Proceedings of the 8th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/2026.35997

Neural Time Series Forecasting: A Problem-Driven Survey from Data Challenges to Model Choice

Chuting Wen

Bangor College, Central South University of Forestry and Technology, Changsha, China
17700293371@163.com

Abstract. Surveys of time series forecasting usually proceed by model family, from recurrent networks to Transformers and foundation models. That chronology is useful, but it gives limited guidance when the practical question is why a forecast fails. This article instead organizes the literature around five recurring difficulties: nonlinear and nonstationary behavior, contamination and structural breaks, uncertainty, long contexts and cross-variable dependence, and limited target-domain data. Studies published between 2014–2025 are compared through the assumptions they make, the settings in which they work, and the failure modes they leave unresolved. The review covers recurrent and probabilistic models, decomposition methods, robust and uncertainty-aware forecasting, Transformer variants, theory-guided methods, and foundation models. Quantitative results are used only when the underlying protocol is sufficiently clear. Across these families, the evidence supports a restrained workflow: begin with a strong simple baseline, identify the dominant source of error, and add complexity only when it addresses that source.

Keywords: Time series forecasting, Neural forecasting, Probabilistic forecasting, Transformer, Foundation model, Uncertainty quantification

1. Introduction

Forecasts schedule electricity generation, set inventory levels, anticipate traffic demand, and detect industrial faults. The task is mathematically similar across these applications, but the source and consequence of error are not. Electricity demand may remain seasonal until a heatwave changes its relation to temperature. Financial returns are noisy and weakly predictable. A sensor stream may drift before resetting after maintenance. Treating these cases as interchangeable benchmark problems obscures the modeling choices that matter in practice.

Neural forecasting has expanded because learned models can represent nonlinear relationships and share information across related series. The literature is often narrated by architecture: recurrent networks, convolutional models, Transformers, and pre-trained systems. This chronology does not map neatly to model selection. Long context and complex cross-variable dependence may justify attention, whereas a short, stable univariate signal may not. Low average point error is also insufficient when a decision depends on an upper quantile or calibrated interval.

Five recurring difficulties organize the discussion: nonlinear and nonstationary dynamics; contamination, outliers, and structural breaks; uncertainty that should be represented rather than

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.35997

hidden in a point estimate; long contexts and multivariate dependence; and limited target-domain data. The categories deliberately cross architectural boundaries. A patched Transformer, a decomposition network, and a foundation model may all target long-horizon forecasting, yet they differ in their assumptions about temporal structure, transfer, and computation.

Coverage centers on neural and hybrid forecasting studies published from 2019–2025, supplemented by earlier work that still anchors probabilistic forecasting or calibration. Priority is given to methods that changed forecasting practice, exposed the limits of stronger-looking architectures, or introduced evaluation principles relevant to the five challenges above. The discussion is not an inventory of every model. Anomaly detection, imputation, and classification appear only where they alter forecasting design or the interpretation of results.

2. A problem-oriented taxonomy of forecasting challenges

Nonlinearity and non-stationarity should not be treated as synonyms. Nonlinearity concerns the mapping from past observations and covariates to future values. Non-stationarity arises when that mapping, the marginal distribution, or seasonal pattern changes over time. Greater network capacity may fit a complicated response, but it does not repair a changed data-generating process. Normalization, rolling retraining, explicit covariates, decomposition, or change-aware validation may then matter more than another attention layer.

Noise also has several forms. Small measurement error, heavy-tailed innovations, isolated outliers, and genuine regime changes call for different responses. Smoothing may reduce random variation, but it can also remove an operationally important peak. Robust losses limit the influence of extreme residuals, while ensembles reduce variance across specifications. Neither approach addresses a persistent break unless the training window, covariates, or model itself is updated after the change.

Forecast uncertainty can be attributed to different sources. Aleatoric uncertainty describes irreducible variation in future observations; epistemic uncertainty reflects limited knowledge of parameters or model structure. The required output depends on the decision. Some applications need conditional quantiles, others need prediction intervals, and a smaller group require a full predictive distribution.

