) = \text{Med}\,\{y_\tau : h(\tau) = h(t), u(\tau) = u(t)\} \end{gathered} \tag{2}$$

where $m(t)$, $d_w(t)$, $h(t)$, and $u(t)$ denote the month, day of week, hour, and minute associated with time t, respectively. Medians are used instead of means to reduce sensitivity to unusually high or low demand observations.

The three profiles are combined into a static temporal reference:

$$P_{\text{static}}(t) = 0.25P_{\text{MHM}}(t) + 0.50P_{\text{DHM}}(t) + 0.25P_{\text{HM}}(t) \tag{3}$$

The larger weight assigned to the day-of-week profile reflects the strong weekly recurrence commonly observed in regional electricity demand.

Because historical profiles may not immediately reflect recent changes in the overall demand level, a normalized load ratio is computed as

$$q_\tau = \frac{y_\tau}{P_{\text{static}}(\tau)} \tag{4}$$

For each forecast day, recent level information is summarized over three retrospective periods. The resulting adjustment factor is

$$r_t = 0.60\text{ Mean }(q_\tau;\ \tau \in \mathcal{H}_1) + 0.30\text{ Mean }(q_\tau;\ \tau \in \mathcal{H}_7) + 0.10\text{ Med }(q_\tau;\ \tau \in \mathcal{H}_{28}) \tag{5}$$

where $\mathcal{H}_1$, $\mathcal{H}_7$, and $\mathcal{H}_{28}$ denote the most recent 1-, 7-, and 28-day historical intervals available before the forecast origin.

The dynamic profile is then obtained by adjusting the static profile according to the recent demand level:

$$P_{\text{dynamic}}(t) = P_{\text{static}}(t)\, r_t \tag{6}$$

For ratio-adjusted lag candidates, the correction factor is conservatively bounded as

$$\tilde{r}_t = \min\,[1.10, \max\,(0.90, r_t)] \tag{7}$$

which prevents short-term level changes from excessively amplifying or suppressing historical lag forecasts.

Additional multi-scale candidates are constructed from these basic temporal sources. A representative four-week forecast is defined as

$$W_4(t) = 0.50L_7(t) + 0.25L_{14}(t) + 0.15L_{21}(t) + 0.10L_{28}(t) \tag{8}$$

The complete candidate pool additionally contains daily–weekly blends, recent same-time combinations, ratio-adjusted lags, a robust median of $L_1(t)$, $L_7(t)$, and $P_{\text{dynamic}}(t)$, and a recent–weekly–dynamic combination. These candidates expose complementary temporal structures before the neural forecast is introduced.

3.2. iTransformer Forecasting Model

The second component of the proposed framework is an iTransformer forecasting model. To ensure a fair comparison with the standalone iTransformer baseline, the principal forecasting configuration is fixed for all candidate configurations. Specifically,

$$L = 288, H = 288, S = 288, N_s = 1 \tag{9}$$

where L is the input length, $H$is the forecasting horizon, $S$is the rolling step size, and N_s is the number of target series. Thus, each model uses the previous 24 h of 5-min demand observations to forecast the complete following 24-h trajectory.

At each forecast origin t, the historical input and future target are defined as

$$\mathbf{X}_t = [y_{t-L+1}, \dots, y_t], \mathbf{Y}_t = [y_{t+1}, \dots, y_{t+H}] \tag{10}$$

The iTransformer generates the multi-step forecast

$$\hat{\mathbf{Y}}_t^{\text{ITR}} = f_{\boldsymbol{\theta}}(\mathbf{X}_t) \tag{11}$$

where $f_{\boldsymbol{\theta}}(\cdot)$ denotes the iTransformer model parameterized by $\boldsymbol{\theta}$.

Within the attention mechanism, query, key, and value representations are combined through scaled dot-product attention:

$$\text{Attention}\,(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{12}$$

where d_k denotes the key dimension.

The code evaluates configurations trained using either MAPE or MAE. The two candidate training objectives are

$$\mathcal{L}_{\text{MAPE}} = \frac{100}{N}\sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{13}$$

and

$$\mathcal{L}_{\text{MAE}} = \frac{1}{N}\sum_{i=1}^{N} | y_i - \hat{y}_i | \tag{14}$$

Although both losses are considered during training, model selection is performed exclusively according to validation MAPE. Let

$$\Theta = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_C\} \tag{15}$$

denote the predefined hyperparameter configurations. The selected iTransformer is

$$\boldsymbol{\theta}^{*}=\arg\min_{\boldsymbol{\theta}_c\in\Theta}\text{MAPE}_{\text{val}}(\boldsymbol{\theta}_c) \tag{16}$$

The search varies hidden size, attention heads, encoder depth, feed-forward dimension, dropout, learning rate, maximum training steps, batch size, scaler type, and training loss, while the input size and forecasting configuration remain unchanged. The final tuned iTransformer forecast is denoted by

$$I(t)=\hat{y}_t^{\text{ITR}}(\boldsymbol{\theta}^{*}) \tag{17}$$

The rolling forecast origins are separated by one complete day,

$$t_j=t_0+jS, S=288 \tag{18}$$

and the trained model is periodically refitted during the test stage to reduce deterioration caused by temporal distribution changes.

3.3. Hybrid Candidate Generation and Selection

The proposed framework does not treat iTransformer and seasonal forecasting as competing alternatives. Instead, the neural forecast is combined with multi-scale temporal candidates to construct hybrid forecasts.

The first hybrid candidate combines the tuned iTransformer forecast with the dynamic historical profile:

$$C_{\text{IP}}(t)=0.80I(t)+0.20P_{\text{dynamic}}(t) \tag{19}$$

A recent-day hybrid candidate is defined as

$$C_{\text{I1}}(t)=0.80I(t)+0.20L_1(t) \tag{20}$$

To incorporate longer recurring behavior, the weekly hybrid forecast is

$$C_{\text{IW}}(t)=0.80I(t)+0.20W_4(t) \tag{21}$$

The recent same-time hybrid candidate is

$$C_{\text{IR}}(t)=0.80I(t)+0.20R_{\text{same}}(t) \tag{23}$$

where

$$R_{\text{same}}(t)=0.45L_1(t)+0.20L_2(t)+0.15L_3(t)+0.20L_7(t) \tag{24}$$

The complete candidate set therefore contains the original multi-scale temporal candidates, the tuned iTransformer forecast, and the four hybrid predictions. Because including all forecasts may introduce unnecessary redundancy, candidates are first ranked according to their validation MAPE:

$$E_j=\frac{100}{N_v}\sum_{i=1}^{N_v}\left|\frac{y_i-\hat{y}_i^{(j)}}{y_i}\right| \tag{25}$$

where E_j denotes the validation MAPE of candidate j.

To reduce redundancy, the linear correlation between two candidate predictions is calculated as

$$\rho_{jk}=\frac{\sum_{i=1}^{N_v}(\hat{y}_i^{(j)}-\bar{y}^{(j)})(\hat{y}_i^{(k)}-\bar{y}^{(k)})}{\sqrt{\sum_{i=1}^{N_v}(\hat{y}_i^{(j)}-\bar{y}^{(j)})^2}\sqrt{\sum_{i=1}^{N_v}(\hat{y}_i^{(k)}-\bar{y}^{(k)})^2}} \tag{26}$$

A candidate is removed when its absolute correlation with an already selected forecast satisfies

$$|\rho_{jk}|\geq\rho_{\max}, \rho_{\max}=0.999 \tag{27}$$

The correlation threshold was fixed at 0.999 as a conservative redundancy criterion. This high value removes only candidates whose validation predictions are nearly identical while retaining forecasts that may exhibit moderately different residual patterns. The threshold was specified before final test evaluation and was not optimized using test observations. Its sensitivity is not examined in the present study and remains a subject for future investigation.

The final selected set is therefore

$$\mathcal{C}^* = \text{TopK}\left\{C_j : E_j \text{ is small}, \ |\rho_{jk}| < 0.999\right\} \tag{28}$$

where at most $K = 10$ candidates are retained, with at least five candidates whenever sufficient valid forecasts are available. The tuned iTransformer forecast is explicitly preserved to ensure that the learned neural component remains represented in the final ensemble.

This two-stage procedure aims to preserve both **accuracy** and **diversity**. Candidate ranking eliminates weak forecasts, whereas correlation pruning reduces the influence of strongly overlapping predictions before weight optimization.

3.4. Validation-Optimized Dynamic Ensemble

After candidate selection, the remaining forecasts are combined through a hierarchical validation-optimized ensemble. The method considers global, load-state-dependent, and state–horizon-dependent weighting rather than assuming that one weight vector is optimal under all forecasting conditions.

For each selected candidate j, an inverse-MAPE prior weight is first calculated as

$$\pi_j = \frac{\left(E_j + \varepsilon\right)^{-1}}{\sum_{k=1}^{K} (E_k + \varepsilon)^{-1}} \tag{29}$$

where E_j is the validation MAPE and ε s a small numerical constant.

Given the candidate matrix **X** and weight vector **w**, ensemble weights are estimated by solving

$$\begin{gathered} \mathrm{w}^* = \arg\min_{\mathrm{w}} \left[\ \text{MAPE}\ (\mathrm{y}, \mathrm{Xw}) + \lambda \parallel \mathrm{w} - \pi \parallel_2^2\right] \\ \text{subject to} \qquad w_j \geq 0, \sum_{j=1}^{K} w_j = 1. \end{gathered} \tag{30}$$

The first term minimizes forecasting error, while the second term discourages extreme deviations from the inverse-MAPE prior. The regularization coefficients are set to 0.005, 0.02, and 0.05 for global, state-specific, and state–horizon-specific optimization, respectively. The constrained problem is solved using SLSQP, with random Dirichlet search used as an additional fallback mechanism. These coefficients were fixed before final test evaluation and were progressively increased from global to state–horizon optimization to impose stronger shrinkage as the number of observations available for local weight estimation decreased. They should therefore be interpreted as design parameters rather than values established through an exhaustive sensitivity analysis. A systematic evaluation of their influence is outside the scope of the present study and is identified as future work.

The predicted load state is determined from the dynamic profile using historical 15th and 85th percentiles:

$$s_t = \begin{cases} \text{Low}, & P_{\text{dynamic}}(t) \leq Q_{0.15}, \\ \text{High}, & P_{\text{dynamic}}(t) \geq Q_{0.85}, \\ \text{Normal}, & \text{otherwise}. \end{cases} \tag{31}$$

The 288-step prediction horizon is divided into four 6-h blocks:

$$b(h) = \begin{cases} B_1, & 1 \leq h \leq 72, \\ B_2, & 73 \leq h \leq 144, \\ B_3, & 145 \leq h \leq 216, \\ B_4, & 217 \leq h \leq 288. \end{cases} \tag{32}$$

The resulting dynamic ensemble forecast is

$$\hat{y}_{t,h}^{\mathrm{DME}} = \begin{cases} \sum_{j=1}^{K} w_j^{(s_t,b(h))}\, \hat{y}_{t,h}^{(j)}, & N_{s_t,b(h)} \geq N_{\min}, \\ \sum_{j=1}^{K} w_j^{(s_t)}\, \hat{y}_{t,h}^{(j)}, & N_{s_t} \geq N_{\min}, \\ \sum_{j=1}^{K} w_j^{(G)}\, \hat{y}_{t,h}^{(j)}, & \text{otherwise}, \end{cases} \tag{33}$$

where $N_{\min} = 250$. Thus, the framework follows the fallback hierarchy

State–Horizon → State → Global.

Rather than automatically adopting the most complex ensemble, six forecasting strategies are evaluated on a separate validation period. The final strategy is selected as

$$m^* = \arg\min_{m \in \mathcal{M}} \text{MAPE}_{\text{select}}\left(\hat{\mathbf{Y}}^{(m)}\right) \tag{34}$$

Where $\mathcal{M} = \{\text{Best Single,Equal,Inverse-MAPE,Global,State,State–Horizon}\}$.

This SAFE-SELECT mechanism allows the framework to revert to a simpler strategy when a more complex dynamic ensemble does not generalize better.

Finally, a shrunk residual bias is estimated for each state–horizon group:

$$\tilde{b}_{s,b} = \frac{n_{s,b}}{n_{s,b} + \kappa}\bar{e}_{s,b} + \frac{\kappa}{n_{s,b} + \kappa}\bar{e}_G \tag{35}$$

where $\bar{e}_{s,b}$ is the local mean residual, $\bar{e}_G$ is the global mean residual, and $\kappa = 30$ controls shrinkage toward the global bias. The final forecast is

$$\hat{y}_{t,h}^{\text{final}} = \text{Clip}\left[\hat{y}_{t,h}^{(m^*)} + \alpha^* \tilde{b}_{s_t,b(h)}\right] \tag{36}$$

Where $\alpha^* \in \{0,0.15,0.30,0.50\}$
is selected using a separate validation subset. If bias correction does not reduce validation MAPE, the procedure selects $\alpha^* = 0$.

From a computational perspective, let N denote the number of aligned validation observations and K the number of retained candidates. Candidate stacking and weighted prediction scale linearly with NK, whereas pairwise correlation screening requires $O(NK^2)$ operations. Because the candidate set is restricted to $K \leq 10$, the candidate-dimension cost remains bounded. The actual computational demand of the optimization stage additionally depends on the number of SLSQP iterations, local state–horizon groups, and fallback searches. Standardized wall-clock time and inference latency were not recorded and are therefore not reported as performance outcomes in this study.

The complete methodology follows a chronological validation protocol. Data from 2022–2024 are used for model training; January–April 2025 are used for iTransformer configuration selection; May 2025 is used for candidate evaluation and ensemble-weight estimation; June 2025 is used for SAFE-SELECT and bias-strength calibration; and observations from July 2025 onward form the final test period. Consequently, test labels are not used for hyperparameter tuning, candidate selection, weight estimation, or final strategy determination.

3.5. Overall Algorithm and Flowchart

The overall procedure of the proposed DME-iTransformer framework is illustrated in Figure 1. The framework follows a sequential and validation-driven forecasting process that integrates data preprocessing, chronological data partitioning, multi-scale temporal candidate construction, iTransformer configuration selection, hybrid candidate generation, candidate screening, dynamic ensemble optimization, strategy selection, and optional residual-bias correction. The principal characteristic of the framework is the strict chronological separation of model-development stages. Consequently, iTransformer tuning, candidate evaluation, ensemble-weight estimation, strategy selection, and bias calibration are completed without using observations from the final test period.

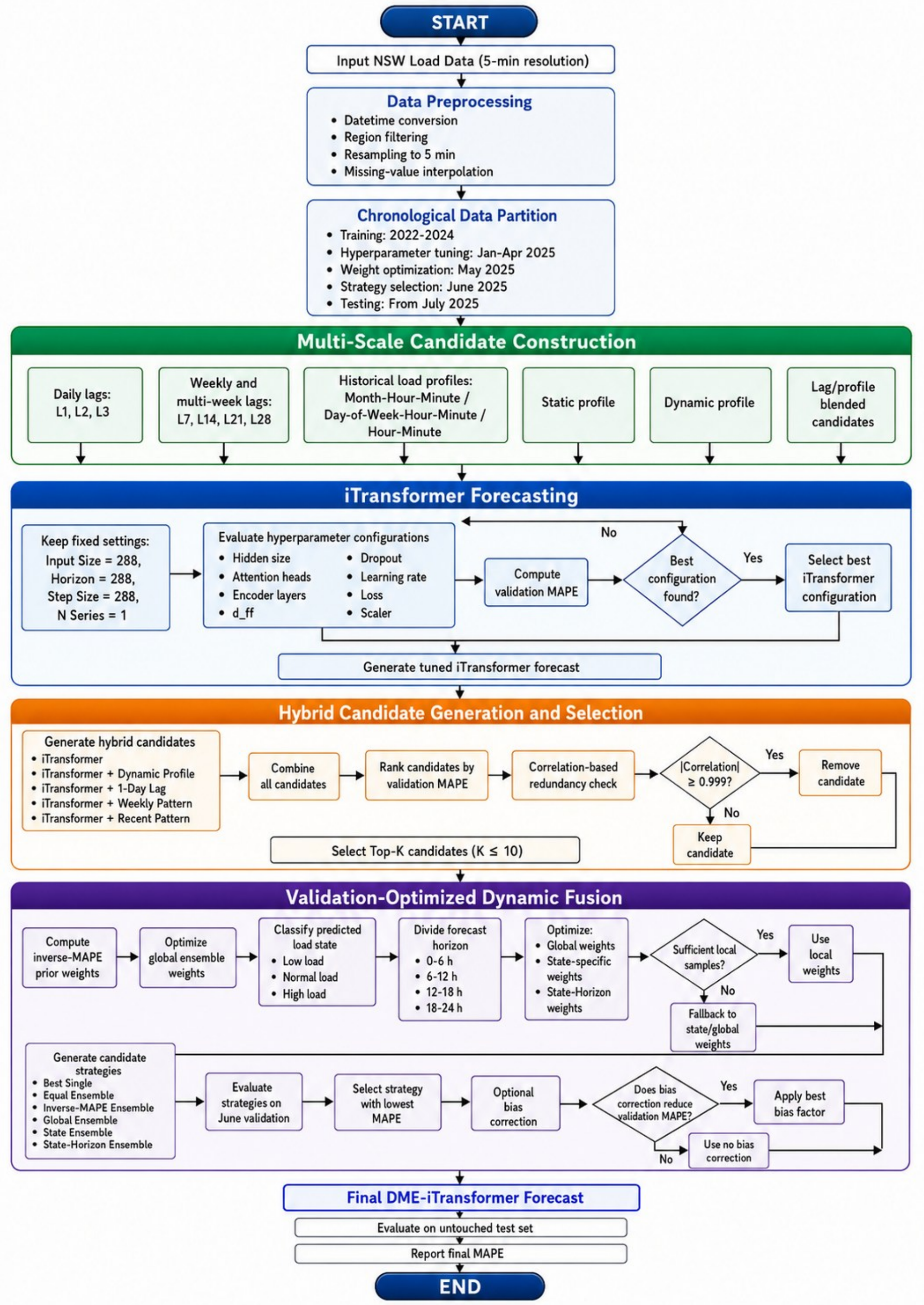


Figure 1. Overall workflow of the proposed Dynamic Multi-Scale Ensemble iTransformer framework for day-ahead electric load forecasting

The complete forecasting procedure consists of four main modules: multi-scale candidate construction, iTransformer forecasting, hybrid candidate generation and selection, and validation-optimized dynamic fusion. After all model configurations and fusion parameters have been determined from the designated training and validation periods, the final forecasting strategy is evaluated once on the untouched test set.

