

# Benchmarking Transformer Architectures for Multi-Horizon Industrial Load Forecasting: A Large-Scale Study with Weather Integration

Narender, Namita Mittal, and Subhrajit

Department of Computer Science and Engineering,  
Malaviya National Institute of Technology, Jaipur, India

**Abstract.** Industrial electricity load forecasting has always been a complex problem due to irregular production cycles, seasonal factors, and heterogeneity in consumption patterns across various manufacturing sectors. Although transformer-based models have shown significant promise in time series forecasting tasks, no systematic testing has been performed on large-scale, multi-sector industrial datasets. This paper makes a move toward filling that gap by offering a comprehensive benchmarking exercise involving the PatchTST, FEDformer, HALO, and DTformer transformer architectures over such an extensive industrial dataset. It applies a rigorous experimental evaluation with statistical significance testing to determine that two of these architectures, DTformer and HALO, significantly outperform baseline approaches, with DTformer achieving even higher accuracy. It is, therefore, newer transformer architectures that have real promise for getting dependable and scalable solutions in industrial load forecasting. Their use can lead to more efficient energy management and improved demand response planning in modern smart grids.

## 1 Introduction

Accurate electricity load forecasting is important in managing industrial energy consumption and operations. With accurate forecasts, industries can schedule their equipment properly, thereby achieving demand response and reducing operating costs. This supports real-time control as well and makes reliable forecasts a requirement for smart grids. Industrial consumption is more complex than residential electricity, typically following a regular day-to-day usage pattern. It varies widely between manufacturing sectors, as well as the mode of operation and production schedule, thus making industrial load forecasting nonlinear and complicated.

Conventional approaches, such as ARIMA [1] and SARIMA [2], as well as machine learning techniques like Random Forest [3] and Support Vector Machine (SVM) [4], heavily rely on manual feature engineering and struggle to model the long-term dependencies inherent in industrial time series data. Transformer-based architectures [8] have significantly succeeded in time series forecasting

in recent years. The self-attention mechanism in transformers helps them learn long-term dependencies efficiently. Although transformers have performed well on benchmark datasets such as ETT and Electricity Load Diagrams, their use for large and diverse industrial datasets has not been explored widely.

The paper compares four transformer-based models, PatchTST, FEDformer, HALO, and DTformer, for multi-horizon industrial load forecasting that incorporates weather data to address the identified challenges. The key contributions of this paper can be enumerated as follows: (1) A novel, large-scale, and heterogeneous dataset containing more than 8.2 million half-hourly observations was gathered from 175 industrial smart meters in Jaipur, India, between 2022 and 2024. (2) A comparative analysis was performed among the four transformer models with respect to loss values, robustness, and computational efficiency. (3) The impact of weather-related variables (temperature and season) on the industrial consumption of electricity was examined. (4) The Diebold–Mariano statistical test was used to confirm the significance of the difference in performances among the models and to ascertain their robustness for real-world deployment.

## 2 Related Work

Previous works have proposed modeling short-term electricity demand using statistical approaches, such as Autoregressive Integrated Moving Average (ARIMA) or Seasonal ARIMA (SARIMA) [1] [2]. These methods perform well on stationary data or short-term forecasting, but were not able to grasp the highly irregular and dynamic nature of industrial electricity demand driven by time-varying production schedules and abrupt changes in demand. Moreover, the linear nature of these approaches limited their performance for more complex and non-linear load profiles and their heavy reliance on parameter tuning often reduced their practical applicability. To better model non-linear relationships between load and external drivers, machine learning algorithms such as Random Forests [3] and Support Vector Machines (SVMs) [4] [5] have been proposed, but these approaches still relied on manual feature engineering and lacked the ability to automatically learn temporal correlations. While the introduction of deep learning architectures such as Long Short-Term Memory (LSTM) [6] or Gated Recurrent Units (GRU) [7] alleviated the problem of learning temporal dependencies, recurrent neural networks still suffered from issues such as vanishing gradients, high computational complexity and poor performance in long-term forecasting.

Transformer-based models [8] have been proposed to address these issues by utilizing self-attention, which enables them to more easily memorize long-term dependencies and train in a parallel manner more quickly. Large-scale models, including Informer [9], Autoformer [10], FEDformer [11], PatchTST [12], DTformer [13] and HALO [14] have achieved impressive results in benchmark datasets, such as ETT and Electricity Load Diagrams. Nevertheless, their usage in industrial energy forecasting is relatively scarce so far. Most of these studies focus on residential or small-scale datasets and tend to avoid examining industrial loads, which are diverse in terms of sectors and processes. The effect of

weather conditions on industrial demand, particularly temperature and seasonal variation, is also explored to a lesser extent; however there has been little or no statistical validation of these models for reliability applications.