Point accuracy and calibration should be examined together. A nominal 95% interval should miss roughly one observation in twenty over an appropriate evaluation sample. After a regime shift, a model may retain a respectable MAE while missing one observation in ten. Conversely, high coverage can be achieved with intervals too wide to guide action. Coverage should therefore be reported with interval width or a proper scoring rule.

Long-horizon and multivariate settings impose a different burden. The model must preserve useful information over many time steps without excessive computation and decide when variables should interact. Sampling frequencies may differ, and much of the recorded history may be irrelevant. Limited target-domain data adds another complication. Transfer and foundation models address scarcity, but raise questions about pretraining overlap, scale mismatch, latency, and the stability of zero-shot rankings.

DOI: 10.54254/2755-2721/2026.35997

3. Methodological responses to forecasting challenges

3.1. Recurrent and hybrid sequence forecasting models

Recurrent neural networks summarize a sequence through a hidden state. LSTM and GRU units mitigate the gradient problems of simple RNNs and remain practical for short- and medium-context forecasting, particularly when observations arrive online. They are well supported in forecasting libraries, accommodate covariates naturally, and fit pipelines that already produce forecasts autoregressively.

DeepAR trains one autoregressive recurrent model across related series and outputs the parameters of a chosen likelihood [1]. A proprietary Amazon dataset (denoted ec in the original paper) contained 500,000 weekly item-sales series. The authors report that training and prediction completed in under 10 hours on one AWS p2.xlarge instance with four CPUs and one GPU, with accuracy improvements of around 15% over the compared methods [1]. These figures describe the original experiment, not present-day latency.

This shared autoregressive design also has costs. Sequential decoding limits parallel execution and can carry early errors through the forecast horizon. Tail behavior depends on the chosen output distribution: a negative-binomial likelihood, for example, is a poor fit when observations are not overdispersed, even if the conditional mean is estimated well.

Temporal Fusion Transformer (TFT) combines LSTM-based local encoders with attention, gating, and variable selection [2]. It is discussed here because recurrent layers encode local structure before attention integrates the wider context. TFT is useful when static attributes, known future inputs, and observed covariates must be represented together. Its attention patterns can support diagnosis but are not causal explanations, and the number of interacting components makes tuning difficult on small datasets.

3.2. Decomposition-based and multi-resolution forecasting

Decomposition can make temporal structure easier to model, although it does not guarantee interpretability.

N-BEATS represents forecasts through fully connected residual blocks and, in its interpretable form, basis expansions for trend and seasonality [3]. This configuration is most convincing when an additive description is plausible. The generic version is better viewed as a strong univariate forecaster because its learned bases need not correspond to physical trend or calendar effects. NHITS adds hierarchical interpolation and multi-rate sampling [4]. Coarse blocks capture low-frequency shape, while finer blocks recover local variation, which is useful when broad trajectory matters more than every short fluctuation.

Related ideas appear inside Transformer architectures. Autoformer repeatedly separates trend and seasonal components and replaces standard self-attention with an autocorrelation mechanism [5]. The design targets recurring lag structure rather than unrestricted token-to-token similarity.

FEDformer moves part of the computation to the frequency domain and selects Fourier or wavelet components intended to retain global periodic behavior [6]. Autoformer and FEDformer can be effective on regular seasonal benchmarks, but their decompositions may change abruptly after a regime shift. The resulting components are best viewed as learned representations, not direct measurements of physical processes.

DOI: 10.54254/2755-2721/2026.35997

3.3. Long-horizon and multivariate forecasting: transformers and linear baselines

Patch-based and variable-token Transformers differ mainly in what they treat as the basic modeling unit. PatchTST groups neighboring observations into patches and applies channel-independent attention, reducing the effective sequence length and postponing cross-variable mixing [7]. TimesNet reshapes temporal patterns into two-dimensional representations in order to model variation within and across periods [8]. iTransformer treats variables, rather than time points, as tokens and uses attention to learn their relations [9]. These choices place the inductive bias in different parts of the data structure.

DLinear provided an influential counterexample to architecture-first model selection [10]. In Table 2 of Zeng et al. [10], its MSE on ETTh1 at a 720-step horizon is 0.472, compared with 1.181 for the Informer result reproduced in the same table, a reduction of about 60% under that protocol. On Traffic at 720 steps, the corresponding values are 0.466 and 0.864, a reduction of about 46% [10]. The comparison depends on the preprocessing and evaluation setup, so it supports careful baseline testing rather than a universal ranking of models.

