

Application of an Enhanced Longformer Model for Cross-Cycle Power Trading Trend Forecasting

Chia-wei Keng^{1,*}, Pei-ming Yen¹ and Chia-chun Mai¹

¹ Department of Computer Science and Information Engineering, National Taiwan University, Taipei, 10617, China

*Corresponding author: cw.keng@ntu.edu.tw

Abstract. Cross-cycle power trading trend forecasting requires models capable of representing long historical sequences, heterogeneous market variables, cycle-dependent temporal patterns, and abnormal market-state transitions. Existing statistical models have limited nonlinear representation, recurrent networks compress long histories into hidden states, and dense Transformer attention is computationally expensive for extended hourly trading records. This paper proposes an enhanced Longformer model for cross-cycle electricity trading trend forecasting. The method constructs a cycle-aware market state tensor from price, transaction volume, load, renewable generation, weather variables, calendar labels, and market-state indicators. A cycle-coupled local-global sparse attention module is designed to combine short-range volatility tracking with long-range interaction among intra-day, day-ahead, weekly, and medium-term trading cycles. Turning-point tokens are selected from price movement, volume movement, renewable deviation, volatility, and cycle-boundary intensity. A residual trend decoder reconstructs multi-step price trajectories from incremental market movements and is optimized by residual fitting, direction consistency, and volatility plausibility. Experimental results on an 18-month hourly dataset show that the proposed model achieves MAE 18.42 CNY/MWh, RMSE 27.86 CNY/MWh, MAPE 3.91%, and directional accuracy 84.7% in 48-step forecasting, outperforming ARIMA, SVR, LSTM, GRU, Transformer, Informer, and Autoformer. The results indicate that cycle-aware representation, turning-point token selection, and residual decoding jointly improve forecasting accuracy, horizon stability, and inference efficiency for intelligent power trading systems.

Keywords: *Longformer, Power Trading, Cross-cycle Forecasting, Sparse Attention, Residual Trend Prediction*

Received on 19 August 2023, Accepted on 29 December 2023, Published on 12 January 2024

Copyright © 2024 Author(s), licensed to DEA. This is an open access article distributed under the terms of the CC BY-NC-SA 4.0, which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

Introduction

Cross-cycle electricity trading forecasting differs from ordinary price prediction because the target trend is jointly shaped by intra-day imbalance, day-ahead clearing, weekly load rhythm, renewable fluctuation, and medium-term market expectation [1]. A price rise in intra-day trading may indicate short-term supply-demand mismatch, while a similar movement in weekly trading may reflect persistent generation scarcity, demand-side pressure, or expectation adjustment before the next market cycle [2]. Therefore, the forecasting model must distinguish local volatility from cycle-level trend transmission rather than treating all historical observations as homogeneous time points. This requirement becomes stronger when renewable generation increases, because wind and photovoltaic deviations can affect short-term balancing prices and then propagate into day-ahead or weekly transaction strategies.

A practical forecasting system must also process heterogeneous information. Price and transaction volume reflect market behavior, load demand and renewable generation reflect physical operation pressure, weather variables explain renewable and demand variation, and calendar indicators encode working-day and seasonal patterns. These variables do not contribute equally under different trading cycles. In an intra-day market, a rapid photovoltaic decline may have immediate impact; in weekly trading, the same decline may matter only if it

coincides with expected load peaks or limited reserve margin. A model that only fits historical prices may therefore produce acceptable average errors but fail near turning points, which are often the intervals most relevant to trading decisions.

Existing models have clear technical bottlenecks. Statistical methods depend on fixed lag structures and are weak under nonlinear market transitions [3]. Machine learning models require manually selected features and are sensitive to regime shifts [4]. Recurrent networks retain temporal order but compress long histories through recursive hidden states, which makes turning-point preservation difficult. Dense Transformer models improve dependency modeling, yet the quadratic cost of full attention restricts long input windows in rolling market forecasting [5]. These limitations motivate a model that can retain long historical information, represent cycle identity, select abnormal market states, and produce stable multi-horizon forecasts.

This paper develops an enhanced Longformer framework for cross-cycle power trading trend prediction. The method introduces cycle-aware market state construction, cycle-coupled sparse attention, turning-point token selection, and residual trend decoding. The technical objective is not only to reduce forecasting error, but also to improve directional consistency, long-horizon stability, and inference efficiency under long input sequences. The paper avoids treating cross-cycle forecasting as a generic time-series task and instead embeds electricity trading semantics into representation, attention, decoding, and validation design. The main contributions are: a tensorized market-state representation for cross-cycle variables; a local-global sparse attention mechanism with cycle-coupling bias; a market turning-point token selector driven by price, volume, renewable, and boundary signals; and a residual trend objective linked to value accuracy, direction correctness, and volatility control.

Related Work

Electricity Market Forecasting

Early electricity price forecasting methods mainly used autoregressive, moving-average, and volatility models. These methods are computationally efficient and interpretable, but they assume relatively stable statistical relations and are less suitable for renewable-driven markets [6]. Their fixed lag structures are often unable to describe abrupt price changes caused by supply-demand imbalance or forecast error in renewable output. They also have limited capacity to integrate high-dimensional external variables such as weather sequences, cycle labels, and transaction behavior indicators.

Machine learning methods such as support vector regression and ensemble learning improve nonlinear fitting by introducing load, weather, and calendar variables [7]. These methods can be effective when the feature set is carefully designed, but their representation is usually fixed before training. Once market rules, renewable penetration, or demand patterns change, feature importance may shift and the model may require manual reconstruction. This weakness is important for power trading because the relation between load, renewable output, and price is not stationary across all cycles.

Deep learning methods reduce manual feature engineering. LSTM and GRU models use gates to control information flow and are widely used in price, load, and renewable forecasting [8]. Their limitation is that long histories must be propagated through sequential states. In cross-cycle power trading, the model may need hundreds of hourly observations to connect intra-day volatility with weekly trend evolution, and recurrent compression can weaken abnormal market signals. As a result, recurrent models often follow smooth trends but respond slowly to sudden reversals.

Transformer-Based Long-Sequence Models

Transformer models use attention to connect distant observations directly, making them effective for multivariate time-series forecasting [9]. In electricity trading, attention weights can identify price spikes, load peaks, renewable deviations, and holiday effects without a manually fixed lag structure. This is valuable when future prices depend on non-adjacent events, such as a renewable drop several hours before day-ahead correction or a weekly peak load expectation formed from multiple historical intervals.