As shown in Figure 1, the procedure begins with New South Wales electricity-demand data recorded at a 5-min resolution. The raw observations undergo datetime conversion, regional filtering, resampling to a regular 5-min time grid, and missing-value interpolation. These preprocessing operations produce a chronologically ordered and temporally consistent load series suitable for high-resolution day-ahead forecasting.

The processed dataset is then divided into five non-overlapping chronological periods. Observations from 2022 to 2024 are used for model training. Data from January to April 2025 are used to evaluate the predefined iTransformer configurations and select the configuration with the lowest validation MAPE. May 2025 is used for forecasting-candidate evaluation, redundancy screening, and ensemble-weight optimization. June 2025 is reserved for final forecasting-strategy selection and residual-bias-strength calibration. Observations from July 2025 onward form the final test set. This organization ensures that every model-development decision is based exclusively on information available before the test period.

The first module constructs a diverse set of deterministic temporal candidates from the historical load series. Daily lags of one, two, and three days represent recent short-term recurrence. Weekly and multi-week lags of seven, fourteen, twenty-one, and twenty-eight days describe recurring temporal behavior over longer periods. Historical load profiles are additionally calculated according to month–hour–minute, day-of-week–hour–minute, and hour–minute groupings. These individual profiles are combined to form a static profile, which is subsequently adjusted using recent load-level information to obtain a dynamic profile. Additional lag–profile combinations are constructed to represent complementary short-term, weekly, multi-week, and profile-based demand patterns.

The second module performs iTransformer forecasting. To maintain a fair experimental setting, the principal forecasting dimensions remain unchanged across all evaluated iTransformer configurations. The input size is fixed at 288 observations, the forecast horizon is 288 observations, the rolling step is 288 observations, and the number of target series is one. Therefore, every configuration uses the preceding 24 h of load observations to forecast the complete load trajectory for the following 24 h.

Within this fixed forecasting setting, several internal iTransformer hyperparameters are evaluated, including hidden size, number of attention heads, encoder depth, feed-forward dimension, dropout rate, learning rate, training loss, and scaling method. This procedure is a predefined validation-based configuration search rather than Bayesian optimization. Each configuration is trained and assessed using validation MAPE, and the configuration producing the lowest validation error is selected. The resulting model is then used to generate the tuned iTransformer forecast for the subsequent candidate-generation stage.

The third module combines the tuned iTransformer forecast with selected deterministic temporal references. Hybrid candidates are generated by blending the iTransformer output with the dynamic profile, one-day lag, weekly pattern, and recent same-time behavior. These hybrid forecasts are merged with the original lag-based, profile-based, and blended temporal candidates to form a unified forecasting pool.

Because the complete candidate pool may contain several forecasts with nearly identical behavior, the candidates are first ranked according to validation MAPE. A correlation-based redundancy analysis is then applied. When the absolute correlation between a candidate and a previously retained forecast is equal to or greater than 0.999, the less accurate candidate is treated as redundant and removed. Otherwise, it is retained. The threshold of 0.999 is used as a conservative criterion that removes only almost-identical forecasts while preserving candidates that may provide complementary error patterns. The screening procedure retains at most ten candidates, while the tuned iTransformer forecast is explicitly preserved in the final candidate set.

The fourth module implements validation-optimized dynamic fusion. The process begins by calculating inverse-MAPE prior weights, which assign greater initial importance to candidates with stronger validation performance. These prior weights are subsequently refined through constrained global ensemble optimization.

Two additional levels of adaptation are then introduced. First, each predicted load condition is classified as low, normal, or high according to thresholds estimated exclusively from historical data. Second, the 288-step day-ahead forecast is divided into four 6-h blocks corresponding to 0–6 h, 6–12 h, 12–18 h, and 18–24 h. Based on these two contextual dimensions, the framework estimates three groups of ensemble weights: global weights, load-state-specific weights, and state–horizon-specific weights.

A hierarchical fallback mechanism is used to reduce the risk of unstable local optimization. When a load-state or state–horizon group contains a sufficient number of validation observations, its corresponding local weight vector is applied. When the available local sample is insufficient, the procedure reverts to a broader state-specific or global weight vector. This design enables context-dependent adaptation where sufficient validation evidence is available while preventing unreliable weight estimation for sparsely represented operating conditions.

After weight optimization, six candidate forecasting strategies are generated: the best individual candidate, equal-weight ensemble, inverse-MAPE ensemble, globally optimized ensemble, load-state-dependent ensemble, and state–horizon-dependent ensemble. These alternatives are evaluated on the independent June 2025 validation period. The strategy with the lowest validation MAPE is selected through the SAFE-SELECT mechanism.

SAFE-SELECT does not assume that the most complex ensemble structure must always provide the best forecasting performance. A simpler individual or global strategy is retained whenever it produces a lower

Vol. 8, No. 4, August 2026, pp. 1259-1298

validation error than the more detailed state- or state–horizon-dependent alternatives. This mechanism limits unnecessary ensemble complexity and reduces the risk that locally optimized weights overfit the weight-estimation period.

The selected strategy is finally subjected to optional residual-bias correction. Candidate bias-strength values, including zero, are evaluated on the strategy-selection validation period. When residual correction decreases validation MAPE, the corresponding bias-strength factor is applied. When no improvement is observed, zero is selected and the forecast remains uncorrected. Thus, bias correction is included only when it provides independent validation evidence of improved forecasting accuracy.

The output of this process is the final DME-iTransformer forecast. Only after the iTransformer configuration, candidate subset, ensemble weights, forecasting strategy, and bias-strength factor have been determined is the framework applied to the untouched test set. The final MAE, MSE, RMSE, MAPE, sMAPE, and R2 values are subsequently calculated from these out-of-sample forecasts.

Overall, Figure 1 shows that DME-iTransformer is not a simple weighted combination of an iTransformer and historical lags. It is a structured forecasting framework in which complementary temporal representations are constructed, evaluated, screened, adaptively weighted, and independently selected through chronologically separated validation stages. This design enables the framework to exploit short-term dependence, daily and weekly recurrence, historical load profiles, and learned nonlinear representations while reducing candidate redundancy, unstable local optimization, unnecessary post-processing, and test-data leakage.

4. EXPERIMENTAL SETUP

This section describes the dataset, preprocessing procedure, forecasting configuration, comparative models, evaluation criteria, and computational environment used to assess the proposed DME-iTransformer. All data partitions were arranged chronologically, and the test period was isolated from hyperparameter tuning, candidate selection, ensemble-weight estimation, and final strategy calibration. This design was adopted to ensure a fair and leakage-free evaluation of the proposed framework.

4.1. Dataset and Data Preparation

The experimental study was conducted using regional electricity-demand data from New South Wales, Australia (Figure 2). The target series corresponds to the NSW1 region and contains total electricity demand measured at a 5-min resolution. The experimental period covers records from January 1, 2022 onward, with observations from July 2025 used exclusively for final testing.

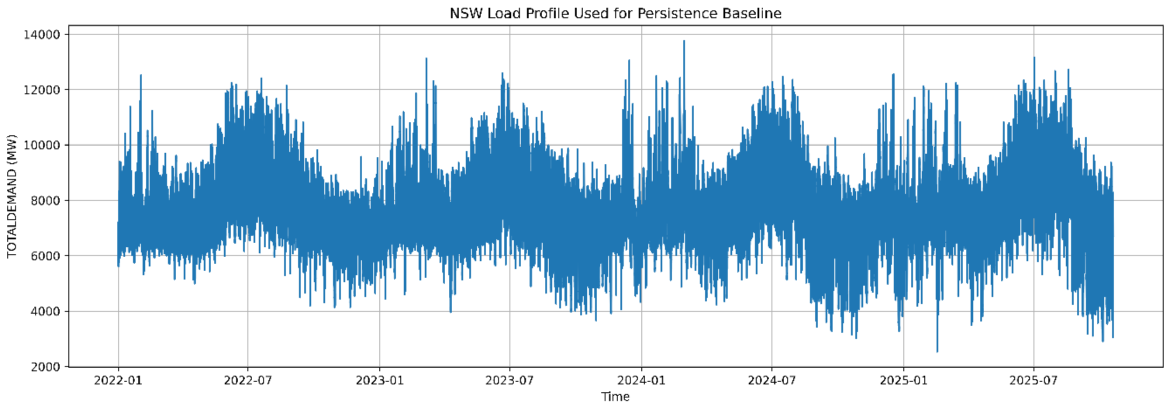


Figure 2. NSW electricity demand profile from 2022 to 2025

The raw dataset was first converted to a consistent chronological format. The timestamp field was transformed into a datetime representation, and the demand variable was converted to a numeric format. Records belonging to the NSW1 region were retained, after which the observations were sorted chronologically and duplicated timestamps were removed.

Because the proposed forecasting problem requires a continuous 5-min time grid, the demand series was resampled at fixed 5-min intervals. Missing observations generated during this procedure were estimated using time-based linear interpolation. For two observed values y_{t_a} and y_{t_b} surrounding a missing observation at time t, the interpolated value is expressed as

$$\tilde{y}_t = y_{t_a} + \frac{t - t_a}{t_b - t_a}\left(y_{t_b} - y_{t_a}\right) \tag{37}$$

where $t_a < t < t_b$. This procedure preserves the local temporal continuity of the load series without introducing future test information into the model-development stages.

After preprocessing, temporal attributes including month, day of week, hour, minute, date, and the corresponding 5-min daily slot were generated. These attributes were used only to construct historical load profiles and organize the forecasting analysis. No weather, price, holiday, or other external variables were included in the present study. Therefore, the proposed method is evaluated as a load-only forecasting framework.

The processed demand series was divided chronologically into five consecutive periods. Data from January 2022 to December 2024 were used for model training. January–April 2025 were used for iTransformer hyperparameter tuning. May 2025 was used for candidate evaluation and ensemble-weight optimization. June 2025 was reserved for final strategy selection and bias-strength calibration. The test period started on July 1, 2025.

This chronological partition can be summarized as

$$\mathcal{D} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{tune}} \cup \mathcal{D}_{\text{weight}} \cup \mathcal{D}_{\text{select}} \cup \mathcal{D}_{\text{test}} \tag{38}$$

where the five subsets are mutually exclusive and ordered in time. This separation prevents test observations from influencing any model-development decision.

4.2. Forecasting Configuration and Fair Comparison

The forecasting task was formulated as high-resolution day-ahead load forecasting. Since the data resolution is 5 min, one complete day contains 288 observations. Therefore, the previous 288 demand values were used to predict the subsequent 288 values.

At forecast origin t, the input sequence is defined as

$$\mathbf{X}_t = [y_{t-287}, y_{t-286}, \dots, y_t] \tag{39}$$

and the corresponding target trajectory is

$$\mathbf{Y}_t = [y_{t+1}, y_{t+2}, \dots, y_{t+288}] \tag{40}$$

Thus, both the historical input and the future forecast represent complete 24-h periods.

To ensure fair comparison, the principal forecasting settings were fixed throughout the experiment:

Table 1. Forecasting Configuration and Backtesting Setup

Parameter	Value	Meaning
Data resolution	5 min	One observation every 5 min
Input size	288	Previous 24 h
Forecast horizon	288	Following 24 h
Step size	288	One forecast origin per day
Number of series	1	NSW1 regional demand

Accordingly

$$L = H = S = 288, N_s = 1 \tag{41}$$

where L, H, S, and N_s denote the input length, forecasting horizon, rolling step size, and number of target series, respectively.

The forecast origins follow

$$t_j = t_0 + jS, S = 288 \tag{42}$$

meaning that consecutive forecasts are separated by one complete day. Each forecast therefore produces the full 288-step load trajectory for the following 24 h.

An internal validation window of 14 days was used during model training:

$$N_{\text{val}} = 14 \times 288 = 4032 \tag{43}$$

observations. During the final test stage, the tuned iTransformer was periodically refitted every 30 forecast windows to reduce performance degradation caused by temporal changes in the demand distribution.

Vol. 8, No. 4, August 2026, pp. 1259-1298

The same target series, temporal resolution, forecast horizon, and test period were used across competing models. The proposed method did not receive a longer historical window or additional external information. Improvements were therefore attributed to model configuration, multi-scale candidate construction, candidate selection, and dynamic fusion rather than to unequal input information (Figure 3).

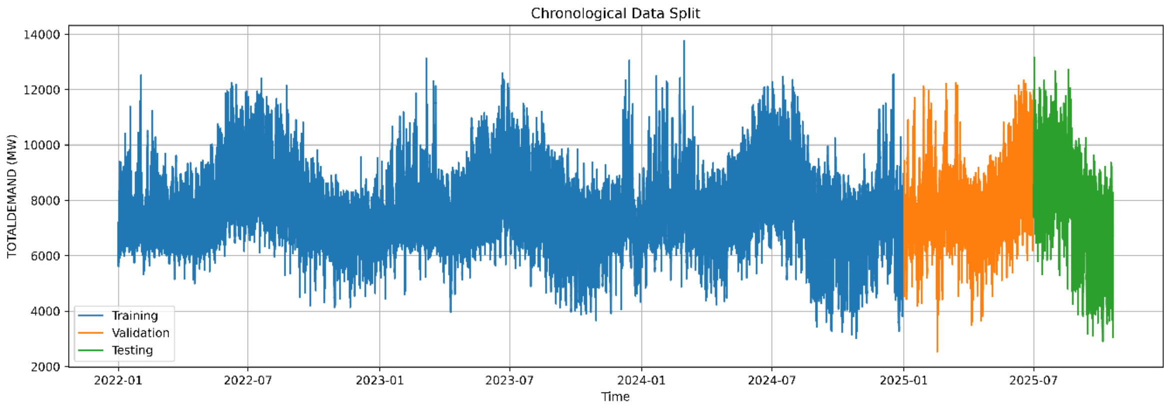


Figure 3. Chronological partition of the experimental dataset

4.3. Baseline Models and Implementation Details

The proposed DME-iTransformer was compared with representative forecasting models from naive, deep-learning, and Transformer-based families. The comparative models were selected to provide both simple seasonal references and nonlinear learned forecasting approaches.The evaluated methods were:

- Persistence Naive, which uses the most recent available demand pattern as the future reference;
- Seasonal Naive, which exploits recurring seasonal behavior and provides a strong benchmark for highly periodic electricity-demand data;
- CNN, which learns local temporal structures using convolutional operations;
- LSTM, which represents sequential dependencies through recurrent memory mechanisms;
- iTransformer, which serves as the standalone Transformer-based benchmark; and
- DME-iTransformer, which integrates the tuned iTransformer forecast with multi-scale lag and profile candidates through validation-optimized ensemble learning. The benchmark set was selected to examine the incremental contribution of the proposed framework relative to simple persistence, explicit seasonal recurrence, conventional deep-learning models, and its own standalone Transformer backbone. It is not intended to provide an exhaustive comparison with every recent Transformer architecture. Models such as PatchTST, TimesNet, FEDformer, and Crossformer were not implemented within the experimental scope of the present study. Therefore, the conclusions are restricted to the six methods evaluated under the common forecasting protocol, while broader Transformer benchmarking is reserved for future work. The comparison therefore covers the progression

$$\text{Naive} \rightarrow \text{Deep Learning} \rightarrow \text{Transformer} \rightarrow \text{Proposed Hybrid Ensemble} \tag{44}$$

For the proposed method, the input size, forecast horizon, rolling step size, and number of target series were fixed for every iTransformer configuration. Only internal model hyperparameters were varied.

Eight predefined configurations were evaluated. The search considered:

Table 2. Hyperparameter Search Space and Candidate Values

Hyperparameter	Candidate values
Hidden size	64, 128, 256
Attention heads	4, 8
Encoder layers	1, 2
Feed-forward dimension	128, 256, 512, 1024
Dropout	0.05, 0.10, 0.15
Learning rate	$10^{-3}, 7\times10^{-4}, 5\times10^{-4}, 3\times10^{-4}$
Maximum training steps	1200, 1500, 1600, 1800
Batch size	32
Window batch size	64
Scaling	Robust, Identity
Training loss	MAPE, MAE

The best iTransformer configuration was selected according to validation MAPE:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}_c \in \Theta} \text{MAPE}_{\text{tune}}(\boldsymbol{\theta}_c) \tag{45}$$

Training used early stopping with a patience of eight validation checks, while validation performance was examined every 50 training steps. Four learning-rate decay stages were applied.

The DME candidate-selection procedure retained at most ten candidates: with $K_{\max} = 10$ a minimum target of five candidates whenever sufficient valid forecasts were available. Candidate pairs with an absolute correlation of at least 0.999 were treated as highly redundant.

For dynamic ensemble optimization, the regularization coefficients were set to $\lambda_G = 0.005, \lambda_S = 0.02, \lambda_{SH} = 0.05$ for global, state-specific, and state–horizon-specific weighting, respectively. A minimum of 250 observations was required before a local weight vector was estimated; otherwise, the framework reverted to a broader state or global weighting strategy.

The constrained optimization procedure used SLSQP together with random Dirichlet search. The global optimization stage used 600 random trials, whereas each local optimization stage used 250 trials.

Finally, the optional residual-corretion strength was selected from $\mathcal{A} = \{0,\ 0.15,\ 0.30,\ 0.50\}$. The inclusion of zero allows the SAFE-SELECT mechanism to completely disable bias correction when no validation improvement is observed.

4.4. Evaluation Methods

Forecasting performance was assessed primarily using mean absolute percentage error (MAPE), because it expresses the average prediction error relative to the actual load magnitude and was also used for iTransformer configuration selection, candidate ranking, ensemble optimization, and final strategy selection.

For N observations, MAPE is defined as

$$\text{MAPE} = \frac{100}{N} \sum_{i=1}^{N} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{46}$$

Lower values indicate better forecasting accuracy.

To provide complementary error information, mean absolute error, root mean squared error, symmetric mean absolute percentage error, and the coefficient of determination were also calculated. RMSE is given by

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \tag{47}$$

while sMAPE is

$$\text{sMAPE} = \frac{100}{N} \sum_{i=1}^{N} \frac{2 \mid y_i - \hat{y}_i \mid}{\mid y_i \mid + \mid \hat{y}_i \mid} \tag{48}$$

The coefficient of determination is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N} (y_i - \bar{y})^2} \tag{49}$$

Although these metrics were computed for diagnostic purposes, MAPE was treated as the principal accuracy criterion throughout the proposed framework.

In addition to overall performance, forecasting behavior was analyzed across individual horizon steps. For forecast step h, horizon-specific MAPE is calculated as

$$\text{MAPE}_h = \frac{100}{N_h} \sum_{i=1}^{N_h} \left| \frac{y_{i,h} - \hat{y}_{i,h}}{y_{i,h}} \right|, h = 1, \dots, 288 \tag{50}$$

This analysis reveals whether forecasting errors increase at longer lead times or remain stable across the 24-h trajectory.