This is the point: complexity must earn its place.

Transformer variants are most defensible when the context is genuinely long, interactions among variables are nonlocal, and the training set is large enough to support stable optimization. With short histories, weak cross-variable dependence, or strict latency limits, a linear, MLP, recurrent, or decomposition model may be preferable. Any claimed gain should be checked against simple baselines using the same split, normalization, and forecast horizon.

3.4. Robust and uncertainty-aware forecasting

Consider day-ahead power scheduling. A forecast can achieve a low monthly MAE and still miss an evening peak that triggers an expensive reserve purchase. In such settings, average error needs to be accompanied by tail-sensitive or event-sensitive measures.

Robust forecasting begins with a judgment about which deviations should be down-weighted. Huber and quantile losses reduce the influence of large residuals during training. Median-based scaling and cautious winsorization can protect preprocessing, although aggressive clipping may remove the event of interest. Forecast combinations offer a different form of protection: averaging models with distinct errors reduces variance, and feature-based weighting can adapt combinations to individual series [11]. Diversity matters more than model count because a set of highly correlated networks can preserve the same blind spots.

Probabilistic forecasts may be produced by likelihood-based networks, quantile regression, Bayesian approximations, or sampling-based generative models. Their assessment should include proper scores and calibration diagnostics rather than only MAE or MSE [12]. For a nominal 95% interval, empirical coverage should be reported together with interval width; one without the other gives an incomplete picture.

Conformal prediction supplies a model-agnostic calibration layer. Adaptive conformal procedures revise interval widths as recent residual behavior changes [13]. They do not compensate for a poor point forecast, and their guarantees depend on the online assumptions being used. Even so, separating prediction from calibration is valuable because the source of an interval adjustment can be examined directly.

DOI: 10.54254/2755-2721/2026.35997

3.5. Knowledge-guided and foundation-model forecasting

Foundation-model forecasting initially borrowed representations and interfaces from language models. In 2023–2024, Time-LLM reprogrammed a largely frozen LLM using patched numerical inputs [14], while LLMTime serialized observations and prompted an LLM to generate future values [15]. These studies broadened the design space, but their computational demands complicated comparisons with dedicated forecasters.

A second line of work pre-trained models directly on time-series corpora. Chronos quantizes scaled values and trains T5-style probabilistic forecasters [16]. Its Mini, Small, Base, and Large variants contain 20M, 46M, 200M, and 710M parameters. Benchmark II contains 27 held-out datasets not used for training. The 710M model obtained aggregate relative WQL of 0.645, compared with 0.639 for task-specific TFT, and relative MASE of 0.823, compared with 0.810 for PatchTST; lower is better for both [16]. Scores were normalized to Seasonal Naive and geometrically aggregated. Chronos was therefore about 1% worse on WQL and 2% worse on MASE than the corresponding supervised specialist. Zero-shot was competitive, not magical.

Computational cost changes how those results should be read. Chronos reports average per-series inference time on a logarithmic scale and across different hardware, so exact latency comparisons are not reliable [16]. One estimate is explicit. The LLMTime baseline using Llama-2 70B required approximately 0.8 s for each generated observation on a p3dn.24xlarge instance. For the Traffic evaluation, which used 862 series, a 24-step horizon, and 20 samples, the authors estimated 92 hours [16]. A method with acceptable accuracy may therefore remain impractical in an operational forecasting service.

Time-series-specific pretraining then developed along several lines. TimesFM uses a patched decoder trained over variable context and horizon lengths [17], while Moirai accommodates multiple sampling frequencies and variable dimensions within one architecture [18]. Sundial, introduced in 2025, replaces discrete value tokenization with flow matching to produce continuous predictive distributions [19]. TSFM-Bench subsequently compared 14 foundation models on 21 datasets from ten domains in zero-, few-, and full-shot settings [20], shifting the discussion from isolated demonstrations toward common evaluation.

Domain structure offers another route to transfer. Conservation penalties, monotonicity constraints, mechanistic state updates, or bounded outputs can improve extrapolation and reduce data needs when the prior knowledge is credible. Soft constraints let observations correct an imperfect prior; rigid or incomplete constraints may instead create systematic error.