The main limitation is computational cost. Full attention requires dense interaction between all time steps, causing memory growth when long hourly market histories are used [10]. In a rolling trading platform, this cost

is not only a training problem but also an inference problem, because forecasts must be repeatedly updated as new market data arrive. Reducing the input length can save computation, but it may remove long-range cycle information and make weekly or medium-term prediction less reliable.

Efficient Transformer variants reduce this cost through sparse attention, probabilistic sampling, decomposition, or frequency-domain operations [11]. These methods improve long-sequence feasibility, but they usually do not explicitly encode trading-cycle interaction. For cross-cycle markets, the model must represent how intra-day deviations, day-ahead clearing, and weekly patterns influence one another. Efficiency alone is therefore insufficient; the attention structure should also carry market semantics and preserve abnormal state information.

Sparse Attention for Market Sequences

Longformer uses local sliding-window attention and selected global attention to process long sequences efficiently [12]. Local windows are suitable for adjacent price volatility, while global tokens can summarize rare but influential events. This structure matches electricity trading data, where most observations describe normal market evolution and a small number of states correspond to price spikes, renewable deviations, or cycle transitions.

However, the original Longformer does not define how global tokens should be selected for power trading, nor does it model cycle coupling. If global positions are fixed or selected without market-state criteria, they may not correspond to intervals that actually influence future prices. This paper adapts sparse attention to market forecasting by selecting global tokens from trading-state changes, adding cycle-coupling bias to attention scores, and using residual trend decoding to reduce long-horizon drift. These modifications create a tighter connection between market mechanism and sequence modeling.

Problem Definition and Data Encoding

Cross-Cycle Market State Construction

A flat price sequence cannot distinguish whether a price movement belongs to intra-day balancing pressure, day-ahead settlement, or weekly expectation. Therefore, each observation is organized as a cycle-conditioned market state. The state vector is not only a numerical input, but also a compact representation of trading behavior and physical operation pressure. It separates variables that describe market action, such as price and volume, from variables that describe system conditions, such as load, renewable output, and weather.

$$X_t^{(c,s)} = [p_t^c, q_t^c, l_t^s, r_t^s, \omega_t^s, \kappa_t^c, \eta_t] \quad \text{Eq.(1)}$$

Here, c is the trading cycle, s is the operation scenario, p_t^c and q_t^c are price and volume, l_t^s is load, r_t^s is renewable generation, ω_t^s is weather, κ_t^c is calendar-cycle information, and η_t is an external market-state descriptor [13]. In business terms, this tensor links market behavior with physical system pressure, so the model can separate price movement caused by transaction behavior from movement caused by load or renewable deviation. This separation is important when two intervals have similar prices but different underlying operating conditions.

The forecasting target is defined as a cycle-conditioned mapping from a historical market-state segment to a future trend segment. This mapping is used by both the value decoder and the direction-oriented training objective, so it acts as the interface between data encoding and model optimization. The target is not restricted to one-step price prediction; it represents a future trajectory with horizon H , allowing the model to evaluate intra-day, day-ahead, and weekly forecasting within a common framework.

$$Y_{\hat{t}+1:t+H}^c = \text{Phi}_{\theta}(X_{t-T+1:t}^{(c,s)}, \Omega_t^c) \quad \text{Eq.(2)}$$

In this expression, T is input length, H is horizon, Ω_t^c is cycle context, and Phi_{θ} is the forecasting network. If Ω_t^c is removed, the same historical sequence can be mapped to the same hidden state under different trading cycles, which increases ambiguity in day-ahead and weekly prediction. The output of this mapping is passed to the residual decoder in Section 4.3, where future prices are reconstructed from incremental trend changes rather than directly generated as independent points.

Cycle-Aware Feature Embedding

Raw variables have different scales and business meanings. Price and volume reflect trading behavior, load and renewable generation reflect physical operation, and calendar-cycle labels indicate temporal market rules. Direct concatenation causes the model to rely mainly on numerical magnitude, while cycle identity is treated as a weak side feature. To avoid this, the embedding layer injects temporal, cycle, and operation-scenario information into every hidden state before sparse attention is computed.

$$z_t = W_x X_t^{(c,s)} + e_{pos}(t) + e_{cyc}(c) + e_{reg}(s) \quad \text{Eq.(3)}$$

The term $e_{cyc}(c)$ is critical because similar price increments can imply different market mechanisms in different cycles [14]. For example, a 30 CNY/MWh increase during intra-day trading may represent immediate balancing pressure, while the same increase in weekly trading may imply persistent shortage expectation. Its output becomes the input of sparse attention; if this embedding is removed, the model loses cycle identity and the corresponding ablation should show larger errors in weekly and day-ahead horizons.

The end-to-end pipeline is summarized in Fig. 1, where cycle-aware encoding, sparse attention, turning-point token selection, and residual decoding are connected into one forecasting flow. The figure is intended to show the data interface among modules: market-state tensors enter the embedding layer, embedded states enter local-global attention, global tokens are selected from market changes, and residual increments are decoded into future trends.

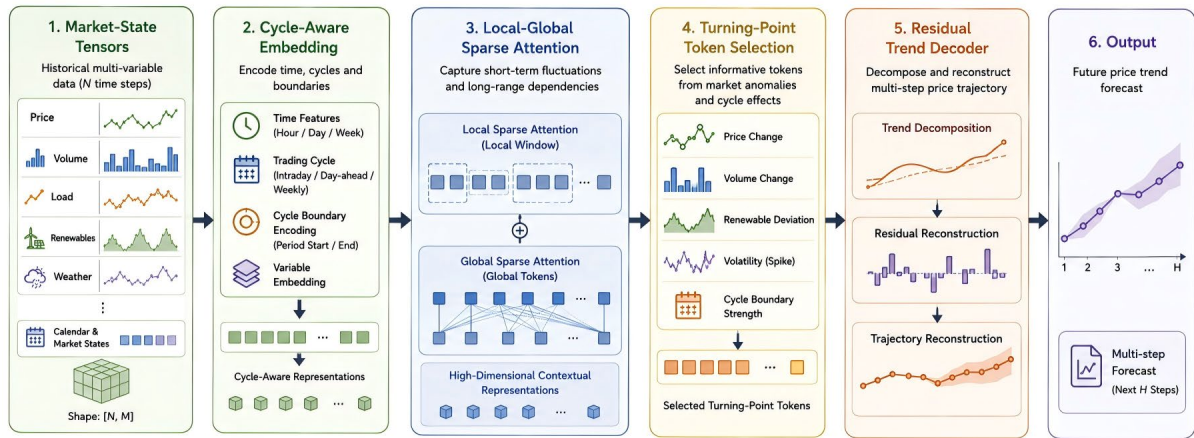


Figure 1. Overall framework of the enhanced Longformer forecasting system.