### 3 Dataset and Preprocessing

#### 3.1 Industrial Smart Meter Dataset and Weather Dataset

This paper used industrial electricity consumption data from 175 smart meters installed at different industries in Jaipur (India). The dataset covers three years (2022–2024) at 30-minute intervals, and includes 8.2 million labels. The meter records 47 features, including voltage (three-phase: VR, VY, VB), current measurements (IR, IY, IB), active power (kW), reactive power (kVAR), power factor, and energy consumption (Wh). Weather data for the same three years, 2022–24, were downloaded through Google Earth Engine to assess the effect of environmental conditions on industrial electricity demand. The weather data set contains daily maximum, minimum, and mean temperature, precipitation, and derived thermal indices.

#### 3.2 Data Cleaning and Data Integration

Energy readings were converted from Wh to kWh for numerical stability, invalid timestamps were removed, and missing values were imputed using time-based interpolation (linear for numerical features and forward/backward fill for categorical ones). All these operations led to a clean and continuous industrial load series. Data from 175 meters were collected at 30-minute intervals and combined by averaging, producing 58,271 unified load observations. Daily weather variables were finally matched with this series through date-based merging, enabling seasonal weather–load analysis.

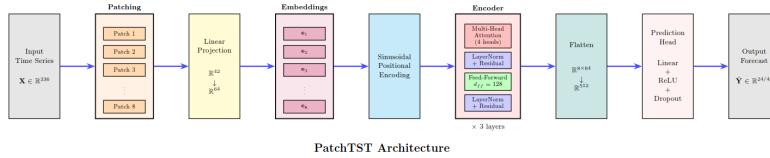
#### 3.3 Sequence Generation and Train–Validation Split

The aggregated load series was transformed into supervised learning sequences using sliding windows, with a 336-step (7-day) lookback to forecast horizons of 24 steps (12 hours) and 48 steps (24 hours). All sequences were min–max normalized to the 0–1 range for stable training of the neural network. The data was split chronologically for the train and test sets, with the last 150 time steps (75 hours) being kept for the test set and the rest used for training. The 24-step horizon resulted in about 57,762 training sequences and 127 testing sequences, with a slight change for the 48-step forecast due to the difference in window length. Sticking to this temporal order makes it possible to test the performance of the models on new, unseen data.

## 4 Methodology

### 4.1 Model Architectures

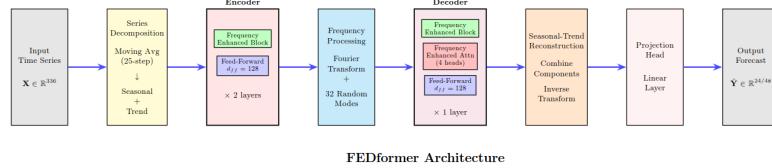
**PatchTST:** The Patch Time Series Transformer employs a patching system that divides the input sequences into non-overlapping subseries tokens, thereby improving efficiency and the information available during computations. The model includes an adaptive patch size mechanism (8 patches in a 336-length sequence) that adjusts automatically based on input length. All patches were linearly projected into a 64-dimensional model space and sinusoidally position-encoded. It consisted of 3 transformer encoder layers, 4 attention heads, and a feed-forward dimension of 128. The multi-head self-attention mechanism processes patch embeddings that encode local and global features within the time sequence. Forecasts of 24 or 48 future time samples were generated using a multi-step prediction head with ReLU activation and dropout regularization.