Performance was also examined by calendar month and load state. Three load regimes were considered:

$$s_t = \begin{cases} \text{Low}, & y_t \le Q_{0.15}, \\ \text{Normal}, & Q_{0.15} < y_t < Q_{0.85}, \\ \text{High}, & y_t \ge Q_{0.85}, \end{cases} \tag{51}$$

where $Q_{0.15}$ and $Q_{0.85}$ are the 15th and 85th percentiles estimated from historical data.

Accordingly, the evaluation was performed at four complementary levels: overall test performance; horizon-wise performance; monthly performance; load-state-specific performance.

This multi-level assessment makes it possible to determine not only whether the proposed model reduces average error, but also whether its performance remains consistent across forecast lead times and demand regimes.

4.5. Computational Environment and Reproducibility

All experiments were implemented in Python 3 and executed in the Google Colab environment. An NVIDIA A100 GPU was used to accelerate the training and inference of the deep-learning models, and the High-RAM runtime option was enabled to support the processing of the high-resolution load series and the evaluation of multiple iTransformer configurations. The same hardware environment was maintained throughout the main experimental runs to reduce variation caused by differences in computational resources.

The forecasting models were implemented using PyTorch and the NeuralForecast library. NumPy and Pandas were used for numerical computation and data preprocessing, while SciPy was employed for constrained ensemble-weight optimization. Matplotlib was used to generate the figures, and OpenPyXL was used to read and export spreadsheet-based data and experimental results. CUDA acceleration was enabled whenever it was available for the corresponding PyTorch operations. The principal hardware and software settings are summarized in Table 3.

Taable 3. Computational Environment and Software Configuration

Component	Setting
Computing platform	Google Colab
Programming language	Python 3
Hardware accelerator	NVIDIA A100 GPU
Memory configuration	High-RAM enabled
Colab runtime version	Recorded for each experimental run
Deep-learning framework	PyTorch
Forecasting library	NeuralForecast
Numerical computation	NumPy
Data processing	Pandas
Constrained optimization	SciPy
Figure generation	Matplotlib
Spreadsheet processing	OpenPyXL
GPU computing interface	CUDA
Random seed	1

To reduce stochastic variation, the random seed was fixed at 1 for the Python random module, NumPy, PyTorch, and the available CUDA operations. The same chronological data partitions, input length, forecasting horizon, rolling interval, validation periods, baseline configurations, and evaluation protocol were maintained across the experiments. Thus, differences in forecasting performance were not caused by changes in the principal data or evaluation settings.

The experimental implementation automatically recorded the Python version, operating platform, CUDA environment, and versions of the principal software libraries for each run. It also stored the evaluated iTransformer configurations, the selected hyperparameters, candidate-ranking results, retained candidate set, optimized ensemble weights, selected forecasting strategy, residual-bias setting, prediction outputs, and final evaluation metrics. These records allow each stage of the forecasting procedure to be traced from model configuration selection to final out-of-sample evaluation.

The experimental outputs were organized into separate directories according to their function. The configs directory contains the selected model configurations and experimental settings. The tables directory stores the configuration-search results, candidate evaluations, optimized weights, and final performance metrics. The data directory contains the processed load series, aligned actual values, and forecasting outputs. The figures_600dpi directory contains the publication-quality figures generated from the final experimental results.

This organization facilitates experiment tracking, comparison among model variants, and independent verification of the reported findings.

The fixed random seed improves the repeatability of the experiments; however, exact numerical replication may still depend on the PyTorch, CUDA, NeuralForecast, and Google Colab runtime versions. Therefore, the precise software versions recorded during the final experimental run should be reported in the archived configuration file and, where space permits, in Table 3.

5. RESULTS AND DISCUSSION

5.1. Benchmark Accuracy and Improvement Analysis

The overall forecasting performance of the six evaluated models is summarized in Table 4. This comparison constitutes the primary quantitative evaluation of the study because all methods were assessed using the same NSW1 demand series, 5-min temporal resolution, 288-step input window, 288-step day-ahead forecasting horizon, rolling step size, and test period. The comparison includes two naive forecasting methods, two conventional deep-learning models, a standalone iTransformer, and the proposed DME-iTransformer. Six complementary metrics are reported: MAE, MSE, RMSE, MAPE, sMAPE, and R^2. Among them, MAPE is treated as the principal performance criterion because it is also used throughout the proposed framework for iTransformer configuration selection, candidate ranking, ensemble optimization, and final strategy selection. Lower values of MAE, MSE, RMSE, MAPE, and sMAPE indicate better forecasting accuracy, whereas a higher R^2 indicates greater explanatory capability.

Table 4. Forecasting performance comparison of different models

Model	MAE	MSE	RMSE	MAPE (%)	sMAPE (%)	R^2
Persistence Naive	1,193.406	2,242,069.293	1,497.354	17.596	15.929	0.2863
Seasonal Naive	570.970	692,704.722	832.289	7.979	7.850	0.7795
CNN	835.153	1,225,256.730	1,106.913	12.419	11.508	0.6100
LSTM	1,396.560	3,210,954.304	1,791.914	19.830	18.508	−0.0221
iTransformer	602.724	657,884.803	811.101	8.691	8.332	0.7906
DME-iTransformer	**478.485**	**476,259.933**	**690.116**	**7.031**	**6.718**	**0.8484**

Table 4 shows that the proposed DME-iTransformer achieves the best performance for all six evaluation metrics. It produces the lowest MAE of 478.485, the lowest MSE of 476,259.933, the lowest RMSE of 690.116, the lowest MAPE of 7.031%, and the lowest sMAPE of 6.718%. It also attains the highest R^2 of 0.8484. The consistency of these results is important because the superiority of the proposed model is not restricted to a single error measure. Instead, the model improves both average absolute accuracy and squared-error-based performance while explaining a larger proportion of the observed load variation.

The MAPE results provide the clearest overall comparison. DME-iTransformer achieves a MAPE of 7.031%, followed by Seasonal Naive at 7.979% and the standalone iTransformer at 8.691%. The remaining models perform considerably worse, with MAPEs of 12.419% for CNN, 17.596% for Persistence Naive, and 19.830% for LSTM. Relative to the strongest conventional benchmark, Seasonal Naive, the proposed model reduces MAPE by approximately 11.88%. Compared with the standalone iTransformer, the reduction reaches approximately 19.10%. These two comparisons are particularly important because Seasonal Naive represents a strong periodic benchmark, whereas iTransformer represents the learned neural backbone of the proposed framework.

The strong performance of Seasonal Naive deserves particular attention. Its MAPE of 7.979% is lower than those of CNN, LSTM, Persistence Naive, and even the standalone iTransformer. This result indicates that the NSW1 demand series contains strong recurring temporal structure and that corresponding historical periods remain highly informative for day-ahead forecasting. Therefore, the Seasonal Naive benchmark should not be regarded merely as a trivial reference. Rather, it constitutes one of the most competitive models in the experiment. Against this strong benchmark, the observed reduction from 7.979% to 7.031% is notable in descriptive terms because the proposed method improves upon an already effective representation of periodic load behavior.

The comparison with iTransformer provides direct evidence regarding the value of the proposed multi-scale ensemble design. The standalone iTransformer achieves a MAPE of 8.691%, whereas DME-iTransformer reduces this value to 7.031%. The corresponding MAE decreases from 602.724 to 478.485, representing a reduction of approximately 20.61%. RMSE decreases from 811.101 to 690.116, corresponding to an improvement of approximately 14.92%, while MSE is reduced by approximately 27.61%. The R^2 value also increases from 0.7906 to 0.8484. These improvements suggest that the additional multi-scale lag and profile information provides complementary predictive information that is not fully captured by the standalone iTransformer.

A similar pattern is observed when the proposed model is compared with Seasonal Naive. DME-iTransformer reduces MAE from 570.970 to 478.485, corresponding to an improvement of approximately 16.20%. MSE decreases from 692,704.722 to 476,259.933, a reduction of approximately 31.25%, while RMSE is reduced by approximately 17.08%. The sMAPE improvement is approximately 14.42%, and R^2 increases from 0.7795 to 0.8484. These findings indicate that the advantage of the proposed framework extends beyond relative percentage error and remains evident under absolute- and squared-error-based criteria.

The squared-error metrics provide additional insight into the ability of the models to control larger forecasting deviations. Although Seasonal Naive achieves a lower MAE and MAPE than the standalone iTransformer, iTransformer obtains slightly lower MSE and RMSE values. Specifically, iTransformer achieves an MSE of 657,884.803 and an RMSE of 811.101, compared with 692,704.722 and 832.289, respectively, for Seasonal Naive. This difference suggests that the two models exhibit distinct forecasting characteristics: Seasonal Naive performs better in average relative and absolute terms, whereas iTransformer appears somewhat more effective at limiting certain larger squared deviations. The proposed DME-iTransformer surpasses both models simultaneously, achieving the lowest values for all absolute, relative, and squared-error metrics.

The R^2 results further support this conclusion. DME-iTransformer obtains the highest value of 0.8484, followed by iTransformer at 0.7906 and Seasonal Naive at 0.7795. Thus, the proposed model improves R^2 by 0.0578 over iTransformer and by 0.0689 over Seasonal Naive. This indicates that the final forecasts more closely reproduce the observed variability of the NSW1 load series. The agreement between the lowest error measures and the highest R^2 strengthens the evidence that the improvement is not caused by one isolated metric.

The CNN provides intermediate performance, with a MAPE of 12.419%, an RMSE of 1,106.913, and an R^2 of 0.6100. Relative to CNN, DME-iTransformer reduces MAPE by approximately 43.39%, MAE by 42.71%, and RMSE by 37.65%. Although the CNN can capture local temporal patterns, its results suggest that local convolutional representations alone are insufficient to fully describe the daily, weekly, and longer-range dependencies present in the high-resolution demand series.

Persistence Naive performs substantially worse, with a MAPE of 17.596% and an R^2 of only 0.2863. DME-iTransformer reduces MAPE by approximately 60.04% relative to this baseline. The large difference between Persistence Naive and Seasonal Naive is also informative. Persistence Naive relies primarily on the most recent demand behavior, whereas Seasonal Naive explicitly exploits recurring temporal structure. The substantially better result of Seasonal Naive reinforces the importance of periodic information in the present forecasting problem and provides further motivation for the multi-scale lag and profile construction used in DME-iTransformer.

Among all evaluated methods, LSTM produces the highest errors, with a MAPE of 19.830%, an RMSE of 1,791.914, and an R^2 of -0.0221. A negative R^2 indicates that, under the squared-error criterion, the LSTM predictions fail to explain the test-set variability effectively. Relative to LSTM, the proposed model reduces MAPE by approximately 64.54%, MAE by 65.74%, and RMSE by 61.49%. This result also shows that increased model complexity alone does not guarantee superior forecasting accuracy. The ability to exploit the temporal structure appropriate to the dataset is more important than simply using a sophisticated neural architecture.

Taken together, the results reveal three important findings. First, the strong performance of Seasonal Naive confirms that high-resolution electricity demand contains pronounced recurring patterns. Second, the standalone iTransformer provides competitive nonlinear forecasting capability but does not outperform the strongest seasonal benchmark in terms of MAPE. Third, the proposed DME-iTransformer achieves the best result by integrating these complementary forecasting perspectives rather than relying exclusively on either a deterministic seasonal reference or a single learned neural model.

The proposed model can therefore be interpreted as bridging the gap between explicit temporal recurrence and learned nonlinear forecasting. The multi-scale lag and profile candidates preserve recurring daily, weekly, and multi-week information, whereas iTransformer provides a learned representation of complex demand dynamics. Candidate selection reduces redundant information, and validation-optimized fusion determines how the retained forecasting sources should be combined. The superior performance across all six metrics suggests that this coordinated framework produces more accurate and stable forecasts than its individual components.

Overall, Table 4 provides the primary quantitative evidence supporting the effectiveness of the proposed DME-iTransformer. The model achieves the lowest MAPE of 7.031%, improving upon the strongest baseline, Seasonal Naive, by 11.88%, and the standalone iTransformer by 19.10%. At the same time, it records the lowest MAE, MSE, RMSE, and sMAPE and the highest R^2. These results confirm that the proposed combination of multi-scale temporal information, tuned iTransformer forecasting, candidate selection, and validation-optimized dynamic fusion provides a clear overall accuracy advantage under the common high-resolution day-ahead forecasting setting.

While Table 4 establishes the overall numerical superiority of DME-iTransformer, aggregate metrics alone do not reveal how forecasting performance changes over time, across forecast horizons, or under different operating conditions. Therefore, the following analyses examine load-profile tracking, residual characteristics, horizon-wise robustness, and peak-load forecasting behavior to determine whether the overall improvement remains consistent beyond the average test-set metrics.

The performance differences reported in this subsection are descriptive comparisons based on aggregate test-set metrics. No formal hypothesis test, confidence-interval analysis, or multiple-comparison correction was conducted. Accordingly, the reported reductions should be interpreted as observed empirical improvements on the NSW1 test period rather than as evidence of statistical significance. Formal paired testing across daily forecasting windows and evaluation over additional regional datasets are reserved for future work.

Direct numerical comparison with accuracy values reported in previous publications was not performed because differences in geographic region, sampling interval, input length, forecast horizon, data partition, external information, and evaluation protocol can make cross-study rankings misleading. The primary comparison in this study is therefore restricted to models evaluated using the same NSW1 data and forecasting protocol. Broader cross-dataset benchmarking is identified as future work.

5.2. Load Profile Tracking and Error Characteristics

To complement the aggregate performance results reported in Table 4, Figure 4 provides a visual comparison of the forecasting behavior of the six models during the first seven days of the test period. The selected interval contains repeated daily load cycles together with substantial variations in peak magnitude and load valleys, allowing the models to be compared in terms of temporal tracking, peak representation, and responsiveness to short-term demand changes. The figure therefore provides a qualitative assessment of whether the numerical improvements reported in Table 4 are also reflected in the predicted load trajectories.

As shown in Figure 4(a), Persistence Naive produces piecewise-constant forecasts and fails to reproduce the pronounced intra-day fluctuations of the actual load. Although it roughly follows changes in the general demand level across consecutive days, it cannot capture either the timing or magnitude of the major peaks and valleys. This behavior explains its relatively high forecasting error reported in Table 4.

Seasonal Naive in Figure 4(b) captures the recurring daily structure considerably better. Its predictions reproduce most of the major rising and falling patterns, confirming the strong periodicity of the NSW1 load series. Nevertheless, visible deviations remain around several local peaks and valleys, particularly when the demand magnitude differs substantially from the corresponding historical seasonal pattern.

The CNN and LSTM exhibit markedly different behaviors. In Figure 4(c), the CNN generates a smooth forecast trajectory and follows the broad daily cycles, but the smoothing effect reduces its ability to reproduce abrupt changes and extreme load levels. In contrast, the LSTM predictions in Figure 4(d) remain close to a nearly constant level and fail to respond adequately to the strong temporal variations in the actual series. This visual behavior is consistent with the comparatively weak LSTM performance reported in Table 4.

The standalone iTransformer in Figure 4(e) provides substantially better temporal tracking. It follows most of the daily cycles and reproduces many of the turning points more accurately than the conventional deep-learning models. However, noticeable deviations are still observed during several high-load and low-load periods, indicating that the learned Transformer representation alone does not fully capture all recurring and regime-dependent variations.

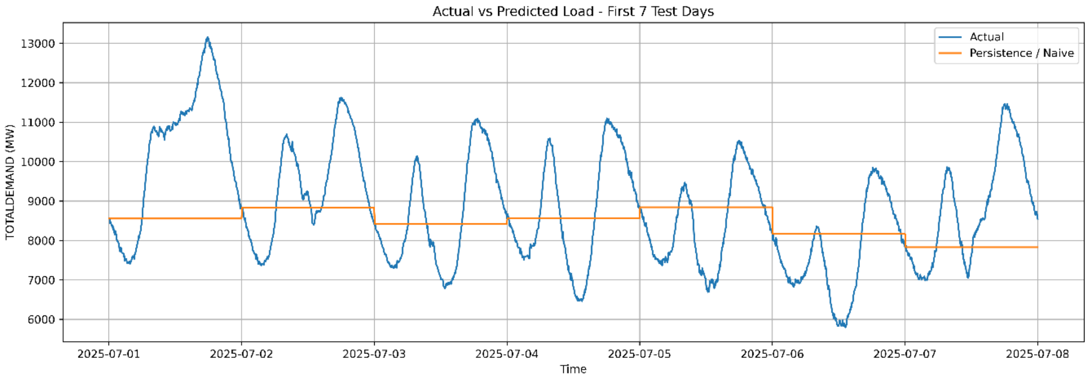
Finally, Figure 4(f) shows that the proposed DME-iTransformer generally follows the actual load trajectory more closely across the seven-day period. The final forecast preserves the temporal structure captured by the tuned iTransformer while better adjusting several peaks, valleys, and intermediate load transitions through the multi-scale candidate and dynamic fusion mechanisms. Although some deviations remain during the most abrupt load changes, the overall trajectory shows closer agreement with the observed demand than the competing models. This visual evidence supports the quantitative findings in Table 4 and indicates that the improvement of DME-iTransformer is reflected not only in lower aggregate errors but also in more consistent tracking of short-term load dynamics.

While Figure 4 demonstrates the temporal tracking capability of the evaluated models, load-profile comparisons alone do not fully reveal the distribution and concentration of forecasting errors. Therefore, the residual characteristics of the six models are examined next.