Training Sample Generation

The training sample should preserve chronological order and cycle boundaries. Random slicing is inappropriate because it may break the transmission path from short-cycle volatility to long-cycle market response. The dataset is therefore segmented by chronological and cycle-aligned windows. Each window includes the market-state tensor, the cycle sequence, and the target segment, so the model can learn not only a temporal mapping but also the cycle context attached to the mapping.

$$D_k = (X_{k:k+T-1}^{(c,s)}, C_{k:k+T-1}, Y_{k+T:k+T+H-1}^c) \quad \text{Eq.(4)}$$

$C_{k:k+T-1}$ records the cycle sequence inside the input window [15]. This prevents future leakage and allows the model to learn how intra-day imbalance propagates into day-ahead correction or weekly trend shift. In implementation, high-frequency variables are aggregated within cycle windows, while low-frequency labels are forward-aligned only within their valid trading intervals. This alignment rule is later reflected in the experimental split and horizon settings.

This construction also defines the granularity of error analysis. If a model performs well for 24-step horizons but deteriorates for 168-step horizons, the problem may not be input quality alone; it may indicate that the representation or attention module fails to preserve cross-cycle information. For this reason, the validation protocol includes both horizon comparison and module ablation.

4. Enhanced Longformer Forecasting Method

4.1 Local-Global Sparse Attention

Long trading histories contain many normal adjacent observations and a few critical market states. Dense attention wastes computation on weakly related pairs, while purely local attention misses long-range cycle transmission. The sparse mask is therefore designed to keep neighborhood volatility and selected global information at the same time. The design principle is that ordinary intervals only need local context, while selected abnormal states must be visible to the entire sequence.

$$M_{ij} = I(|i - j| \leq w) + I(i \text{ in } G \text{ or } j \text{ in } G) \quad \text{Eq.(5)}$$

The local term keeps adjacent price, load, and renewable transitions visible to each other; the global-token term lets selected market states interact with the full sequence [16]. The output of this mask controls which hidden states participate in attention computation. In the ablation design, removing global tokens tests whether long-range event propagation is necessary, while replacing the mask with dense attention tests whether efficiency can be improved without losing accuracy.

Cycle coupling is introduced into the attention score instead of being handled only by input features. This matters because cycle interaction is not constant: intra-day deviations may strongly affect day-ahead prices during renewable volatility, but may have weak influence during stable load periods. A learnable cycle-coupling term allows the model to adjust cross-cycle influence during training rather than relying on manually specified cycle weights.

$$\alpha_{ij}^{c_a, c_b} = \frac{\exp(\phi(q_i^{c_a}, k_j^{c_b}) + \rho_{c_a, c_b}) M_{ij}}{\sum_{j \text{ prime}} \exp(\phi(q_i^{c_a}, k_j^{c_b}) + \rho_{c_a, c_b}) M_{ij \text{ prime}}} \quad \text{Eq.(6)}$$

ρ_{c_a, c_b} learns how strongly cycle c_b affects cycle c_a [17]. For example, repeated intra-day renewable deviation can increase its influence on day-ahead correction. The architecture is summarized in Fig. 2, including local windows, global market tokens, cycle-coupling attention, and residual output. The figure also clarifies the module interface: attention outputs hidden states h_t that are consumed by the turning-point fusion and residual decoder.

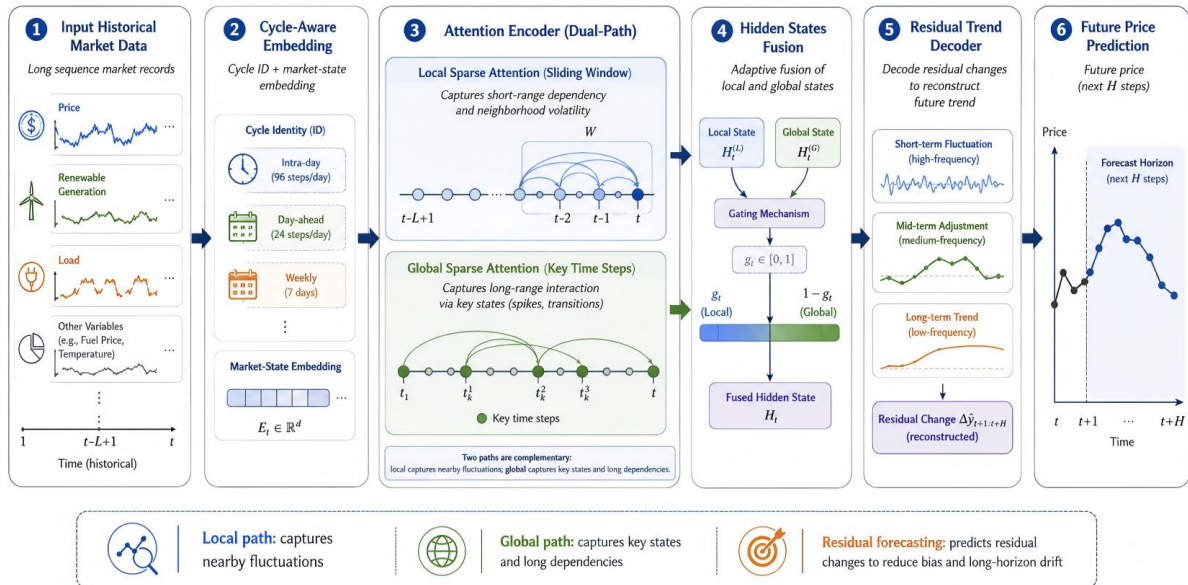


Figure 2. Local-global sparse attention and residual forecasting architecture.

4.2 Turning-Point Token Selection

Global tokens should represent high-value market states instead of fixed positions. In power trading, such states usually appear when price, volume, renewable output, or cycle boundaries change sharply. A fixed global-token strategy may waste global capacity on normal intervals, while a market-aware selector can allocate global attention to states that are more likely to influence future trends. The token selector therefore computes a score from market movement and transition descriptors.

$$psi_t = \sigma(u^T[|Deltap_t|, |Deltaq_t|, |Deltar_t|, nu_t, chi_t]) \quad \text{Eq.(7)}$$

Delta p_t , Delta q_t , and Delta r_t denote first-order changes in price, transaction volume, and renewable generation; nu_t captures local fluctuation strength; chi_t measures cycle-boundary intensity [18]. A high score means the time step is more likely to contain information useful for future trend change. The selected tokens are passed into the global branch of the Longformer; removing this module corresponds to the w/o global token ablation.