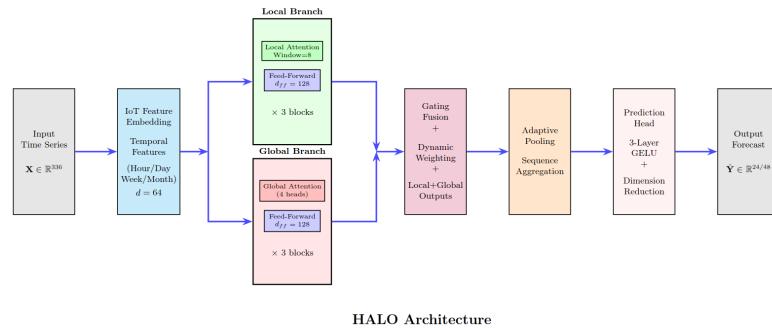
**Fig. 1.** PatchTST Architecture

**FEDformer:** The Frequency-Enhanced Decomposition Transformer utilizes a Fourier block, enabling frequency-domain techniques, as well as trends and seasonal decomposition. It utilizes a 25-timestep moving average window for trend and season decomposition and 32 random frequency modes for increased representation space. Furthermore, randomized time-window sampling (25 timesteps) increases robustness. The seasonal layers in the encoder filter seasonal elements via frequency-enhanced blocks while the decoder applies frequency enhancement within self-attention and cross-attention. The architecture consists of 2 encoder layers and 1 decoder layer, each having 4 attention heads and 128 in the feed-forward dimension. Its improvement lies in the capability to decompose time series into interpretable and meaningful seasonal and trend terms, along with time periodicity and consistent time.

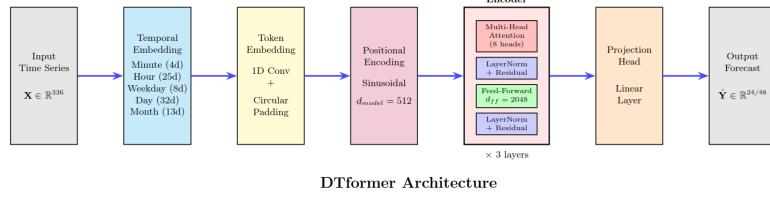
**HALO:** HALO integrates local and global attention schemes in a fusion-based transformer framework. Localized attention, implemented in a window of 8 timesteps, captures the short-term dependencies while the globalized attention scheme captures the long-term periodicity of the quantities of interest. A gating mechanism is in place to dynamically fuse the output of the local and

**Fig. 2.** FEDformer Architecture

global fan-out mechanisms. The overall architecture is characterized by 3 HALO transformer blocks and 64-dimensional embeddings with 4 attention heads using a feed-forward 128-dimensional element. IoT-specific embedding mechanisms were implemented to extract temporal features, such as hours, days, weeks, and months. The multistep prediction head contains GELU activations and an adaptive pooling over sequences, generating forecasts produced by a three-layer prediction module with an ablation on dimensionality.

**Fig. 3.** HALO Architecture

**DTformer:** DTformer incorporates rich temporal embeddings that model dependencies that depend on time. Separate embedding layers are formed for minute (4D), hour (25D), weekday (8D), day (32D), and month (13D) embeddings. Token embeddings were given using 1D convolution with circular padding. Positional vectors were learned using sinusoidal encoding. The encoder consists of 3 layers, with 8 attention heads, where the model space has a width of 512, while the feed-forward model space has a dimensionality of 2048. The virtues of DTformer stem from its ability to encapsulate high-capacity temporal embeddings, which enable it to represent calendar dependencies and cyclical effects in industrial electricity consumption.

**Fig. 4.** DTformer Architecture

## 4.2 Experimental Setup

**Input Sequence Length and Forecast Horizon:** All models were trained using an input sequence of 336 time steps corresponding to 7 days of half-hourly readings. This setup enables the models to learn about weekly and daily usage patterns in industrial electricity demand. Two forecasting horizons were used: 24 time steps (12 hours) for very short-term forecasts and 48 time steps (24 hours) for short-term forecasts.

**Optimization and Training Configuration:** The Adam optimizer was used to train PatchTST, FEDformer, and HALO with a learning rate of  $1 \times 10^{-3}$  and a weight decay of  $1 \times 10^{-5}$ . Due to their moderate parameter sizes, these models exhibited stable convergence under this configuration. DTformer, having a significantly larger model capacity ( $\sim 9.5M$  parameters), required a smaller learning rate of  $1 \times 10^{-4}$  with the Adam optimizer to ensure stability and prevent gradient explosion. Batch sizes of 16 and 32 were systematically evaluated, enabling the selection of a configuration that balanced computational efficiency with reliable convergence for all models. To study training stability, each model was trained for 30, 50, and 70 epochs. Early stopping, monitored using the validation loss, was applied with a patience of 20 epochs to avoid overfitting. Regularization techniques included layer normalization within transformer blocks, gradient clipping to prevent exploding gradients, and dropout rates of 0.1 for PatchTST, FEDformer, and HALO, and 0.05 for DTformer.

