Proceedings of CONF-MLA 2026 Symposium: Learning and Decision Making in Multi Agent Software Systems
DOI: 10.54254/2755-2721/2026.BA36296

A Review of Transformer Models for Time Series Forecasting

Xuhui Ren

School of Mathematics and Data Science, Jiangnan University, Wuxi, China
Renxuhui2027@126.com

Abstract. Long-term time series forecasting is important in energy scheduling, traffic management, meteorological monitoring, and industrial operations. However, conventional statistical models and recurrent neural networks have limitations in modeling long-range dependencies, complex periodic patterns, and multivariate relationships. This review compares three representative Transformer-based models—Informer, Autoformer, and PatchTST—through literature synthesis and comparative analysis. Informer reduces long-sequence computation through sparse attention, Autoformer strengthens periodic modeling through series decomposition and Auto-Correlation, and PatchTST improves input representation through patch-based tokenization. Public results on the Electricity dataset show that PatchTST outperforms earlier Transformer-based models, while DLinear achieves comparable errors. These findings indicate that model complexity is not the sole determinant of forecasting performance and that look-back windows and experimental protocols also affect model comparisons. The review further discusses computational cost, non-stationarity, and cross-variable modeling.

Keywords: Transformer, time series forecasting, Informer, Autoformer, PatchTST

1. Introduction

Time-series data are widely generated in energy, transportation, meteorological, and industrial systems. Their trends, periodic changes, sudden fluctuations, and interactions between variables make long-term forecasting difficult. Traditional methods such as ARIMA are simple and interpretable, but their assumptions of linearity and stationarity limit their performance on nonlinear and multivariate data [1]. RNNs, LSTMs, and GRUs can model nonlinear relationships more effectively, but their sequential structure restricts parallel computation and makes long-range information harder to capture [1].

Transformer provides another approach by using self-attention to connect different temporal positions directly while allowing parallel processing [2]. However, full attention becomes expensive when the input sequence is long, and the original architecture does not explicitly model common time-series characteristics such as trends and periodicity. Informer, Autoformer, and PatchTST address these problems from different perspectives: sparse attention, temporal decomposition, and patch-based representation, respectively [3-5].

This review compares the three models in terms of their mechanisms, reported results, and experimental settings. It also considers look-back windows, data splits, normalization, and

DOI: 10.54254/2755-2721/2026.BA36296

hyperparameter tuning, since these factors may affect the conclusions drawn from model comparisons.

2. Fundamentals of time series forecasting and transformer models

2.1. Basic concepts and typical tasks in time series forecasting

A time series is a sequence of observations recorded in chronological order, such as electricity load, traffic flow, temperature, and equipment status. Forecasting uses historical observations to estimate future values while accounting for trends, seasonality, periodic patterns, fluctuations, and random disturbances [1]. Tasks may be univariate or multivariate, single-step or multi-step, and short-term or long-term. Because long-term and multi-step forecasting are more sensitive to distant information, accumulated errors, and distribution shifts, input length, forecast horizon, sampling frequency, and train–test splits should be considered when comparing results across studies.

2.2. Conventional time series forecasting methods and their inherent limitations

Statistical methods such as moving averages, exponential smoothing, and ARIMA are computationally efficient and interpretable but rely strongly on linearity and stationarity assumptions, limiting their ability to model nonlinear dynamics and multivariate interactions [1]. RNNs, LSTMs, and GRUs improve nonlinear modeling, while DeepAR supports probabilistic forecasting across related series [6]. However, recurrent networks remain limited by sequential computation and long-range information transmission. Temporal Fusion Transformer improves multi-horizon forecasting and interpretability but increases architectural and tuning complexity [7]. These limitations motivate the use of Transformer architectures that support parallel processing and direct long-range dependency modeling.