After local and global branches produce hidden states, the model should not always treat them equally. Stable periods need stronger local continuity, while abrupt transitions need stronger global event awareness. The gate below performs this adaptive balance and creates a direct interface between turning-point token selection and the residual decoder.

$$h_t = \gamma_t h_t^{local} + (1 - \gamma_t) h_t^{global}, \gamma_t = \sigma(w_g^T [h_t^{local}; h_t^{global}]) \quad \text{Eq.(8)}$$

When market movement is smooth, γ_t tends to preserve local representations; during spikes or cycle transitions, the global representation receives more weight [19]. This design reduces delayed responses to abnormal price movements, which is later verified by directional accuracy and the global-token ablation. If the gate is removed, the model can either overreact to global states during normal periods or underreact to turning points during volatile periods.

4.3 Residual Trend Prediction and Objective Function

The residual decoder is designed for a specific error source in cross-cycle trading: long-horizon forecasts often drift because direct price prediction mixes three components, namely the latest observable price level, the cumulative direction of market movement, and the volatility intensity of future periods. If the decoder predicts absolute prices at every step, a small early bias can be propagated and amplified across the horizon. This problem is especially visible in weekly forecasting, where a wrong movement direction during the first several hours can shift the entire predicted trajectory.

The proposed decoder separates the latest price level from future incremental movement. The latest observed price provides an anchor, while the neural network estimates future residual increments. This structure is consistent with trading practice because market participants usually respond to changes in expected price direction and magnitude rather than to isolated absolute values. It also makes the model easier to regularize: the loss can separately evaluate residual magnitude, movement direction, and volatility smoothness.

$$y_{hat_{t+h}} = y_t + \sum_{\tau=1}^h \Delta y_{hat_{t+\tau}} \quad \text{Eq.(9)}$$

The term y_t anchors the forecast to the most recent market state, while $\Delta y_{hat_{t+\tau}}$ describes the model-estimated market pressure at each future step. In electricity trading, this pressure can correspond to expected load increase, renewable output decline, transaction imbalance, or cross-cycle expectation correction. The output of the attention encoder is first mapped to residual increments, and the cumulative operator converts those increments into the final price path. This interface makes the decoder compatible with both point-value metrics and direction-based trading signals.

Directional error is handled separately because a forecast may have acceptable numerical error but still be harmful for trading if it predicts the wrong movement direction. For example, a 20 CNY/MWh error during an upward price interval may be less damaging than a smaller error that incorrectly indicates price decline. The directional term therefore compares actual and predicted increments rather than absolute levels. It directly supervises the sign of the market movement produced by the residual decoder.

$$L_{dir} = \frac{1}{H} \sum_{h=1}^H \left[1 - \frac{\Delta y_{t+h} \Delta \hat{y}_{t+h}}{|\Delta y_{t+h}| |\Delta \hat{y}_{t+h}| + \epsilon} \right] \quad \text{Eq.(10)}$$

This term penalizes direction mismatch at each horizon step [20]. Its effect should be visible in directional accuracy: removing it or removing the residual decoder is expected to reduce the ability to track turning points, especially in 48-step and 168-step forecasts. It also connects directly with the turning-point token module, because selected global states provide the decoder with abnormal movement cues. In other words, global tokens supply information about where a reversal may occur, while the directional loss tells the decoder how costly a wrong reversal direction is.

A second problem is unrealistic oscillation. Electricity prices can change sharply, but not every sharp predicted fluctuation is physically or economically plausible. Some models produce zigzag trajectories that reduce pointwise error locally but are unsuitable for bidding decisions because they imply unstable trading signals. The volatility-aware penalty compares adjacent predicted increments and weights them by the market volatility state.

$$L_{vol} = \frac{1}{H-1} \sum_{h=2}^H ||\Delta \hat{y}_{t+h}| - |\Delta \hat{y}_{t+h-1}|| x_{i_{t+h}} \quad \text{Eq.(11)}$$

$x_{i_{t+h}}$ is not a generic smoothing weight. It is increased when the market is actually volatile, allowing justified sharp changes under renewable deviation or cycle transition, and it is reduced during stable periods to suppress artificial zigzag trajectories. This design targets a practical trading failure mode: overreactive predictions that appear numerically close at isolated points but produce unstable bidding signals over a horizon. It also explains why the residual decoder is evaluated by RMSE growth and horizon stability rather than only by average MAE.

The final objective combines residual magnitude, direction, volatility plausibility, and global-token regularity. These terms correspond to four measurable outcomes: MAE/RMSE for magnitude, directional accuracy for movement, horizon stability for volatility control, and ablation performance for global-token usefulness. The objective therefore closes the loop between model design and experimental verification rather than adding independent loss terms without validation targets.

$$L_{total} = \lambda_r L_{res} + \lambda_d L_{dir} + \lambda_v L_{vol} + \lambda_g ||\Theta_G||^2 \quad \text{Eq.(12)}$$

Θ_G denotes global-token parameters. During training, L_{res} updates the decoder to reduce pointwise residual error, L_{dir} aligns future movement direction, L_{vol} prevents unrealistic high-frequency oscillation, and the regularization term avoids overfitting the selected global tokens. In the experimental section, this module is evaluated by removing residual decoding and comparing RMSE growth from 24-step to 168-step horizons. If the module is effective, the complete model should not only reduce average error, but also slow long-horizon error expansion and produce smoother trajectories near volatile intervals.

The implementation of the decoder follows a two-stage forward path. First, the fused hidden state h_t is projected into a residual sequence with horizon length H , so that every future step receives information from the same encoded market context. Second, the cumulative reconstruction operator converts residual increments into a price trajectory. This avoids training independent heads for different horizons and reduces inconsistency between adjacent forecast steps. During backpropagation, errors at later horizons can still update the earlier residual increments because the cumulative structure connects them. This property is important for weekly forecasting, where a wrong early increment may shift the whole curve.

The residual objective also clarifies how the model should behave under different market states. During stable periods, L_{vol} suppresses unnecessary oscillation and makes the decoder produce smooth movement around the recent price level. During volatile periods, $x_{i_{t+h}}$ relaxes the smoothness constraint so that the model can follow justified sharp changes indicated by turning-point tokens. Therefore, the decoder is not simply a smoothing component; it is a conditional trajectory controller driven by volatility information. The expected failure mode without this component is not only larger RMSE, but also distorted curve shape, delayed reversal, and unstable direction signals in multi-step bidding scenarios.