## 4.3 Diebold–Mariano (DM) Test

Consider two forecasting models, A and B, producing predictions  $\hat{y}_{A,t}$  and  $\hat{y}_{B,t}$  for the true observed value  $y_t$  at time  $t$ , where  $t = 1, 2, \dots, T$  and  $T$  is the total number of test instances.

The forecasting errors for both models are as follows:

$$e_{A,t} = y_t - \hat{y}_{A,t}, \quad e_{B,t} = y_t - \hat{y}_{B,t} \quad (1)$$

Using Mean Squared Error (MSE) as the loss function, the pointwise losses are computed as:

$$L_{A,t} = e_{A,t}^2, \quad L_{B,t} = e_{B,t}^2 \quad (2)$$

The loss differential at each time step is then defined as:

$$d_t = L_{A,t} - L_{B,t} \quad (3)$$

A positive  $d_t$  indicates that Model A produced a larger squared error than Model B at time  $t$ , implying better performance of Model B.

The Diebold–Mariano (DM) test statistic is defined as:

$$DM = \frac{\bar{d}}{\sqrt{\widehat{\text{Var}}(\bar{d})}} \quad (4)$$

where  $\bar{d}$  represents the sample mean of the loss differential series, and  $\widehat{\text{Var}}(\bar{d})$  denotes its estimated variance accounting for autocorrelation in forecast errors. The corresponding  $p$ -value is obtained from the Student's  $t$ -distribution:

$$p = 2[1 - F_t(|DM|)] \quad (5)$$

where  $F_t(\cdot)$  is the cumulative distribution function (CDF) of the  $t$ -distribution. If  $p < 0.05$ , it indicates that the two Models perform significantly differently; otherwise, their forecasting performances are statistically similar. In this way, the DM test provides a robust statistical basis for pairwise model comparison, which complements conventional measures of standard error, as it evaluates whether the differences in MSE are statistically significant or not.

## 5 Results and Analysis

### 5.1 Modelwise Performance

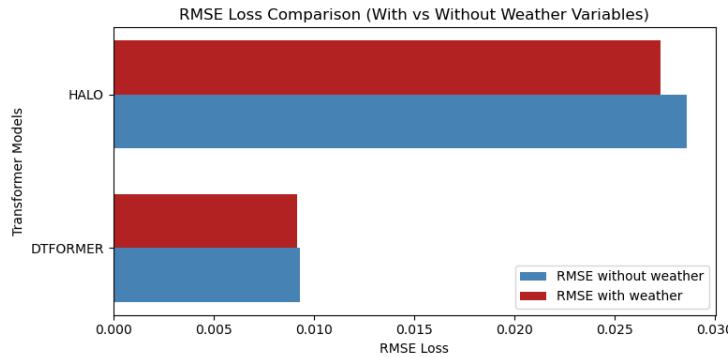
The averaged cumulative Mean Squared Error (MSE) over all 12 configurations of the experiment (sequence length = 336; prediction horizons = 24, 48; batch sizes = 16, 32; epochs = 30, 50, 70) proves that DTformer is more stable and represents the minimum possible value of cumulative average MSE over the entire grid of experiments.

**Table 1.** Model Performance Comparison

Model	Mean Squared Error	Root Mean Squared Error
PatchTST	$0.000588 \pm 0.000427$	$0.022991 \pm 0.007694$
FEDformer	$0.000306 \pm 0.000181$	$0.016782 \pm 0.004933$
HALO	$0.000266 \pm 0.000019$	$0.028612 \pm 0.040888$
DTformer	$0.000088 \pm 0.000023$	$0.009307 \pm 0.001152$

Calendar minute-level embeddings of DTformer explain its superior performance, achieving an MSE of  $0.000088 \pm 0.000023$  compared to the sinusoidal and

frequency-based encodings of PatchTST and FEDformer, respectively. The features directly map onto industrial production schedules by incorporating minute, hour, weekday, day, and month information. Such explicit temporal grounding outperforms HALO’s coarse embeddings, which reach only  $0.000266 \pm 0.000019$  across 12 consistent runs in all configurations. DTformer, with a parameter space of 9.5M, can learn complex interactions across data with heterogeneity represented by 175 meters and regularizes well under stable training setups ( $LR = 1 \times 10^{-4}$ , dropout = 0.05). The significant enhancement of the DTformer’s forecasting performance can directly result in valuable operational advantages for industrial energy management. At both the short and long horizons, enhanced predictability enables industries to make informed decisions about relocating energy-intensive processes from peak tariff periods to non-peak, lower-cost windows, which will lower overall energy costs and improve equipment scheduling.