4. Cross-method comparison and model selection

Model selection is more useful as a sequence of tests than as a contest among architecture labels. Seasonal-naive and regularized linear forecasts provide the initial reference. Decomposition becomes attractive when multiple scales or changing seasonal structure dominate; probabilistic outputs are necessary when decisions depend on risk. Attention and variable-token models need evidence of long context or cross-variable dependence, while foundation models are most relevant when target data are scarce or one service must operate across many domains. Accuracy gains should be weighed against training, latency, and maintenance costs, as summarized in Table 1.

DOI: 10.54254/2755-2721/2026.35997

Table 1. Problem-driven model selection guide

Dominant condition	Reasonable starting point	Why it fits	What must be checked
Stable short/medium horizon	Seasonal naive, linear/MLP, LSTM/GRU	Low cost; sufficient local memory	Sensitivity to window length and scaling
Many related series; distribution required	DeepAR [1], TFT [2]	Shared learning; covariates and likelihoods or quantiles	Likelihood choice, calibration, autoregressive error
Strong seasonality or multi-scale horizon	N-BEATS [3], NHITS [4], Autoformer [5]	Separates or learns temporal scales	Component stability after regime changes
Long context or complex multivariate dependence	PatchTST [7], TimesNet [8], iTransformer [9]	Efficient long-range or cross-variable modeling	Gain over DLinear [10], training variance, latency
Outliers, drift, or high operational risk	Robust loss, diverse ensemble, conformal calibration [13]	Limits extreme-residual influence; calibrates intervals	Whether extremes are errors or real events; post-shift coverage
Limited target data or many domains	Chronos [16], TimesFM [17], Moirai [18], Sundial [19]	Zero-/few-shot transfer and reusable infrastructure	Pretraining overlap, scale mismatch, inference cost
Known physical or operational constraints	Theory-guided or hybrid model	Improves data efficiency and extrapolation	Constraint misspecification and transferability

The table also shows that method families solve different parts of the problem. Recurrent and decomposition models make stronger local or additive assumptions and usually need less data; Transformer and foundation models relax those assumptions but must justify the added capacity through long context, multivariate interaction, or transfer. Robust losses and calibration layers cut across this distinction because they change sensitivity to errors and uncertainty rather than representation capacity. Rolling-origin evaluation is generally more informative than a single random split, and the test period should include plausible distributional changes. Metrics must match the decision: MAE is easy to interpret, squared error emphasizes large misses, quantile loss represents asymmetric risk, and proper scores assess full distributions. Latency, memory, retraining frequency, and sensitivity to missing covariates should be reported with predictive accuracy.

5. Research gaps and future directions

Distribution shift remains underrepresented in forecasting evaluation. Under the unified TSFM-Bench protocol, zero-shot time-series pre-trained models outperformed full-shot task-specific models on 10 of 21 datasets [20]. In the 5% few-shot setting, evaluated on 14 long-term forecasting datasets, pre-trained models led the LLM-based and task-specific alternatives on 13 datasets [20]. The pattern is important but uneven: no model dominated every setting, and several pre-trained models improved little or deteriorated when more adaptation data were supplied.

DOI: 10.54254/2755-2721/2026.35997

Standardization can also change familiar rankings. In TSFM-Bench's ETTh1 5% few-shot results, averaged over prediction horizons of 96, 192, 336, and 720 steps, Time-LLM records an MSE of 0.674 and DLinear 0.476 [20]. This does not rule out LLM-based forecasting; it shows how quickly apparent gains can narrow once splits, normalization, sampling, and horizons are aligned. Future benchmarks should compare zero-shot, adapted, and task-specific models under the same protocol and document checks for training-data overlap.

Uncertainty becomes harder when users act on an entire forecast path. Marginal intervals at individual horizons do not represent the joint risk of a trajectory, yet grid operators and inventory planners make path-dependent decisions. Efficient joint calibration, tail-sensitive objectives, and decision-aware scoring remain open problems.

Adaptation costs are seldom reported with the same care as forecast error. Foundation models may reduce task-specific training while increasing inference latency and obscuring when retraining is needed. Parameter-efficient adaptation, retrieval of related series, online updating, and compact edge models are promising, but efficiency claims should include wall-clock time, memory use, and hardware conditions.