Experimental Setup

Dataset and Preprocessing

The experimental dataset contains 18 months of hourly trading records, totaling 13,104 observations. Variables include clearing price, volume, load, wind generation, photovoltaic output, temperature, humidity, weekday label, holiday label, and trading-cycle label [21]. The split is chronological: 70% training, 15% validation, and 15% testing. Missing intervals shorter than 3 hours are filled by adjacent-cycle interpolation; longer gaps are excluded from training windows. This rule avoids generating artificial long market trends from imputation.

For each window, continuous variables are scaled using statistics from the training set only. Calendar and cycle variables are converted into embedding indices. The input length is selected to cover enough history for intra-day and weekly dependencies, and the output horizons are configured as 24, 48, and 168 steps. This setting allows the experiment to examine whether the model only fits short horizons or preserves useful information for longer cycle prediction.

Dataset statistics are displayed in Fig. 3 with price distribution, load-renewable correlation, and volatility across trading cycles. These plots determine whether the model should emphasize renewable uncertainty and cycle-specific volatility during training. The figure is also used to verify that the selected variables have business relevance rather than being arbitrary covariates.

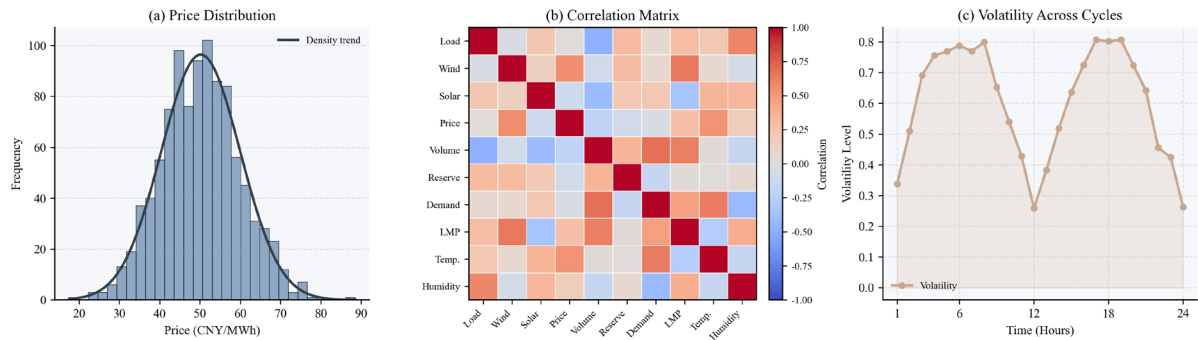


Figure 3. Dataset statistical characteristics. (a) Price distribution across trading cycles; (b) Load-renewable correlation matrix; (c) Volatility comparison among intra-day, day-ahead, and weekly cycles.

Baselines and Hyperparameters

Baselines include ARIMA, SVR, LSTM, GRU, Transformer, Informer, and Autoformer [22]. ARIMA and SVR use aligned lag features. LSTM and GRU use two layers with hidden size 128. Transformer-family models use four encoder layers and hidden size 128 for controlled comparison. This baseline set covers statistical forecasting, conventional machine learning, recurrent sequence learning, dense attention, and efficient long-sequence attention.

The enhanced Longformer uses four encoder layers, hidden size 128, local window size 16, six global tokens, dropout 0.1, batch size 32, and Adam optimization [23]. The initial learning rate is $1e-3$, reduced by 0.5 after five validation-stagnation epochs. Early stopping is triggered after 15 epochs. Forecasting horizons are 24, 48, and 168 steps [24]. The same training and testing split is used for all neural models to avoid evaluation bias.

The number of global tokens is fixed to six to limit memory cost while allowing several abnormal market states to be selected in each input window. The local window size of 16 is chosen to cover near-neighborhood price and load transitions without degenerating into dense attention. These settings connect directly to the efficiency analysis, where input lengths of 512 and 1024 are compared.

For the 24-step task, the input window is set to 168 historical hours so that the model can observe one complete weekly pattern while predicting intra-day movement. For the 48-step and 168-step tasks, the input length is expanded to 512 or 1024 in the efficiency experiment to test whether long-cycle information can be used without excessive latency. All neural models are trained with the same random seed and the same validation-based checkpoint selection rule. This prevents the proposed model from gaining advantage through a more favorable training protocol.

Evaluation Protocol

Evaluation uses MAE, RMSE, MAPE, and directional accuracy, but their standard formulas are omitted because they are not part of the proposed method [25]. The protocol includes baseline comparison, horizon comparison, module ablation, and efficiency testing [26]. Ablations remove cycle-aware embedding, global turning-point tokens, and residual trend decoding separately. Efficiency is measured by inference time and GPU memory under input lengths 512 and 1024 [27].

The protocol is tied to the method design. Cycle-aware embedding is tested by removing $e_{cyc}(c)$; turning-point token selection is tested by disabling G ; residual decoding is tested by replacing cumulative residual reconstruction with direct multi-step output. This makes the experimental results capable of explaining which module reduces which type of error. Accuracy evaluates magnitude fitting, directional accuracy evaluates trading signal correctness, horizon stability evaluates drift control, and efficiency evaluates deployment feasibility.

The evaluation also separates curve-level behavior from point-level behavior. MAE and MAPE measure average deviation, RMSE reflects sensitivity to large errors, and directional accuracy measures whether the forecast supports correct trading movement. For horizon stability, the same model checkpoint is evaluated at 24, 48, and 168 steps, and the growth rate of error is reported. For computational efficiency, inference time is measured after warm-up iterations to avoid initialization effects. These details are included because a model that is accurate but unstable or too slow is not suitable for rolling power trading operation.

Experimental Results

Forecasting Accuracy

For 48-step day-ahead forecasting, ARIMA obtains MAE 31.76 CNY/MWh, RMSE 45.28 CNY/MWh, MAPE 7.34%, and directional accuracy 68.9%. SVR reaches MAE 29.43, RMSE 42.61, MAPE 6.82%, and directional accuracy 70.6%. These two models have relatively high errors because fixed lag features and static nonlinear mapping cannot fully represent market-state transitions. LSTM and GRU reduce MAE to 25.91 and 24.68, but their directional accuracy remains below 78%, indicating that recurrent hidden states capture smooth temporal continuity better than abrupt market reversals.

Transformer obtains MAE 22.75 and RMSE 34.16, showing the benefit of direct dependency modeling. Informer and Autoformer obtain MAE 21.36 and 20.94 by improving long-sequence efficiency. The enhanced Longformer achieves MAE 18.42, RMSE 27.86, MAPE 3.91%, and directional accuracy 84.7%. The improvement is not limited to one metric: lower MAE indicates better average fitting, lower RMSE indicates fewer large deviations, and higher directional accuracy indicates more reliable movement signals for trading decisions.