**Fig. 5.** Weather-Integration

The RMSE loss comparison between DTformer and HALO shows that the error is significantly reduced when weather features are incorporated. HALO’s RMSE decreases substantially from approximately 0.028612 to 0.027914, while DTformer also shows a improvement, with RMSE reducing from about 0.009307 to 0.009148.

**Table 2.** Model’s Efficiency (Input Length 336, Prediction Length 48, Batch Size 32)

Model	Parameters	Inference Time (ms)	Training Time (s/epoch)
PatchTST	106.8K	5.02	16.39
HALO	206.4K	46.31	119.46
FEDformer	1.69M	59.64	234.99
DTformer	9.51M	186.78	503.08

## 5.2 Diebold–Mariano(DM) Test Results

Pairwise Diebold–Mariano testing revealed a clear performance hierarchy among the transformer models:

$$\text{HALO} > \text{DTformer} > \text{PatchTST} > \text{FEDformer}.$$

Configurations				
S. No.	Sequence Length	Prediction Length	Batch Size	No. of Epochs
1	336	24	16	30
2	336	24	16	50
3	336	24	16	70
4	336	24	32	30
5	336	24	32	50
6	336	24	32	70
7	336	48	16	30
8	336	48	16	50
9	336	48	16	70
10	336	48	32	30
11	336	48	32	50
12	336	48	32	70

DTformer vs HALO		DTformer vs PatchTST		DTformer vs FEDformer	
DM Stat	p-value	DM Stat	p-value	DM Stat	p-value
-0.810892e-06	0.400653e49	-0.477181e3	0.612050e03	-0.7004e46e39	0
1.971747e14	0.604893e33B	-3.49798e494	0.001933e07	-10.1212e066	0
-0.664272e001	0	-12.2544e08	0	-17.4651e041	0
-3.405058e001	0.000674e26	2.39986e02	0.000926e17	-21.6981e38	0
7.374947e15	1.641e-12	-7.30561e59	2.765e-10	-10.9920e992	0
-4.125384e42	0.23742e83	-13.8076e99	0	-8.70052e071	0
1.819474e18	0.064909e007	1.78323e54	0.074206e39	-17.9817e273	0
9.265234e05B	0	-9.00431e74	0	-12.8721e784	0
-1.294002e24	0.169564e05	-9.36482e393	0	-23.0735e549	0
-7.364002e95	6.211e-14	-16.7679e025	0	-6.92329e78	4.431e-10
2.029002e12	0.044818e0	-3.88689e36	0.000104e49	-27.3793e061	0
1.471484e05	2.313e-10	-7.60037e275	8.441e-10	-27.8805e496	0

HALO vs PatchTST		HALO vs FEDformer		PatchTST vs FEDformer	
DM Stat	p-value	DM Stat	p-value	DM Stat	p-value
1.016793e55	0.288191e32	-8.4160e073	0	-8.6098e061	0
-6.842088e5	7.793e-12	-10.8620e169	0	-10.6282e997	0
-11.7008e58	0	-14.7961e435	0	4.79374e337	1.921e-9
9.235283e37	0	-26.7070e396	0	-36.0274e334	0
-10.4083e7	0	-14.0479e044	0	-3.9546e492	7.661e-7
-45.27688e3	0	-9.7921e193	0	-2.03664e891	0.03348e72
-1.223168e58	0.23212e7997	-18.8279e944	0	-16.7396e429	0
-9.44742e37	0	-20.3104e077	0	7.5451e807	4.223e-14
-9.85661e47	0	-22.8391e049	0	-4.4318e635	0
14.6699e89e	0	-4.0738e183	6.661e-10	-0.0107e9029	0.152339972
-9.060004e79	0	-45.5689e504	0	-24.7695e55	0
-10.000000e32	0	-48.3285e593	0	-16.7341e771	0