Interpretability claims also need an operational test. A useful explanation should separate seasonal evidence, recent anomalies, exogenous drivers, and model uncertainty, then show how that information changes a decision—for example, by rejecting an input, widening an interval, requesting human review, or triggering retraining. Attention maps alone rarely establish that connection.

6. Conclusion

Neural time series forecasting is better understood as a set of responses to distinct data problems than as a progression toward ever larger architectures. Recurrent, decomposition, Transformer, robust, probabilistic, theory-guided, and foundation models each encode assumptions that help in some settings and mislead in others. Strong baselines, realistic temporal shifts, and cost-aware evaluation remain necessary. A defensible workflow identifies the main source of error, uses the least complex model that addresses it, and evaluates accuracy together with calibration and operational constraints. Quantitative comparisons are informative only when the dataset, horizon, aggregation, and hardware conditions are stated.

References

[1] Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting, 36*(3), 1181–1191.

[2] 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.

[3] Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2020). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *Proceedings of the International Conference on Learning Representations*.

[4] 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. In *Proceedings of the AAAI Conference on Artificial Intelligence, 37*(6), 6989–6997. https://doi.org/10.1609/aaai.v37i6.25854

[5] Wu, H., Xu, J., Wang, J., & Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems, 34*, 22419–22430.

[6] Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., & Jin, R. (2022). FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning* (pp. 27268–27286).

[7] Nie, Y., Nguyen, N. H., Sinthong, P., & Kalagnanam, J. (2023). A time series is worth 64 words: Long-term forecasting with transformers. In *Proceedings of the 11th International Conference on Learning Representations*.

DOI: 10.54254/2755-2721/2026.35997

[8] Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., & Long, M. (2023). TimesNet: Temporal 2D-variation modeling for general time series analysis. In *Proceedings of the 11th International Conference on Learning Representations*.
[9] Liu, Y., et al. (2024). iTransformer: Inverted transformers are effective for time series forecasting. In *Proceedings of the 12th International Conference on Learning Representations*.
[10] Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023). Are transformers effective for time series forecasting? In *Proceedings of the 37th AAAI Conference on Artificial Intelligence, 37*(9), 11121–11128.
[11] Montero-Manso, P., Athanasopoulos, G., Hyndman, R. J., & Talagala, T. S. (2020). FFORMA: Feature-based forecast model averaging. *International Journal of Forecasting, 36*(1), 86–92.
[12] Gneiting, T., & Katzfuss, M. (2014). Probabilistic forecasting. *Annual Review of Statistics and Its Application, 1*, 125–151.
[13] Zaffran, M., Féron, O., Goude, Y., Josse, J., & Dieuleveut, A. (2022). Adaptive conformal predictions for time series. In *Proceedings of the 39th International Conference on Machine Learning* (pp. 25834–25866).
[14] Jin, M., et al. (2024). Time-LLM: Time series forecasting by reprogramming large language models. In *Proceedings of the 12th International Conference on Learning Representations*.
[15] Gruver, N., Finzi, M., Qiu, S., & Wilson, A. G. (2023). Large language models are zero-shot time series forecasters. In *Advances in Neural Information Processing Systems, 36*, 19622–19635.
[16] Ansari, A. F., et al. (2024). Chronos: Learning the language of time series. *Transactions on Machine Learning Research*.
[17] Das, A., Kong, W., Sen, R., & Zhou, Y. (2024). A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning* (pp. 10148–10167).
[18] Woo, G., Liu, C., Kumar, A., Xiong, C., Savarese, S., & Sahoo, D. (2024). Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning* (pp. 53140–53164).
[19] Liu, Y., Qin, G., Shi, Z., Chen, Z., Yang, C., Huang, X., Wang, J., & Long, M. (2025). Sundial: A family of highly capable time series foundation models. In *Proceedings of the 42nd International Conference on Machine Learning* (pp. 39295–39317).
[20] Li, Z., et al. (2025). TSFM-Bench: A comprehensive and unified benchmark of foundation models for time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 5595–5606). https://doi.org/10.1145/3711896.3737442