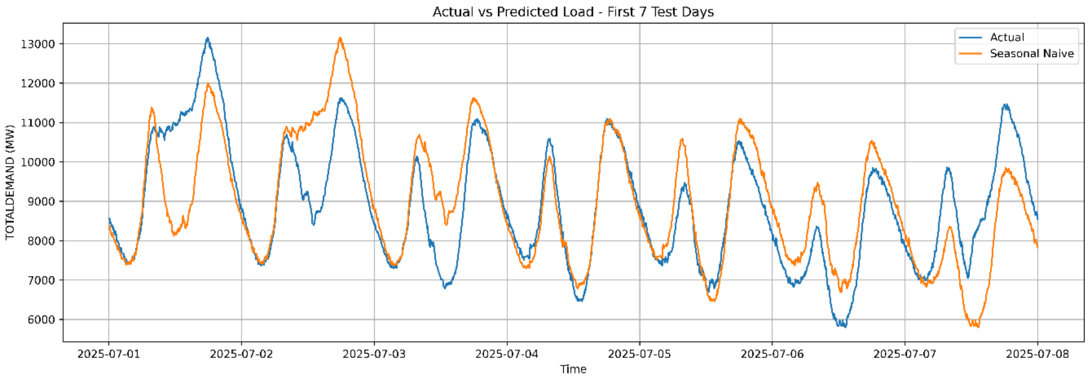
To further examine the error characteristics of the evaluated models, Figure 5 presents the residual distributions over the test period. The residual is defined as the difference between the actual and predicted load, such that positive values indicate underestimation and negative values indicate overestimation. A desirable forecasting model should produce a narrow distribution with a pronounced concentration around zero

Vol. 8, No. 4, August 2026, pp. 1259-1298

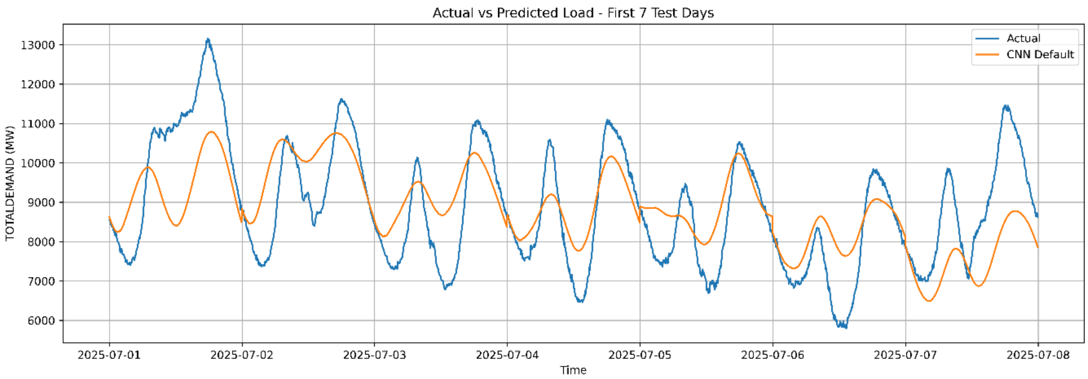
and relatively short tails. Unlike the load-profile comparison in Figure 4, this analysis provides a more direct view of the dispersion, bias, and frequency of large forecasting errors.



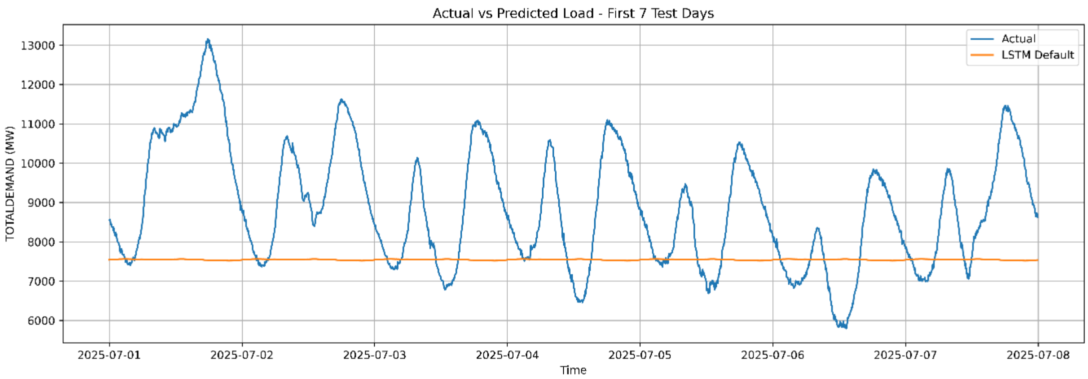
(a) Persistence Naive



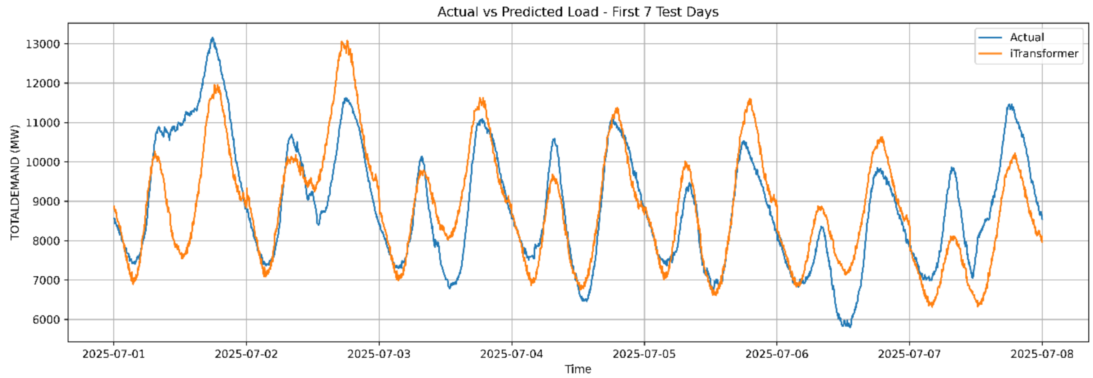
(b) Seasonal Naive



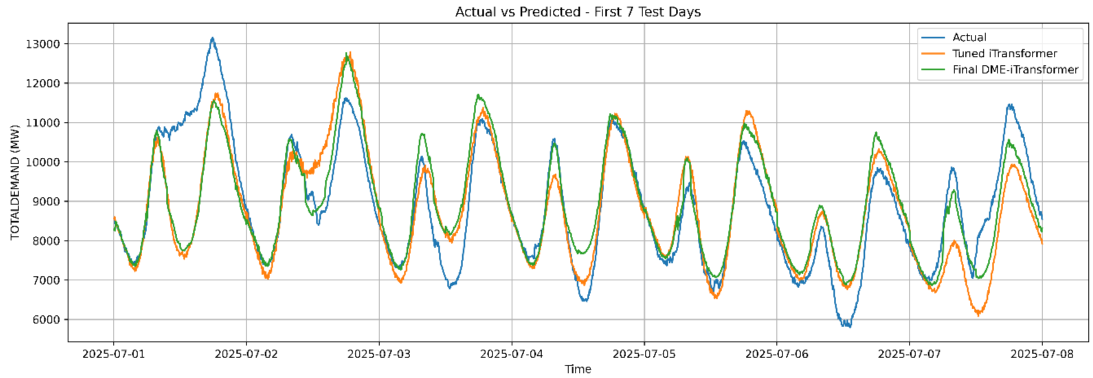
(c) CNN



(d) LSTM



(e) iTransformer



(f) DME-iTransformer

Figure 4. Actual and predicted load profiles during the first seven test days: (a) Persistence Naive, (b) Seasonal Naive, (c) CNN, (d) LSTM, (e) iTransformer, and (f) DME-iTransformer

As shown in Figure 5(a), Persistence Naive produces a broad and irregular residual distribution extending across a wide error range. The presence of substantial frequencies on both sides of zero indicates that this model frequently produces both overestimation and underestimation errors. This dispersed pattern is consistent with its limited ability to follow the rapidly changing demand profile observed in Figure 4(a).

The Seasonal Naive model in Figure 5(b) exhibits a considerably sharper concentration around zero. Most residuals are located within a narrower central region, indicating that seasonal recurrence provides a strong approximation of the actual demand pattern. However, the remaining tails on both sides show that large errors can still occur when current load conditions differ from the corresponding historical seasonal pattern.

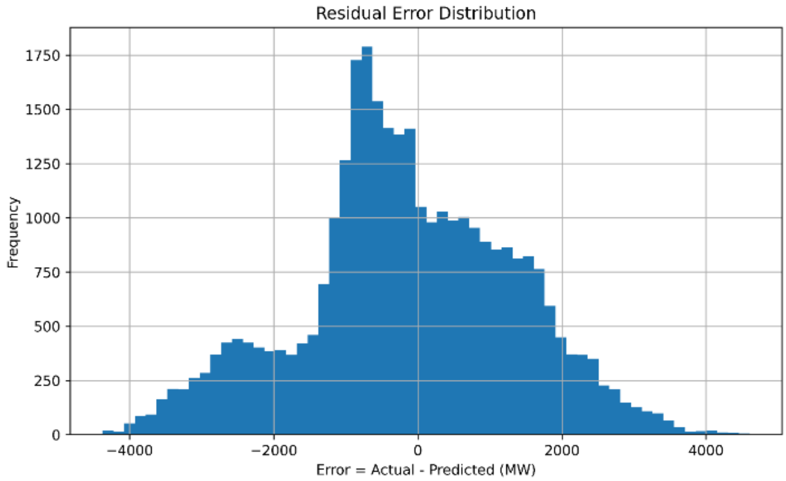
The CNN distribution in Figure 5(c) is also centered close to zero but is visibly broader than that of Seasonal Naive. The extended negative and positive tails suggest that the smoothing behavior observed in Figure 4(c) leads to larger deviations during rapid load transitions and extreme demand levels. Thus, although CNN captures the overall temporal pattern, its residual spread reflects limited precision during more variable operating conditions.

Figure 5(d) shows that LSTM produces the widest and most dispersed residual distribution among the learning-based models. The long tails and relatively broad central region indicate frequent large forecasting deviations, which is consistent with its weak temporal tracking in Figure 4(d) and the relatively high errors reported in Table 4.

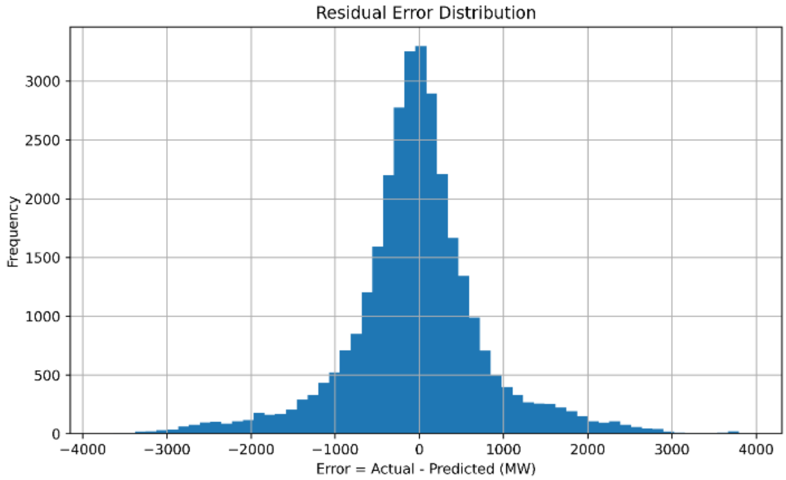
The standalone iTransformer in Figure 5(e) presents a much more compact distribution, with a clear peak near zero and a lower frequency of large residuals. This confirms its stronger ability to represent the temporal dynamics of the load series compared with CNN and LSTM. Nevertheless, noticeable tails remain, indicating that some load transitions and extreme variations are still difficult to predict accurately.

Finally, the proposed DME-iTransformer in Figure 5(f) exhibits the sharpest concentration of residuals around zero among the six models. Its pronounced central peak indicates that a larger proportion of predictions produce relatively small errors, while the overall residual spread is reduced compared with the competing approaches. Although some extreme residuals remain, the tighter central concentration supports the lower MAE, RMSE, MAPE, and sMAPE reported in Table 4. These findings suggest that the combination of multi-scale temporal candidates, tuned iTransformer forecasts, and validation-optimized dynamic fusion not only reduces the average forecasting error but also improves the overall concentration and stability of the residual distribution.

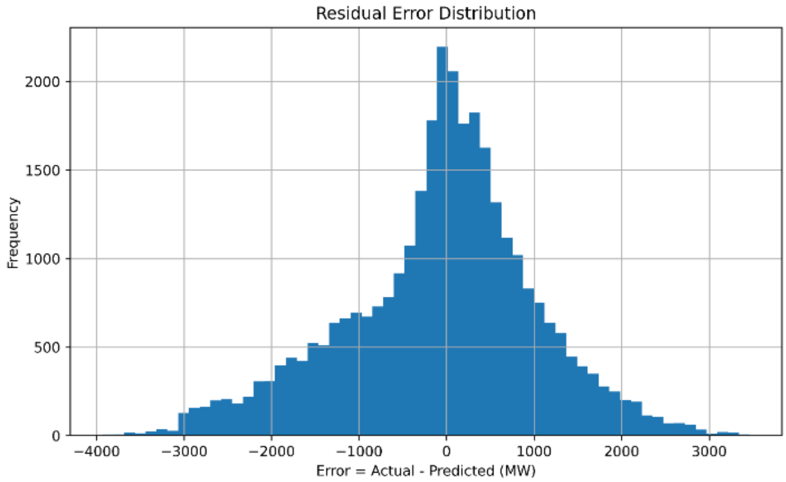
Although Figure 4 and Figure 5 demonstrate improved temporal tracking and error concentration, these aggregate visualizations do not show whether the forecasting advantage remains consistent across the entire 288-step day-ahead horizon. Therefore, the next analysis examines the horizon-wise forecasting performance of all six models.



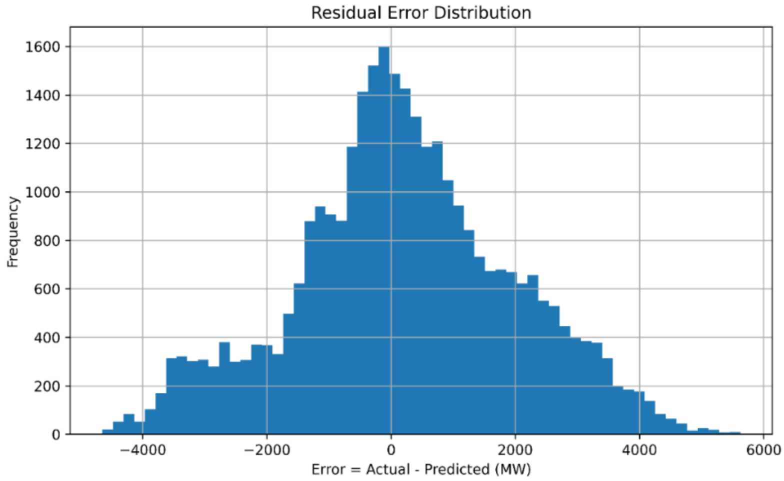
(a) Persistence Naive



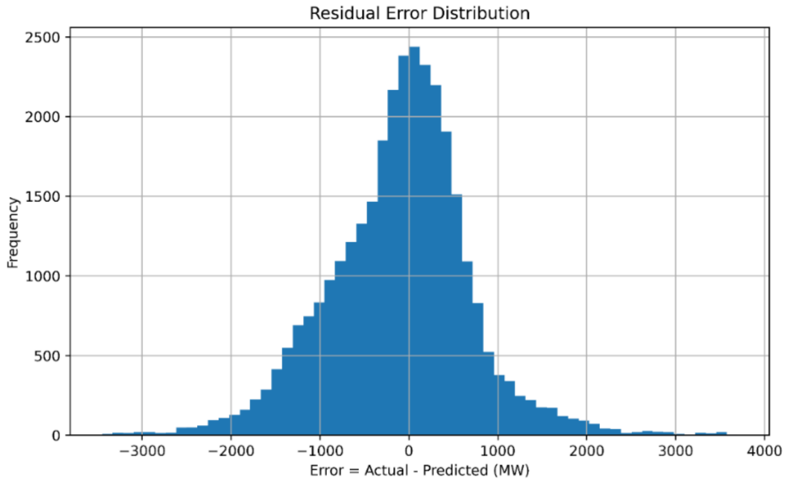
(b) Seasonal Naive



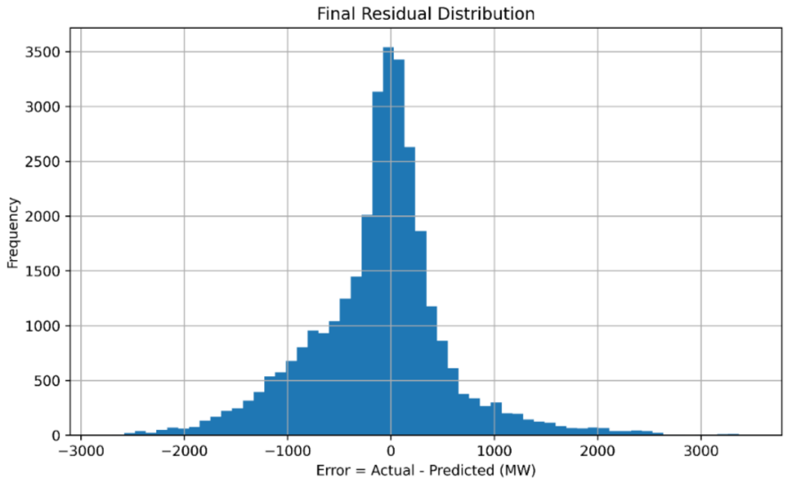
(c) CNN



(d) LSTM



(e) iTransformer

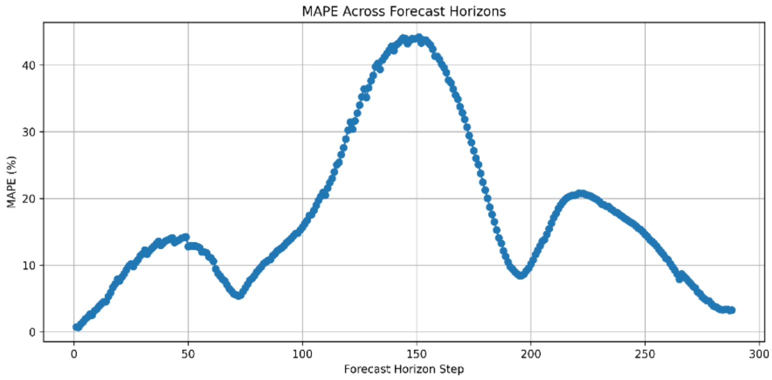


(f) DME-iTransformer

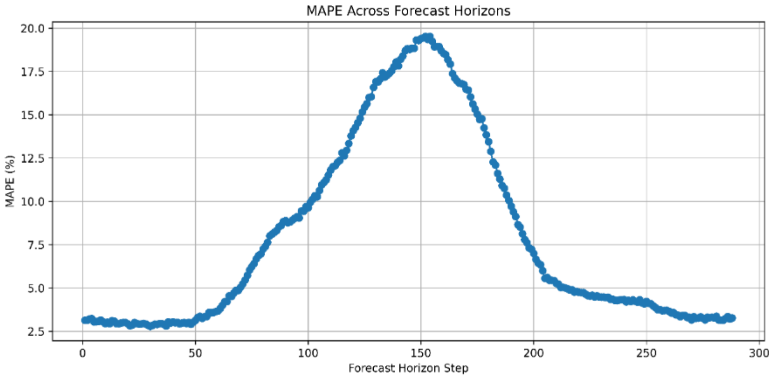
Figure 5. Residual distributions of the evaluated models

5.3. Forecast Robustness Across Horizons and Load Regimes

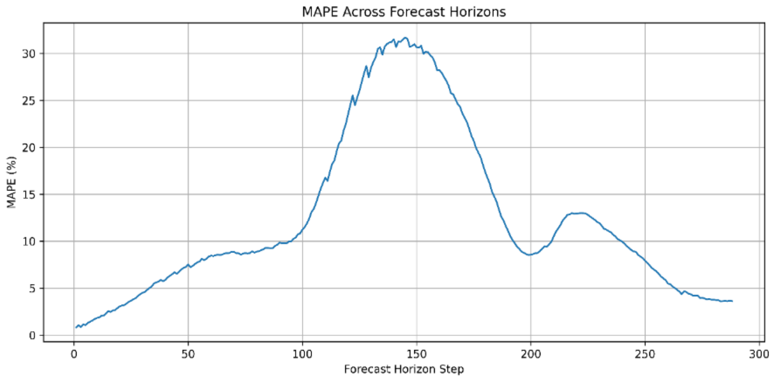
To investigate whether forecasting accuracy remains consistent throughout the 24-h prediction period, Figure 6 compares the horizon-wise MAPE of the six evaluated models over all 288 forecast steps. Since each step corresponds to 5 min, the figure reveals how prediction difficulty changes from the beginning to the end of the day-ahead horizon. A robust forecasting model should maintain relatively low errors across the entire horizon rather than achieving good performance only at selected lead times.