2.3. Core transformer architecture and its suitability for time series forecasting

The original Transformer mainly consists of self-attention, multi-head attention, positional encoding, and feed-forward networks [2]. The input is projected into query Q, key K, and value V, and scaled dot-product attention is defined in Eq. (1). Self-attention directly captures dependencies between temporal positions while supporting parallel computation, providing shorter information paths than recurrent networks [8].

$$\mathrm{Attention}\left(\mathrm{Q},\mathrm{K},\mathrm{V}\right)=\mathrm{softmax}\left(\frac{\mathrm{QK}^{\mathrm{T}}}{\sqrt{\mathrm{d_k}}}\right)\mathrm{V} \tag{1}$$

The original Transformer nevertheless has two major limitations. First, full attention has quadratic computational and memory complexity with respect to sequence length. Second, positional encoding provides order information but does not explicitly capture trends, periodicity, or local continuity. The competitive performance of DLinear further indicates that long-range dependency modeling alone does not determine forecasting performance [9]. These limitations motivate three representative directions: Informer improves attention efficiency, Autoformer models temporal structure, and PatchTST improves input representation [3-5].

DOI: 10.54254/2755-2721/2026.BA36296

3. Representative transformer-based models for time series forecasting

3.1. Informer: long-sequence optimization based on sparse attention

Informer mainly aims to reduce the high computational cost of self-attention for long input sequences [3]. ProbSparse attention does not treat all queries equally. Instead, it gives more attention to queries that are likely to produce more informative attention distributions. In the original study, query sparsity is described using Kullback–Leibler divergence and then approximated by a max–mean criterion, as shown in Eq. (2). Queries with larger differences between their maximum and average relevance scores are more likely to enter the Top-u set. The main computation is therefore concentrated on approximately u=c ln L_Q active queries, reducing the complexity to about O(L log L) [3]. Self-attention distilling further reduces intermediate representations, while the generative decoder outputs the forecasting horizon in one forward pass.

$$M(q_i, K) = \max_j s_{ij} - \frac{1}{L_K} \sum_{j=1}^{L_K} s_{ij} \tag{2}$$

The sparsity assumption also introduces a limitation. If useful information is spread across many moderately related positions, some dependencies may receive less attention after query selection. This is more likely to occur in noisy data or series with weak periodicity. Therefore, Informer is mainly valuable for improving the efficiency of long-sequence processing rather than guaranteeing better accuracy on every dataset.

3.2. Autoformer: series decomposition and auto-correlation modeling

Autoformer focuses more directly on the temporal structure of time-series data [4]. It introduces progressive decomposition into the network to separate trend information from seasonal or residual components. The Auto-Correlation mechanism then searches for repeated patterns across different time delays. It uses the fast Fourier transform to estimate correlations, selects important delays, and combines the corresponding shifted representations according to their correlation strengths, as summarized in Eq. (3).

$$TDA(V) = \sum_{i=1}^{k} w_i \mathrm{Roll}(V, \tau_i) \tag{3}$$

Compared with point-wise self-attention, Autoformer places greater emphasis on repeated temporal patterns. When daily, weekly, or seasonal cycles remain relatively stable, these historical patterns can provide useful information for forecasting. However, sudden weather changes, holidays, or changes in operating conditions may weaken previously identified periodic relationships. The decomposition window also needs to be chosen carefully: a large window may smooth short-term changes, while a small one may fail to capture a stable trend. Therefore, Autoformer is effective for regular temporal patterns but remains sensitive to periodic stability and decomposition settings.

3.3. PatchTST: patch-based representation and channel-independent modeling

PatchTST takes a different approach by changing the input representation rather than redesigning the attention mechanism [5]. Consecutive observations are grouped into patches, and each patch is treated as a token. This reduces the number of tokens processed by the Transformer while retaining

DOI: 10.54254/2755-2721/2026.BA36296

local temporal information. As a result, longer historical windows can be used without increasing attention cost as quickly as when every time point is treated as a separate token.

PatchTST also uses channel-independent modeling, where different variables are processed separately while sharing model parameters. This design can reduce interference between variables, but it may also weaken the modeling of cross-variable lead–lag relationships. Patch length creates another trade-off. Short patches preserve more temporal detail but reduce the computational advantage, whereas long patches may combine abrupt changes and stable periods in the same segment. Therefore, an appropriate patch size is needed to balance local information, historical context, and computational cost [5].