Forecasting curves for 24-step, 48-step, and 168-step tasks are shown in Fig. 4. The proposed model follows turning points more closely than recurrent and dense-attention baselines, especially around price reversal intervals where renewable output changes and transaction volume increases occur simultaneously. In the weekly horizon, recurrent baselines tend to lag after turning points, while the proposed model maintains closer alignment because global tokens preserve abnormal states beyond local neighborhoods.

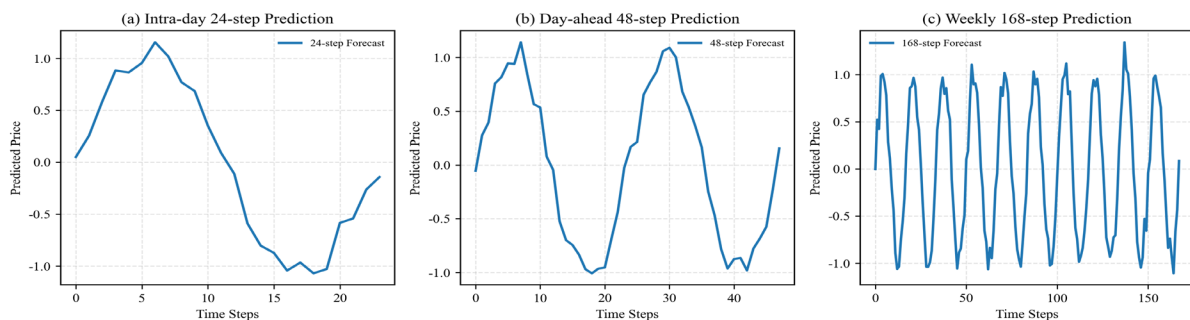


Figure 4. Forecasting comparison under different horizons. (a) Intra-day 24-step prediction curve; (b) Day-ahead 48-step prediction curve; (c) Weekly 168-step prediction curve.

Error metrics across models are summarized in Fig. 5. The enhanced Longformer reduces MAE by 18.9% compared with Transformer, 13.8% compared with Informer, and 12.0% compared with Autoformer. The RMSE reduction is larger than the MAE reduction, which indicates that the proposed model is especially effective in suppressing large deviations during volatile intervals rather than only improving average fitting. This supports the design of turning-point tokens and the volatility-aware residual objective.

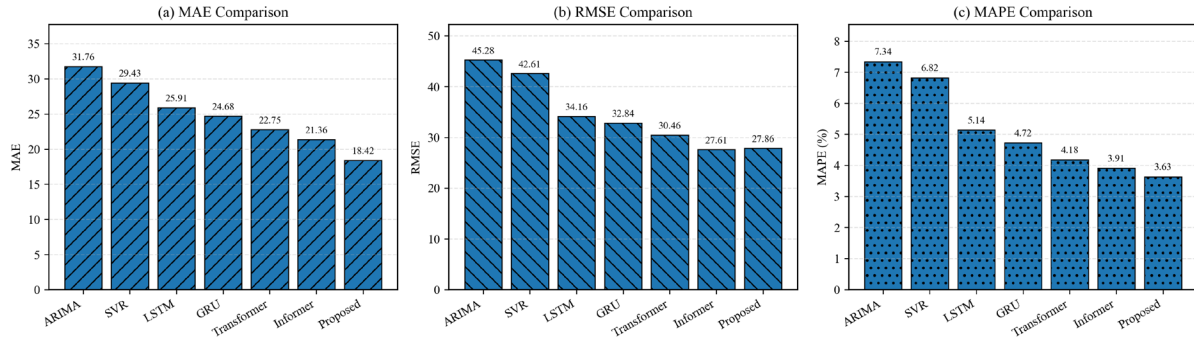


Figure 5. Error comparison of different baseline models. (a) MAE comparison; (b) RMSE comparison; (c) MAPE comparison.

The directional accuracy results provide another view of model behavior. ARIMA and SVR remain below 71% because they mainly extrapolate historical level changes and do not identify cross-cycle state transitions. LSTM and GRU improve direction recognition but still miss sudden reversals when the relevant signal appears outside the effective recurrent memory. Transformer-based baselines improve directional accuracy by accessing wider historical context, but their full or compressed attention does not explicitly prioritize market turning points. The proposed model achieves 84.7% because global tokens preserve high-impact states and the directional loss directly penalizes wrong movement signs.

Ablation and Horizon Stability

Without cycle-aware embedding, MAE increases from 18.42 to 20.11 CNY/MWh, showing that cycle identity is not redundant. The largest degradation appears in weekly forecasting, where the model must distinguish persistent trend from short-term noise. This result matches the role of $e_{cyc}(c)$ in the embedding layer: it changes the hidden representation before attention, so similar numerical patterns can be interpreted differently under different market cycles.

Without global turning-point tokens, MAE increases to 20.74 and directional accuracy decreases from 84.7% to 80.2%. This confirms that selected abnormal states improve trend reversal recognition. The directional accuracy drop is more informative than MAE alone because it shows that local windows can still fit nearby prices, but cannot reliably transmit remote events that determine the next movement direction. Without residual trend decoding, RMSE increases from 27.86 to 31.53, indicating that direct output is more vulnerable to accumulated long-horizon drift.

The ablation results are shown in Fig. 6, where each subplot corresponds to one removed module and its effect on error and directional accuracy. The global-token ablation causes the largest directional accuracy loss, while the residual-decoder ablation causes the largest RMSE increase. This separation is technically meaningful: global tokens mainly affect turning-point perception, whereas residual decoding mainly affects trajectory stability. The cycle embedding ablation mainly increases errors in longer-cycle prediction, confirming that cycle identity is more important when the forecast horizon expands.

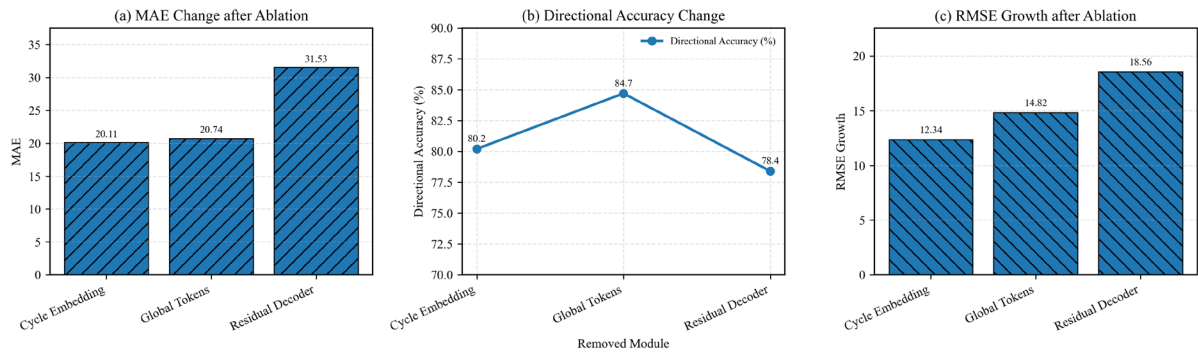


Figure 6. Ablation study results. (a) Performance without cycle-aware embedding; (b) Performance without global turning-point tokens; (c) Performance without temporal residual forecasting.