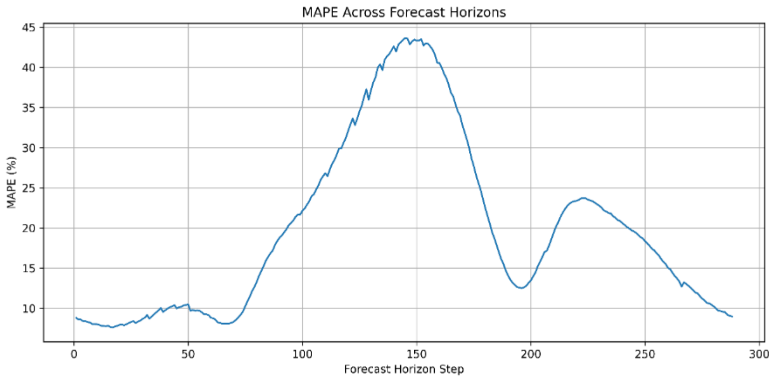
(a) Persistence Naive



(b) Seasonal Naive

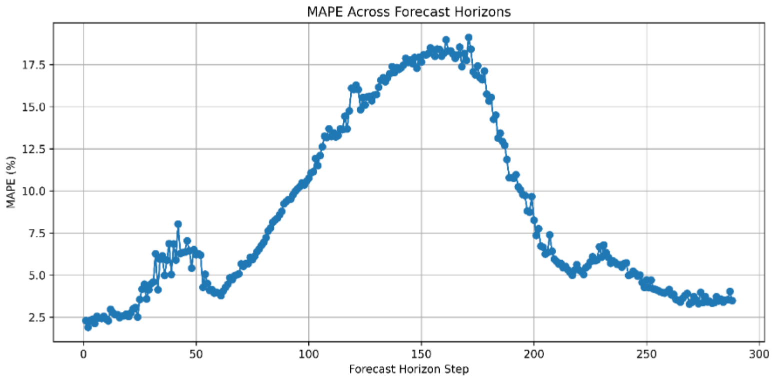


(c) CNN

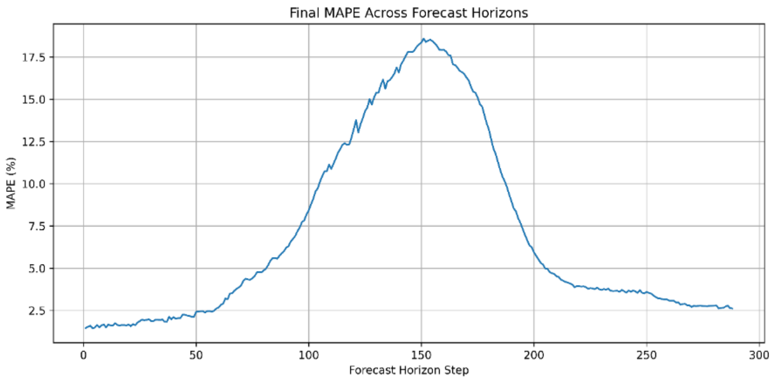


(d) LSTM

Vol. 8, No. 4, August 2026, pp. 1259-1298



(e) iTransformer



(f) DME-iTransformer

Figure 6. Horizon-wise MAPE comparison of all forecasting models over the 288-step day-ahead horizon: (a) Persistence Naive, (b) Seasonal Naive, (c) CNN, (d) LSTM, (e) iTransformer, and (f) DME-iTransformer

Figure 6(a) shows that Persistence Naive exhibits substantial error variation across the forecasting horizon. The MAPE is relatively low at the earliest steps but increases sharply during the middle portion of the horizon, reaching its highest level around steps 140–160. A second increase is also observed later in the day. This pattern indicates that simply carrying forward recent demand information is inadequate when the future load trajectory departs from the current demand level.

Seasonal Naive in Figure 6(b) produces a considerably lower and more stable error profile. Its MAPE remains relatively small during the early and late forecast steps, although a clear increase is again observed around the middle of the horizon. This confirms that recurring seasonal information is highly useful for the NSW1 series, but its accuracy deteriorates during periods characterized by stronger changes in load magnitude.

The CNN results in Figure 6(c) follow a similar general pattern but with a larger mid-horizon error peak. The error increases gradually from the early forecast steps, reaches its maximum around the central part of the horizon, and then declines toward the end. This behavior suggests that the CNN captures short-term temporal structure reasonably well but becomes less effective during the more variable part of the daily load cycle.

As shown in Figure 6(d), LSTM presents the highest horizon-wise errors among the learning-based models. Its MAPE remains elevated over a broad portion of the forecast horizon and increases markedly around the middle steps. The wide high-error region is consistent with the weak overall performance reported in Table 4 and indicates limited ability to adapt to changing intra-day demand dynamics.

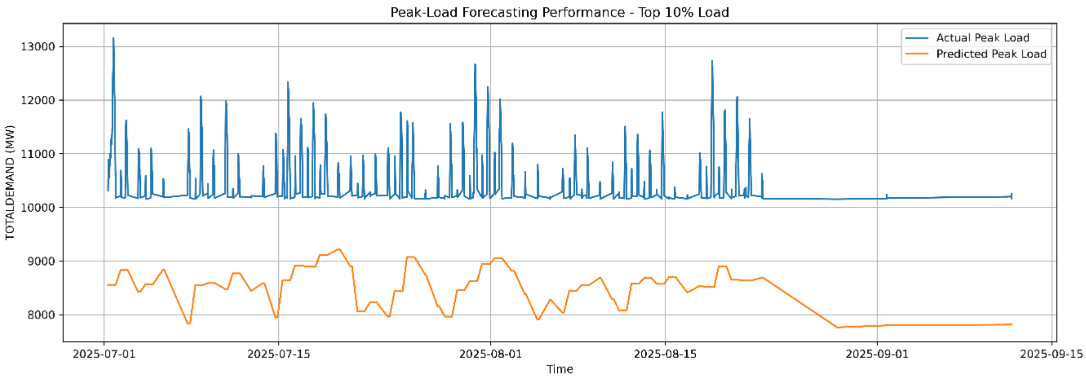
The standalone iTransformer in Figure 6(e) achieves a much lower error profile than CNN and LSTM. Although its MAPE also rises around the middle of the horizon, the overall magnitude remains clearly reduced. This result demonstrates the stronger ability of the Transformer architecture to represent long-range temporal dependencies, while also showing that certain intra-day periods remain particularly difficult to forecast.

Finally, Figure 6(f) shows that DME-iTransformer maintains the lowest overall horizon-wise error pattern. The MAPE remains particularly low during the early forecast steps and decreases again toward the end of the horizon. Although the central portion of the day continues to be the most challenging region, the proposed model substantially reduces the error magnitude compared with Persistence Naive, CNN, and LSTM and remains competitive with the strongest baselines throughout the horizon. This result indicates that the multi-scale temporal candidates and dynamic ensemble mechanism improve forecasting robustness across different lead times rather than only reducing the overall average error.

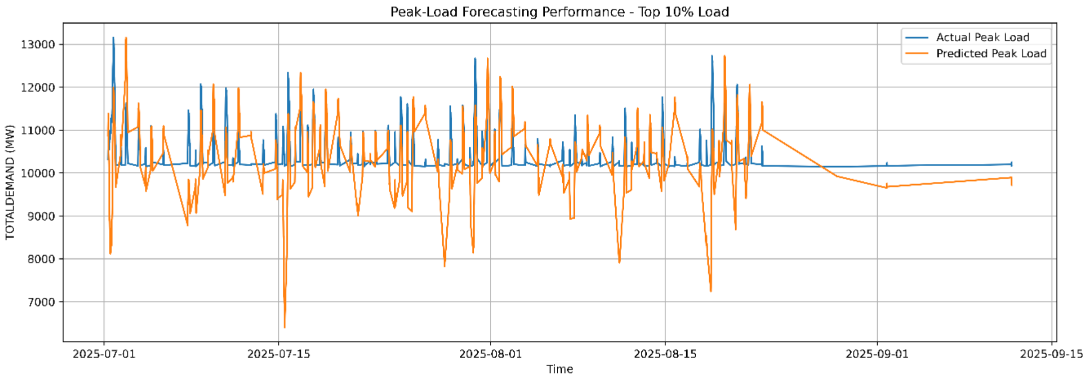
Vol. 8, No. 4, August 2026, pp. 1259-1298

Overall, Figure 6 reveals a common characteristic across all models: the middle portion of the 288-step horizon is considerably more difficult to forecast than the early and late periods. However, the magnitude of this increase differs substantially among the models. DME-iTransformer provides the most favorable overall horizon-wise profile, supporting the conclusion that its improvement is distributed across the day-ahead forecasting horizon rather than being concentrated at only a few forecast steps.

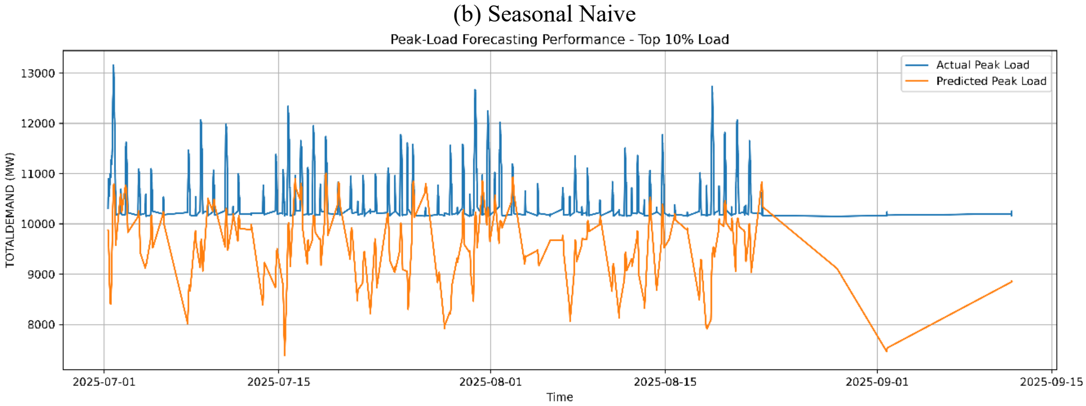
To assess model behavior under the most demanding operating conditions, Figure 7 compares the forecasting performance of the six models for observations belonging to the top 10% of actual load values. Peak-load forecasting is particularly challenging because high-demand periods are often associated with rapid changes in magnitude and relatively infrequent extreme values. A reliable forecasting model should therefore reproduce both the general high-load level and the short-duration peaks without systematic underestimation or excessive fluctuations.



(a) Persistence Naive

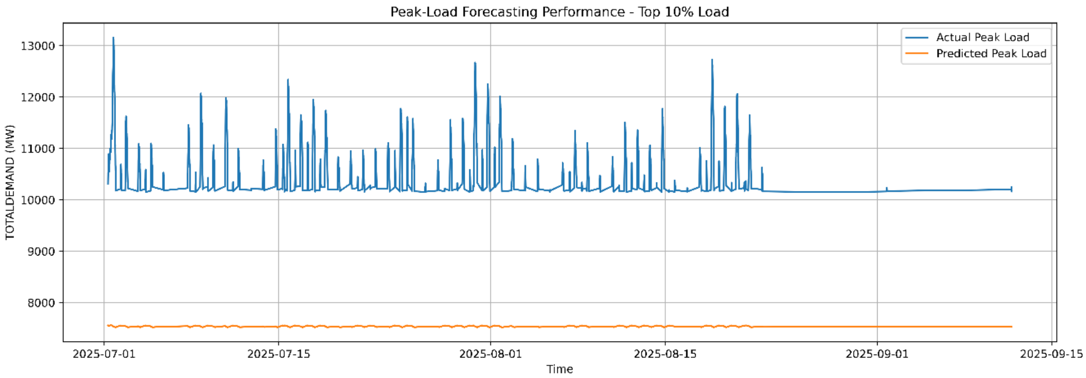


(b) Seasonal Naive

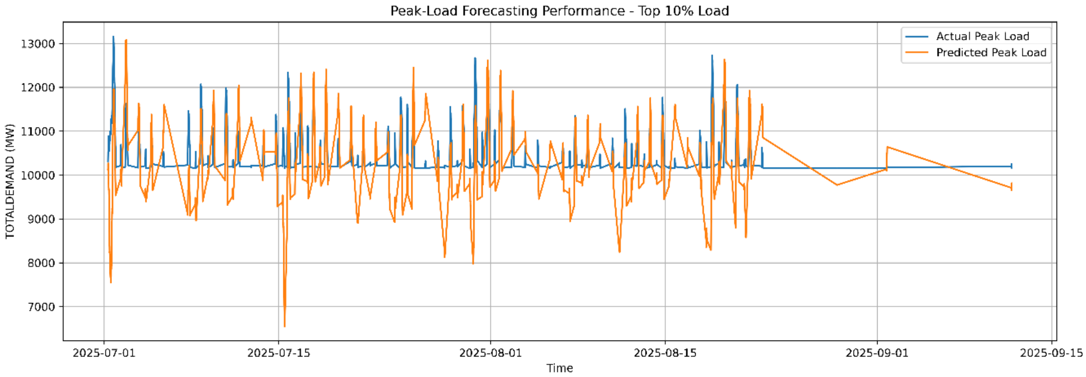


(c) CNN

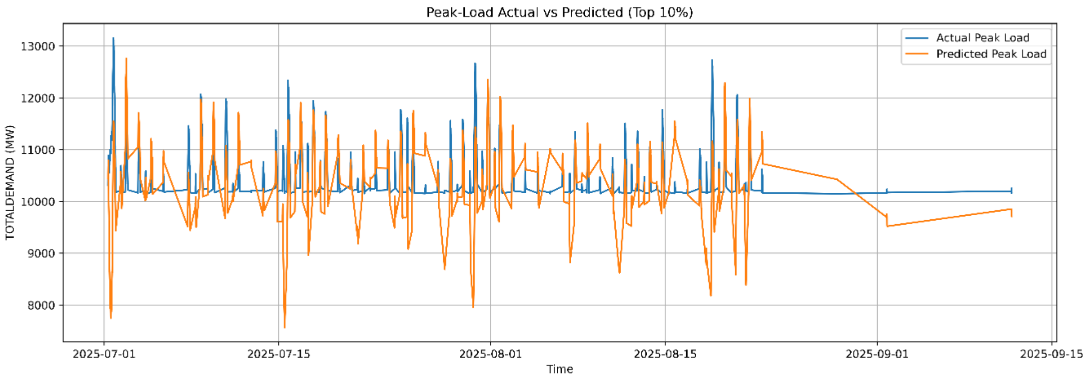
Vol. 8, No. 4, August 2026, pp. 1259-1298



(d) LSTM



(e) iTransformer



(f) DME-iTransformer

Figure 7. Peak-load forecasting behavior of the six forecasting models: (a) Persistence Naive, (b) Seasonal Naive, (c) CNN, (d) LSTM, (e) iTransformer, and (f) DME-iTransformer

Figure 7(a) shows that Persistence Naive consistently underestimates the high-load observations. Its predictions remain well below the actual peak-load level and fail to reproduce the numerous short-duration spikes observed throughout the test period. Although some gradual changes in the predicted level are visible, the method cannot adequately respond to the magnitude of the highest demand events.

Seasonal Naive in Figure 7(b) provides a much closer representation of the peak-load range. It follows many of the major fluctuations and occasionally reproduces individual high-demand events. However, the predictions remain highly variable, with several pronounced overestimation and underestimation episodes. This behavior indicates that historical seasonal recurrence is informative for high-load forecasting, but mismatches between current and previous seasonal conditions can still produce substantial errors.

The CNN results in Figure 7(c) show partial tracking of the peak-load pattern. The model captures some temporal variations but generally predicts below the actual high-load level, particularly during the largest peaks. The smoothed forecasting behavior observed earlier is also evident here, suggesting that the CNN has difficulty reproducing abrupt high-demand excursions.

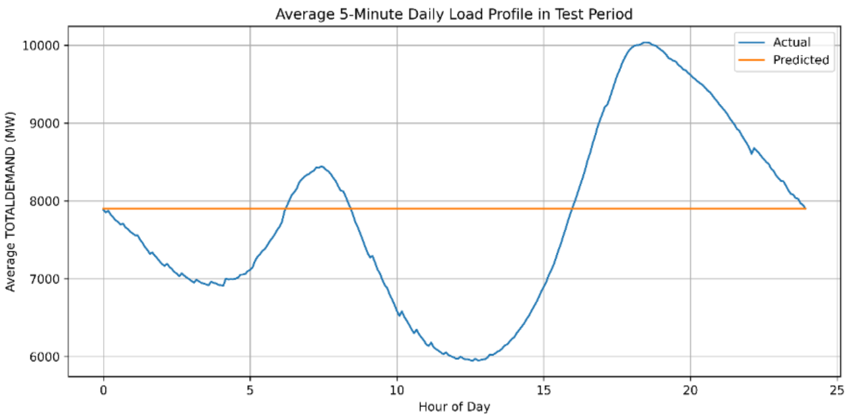
As shown in Figure 7(d), LSTM performs poorly under peak-load conditions. Its predictions remain almost constant and substantially below the actual top-10% load values. The model therefore fails to respond to either the changing level or the short-term spikes of the peak-demand sequence, which is consistent with its weak overall performance in Table 4 and its high horizon-wise errors in Figure 6.