Overall, these three models should not be regarded as simple successive upgrades. Informer mainly focuses on computational efficiency, Autoformer emphasizes temporal structure, and PatchTST changes the way time-series inputs are represented [3-5].

Table 1. Mechanistic comparison of Informer, autoformer, and PatchTST

Model	Primary Improvement Direction	Core Mechanism	Main Advantages	Main Limitation
Informer	Attention efficiency	ProbSparse attention and self-attention distilling	Reduces the computational cost of long-sequence modeling	Distributed weak dependencies may be underrepresented
Autoformer	Temporal-structure modeling	Series decomposition and Auto-Correlation	Explicitly exploits trend and periodic patterns	Periodicity shifts and decomposition parameters may affect performance
PatchTST	Input representation	Patching and channel-independent modeling	Captures local patterns while enabling longer look-back windows	Sensitive to patch configuration and limited in cross-variable interaction modeling

4. Applications and comparison of representative transformer-based time-series models

4.1. Energy load forecasting

Energy load exhibits daily, weekly, and seasonal patterns while also being influenced by temperature, holidays, and production activities. Informer, Autoformer, and PatchTST have been evaluated on datasets such as Electricity and ETT [3-5], emphasizing long-window efficiency, periodic modeling, and local-pattern representation, respectively. However, benchmark datasets cannot fully capture distribution shifts caused by extreme weather or operational changes, so lower benchmark errors do not necessarily guarantee better real-world deployment performance.

DOI: 10.54254/2755-2721/2026.BA36296

4.2. Traffic flow and meteorological forecasting

Traffic flow contains daily and weekly periodicity but is also affected by accidents, weather, construction, and road-network topology, meaning that temporal modeling often needs to be combined with spatial or graph-based mechanisms. Meteorological series similarly combine periodicity and abrupt changes with spatial propagation, topography, and physical constraints. Informer, Autoformer, and PatchTST can model different temporal properties [3-5], but real-world traffic and weather forecasting generally requires additional spatial or domain-specific information.

4.3. Comprehensive comparison, quantitative evidence, and experimental settings

Long-term forecasting commonly uses benchmark datasets such as ETT, Electricity, Traffic, and Weather, with MSE and MAE as standard evaluation metrics [3-5, 9].

Table 2 presents publicly reported Electricity results from the PatchTST study [5]. At the 96-step horizon, PatchTST/64 achieves an MSE/MAE of 0.129/0.222, compared with 0.196/0.313 for Autoformer and 0.304/0.393 for Informer. At 720 steps, the corresponding results are 0.197/0.290, 0.236/0.342, and 0.351/0.427 [5], indicating a clear benefit from patch-based input representation.

Table 2. Publicly reported long-term forecasting results on the electricity dataset (MSE/MAE)

Forecast Horizon	Informer	Autoformer	PatchTST/64	DLinear	Metric
96	0.304 / 0.393	0.196 / 0.313	0.129 / 0.222	0.140 / 0.237	MSE / MAE
720	0.351 / 0.427	0.236 / 0.342	0.197 / 0.290	0.203 / 0.301	MSE / MAE

Note: PatchTST/64 uses a 512-step look-back window, whereas DLinear uses 336 steps; Informer and Autoformer were re-evaluated under multiple look-back settings [5].

DLinear remains highly competitive, achieving 0.140/0.237 at 96 steps and 0.203/0.301 at 720 steps, with MSE gaps of only 0.011 and 0.006 from PatchTST [5, 9]. This shows that complex attention is not necessary for competitive performance. Because the models use different experimental settings, error differences should be interpreted together with look-back length, normalization, and hyperparameter tuning rather than attributed solely to architecture [5].