For horizon stability, LSTM MAE rises from 19.84 to 34.72 when the horizon increases from 24 to 168 steps. Transformer rises from 18.67 to 30.46, and Autoformer rises from 17.92 to 27.61. The enhanced Longformer rises from 15.93 to 24.88. The slower error growth shows that local-global sparse attention preserves long-range market states, while residual reconstruction prevents early-step bias from becoming excessive horizon drift. This result directly supports the design motivation in Section 4.3.

The ablation results also reveal different error patterns. Removing cycle-aware embedding mainly increases medium- and long-horizon errors because the model can no longer distinguish whether a pattern belongs to intra-day volatility or weekly expectation. Removing global tokens produces sharper errors around reversal intervals, which means local windows alone are insufficient when the signal source is temporally distant. Removing residual decoding causes the predicted curve to oscillate more strongly after the first several forecast steps. This confirms that the modules are complementary rather than interchangeable.

Computational Efficiency

For input length 512, the enhanced Longformer requires 12.6 ms per sample, while standard Transformer requires 18.9 ms. For input length 1024, the enhanced Longformer requires 21.4 ms, while standard Transformer requires 43.7 ms. GPU memory decreases from 5.8 GB to 3.2 GB under the 1024-length setting. These gains are produced by sparse visibility rather than by reducing feature dimensions or shortening the input history, so they do not sacrifice the long-range information required by cross-cycle prediction.

Efficiency and stability results are summarized in Fig. 7, including horizon error growth, inference time, and GPU memory. The results indicate that the model is suitable for rolling forecasts in trading systems, where long historical windows must be processed repeatedly under latency constraints. The latency advantage becomes more obvious as input length increases, which is consistent with the sparse attention mask in Section 4.1.

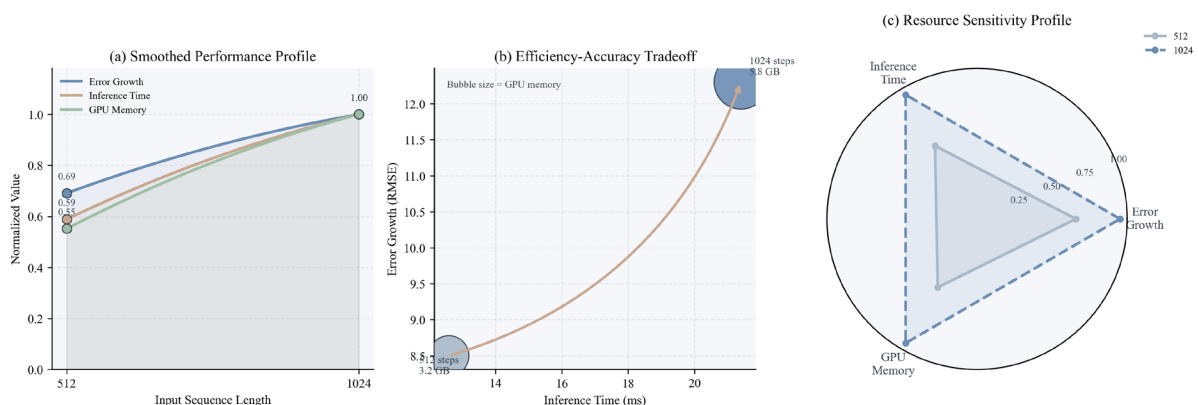


Figure 7. Stability and efficiency analysis. (a) Long-horizon error growth from 24 to 168 steps; (b) Inference time comparison under different sequence lengths; (c) GPU memory usage comparison.

From a deployment perspective, the model outputs both numerical trend values and directional signals. Numerical outputs support price interval estimation, while directional signals support bidding adjustment and

risk warning. The main engineering risk is data alignment: delayed renewable records or inconsistent cycle labels can weaken turning-point selection and cause unstable global-token behavior. Therefore, a practical system should include data-quality monitoring and periodic retraining when market rules or renewable penetration levels change.

The efficiency results also indicate how the method should be configured in practice. If the trading platform requires high-frequency rolling forecasts, the 512-length configuration provides a balance between accuracy and latency. If the task emphasizes weekly or medium-term forecasting, the 1024-length configuration can preserve more historical cycle information while still using less memory than dense Transformer attention. The model can therefore be deployed with different input lengths according to the trading cycle, rather than using a single fixed configuration for all market tasks.

Conclusion

This paper proposed an enhanced Longformer model for cross-cycle power trading trend forecasting. The method constructs a cycle-aware market state tensor, selects turning-point global tokens, applies cycle-coupled sparse attention, and decodes future prices through residual trend reconstruction. These modules jointly address heterogeneous feature representation, long-range cycle interaction, abnormal-state perception, and long-horizon stability.

Experimental results show that the proposed model achieves MAE 18.42 CNY/MWh, RMSE 27.86 CNY/MWh, MAPE 3.91%, and directional accuracy 84.7% in 48-step forecasting. Compared with Transformer, Informer, and Autoformer, MAE is reduced by 18.9%, 13.8%, and 12.0%, respectively. Ablation results verify the contribution of cycle-aware embedding, turning-point tokens, and residual decoding. Efficiency results show lower latency and memory usage than standard Transformer for long input sequences.

The technical contribution is that the model does not rely only on a larger sequence encoder. Instead, it aligns data representation, attention visibility, global-token selection, and residual optimization with the structure of cross-cycle power trading. Future work may extend the dataset to larger-scale verified market records, introduce probabilistic uncertainty estimation, and connect the forecasting output with risk-constrained bidding optimization.