The standalone iTransformer in Figure 7(e) exhibits a marked improvement over the conventional deep-learning models. Its predictions generally remain within the high-load operating range and reproduce many of the temporal fluctuations more effectively. Nevertheless, several local peaks are still missed or inaccurately estimated, showing that the Transformer backbone alone does not completely capture all extreme-load variations.

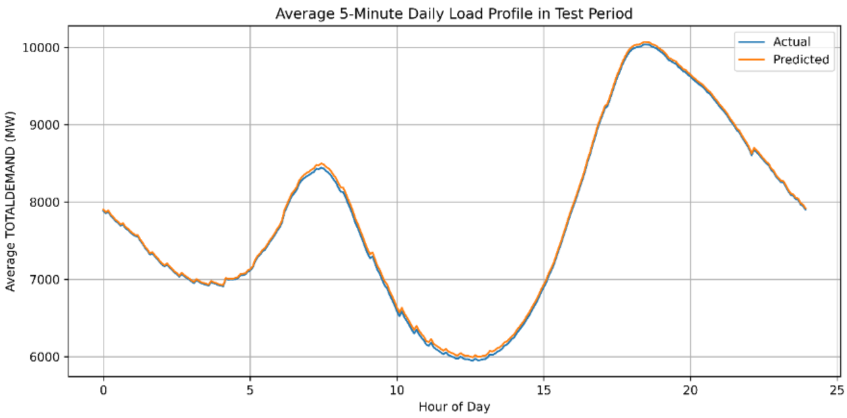
Finally, Figure 7(f) shows that DME-iTransformer provides the closest overall correspondence with the actual peak-load sequence. The predicted values generally remain within the appropriate high-demand range and follow many of the observed changes more closely than the other models. Although some individual extreme peaks are still underestimated or overestimated, the proposed model reduces the persistent low bias observed in Persistence Naive, CNN, and LSTM while maintaining more stable peak tracking than the purely seasonal approach.

Overall, Figure 7 confirms that peak-load forecasting remains considerably more difficult than average operating-condition forecasting. However, the proposed DME-iTransformer shows the most balanced behavior across these high-demand observations by combining learned nonlinear forecasting with multi-scale temporal information. This result complements the overall accuracy findings in Table 4 and indicates that the advantage of the proposed method is also preserved under demanding peak-load conditions.

To examine whether the evaluated models can reproduce the typical intra-day demand structure, Figure 8 compares the average 5-min daily load profile of the actual observations with the corresponding model predictions over the test period. Unlike the previous figures, which focus on individual trajectories, residuals, and forecast horizons, this analysis emphasizes the ability of each method to represent the characteristic daily load cycle, including the early-morning decline, morning peak, midday valley, afternoon ramp, and evening peak.

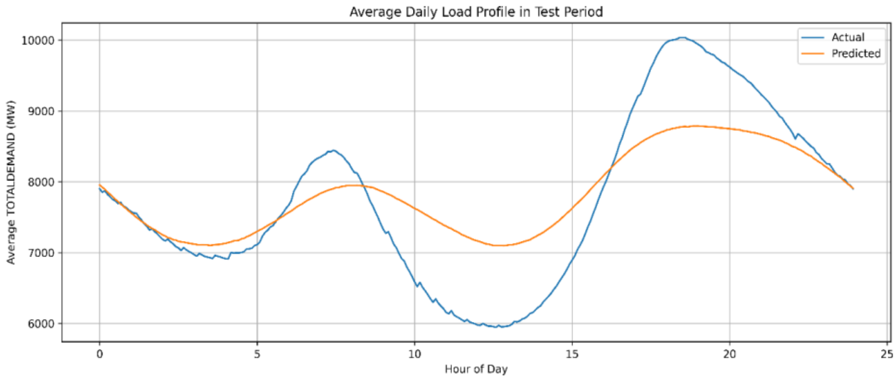


(a) Persistence Naive

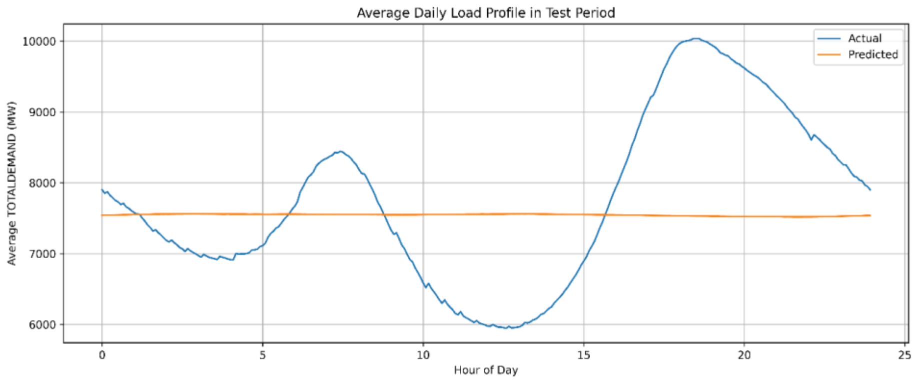


(b) Seasonal Naive

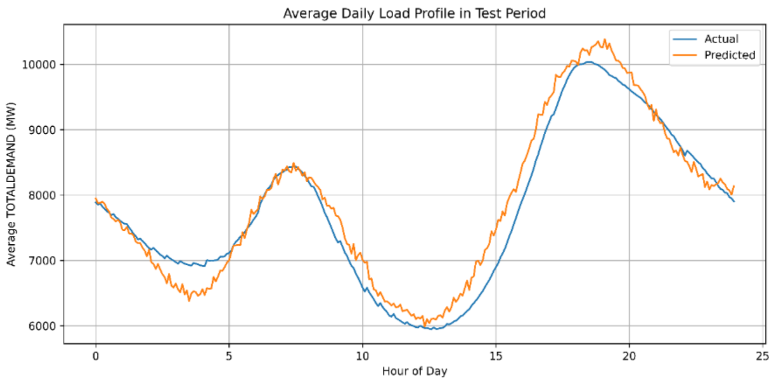
Vol. 8, No. 4, August 2026, pp. 1259-1298



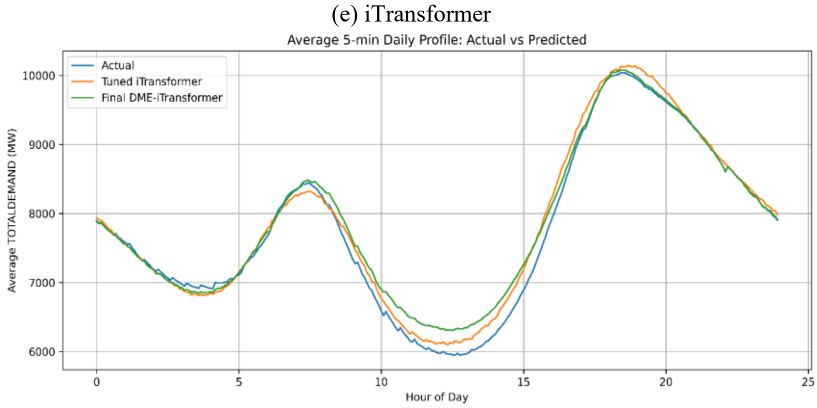
(c) CNN



(d) LSTM



(e) iTransformer



(f) DME-iTransformer

Figure 8. Average 5-min daily load profiles of the actual demand and the six forecasting models: (a) Persistence Naive, (b) Seasonal Naive, (c) CNN, (d) LSTM, (e) iTransformer, and (f) DME-iTransformer

Figure 8(a) shows that Persistence Naive produces an almost constant average profile and therefore fails to reproduce the pronounced intra-day structure of the actual demand. The morning peak, midday decline, and dominant evening peak are almost entirely absent from the prediction. This result further illustrates why a simple persistence mechanism is inadequate for high-resolution day-ahead forecasting when substantial changes occur within the daily cycle.

Seasonal Naive in Figure 8(b) reproduces the average daily profile with remarkable accuracy. The predicted curve closely follows the actual trajectory across nearly the entire 24-h period, including the morning peak, midday valley, afternoon ramp, and evening maximum. This close agreement confirms the strong daily periodicity of the NSW1 demand series and explains why Seasonal Naive performs as the strongest baseline in Table 4. Nevertheless, the average-profile agreement should be interpreted together with the earlier results, since averaging can conceal forecasting errors occurring on individual days.

The CNN result in Figure 8(c) captures the broad shape of the daily cycle but considerably smooths its main features. The morning peak is underestimated, the midday valley is predicted too high, and the evening peak is also substantially reduced. This behavior is consistent with the smoothing effect observed in the seven-day trajectories and indicates that the CNN learns the overall daily pattern but has difficulty preserving the full amplitude of intra-day demand variations.

A more pronounced limitation is observed for LSTM in Figure 8(d). The predicted average profile remains nearly flat throughout the day and does not reproduce the major temporal structures of the actual load. The inability to represent the morning rise, midday decline, and evening peak is consistent with the weak point-wise and overall performance previously observed for this model.

The standalone iTransformer in Figure 8(e) provides a substantially better representation of the average daily demand cycle. It captures the main turning points and follows the actual curve throughout most of the day. Some deviations remain, particularly around the early-morning valley and the timing and magnitude of the evening peak, where the predicted profile tends to shift slightly relative to the actual demand.

Finally, Figure 8(f) shows that DME-iTransformer preserves the principal daily structure while providing a closer alignment with the actual average profile. The proposed forecast follows the early-morning decline, morning peak, midday valley, afternoon increase, and evening peak with good consistency. Compared with the standalone iTransformer, the final DME-iTransformer curve better adjusts several parts of the daily cycle, particularly around the morning transition and midday-to-afternoon recovery. Some small deviations remain near the valley and peak regions, but the overall profile remains closely aligned with the actual demand.

Overall, Figure 8 confirms that the models differ substantially in their ability to represent the characteristic intra-day load shape. Persistence Naive and LSTM fail to reproduce the daily cycle, whereas CNN captures only a smoothed approximation. Seasonal Naive demonstrates the strength of explicit periodic information, while iTransformer provides a more flexible learned representation of daily dynamics. DME-iTransformer combines these complementary advantages and retains the characteristic load structure while improving overall forecasting accuracy, as demonstrated quantitatively in Table 4.

5.4. Internal Component-Wise Performance Analysis

To examine how forecasting performance changes across the main internal configurations of the proposed framework, Figure 9 compares the test MAPE obtained from the tuned iTransformer backbone, candidate-based alternatives, optimized fusion strategies, SAFE-SELECT output, and final forecast. In contrast to the preceding figures, which compare DME-iTransformer with external benchmark models, this analysis focuses exclusively on the internal development of the proposed method. The comparison illustrates how the forecasting performance changes when moving from the tuned iTransformer backbone to candidate-based forecasting, global fusion, load-state adaptation, state–horizon weighting, and the final SAFE-SELECT output.

Figure 9 first shows that the tuned iTransformer provides a competitive starting point but has the highest MAPE among the internal variants considered. The Best Validation Single strategy achieves a noticeable reduction, indicating that some multi-scale lag or profile-based candidates contain useful information that is not fully captured by the Transformer backbone alone. This finding supports the motivation for constructing a diverse candidate pool rather than relying exclusively on a single learned forecasting model.

The Global Optimized strategy does not outperform the best individual validation candidate. This result is important because it demonstrates that combining multiple candidates through a single global weight vector does not automatically improve forecasting accuracy. A fixed ensemble may be unable to account for changes in demand regime and forecast lead time, particularly when different candidates become more informative under different operating conditions.

A clearer improvement is observed after introducing load-state-dependent optimization. The State Optimized strategy reduces MAPE relative to the global ensemble, suggesting that candidate relevance changes between low-, normal-, and high-load conditions. The State-Horizon Optimized strategy further improves performance by allowing the ensemble weights to adapt jointly to the predicted load state and to different segments of the 288-step forecasting horizon. This indicates that the usefulness of individual forecasting candidates is not constant throughout the day-ahead prediction period.

The Selected Strategy PreBias configuration achieves the lowest or nearly lowest MAPE among the internal alternatives. This result highlights the role of the SAFE-SELECT mechanism, which does not assume that the most complex ensemble structure must always be the best. Instead, the final strategy is selected according to validation performance, allowing the framework to retain a simpler alternative whenever a more complex configuration fails to provide a clear advantage.

The Final DME-iTransformer maintains performance very close to the best internal strategy and achieves an overall test MAPE of approximately 7.03%. The small difference between the pre-bias selected strategy and the final output indicates that the main performance gain originates from candidate construction, adaptive state–horizon fusion, and validation-based strategy selection rather than from the final residual-adjustment step alone. In other words, the improvement is produced by the overall coordination of the framework rather than by a single post-processing operation.

Overall, Figure 9 demonstrates that the proposed method benefits most from moving beyond a fixed global ensemble toward context-dependent weighting and validation-based strategy selection. The results also show that performance improvement is not strictly monotonic across all intermediate configurations, which confirms the importance of SAFE-SELECT. More complex fusion is retained only when it provides evidence of better validation performance. This design helps prevent unnecessary ensemble complexity and supports the robustness of the final DME-iTransformer.

To clarify the contribution of the candidate-generation stage, Figure 10 compares the validation MAPE of the main forecasting candidates before the final ensemble optimization. The candidate pool includes individual lag-based forecasts, calendar-conditioned load profiles, multi-scale blends, the tuned iTransformer, and hybrid combinations between iTransformer and historical temporal references. Since lower MAPE indicates better performance, the figure reveals which forecasting mechanisms provide the most accurate individual predictions and which sources may offer complementary information for the subsequent candidate-selection and dynamic-fusion stages.

As shown in Figure 10, the lowest validation errors are obtained by several multi-source temporal candidates, particularly profile_lag_blend, recent_weekly_dynamic_blend, and robust_triplet_median. These candidates combine information from recent daily behavior, weekly recurrence, and load-profile structure rather than relying on a single historical reference. Their favorable performance indicates that the NSW1 demand series benefits from jointly representing short-term continuity and repeated multi-scale temporal patterns.

The profile_dynamic candidate also achieves competitive validation accuracy. Compared with fixed profile estimates such as profile_static and profile_dow_hm, the dynamic profile adjusts the historical load shape according to the recent demand level. Its lower validation error therefore suggests that adapting recurring profiles to current operating conditions is more effective than using fixed seasonal averages alone.

The tuned iTransformer and its hybrid variants occupy the middle part of the ranking. In particular, itr_recent_blend, itr_profile_blend, itr_weekly_blend, and itr_lag1_blend remain competitive with the standalone itr_tuned forecast. This result indicates that the Transformer representation is useful but can be further complemented by explicit historical references. At the same time, no single iTransformer-based candidate dominates all other alternatives, which supports the use of an ensemble rather than relying exclusively on the neural backbone.

In contrast, individual candidates such as lag_7day, profile_dow_hm, and profile_static exhibit higher validation MAPE values. Although these candidates contain meaningful seasonal information, their standalone performance is more limited because they cannot fully adapt to changes in demand level and short-term operating conditions. Nevertheless, such candidates may still contribute useful diversity when their errors are not excessively correlated with those of stronger predictors.

An important point is that Figure 10 represents individual validation accuracy only. The final candidate set is not selected solely according to MAPE ranking. The proposed framework also applies correlation-based pruning to remove highly redundant forecasts and explicitly preserves the tuned iTransformer representation. Therefore, a candidate with slightly higher individual MAPE may still be valuable if it provides complementary information that improves the ensemble forecast.

Overall, Figure 10 demonstrates that the strongest individual candidates arise from different forecasting mechanisms rather than from a single model family. The competitive performance of multi-scale lag/profile combinations, dynamic profiles, and iTransformer-based hybrids justifies the construction of a diverse candidate pool. This diversity provides the basis for the subsequent global, load-state, and state–horizon weight optimization stages of the proposed DME-iTransformer.

The analyses in Figure 9 and Figure 10 provide component-wise diagnostic evidence but do not constitute a fully controlled ablation in which each module is independently removed while all remaining configurations are held constant. Therefore, the separate causal contributions of the dynamic profile, correlation-based pruning, SAFE-SELECT, and residual-bias correction should not be inferred beyond the internal variants explicitly reported. A complete module-by-module ablation is reserved for future work.

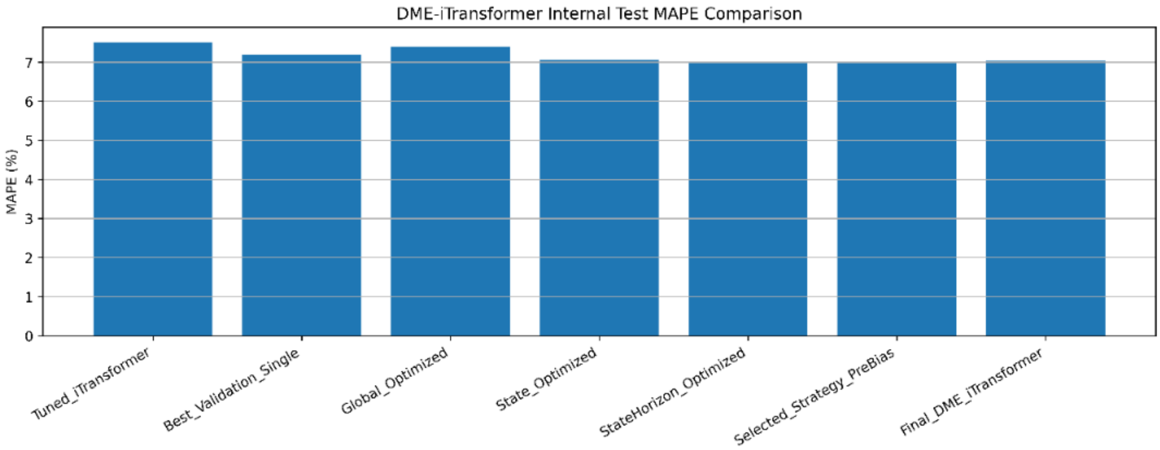


Figure 9. Progressive MAPE changes across the main internal configurations of the proposed DME-iTransformer framework

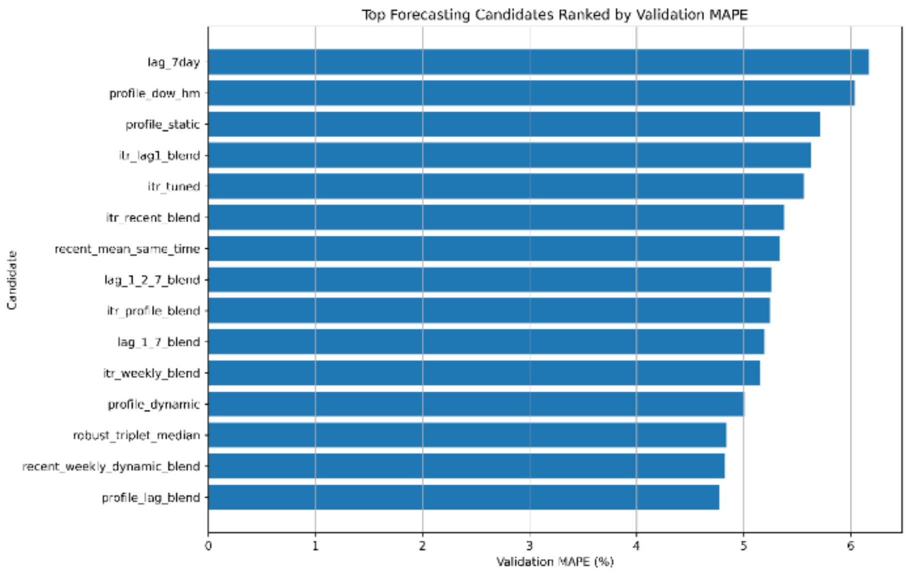


Figure 10. Validation MAPE ranking of the main forecasting candidates

5.5. Adaptive Weight Interpretation

To investigate how the proposed ensemble adapts its forecasting mechanism to different operating conditions, Figure 11 visualizes the optimized weights assigned to the selected candidates under global, load-state, and state–horizon contexts. The color intensity represents the relative contribution of each candidate, with brighter cells indicating larger optimized weights. Unlike the previous analyses, which focused on forecasting accuracy, this figure provides insight into the internal decision structure of DME-iTransformer and shows whether the importance of individual forecasting sources remains fixed or changes according to load conditions and forecast lead time.