HALO		PatchTST	
DM Stat	p-value	DM Stat	p-value

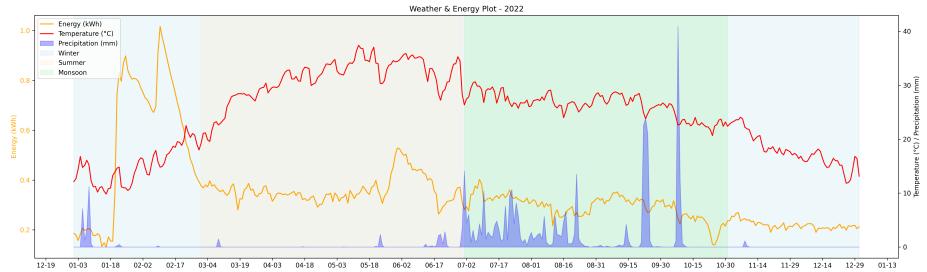
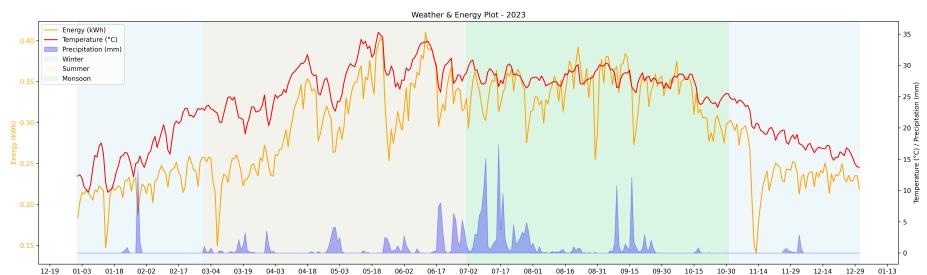
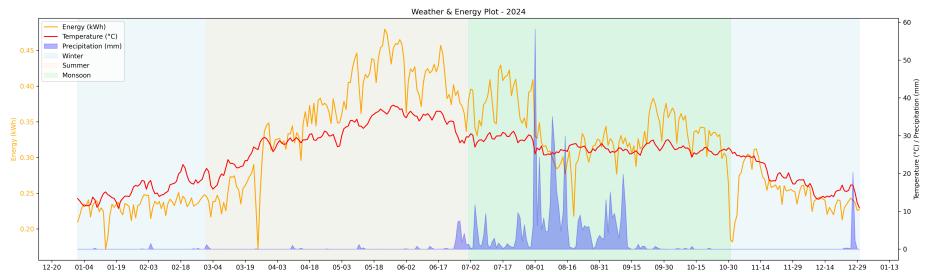
Fig. 6. DM-Stats and p-value

DTformer's comparable performance to HALO can be attributed to HALO's local attention, which captures short-term fluctuations, and its global attention, which models long-term periodicity. This fusion-based design allows HALO to effectively extract key temporal patterns in industrial load data, despite having far fewer parameters. Its lightweight architecture, combined with temporal embeddings for hours, days, weeks, and months, enables HALO to represent the same dominant seasonal and operational dynamics that DTformer models within a larger framework.

## 5.3 Weather–Load Trend Analysis

The seasonal analysis for 2022–2024 reveals distinct weather-driven trends in industrial power use. Overall, precipitation and temperature exert a major influence on electricity consumption and clear seasonal patterns are observable.

Temperature is the key meteorological driver of industrial demand, exhibiting strong seasonal variations. The monsoon displays the closest correlation ( $r = 0.517$ ,  $p = 3.1 \times 10^{-20}$ ) with demand, followed by winter ( $r = 0.352$ ,  $p = 1.82 \times 10^{-14}$ ) and summer ( $r = 0.237$ ,  $p = 1 \times 10^{-7}$ ). Lag effects persist across several lags but are particularly pronounced at lag 3 in all seasons, with the monsoon

**Fig. 7.** Weather Integration-2022**Fig. 8.** Weather Integration-2023**Fig. 9.** Weather Integration-2024

again showing the strongest lag 3 correlation ( $r = 0.441$ ). Precipitation does not influence winter ( $r = -0.059, p = 0.211$ ) or monsoon ( $r = -0.050, p = 0.404$ ), and has only a very small (though non-negligible) impact during summer ( $r = 0.129, p = 0.004$ ). If correlation alone determined behaviour, summer average loads should be the largest and their variations the smallest; however, this is not observed in the data. While seasonal averages broadly follow the expected trend, with monsoon loads being the highest, at an average demand of 137.7 kWh (standard deviation = 31.6), summer loads show the greatest variability, with a standard deviation of 53.8 kWh.