5. Current challenges and future research directions

5.1. Computational cost and model complexity

Long input sequences increase both computational cost and model complexity. Informer, Autoformer, PatchTST, and FEDformer improve efficiency or temporal modeling through different mechanisms [3-5, 10], but additional components are useful only when their gains justify training, memory, and tuning costs. The competitiveness of DLinear further suggests that simple baselines should remain an important reference [5, 9]. Model comparisons should therefore consider resource costs in addition to MSE and MAE.

5.2. Data adaptation, generalization, and interpretability

Real-world time series are often non-stationary and contain missing values, anomalies, and changing cross-variable relationships. PatchTST may miss cross-variable lead–lag dependencies, whereas iTransformer directly models inter-variable relationships [12]. Attention weights indicate association

DOI: 10.54254/2755-2721/2026.BA36296

rather than causality, while TFT improves interpretability at the cost of additional complexity [7]. Generalization should therefore be tested under rolling windows, anomalies, and distribution shifts.

5.3. Trends toward lightweight, hybrid, and multi-task forecasting

Recent research increasingly focuses on how time series should be represented rather than simply designing more complex attention mechanisms. PatchTST changes temporal token scale, TimesNet transforms one-dimensional time series into two-dimensional representations to capture multi-periodic variations [11], and iTransformer treats variables as tokens [12]. These developments suggest that input organization and inductive bias may be as important as network depth. Future work should also investigate self-supervised and multi-task learning while adopting more standardized reporting of look-back settings, normalization procedures, random seeds, and computational resources.

6. Conclusion

This review examines three core issues: why Transformer architectures are suitable for time-series forecasting, how Informer, Autoformer, and PatchTST improve the original architecture, and whether greater architectural complexity necessarily leads to better performance. Transformer supports parallel sequence processing and long-range dependency modeling [2], while Informer improves computational efficiency through sparse attention [3], Autoformer explicitly models temporal structure through series decomposition and Auto-Correlation [4], and PatchTST improves input representation through patching and channel-independent modeling [5]. Public results on the Electricity dataset show that PatchTST outperforms earlier Transformer-based models, while DLinear achieves comparable errors [5, 9]. These findings indicate that model complexity alone is not a reliable predictor of forecasting performance and that input representation, look-back window, and experimental protocol also play important roles.

This review has two main limitations. First, the quantitative analysis relies primarily on publicly reported results, and not all models were retrained under a unified implementation and identical hyperparameter settings. Second, the analysis focuses mainly on Informer, Autoformer, and PatchTST and therefore does not cover all recent architectures. Future research should establish more standardized experimental protocols, improve robustness to non-stationarity and cross-variable interactions, and further investigate the relationship between input representation and model architecture. These directions can help distinguish genuine architectural improvements from gains introduced by experimental settings and improve the reliability and reproducibility of long-term forecasting research.

References

[1] Wen, Q., Zhou, T., Zhang, C., et al. (2023). Transformers in time series: A survey. *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 6778-6786.
[2] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*.
[3] Zhou, H., Zhang, S., Peng, J., et al. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence, 35*(12), 11106-11115.
[4] Wu, H., Xu, J., Wang, J., & Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems, 34*, 22419-22430.
[5] Nie, Y., Nguyen, N. H., Sinthong, P., & Kalagnanam, J. (2023). A time series is worth 64 words: Long-term forecasting with transformers. *International Conference on Learning Representations*.

DOI: 10.54254/2755-2721/2026.BA36296

[6] Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting, 36*(3), 1181-1191.
[7] 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.
[8] Ahmed, S., Nielsen, I. E., Tripathi, A., et al. (2022). Transformers in time-series analysis: A tutorial. arXiv preprint arXiv:2205.01138.
[9] 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.
[10] Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., & Jin, R. (2022). FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. *Proceedings of the 39th International Conference on Machine Learning*, 27268-27286.
[11] Wu, H., Hu, T., Liu, Y., et al. (2023). TimesNet: Temporal 2D-variation modeling for general time series analysis. *International Conference on Learning Representations*.
[12] Liu, Y., Hu, T., Zhang, H., et al. (2024). iTransformer: Inverted transformers are effective for time series forecasting. *International Conference on Learning Representations*.