At the global level, the ensemble distributes its weights across several candidates rather than relying on a single predictor. In particular, profile_lag_blend, recent_weekly_dynamic_blend, robust_triplet_median, and profile_dynamic receive visible contributions. This distributed weighting indicates that the overall forecast benefits from combining recent load behavior, weekly recurrence, robust multi-source information, and dynamic profile structure.

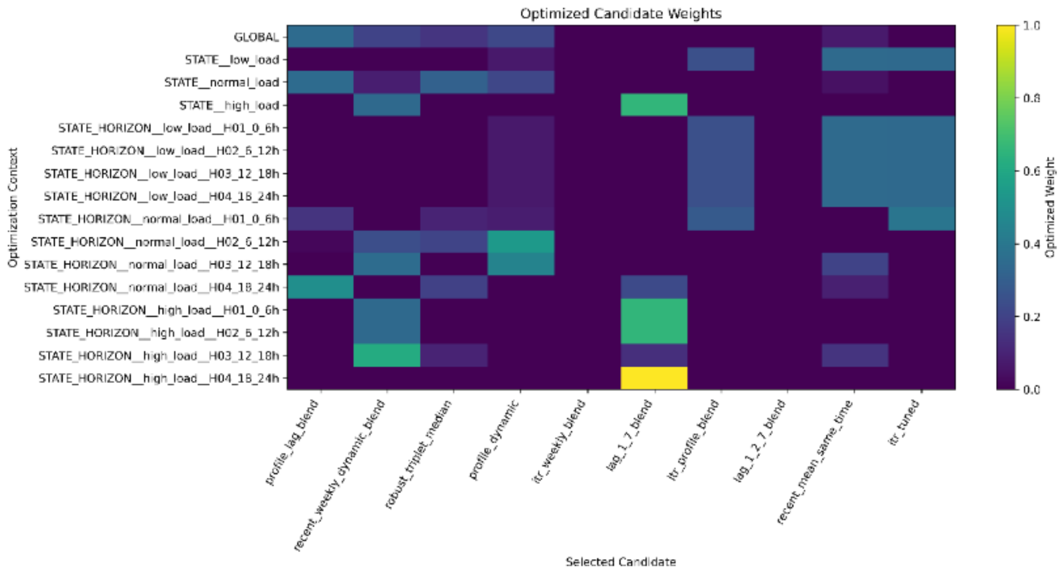


Figure 11. Optimized candidate weights across global, load-state, and state–horizon forecasting conditions

The state-specific rows reveal a clear change in candidate importance across different demand regimes. Under low-load conditions, greater emphasis is placed on iTransformer-related and recent-history candidates, particularly itr_profile_blend, recent_mean_same_time, and itr_tuned. Under normal-load conditions, the weights are spread more broadly across profile- and lag-based candidates. In contrast, high-load conditions are dominated mainly by lag_1_7_blend and recent_weekly_dynamic_blend, indicating that recent daily and weekly recurrence becomes especially informative when forecasting elevated demand levels.

A more detailed adaptive pattern appears in the state–horizon rows. For low-load periods, the model consistently assigns substantial weight to itr_tuned, recent_mean_same_time, and itr_profile_blend across several horizon blocks. This suggests that the combination of the learned Transformer forecast and recent same-time behavior is particularly useful for representing low-demand periods throughout the day-ahead trajectory.

For normal-load conditions, the weighting pattern changes with the forecast horizon. profile_dynamic becomes more influential during several intermediate horizon blocks, while profile_lag_blend gains importance in later periods. This transition indicates that the ensemble does not treat the entire 288-step forecast horizon uniformly; instead, it shifts between profile-based and recent-history-based predictors as the forecast progresses.

The most pronounced specialization occurs under high-load conditions. The ensemble assigns large weights to recent_weekly_dynamic_blend and lag_1_7_blend, with the latter becoming dominant in some horizon blocks and receiving an almost exclusive contribution in the final high-load horizon segment. This behavior suggests that the most recent daily and weekly demand patterns provide particularly strong information for forecasting high-load periods, especially at specific lead times.

Another important observation is that several candidates receive near-zero weights in many contexts. This sparsity shows that the optimization procedure does not simply average all retained candidates. Instead, it selectively activates only the most useful forecasting sources for each operating condition. Therefore, the role of the candidate-selection stage is complemented by a second level of adaptation in which the contribution of the retained candidates changes dynamically.

Overall, Figure 11 provides direct evidence for the adaptive nature of the proposed DME-iTransformer. The optimized weights vary substantially across global, load-state, and state–horizon contexts, confirming that no single candidate is consistently optimal under all conditions. The framework therefore improves forecasting performance by dynamically selecting different combinations of temporal profiles, lag structures, and iTransformer-based forecasts according to the current demand regime and forecast position. This adaptive weighting behavior helps explain the performance improvements observed in Table 4 and the preceding figures.

The present study did not record standardized wall-clock training time, per-window inference latency, peak GPU-memory consumption, or repeated timing measurements under a controlled runtime. Consequently, the results in this subsection should be interpreted exclusively as evidence of adaptive weight allocation and not as evidence of computational efficiency. The practical computational cost of candidate generation, correlation screening, local ensemble optimization, SAFE-SELECT, and residual-bias correction remains to be quantified in future work under fixed hardware and software conditions.

5.6. Scope and Limitations of the Experimental Evaluation

The experimental evidence should be interpreted within several boundaries. First, the benchmark set contains Persistence Naive, Seasonal Naive, CNN, LSTM, standalone iTransformer, and the proposed DME-iTransformer, but it does not include additional recent Transformer architectures. Second, the performance differences are based on descriptive aggregate metrics and were not subjected to formal statistical significance testing. Third, although internal forecasting variants and candidate rankings are reported, a fully controlled module-by-module ablation was not conducted. Fourth, standardized computational time, inference latency, and memory consumption were not measured. Fifth, the experiment was restricted to one regional load series and used historical load information only.

These limitations do not alter the numerical results obtained under the adopted NSW1 forecasting protocol, but they restrict the breadth of the conclusions. The findings demonstrate that DME-iTransformer outperformed the five evaluated baseline methods under the specified dataset, input length, forecasting horizon, and chronological partitions. They should not be interpreted as establishing universal superiority over all recent forecasting architectures, electricity systems, or computational environments.

6. CONCLUSION

This study proposed a Dynamic Multi-Scale Ensemble iTransformer (DME-iTransformer) framework for deterministic, high-resolution day-ahead electric load forecasting. The framework integrates a tuned iTransformer with daily, weekly, and multi-week lag information, static and dynamically adjusted historical load profiles, and hybrid Transformer–temporal candidates. Validation-based candidate ranking, correlation-based redundancy removal, context-dependent weight optimization, SAFE-SELECT strategy selection, and optional residual-bias correction were incorporated into a chronologically separated forecasting procedure. This design ensured that model configuration, candidate selection, weight estimation, and strategy calibration were completed without using information from the final test period.

The proposed framework achieved the best overall performance among the six evaluated methods on the NSW1 electricity-demand dataset recorded at 5-min intervals. DME-iTransformer obtained an MAE of 478.485 MW, an MSE of 476,259.933 MW^2, an RMSE of 690.116 MW, a MAPE of 7.031%, an sMAPE of 6.718%, and an R^2 of 0.8484. Compared with Seasonal Naive, the strongest conventional benchmark, the proposed method reduced MAPE by 11.88%. Relative to the standalone iTransformer, it reduced MAPE by 19.10%. The simultaneous improvement in absolute, squared, relative, and goodness-of-fit measures indicates that the proposed framework provides a consistent accuracy advantage rather than an improvement restricted to one evaluation metric.

The additional analyses support three main findings. First, explicit temporal recurrence remains highly informative for high-resolution load forecasting, as demonstrated by the competitive performance of Seasonal Naive and the strong validation results of lag- and profile-based candidates. Second, the standalone iTransformer captures nonlinear demand dynamics effectively but does not explicitly preserve all daily, weekly, and multi-week references required for the forecasting task. Third, candidate relevance changes across load states and forecast-horizon segments. The optimized-weight analysis showed that no individual forecasting source was dominant under all conditions, thereby justifying the use of context-dependent fusion.

The principal methodological contribution of this study is the integration of learned nonlinear forecasting and explicit multi-scale temporal recurrence within a validation-driven ensemble framework. Unlike fixed-weight ensembles, the proposed method allows candidate contributions to vary according to predicted load conditions and positions within the 288-step day-ahead horizon. In addition, SAFE-SELECT prevents the automatic adoption of unnecessary ensemble complexity by retaining a simpler forecasting strategy whenever it performs better on independent validation data. The results therefore contribute new evidence that adaptive combination and complexity control are both important when integrating Transformer forecasts with deterministic temporal references.

From a practical perspective, DME-iTransformer provided more accurate tracking of daily demand trajectories, a narrower residual distribution, improved representation of the average intraday load profile, and more balanced performance during high-load conditions. However, forecasting errors remained higher in the middle portion of the day-ahead horizon, indicating that rapid intraday transitions and peak-related variations remain difficult to predict. The framework should therefore be regarded as an improved deterministic forecasting approach rather than a complete solution for all operating conditions.

The present study has several limitations. First, the experiments were restricted to a single regional electricity-demand series, and the model inputs did not include weather conditions, holidays, electricity prices, renewable-generation information, or other external variables. Second, the benchmark set did not include additional recent Transformer architectures, and the reported conclusions are therefore limited to the six evaluated methods. Third, the observed performance differences were assessed using descriptive forecasting metrics without formal statistical significance testing. Fourth, the internal component comparisons do not constitute a fully controlled module-by-module ablation. Fifth, standardized training time, inference latency, and memory consumption were not systematically recorded. Finally, the study produced point forecasts only and did not generate calibrated prediction intervals or other probabilistic outputs. These limitations restrict the generalization of the findings beyond deterministic, load-only day-ahead forecasting for the NSW1 dataset under the adopted experimental protocol.

Future research should evaluate the proposed framework across multiple regions, temporal resolutions, seasons, and electricity systems. Broader benchmarking should include recent Transformer-based architectures such as PatchTST, TimesNet, FEDformer, and Crossformer under the same data partitions and forecasting settings. Formal paired statistical tests and confidence-interval analyses should be performed across daily forecasting windows. A controlled ablation study should independently remove the dynamic profile, candidate-screening mechanism, state–horizon fusion, SAFE-SELECT, and residual-bias correction to quantify their individual contributions. Standardized computational profiling should additionally report training time, inference latency, post-processing time, and memory consumption. Other extensions include the incorporation of carefully selected exogenous variables, probabilistic forecasting with calibrated prediction intervals, online candidate adaptation, and concept-drift detection.

REFERENCES

[1] D. Kiruthiga and V. Manikandan, "Levy flight-particle swarm optimization-assisted BiLSTM + dropout deep learning model for short-term load forecasting," *Neural Comput. Appl.*, vol. 35, no. 3, pp. 2679–2700, 2023, https://doi.org/10.1007/s00521-022-07751-y.

[2] W. Lin, D. Wu, and M. Jenkin, "Electric Load Forecasting for Individual Households via Spatial-temporal Knowledge Distillation," *IEEE Trans. Power Syst.*, vol. 40, no. 1, pp. 572–584, 2024, https://doi.org/10.1109/TPWRS.2024.3393926.

[3] S. Islam, T. M. R. Bashar, and M. Munem, "Enhanced very short-term load forecasting with multi-lag feature engineering and prophet-XGBoost-CatBoost architecture," *Energy*, vol. 335, no. June, p. 137981, 2025, https://doi.org/10.1016/j.energy.2025.137981.

[4] M. Pleños, "Time Series Forecasting Using Holt-Winters Exponential Smoothing: Application to Abaca Fiber Data," *Zesz. Nauk. SGGW w Warszawie - Probl. Rol. Światowego*, vol. 22, no. 2, pp. 17–29, 2022, https://doi.org/10.22630/prs.2022.22.2.6.

[5] S. Smyl, G. Dudek, and P. Pełka, "Contextually enhanced ES-dRNN with dynamic attention for short-term load forecasting," *Neural Networks*, vol. 169, no. December 2022, pp. 660–672, 2024, https://doi.org/10.1016/j.neunet.2023.11.017.

[6] S. Smyl, G. Dudek, and P. Pełka, "Contextually enhanced ES-dRNN with dynamic attention for short-term load forecasting," *Neural Networks*, vol. 169, no. September 2023, pp. 660–672, 2024, https://doi.org/10.1016/j.neunet.2023.11.017.

[7] D. Liu, X. Lin, H. Liu, and J. Zhu, "A coupled framework for power load forecasting with Gaussian implicit spatio temporal block and attention mechanisms network," *Comput. Electr. Eng.*, vol. 123, no. PD, p. 110263, 2025, https://doi.org/10.1016/j.compeleceng.2025.110263.

[8] G. Tziolis *et al.*, "Direct short-term net load forecasting in renewable integrated microgrids using machine learning: A comparative assessment," *Sustain. Energy, Grids Networks*, vol. 37, no. December 2023, p. 101256, 2024, https://doi.org/10.1016/j.segan.2023.101256.

[9] F. Gao, J. Song, and X. Shao, "Short-term interval-valued load forecasting with a combined strategy of iHW and multioutput machine learning," *Ann. Oper. Res.*, vol. 346, no. 3, pp. 2009–2033, 2025, https://doi.org/10.1007/s10479-024-06446-y.

[10] M. S. Iqbal and M. Adnan, "Energy & Buildings Edge computing and transfer learning-based short-term load forecasting for residential and commercial buildings," *Energy Build.*, vol. 329, p. 115273, 2025, https://doi.org/10.1016/j.enbuild.2025.115273.

[11] M. Heleno, W. Zhang, K. Sun, L. R. Garcia, and T. Hong, "Energy & Buildings A cross-dimensional analysis of data-driven short-term load forecasting methods with large-scale smart meter data," *Energy Build.*, vol. 344, p. 115909, 2025, https://doi.org/10.1016/j.enbuild.2025.115909.

[12] L. Xiao, Q. Bai, and B. Wang, "A dynamic multi-model transfer based short-term load forecasting," *Appl. Soft Comput.*, vol. 159, p. 111627, 2024, https://doi.org/10.1016/j.asoc.2024.111627.

[13] T. H. M. S. Rashid *et al.*, "Energy Conversion and Management : X A data-driven framework for seasonal PV forecasting and coordinated EV charging to improve hosting capacity in unbalanced distribution grids," *Energy Convers. Manag. X*, vol. 30, p. 101708, 2026, https://doi.org/10.1016/j.ecmx.2026.101708.

[14] P. Aduama, A. S. Al-sumaiti, V. K. Saini, and R. Kong, "Energy and AI Explainable optimized voting ensemble for photovoltaic power forecasting," *Energy AI*, vol. 24, no. January, p. 100690, 2026, https://doi.org/10.1016/j.egyai.2026.100690.

[15] A. Sharma and S. K. Jain, "A Novel Two-Stage Framework for Mid-Term Electric Load Forecasting," *IEEE Trans. Ind. Informatics*, vol. 20, no. 1, pp. 247–255, 2024, https://doi.org/10.1109/TII.2023.3259445.

[16] Z. Zhang, C. Zhang, Y. Dong, and W. Hong, "Knowledge-Based Systems Bi-directional gated recurrent unit enhanced twin support vector regression with seasonal mechanism for electric load forecasting," *Knowledge-Based Syst.*, vol. 310, p. 112943, 2025, https://doi.org/10.1016/j.knosys.2024.112943.

[17] G. Duan, Y. Du, Y. Shang, H. Xue, and R. Zhang, "Research on Support Vector Regression Short-Time Traffic Flow Prediction Model for Secondary Roads Based on Associated Road Analysis," *Appl. Sci.*, vol. 15, no. 4, 2025, https://doi.org/10.3390/app15041779.

[18] R. Rodríguez-Pérez and J. Bajorath, "Evolution of Support Vector Machine and Regression Modeling in Chemoinformatics and Drug Discovery," *J. Comput. Aided. Mol. Des.*, vol. 36, no. 5, pp. 355–362, 2022, https://doi.org/10.1007/s10822-022-00442-9.

[19] Y. Lu and G. Wang, "A load forecasting model based on support vector regression with whale optimization algorithm," *Multimed. Tools Appl.*, vol. 82, no. 7, pp. 9939–9959, 2023, https://doi.org/10.1007/s11042-022-13462-2.

[20] J. Luo, Y. Zheng, T. Hong, A. Luo, and X. Yang, "Fuzzy support vector regressions for short-term load forecasting," *Fuzzy Optim. Decis. Mak.*, vol. 23, no. 3, pp. 363–385, 2024, https://doi.org/10.1007/s10700-024-09425-x.

[21] N. Tsalikidis, A. Mystakidis, C. Tjortjis, P. Koukaras, and D. Ioannidis, "Energy load forecasting: one-step ahead hybrid model utilizing ensembling: N. Tsalikidis et al.," *Computing*, vol. 106, no. 1, pp. 241-273, 2024. https://doi.org/10.1007/s00607-023-01217-2.