## 6 Conclusion and Future Work

### 6.1 Model Performance Comparison

DTformer achieved the lowest prediction error (MSE:  $0.000088 \pm 0.000023$ , RMSE: 0.0093), based on a comparative evaluation involving PatchTST, FEDformer, HALO and DTformer on aggregated industrial smart meter data with integrated weather information. This represents an improvement of 85.034% over PatchTST, 71.242% over FEDformer, and 66.917% over HALO. The Diebold-Mariano statistical tests further confirmed that DTformer and HALO performed significantly better than PatchTST and FEDformer, with HALO occasionally matching or marginally surpassing DTformer in certain configurations. DTformer have high predictive accuracy, they are still not easy to implement in real-time industrial environments due to their heavy computational requirements and long inference times. PatchTST enables faster inference but sacrifices accuracy, making it a less viable option in situations where accurate predictions are crucial. These issues underscore the need for model compression and optimization to make high-performance transformers feasible for deployment. The second limitation of the work is the reliance on a single averaged time series that represents all industrial sites. While this aggregation results in a cleaner, less noisy signal that is ideal for regional forecasting and grid-level planning, it also masks site-specific variations, peak events, changes at the equipment level, and operational schedules. As a result, the models cannot capture industry-specific patterns or identify anomalies, thus diminishing their applicability to site-level operational decision-making.

### 6.2 Future Work

In order to facilitate the use of HALO and DTformer in real-time industrial scenarios, model compression and acceleration techniques research will include knowledge distillation, sparse or low-rank attention, token reduction, and factorised temporal embeddings to attenuate model size and inference time while still preserving accuracy. Federated learning is another option that allows privacy protection in multi-site collaboration, but conversing and system heterogeneity issues have to be resolved first. Integration of these optimized models with

SCADA and smart-grid systems will be a step further in enabling extensive, real-time forecasting. Future work will overcome the limitations of an average load series by incorporating data from several cities, different climates, and various industrial categories to create cluster- or industry-specific forecasting models. Hybrid architectures with site-specific output heads might also enable transformers to keep global patterns while understanding localized behaviour more efficiently.

## References

1. Nepal, B., Yamaha, M., Yokoe, A., Yamaji, T.: Electricity load forecasting using clustering and ARIMA model for energy management in buildings. *Jpn. Archit. Rev.* **3**, 62–76 (2020). <https://doi.org/10.1002/2475-8876.12135>
2. Liu, X., Lin, Z., Feng, Z.: Short-term offshore wind speed forecast by seasonal ARIMA-A comparison against GRU and LSTM. *Energy* **227**, 120492 (2021). <https://doi.org/10.1016/j.energy.2021.120492>
3. Breiman, L.: Random forests. *Mach. Learn.* **45**, 5–32 (2001)
4. Selakov, A., Ilic, S., Vukmirovic, S., Kulic, F., Erdeljan, A., Gorecan, Z.: A comparative analysis of SVM and ANN based hybrid model for short term load forecasting. In: IEEE Transmission and Distribution Conference and Exposition (T&D), pp. 1–6. IEEE, USA (2012)
5. Saini, L.M., Aggarwal, S.K., Kumar, A.: Parameter optimization using genetic algorithm for support vector machine-based price forecasting model in national electricity market. *IET Gener. Transm. Distrib.* **4**(1), 36–49 (2010)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997)
7. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv:1412.3555* (2014)
8. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 30, pp. 5998–6008. Curran Associates (2017)
9. Zhou, H., et al.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proc. AAAI Conf. Artificial Intelligence, vol. 35(12), pp. 11106–11115 (2021)
10. Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 34, pp. 22419–22430. Curran Associates (2021)
11. Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R.: FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In: Int. Conf. Machine Learning (ICML), pp. 27268–27286. Baltimore, MD, USA (2022)
12. Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J.: A time series is worth 64 words: Long-term forecasting with transformers. In: Int. Conf. Learning Representations (ICLR), pp. 11848–11867 (2023)
13. Fan, J., et al.: Optimizing attention in a transformer for multihorizon, multienergy load forecasting in integrated energy systems. *IEEE Trans. Ind. Inform.* **20**(8), 10238–10248 (2024). <https://doi.org/10.1109/TII.2024.3361949>
14. Pan, C., Zhang, C., Ngai, E.C.H., Liu, J., Li, B.: HALO: HVAC load forecasting with industrial IoT and local-global-scale transformer. *IEEE Internet Things J.* **11**(17), 28307–28320 (2024). <https://doi.org/10.1109/JIOT.2024.3356410>