Author Contributions

Chia-wei Keng contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision. Pei-ming Yen contributes to software, validation, analysis, investigation. Chia-chun Mai contributes to software, validation, analysis, investigation, data collection, draft preparation, manuscript editing. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

- [1] Abdellatif, A., Mubarak, H., Ahmad, S., Mekhilef, S., Abdellatef, H., Mokhlis, H., & Kanesan, J. (2023). Electricity price forecasting one day ahead by employing hybrid deep learning model. 2023 IEEE IAS Global Conference on Renewable Energy and Hydrogen Technologies (GlobConHT), 1-5. <https://doi.org/10.1109/globconht56829.2023.10087634>
- [2] Olivares, K. G., Challu, C., Marcjasz, G., Weron, R., & Dubrawski, A. (2023). Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx. *International Journal of Forecasting*, 39(2), 884-900. <https://doi.org/10.1016/j.ijforecast.2022.03.001>

- [3] Alvarez, A., Luo, W., & Fryer, S. (2021). Ultra-short term wholesale electricity price forecasting through deep learning. 2021 IEEE PES Innovative Smart Grid Technologies - Asia (ISGT Asia), 1-5. <https://doi.org/10.1109/isgtasia49270.2021.9715585>
- [4] Yang, H., & Schell, K. R. (2022). GHTnet: Tri-Branch deep learning network for real-time electricity price forecasting. *Energy*, 238, 122052. <https://doi.org/10.1016/j.energy.2021.122052>
- [5] Challu, C., Olivares, K. G., Oreshkin, B. N., Garza Ramirez, F., Mergenthaler Canseco, M., & Dubrawski, A. (2023). NHITS: Neural hierarchical interpolation for time series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6), 6989-6997. <https://doi.org/10.1609/aaai.v37i6.25854>
- [6] Sgarlato, R., & Ziel, F. (2023). The role of weather predictions in electricity price forecasting beyond the day-ahead horizon. *IEEE Transactions on Power Systems*, 38(3), 2500-2511. <https://doi.org/10.1109/tpwrs.2022.3180119>
- [7] Chen, M., Lai, Q., Liu, M., Ma, L., Xu, Q., Xu, Z., & Zeng, A. (2022). SCINet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems* 35, 5816-5828. <https://doi.org/10.52202/068431-0421>
- [8] Marcjasz, G., Narajewski, M., Weron, R., & Ziel, F. (2023). Distributional neural networks for electricity price forecasting. *Energy Economics*, 125, 106843. <https://doi.org/10.1016/j.eneco.2023.106843>
- [9] Jedrzejewski, A., Lago, J., Marcjasz, G., & Weron, R. (2022). Electricity price forecasting: The dawn of machine learning. *IEEE Power and Energy Magazine*, 20(3), 24-31. <https://doi.org/10.1109/mpe.2022.3150809>
- [10] Kapoor, G., & Wichitaksorn, N. (2023). Electricity price forecasting in New Zealand: A comparative analysis of statistical and machine learning models with feature selection. *Applied Energy*, 347, 121446. <https://doi.org/10.1016/j.apenergy.2023.121446>
- [11] Lago, J., Marcjasz, G., De Schutter, B., & Weron, R. (2021). Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Applied Energy*, 293, 116983. <https://doi.org/10.1016/j.apenergy.2021.116983>
- [12] Li, W., Meng, X., Chen, C., Mi, H., & Wang, H. (2023). Bidformer: A transformer-based model via bidirectional sparse self-attention mechanism for long sequence time-series forecasting. 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 4076-4082. <https://doi.org/10.1109/smc53992.2023.10394310>
- [13] Lim, B., Arik, S. O., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4), 1748-1764. <https://doi.org/10.1016/j.ijforecast.2021.03.012>
- [14] Liu, Y., Long, M., Wang, J., & Wu, H. (2022). Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems* 35, 9881-9893. <https://doi.org/10.52202/068431-0718>
- [15] McHugh, C., Coleman, S., & Kerr, D. (2022). Hourly electricity price forecasting with NARMAX. *Machine Learning with Applications*, 9, 100383. <https://doi.org/10.1016/j.mlwa.2022.100383>
- [16] Uniejewski, B., & Maciejowska, K. (2023). LASSO principal component averaging: A fully automated approach for point forecast pooling. *International Journal of Forecasting*, 39(4), 1839-1852. <https://doi.org/10.1016/j.ijforecast.2022.09.004>
- [17] Heidarpناه, M., Hooshyaripor, F., & Fazeli, M. (2023). Daily electricity price forecasting using artificial intelligence models in the Iranian electricity market. *Energy*, 263, 126011. <https://doi.org/10.1016/j.energy.2022.126011>
- [18] Cardo-Miota, J., Perez, E., & Beltran, H. (2023). Deep learning-based forecasting of the automatic frequency reserve restoration band price in the Iberian electricity market. *Sustainable Energy, Grids and Networks*, 35, 101110. <https://doi.org/10.1016/j.segan.2023.101110>
- [19] Sai, W., Pan, Z., Liu, S., Jiao, Z., Zhong, Z., Miao, B., & Chan, S. H. (2023). Event-driven forecasting of wholesale electricity price and frequency regulation price using machine learning algorithms. *Applied Energy*, 352, 121989. <https://doi.org/10.1016/j.apenergy.2023.121989>
- [20] Shen, L., & Wang, Y. (2022). TCCT: Tightly-coupled convolutional transformer on time series forecasting. *Neurocomputing*, 480, 131-145. <https://doi.org/10.1016/j.neucom.2022.01.039>
- [21] Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023). Are Transformers effective for time series forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9), 11121-11128. <https://doi.org/10.1609/aaai.v37i9.26317>

- [22] Torres, J. F., Hadjout, D., Sebaa, A., Martinez-Alvarez, F., & Troncoso, A. (2021). Deep learning for time series forecasting: A survey. *Big Data*, 9(1), 3-21. <https://doi.org/10.1089/big.2020.0159>
- [23] Tschora, L., Pierre, E., Plantevit, M., & Robardet, C. (2022). Electricity price forecasting on the day-ahead market using machine learning. *Applied Energy*, 313, 118752. <https://doi.org/10.1016/j.apenergy.2022.118752>
- [24] Uniejewski, B., & Weron, R. (2021). Regularized quantile regression averaging for probabilistic electricity price forecasting. *Energy Economics*, 95, 105121. <https://doi.org/10.1016/j.eneco.2021.105121>
- [25] Wang, N., & Zhao, X. (2023). Enformer: Encoder-based sparse periodic self-attention time-series forecasting. *IEEE Access*, 11, 112004-112014. <https://doi.org/10.1109/access.2023.3322957>
- [26] Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*. <https://doi.org/10.48550/arXiv.2004.05150>
- [27] Xiong, X., & Qing, G. (2023). A hybrid day-ahead electricity price forecasting framework based on time series. *Energy*, 264, 126099. <https://doi.org/10.1016/j.energy.2022.126099>