[22] M. Dostmohammadi, M. Z. Pedram, S. Hoseinzadeh, and D. A. Garcia, "A GA-stacking ensemble approach for forecasting energy consumption in a smart household: A comparative study of ensemble methods," *J. Environ. Manage.*, vol. 364, p. 121264, 2024, https://doi.org/10.1016/j.jenvman.2024.121264.

[23] A. Lindholm, N. Wahlström, F. Lindsten, and T. B. Schön, "Machine Learning," *Mach. Learn.*, 2022, https://doi.org/10.1017/9781108919371.

[24] R. Bareth, M. Kochar, and A. Yadav, "Comparative Analysis of different Machine learning Models for Load Forecasting," *2023 IEEE IAS Glob. Conf. Renew. Energy Hydrog. Technol. GlobConHT 2023*, pp. 1–5, 2023, https://doi.org/10.1109/GlobConHT56829.2023.10087406.

[25] Y. Chen, C. Lin, Y. Zhang, J. Liu, and D. Yu, "Day-ahead load forecast based on Conv2D-GRU_SC aimed to adapt to steep changes in load," *Energy*, vol. 302, p. 131814, 2024, https://doi.org/10.1016/j.energy.2024.131814.

[26] M. Goyal and M. Pandey, "A Systematic Analysis for Energy Performance Predictions in Residential Buildings Using Ensemble Learning," *Arab. J. Sci. Eng.*, vol. 46, no. 4, pp. 3155-3168, 2020, https://doi.org/10.1007/s13369-020-05069-2.

[27] H. Osman, A. Azab, A. Alghazi, S. Duffuaa, and F. Baki, "Results in Engineering Forecasting electric vehicle charging loads using random forest and gene expression programming ensemble models," *Results Eng.*, vol. 28, p. 108369, 2025, https://doi.org/10.1016/j.rineng.2025.108369.

[28] V. Veeramsetty, K. R. Reddy, M. Santhosh, A. Mohnot, and G. Singal, "Short-term electric power load forecasting using random forest and gated recurrent unit," *Electr. Eng.*, vol. 104, no. 1, pp. 307–329, 2022, https://doi.org/10.1007/s00202-021-01376-5.

[29] M. Mohamed, F. E. Mahmood, M. A. Abd, M. Rezkallah, and A. Hamadi, "Load Demand Forecasting Using eXtreme Gradient Boosting (XGboost)," *2023 IEEE Ind. Appl. Soc. Annu. Meet.*, pp. 1–7, 2023, https://doi.org/10.1109/IAS54024.2023.10406613.

[30] P. Saini and S. K. Parida, "A novel probabilistic gradient boosting model with multi-approach feature selection and iterative seasonal trend decomposition for short-term load forecasting," *Energy*, vol. 294, p. 130975, 2024, https://doi.org/10.1016/j.energy.2024.130975.

[31] Z. Qinghe, X. Wen, H. Boyan, W. Jong, and F. Junlong, "Optimised extreme gradient boosting model for short term electric load demand forecasting of regional grid system," *Sci. Rep.*, vol. 12, no. 1, pp. 1–12, 2022, https://doi.org/10.1038/s41598-022-22024-3.

[32] P. Saini and S. K. Parida, "A novel probabilistic gradient boosting model with multi-approach feature selection and iterative seasonal trend decomposition for short-term load forecasting," *Energy*, vol. 294, p. 130975, 2024, https://doi.org/10.1016/j.energy.2024.130975.

[33] Z. Li *et al.*, "Fuel economy and emission optimization of spark-ignition pure methanol engine based on gradient boosting neural network and many-objective Bayesian optimization," *Fuel*, vol. 426, p. 139573, 2026, https://doi.org/10.1016/j.fuel.2026.139573.

[34] Z. Song, F. Xiao, Z. Chen, and H. Madsen, "Energy and AI Probabilistic ultra-short-term solar photovoltaic power forecasting using natural gradient boosting with attention-enhanced neural networks," *Energy AI*, vol. 20, no. March, p. 100496, 2025, https://doi.org/10.1016/j.egyai.2025.100496.

[35] S. Lei and F. Wang, "A Two-stage Short-term Load Forecasting Method Based on Comprehensive Similarity Day Selection and CEEMDAN-XGBoost Error Correction," *2023 Int. Conf. Futur. Energy Solut.*, pp. 1–6, https://doi.org/10.1109/FES57669.2023.10183307.

[36] L. Zhang and D. Jánošík, "Enhanced short-term load forecasting with hybrid machine learning models: CatBoost and XGBoost approaches," *Expert Syst. Appl.*, vol. 241, 2024, https://doi.org/10.1016/j.eswa.2023.122686.

[37] Q. Yang, Y. Lin, S. Kuang, and D. Wang, "A novel short-term load forecasting approach for data-poor areas based on K-MIFS-XGBoost and transfer-learning," *Electr. Power Syst. Res.*, vol. 229, p. 110151, 2024, https://doi.org/10.1016/j.epsr.2024.110151.

[38] R. Banik and A. Biswas, "Enhanced renewable power and load forecasting using RF-XGBoost stacked ensemble," *Electr. Eng.*, vol. 106, no. 4, pp. 4947–4967, 2024, https://doi.org/10.1007/s00202-024-02273-3.

[39] Z. Wang, L. Chen, and C. Wang, "Weekly periodicity feature-driven CNN-GRU and LightGBM hybrid model for short-term multi-step building load forecasting," *Energy*, vol. 355, p. 141163, 2026, https://doi.org/10.1016/j.energy.2026.141163.

[40] H. Auto-tuning, S. Lei, and F. Wang, "A Short-term Net Load Forecasting Method Based on Two-stage Feature Selection and LightGBM with," *2023 IEEE/IAS 59th Ind. Commer. Power Syst. Tech. Conf.*, pp. 1–6, https://doi.org/10.1109/ICPS57144.2023.10142095.

[41] L. Yang, Q. Duan, B. Pan, and Z. Wang, "Post-pandemic holiday load forecasting considering social factors influences based on TL-Informer-LightGBM model," *Electr. Power Syst. Res.*, vol. 247, p. 111861, 2025, https://doi.org/10.1016/j.epsr.2025.111861.

[42] X. Yao, X. Fu, and C. Zong, "Short-Term Load Forecasting Method Based on Feature Preference Strategy and LightGBM-XGboost," *IEEE Access*, vol. 10, pp. 75257–75268, 2022, https://doi.org/10.1109/ACCESS.2022.3192011.

[43] B. Predić *et al.*, "Cloud-load forecasting via decomposition-aided attention recurrent neural network tuned by modified particle swarm optimization," *Complex Intell. Syst.*, vol. 10, no. 2, pp. 2249–2269, 2024, https://doi.org/10.1007/s40747-023-01265-3.

[44] W. Waheed, Q. Xu, M. Aurangzeb, S. Iqbal, S. Hanif, and Z. M. S. Elbarbary, "Heliyon Empowering data-driven load forecasting by leveraging long short-term memory recurrent neural networks," *Heliyon*, vol. 10, no. 24, p. e40934, 2024, https://doi.org/10.1016/j.heliyon.2024.e40934.

[45] X. He, W. Zhao, Z. Gao, Q. Zhang, and W. Wang, "A hybrid prediction interval model for short-term electric load forecast using Holt-Winters and Gate Recurrent Unit," *Sustain. Energy, Grids Networks*, vol. 38, p. 101343, 2024, https://doi.org/10.1016/j.segan.2024.101343.

[46] Q. Shen, L. Mo, G. Liu, J. Zhou, Y. Zhang, and P. Ren, "Short-Term Load Forecasting Based on Multi-Scale Ensemble Deep Learning Neural Network," *IEEE Access*, vol. 11, pp. 111963–111975, 2023, https://doi.org/10.1109/ACCESS.2023.3322167.

[47] V. Gundu and S. P. Simon, "Quality analysis of combined similar day and day ahead short-term load forecasting using recurrent neural networks," *Sadhana - Acad. Proc. Eng. Sci.*, vol. 47, no. 1, 2022, https://doi.org/10.1007/s12046-022-01825-2.

[48] U. Javed *et al.*, "A novel short receptive field based dilated causal convolutional network integrated with Bidirectional LSTM for short-term load forecasting," *Expert Syst. Appl.*, vol. 205, no. February, p. 117689, 2022, https://doi.org/10.1016/j.eswa.2022.117689.

[49] P. Zeng, L. Qi, C. Hu, and X. Ye, "A high-efficiency voltage stabilization strategy for energy harvesting systems integrating a CNN – LSTM prediction model and a buck – boost closed-loop circuit," *Electr. Power Syst. Res.*, vol. 259, p. 113174, 2026, https://doi.org/10.1016/j.epsr.2026.113174.

[50] W. Xiong, "Bus load forecasting based on maximum information coefficient and CNN-LSTM model," *2023 IEEE Int. Conf. Image Process. Comput. Appl.*, pp. 659–663, 2023, https://doi.org/10.1109/ICIPCA59209.2023.10257944.

[51] J. Zhu, J. Yang, M. Peng, X. Cui, and X. Liang, "A CNN-LSTM-attention (CLA) hybrid model with PID-based search algorithm (PSA) for heat load forecasting in district heating system," *J. Build. Eng.*, vol. 111, no. 135, p. 113325, 2025, https://doi.org/10.1016/j.jobe.2025.113325.

[52] Z. Chen, Y. Wang, L. Wang, and R. Gong, "A CNN-LSTM-SE model for the short-term roll and heave forecasting of a," *Ocean Eng.*, vol. 350, p. 124264, 2026, https://doi.org/10.1016/j.oceaneng.2026.124264.

[53] J. Dong, Y. Jiang, P. Chen, J. Li, Z. Wang, and S. Han, "International Journal of Electrical Power and Energy Systems Short-term power load forecasting using bidirectional gated recurrent units-based adaptive stacked autoencoder," *Int. J. Electr. Power Energy Syst.*, vol. 165, p. 110459, 2025, https://doi.org/10.1016/j.ijepes.2025.110459.

[54] A. Sharma and S. K. Jain, "A Novel Two-Stage Framework for Mid-Term Electric Load Forecasting," in *IEEE Transactions on Industrial Informatics*, vol. 20, no. 1, pp. 247-255, 2024, https://doi.org/10.1109/TII.2023.3259445.

[55] X. Li, S. Wang, and Z. Chen, "Hierarchical reconciliation of convolutional gated recurrent units for unified forecasting of branched and aggregated district heating loads," *Energy*, vol. 313, p. 134097, 2024, https://doi.org/10.1016/j.energy.2024.134097.

[56] H. I. El-azab, R. A. Swief, N. H. El-amary, and H. K. Temraz, "Machine and deep learning approaches for forecasting electricity price and energy load assessment on real datasets," *Ain Shams Eng. J.*, vol. 15, no. 4, p. 102613, 2024, https://doi.org/10.1016/j.asej.2023.102613.

[57] X. Li, H. Guo, L. Xu, and Z. Xing, "Bayesian-Based Hyperparameter Optimization of 1D-CNN for Structural Anomaly Detection," *Sensors*, vol. 23, no. 11, 2023, https://doi.org/10.3390/s23115058.

[58] S. Zhang, R. Chen, J. Cao, and J. Tan, "A CNN and LSTM-based multi-task learning architecture for short and medium-term electricity load forecasting," *Electr. Power Syst. Res.*, vol. 222, no. October 2022, p. 109507, 2023, https://doi.org/10.1016/j.epsr.2023.109507.

[59] W. Zeng *et al.*, "Hybrid CEEMDAN-DBN-ELM for online DGA serials and transformer status forecasting," *Electr. Power Syst. Res.*, vol. 217, 2023, https://doi.org/10.1016/j.epsr.2023.109176.

[60] S. A. Alsamraee and S. Khanna, "A hybrid transformer – TCN – GRU based model for thermal load forecasting of a large university campus," *Energy*, vol. 344, p. 140114, 2026, https://doi.org/10.1016/j.energy.2026.140114.

[61] P. Ran, K. Dong, X. Liu, and J. Wang, "Short-term load forecasting based on CEEMDAN and Transformer," *Electr. Power Syst. Res.*, vol. 214, p. 108885, 2023, https://doi.org/10.1016/j.epsr.2022.108885.

[62] J. Chen, D. Wu, Q. Ma, Z. Zhao, Y. Wu, and Y. Ren, "ACTNet : Adapting CNN-Transformer-based foundation model for remote sensing change detection," *Pattern Recognition*, vol. 180, 2026, https://doi.org/10.1016/j.patcog.2026.114066.

[63] Z. Tian, W. Liu, W. Jiang, and C. Wu, "CNNs-Transformer based day-ahead probabilistic load forecasting for weekends with limited data availability," *Energy*, vol. 293, p. 130666, 2024, https://doi.org/10.1016/j.energy.2024.130666.

[64] I. U. Khalil, A. Ul Haq, and N. ul Islam, "A deep learning-based transformer model for photovoltaic fault forecasting and classification," *Electr. Power Syst. Res.*, vol. 228, p. 110063, 2024, https://doi.org/10.1016/j.epsr.2023.110063.

[65] Y. Zhou, L. Wang, J. Zhao, and W. Li, "A short-term photovoltaic power prediction method based on the BSLO dual-optimized VMD-CNN-iTransformer-LSTM combined model," *Measurement*, vol. 277, p. 121600, 2026, https://doi.org/10.1016/j.measurement.2026.121600.

[66] B. Fang, L. Xu, Y. Luo, Z. Luo, and W. Li, "A method for short-term electric load forecasting based on the FMLP-iTransformer model," *Energy Reports*, vol. 12, pp. 3405–3411, 2024, https://doi.org/10.1016/j.egyr.2024.09.023.

[67] F. Li, Y. Yu, H. Zhou, and R. Dong, "DFIMformer : Dynamic frequency-enhanced iTransformer for multiscale time series forecasting," *Inf. Syst.*, vol. 141, p. 102747, 2026, https://doi.org/10.1016/j.is.2026.102747.

[68] B. Fang, L. Xu, Y. Luo, Z. Luo, and W. Li, "A method for short-term electric load forecasting based on the FMLP-iTransformer model," *Energy Reports*, vol. 12, pp. 3405–3411, 2024, https://doi.org/10.1016/j.egyr.2024.09.023.

[69] H. Nagpal, A. Soppelsa, H. Madsen, and R. Fedrizzi, "Energy & Buildings An adaptive multi-seasonal ARIMA approach for domestic hot water load forecasting : A pilot study," *Energy Build.*, vol. 347, p. 116226, 2025, https://doi.org/10.1016/j.enbuild.2025.116226.

[70] L. Ji, Y. Zou, K. He, and B. Zhu, "ScienceDirect Carbon Carbon futures futures price price forecasting forecasting based based with with ARIMA-CNN- LSTM model LSTM model," *Procedia Comput. Sci.*, vol. 162, pp. 33–38, 2020, https://doi.org/10.1016/j.procs.2019.11.254.

[71] F. Khosravi and M. Shadaram, "Retentive Time Series : A Scalable Machine Learning Model for Traffic Prediction in Elastic Optical Networks," *IEEE Access*, vol. 13, pp. 1–17, 2025, https://doi.org/10.1109/ACCESS.2025.3585076.

[72] A. A. Sleem and M. N. Syed, "Novel Predictive Factors to Boost Short-Term Electricity Demand Forecasting Using Machine Learning Models," in *IEEE Access*, vol. 13, pp. 147265-147290, 2025, https://doi.org/10.1109/ACCESS.2025.3600864.

[73] S. Khaleghian, T. Doan, J. O. E. Knox, A. Harris, M. Sartipi, and S. Member, "Data-Driven Insights Into EV Charging Patterns : Machine Learning Models Reveal Key Predictors of Station Utilization in Tennessee," *IEEE Access*, vol. 13, pp. 51458–51483, 2025, https://doi.org/10.1109/ACCESS.2025.3553086.

[74] E. A. Alghamdi, M. H. Alkinani, E. A. L. I. Aldhahri, and N. Ayub, "REST Network : An Ensemble Deep Learning Approach for EV Charging Load Forecasting in Artificial Port Supply Chains," *IEEE Access*, vol. 13, pp. 133128–133144, 2025, https://doi.org/10.1109/ACCESS.2025.3589292.

[75] M. Alaraj, C. Martins, M. Radi, M. Darwish, and M. Majdalawieh, "Spatial – Temporal Deep Learning for Electric-Vehicle Charging Demand : An Exploratory Study of Graph Convolutional and LSTM Networks Performance," *IEEE Access*, vol. 13, pp. 202203–202213, 2025, https://doi.org/10.1109/ACCESS.2025.3636433.

[76] L. Peng, S. Yun, X. Liu, and G. Fan, "Uncertainty quantification and load forecasting based on CPO-VMD and hybrid deep learning," *Electr. Power Syst. Res.*, vol. 252, p. 112444, 2026, https://doi.org/10.1016/j.epsr.2025.112444.

[77] J. Feng *et al.*, "Transformer Abnormal State Identification Based on TCN-Transformer Model in Multiphysics," *IEEE Access*, vol. 13, pp. 44775–44788, 2025, https://doi.org/10.1109/ACCESS.2025.3546341.

[78] H. J. Kim, D. Kim, H. Tak, and J. Y. Lee, "Global-local attention-enabled multiple decoder Transformer for multi-energy load forecasting in user-level integrated energy system," *Appl. Energy*, vol. 396, p. 126255, 2025, https://doi.org/10.1016/j.apenergy.2025.126255.

[79] S. Wu, Z. Wang, and X. Liu, "Temporal structure-preserving transformer for industrial load forecasting," *Neural Networks*, vol. 192, p. 107887, 2025, https://doi.org/10.1016/j.neunet.2025.107887.

AUTHOR BIOGRAPHY

Nguyen Anh Tuan, received a B.Sc. degree from the Industrial University of Ho Chi Minh City, Vietnam 2012, and an M.Sc. degree from the HCMC University of Technology and Education, Vietnam, in 2015. He is a Lecturer with the Faculty of Electrical Engineering Technology at the Industrial University of Ho Chi Minh City, Vietnam. His main research interests include power quality and load forecasting.
Email: nguyenanhtuan@iuh.edu.vn

Vol. 8, No. 4, August 2026, pp. 1259-1298

Trung Dung Nguyen, was born in 1976 in Vietnam. He received a Master's degree in Electrical Engineering from Ho Chi Minh City University of Technology and Education, Vietnam, in 2017. He lectures at the Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam. His research focuses on applications of metaheuristic algorithms in power system optimization, optimal control, and model predictive control.
Email: nguyentrungdung@iuh.edu.vn