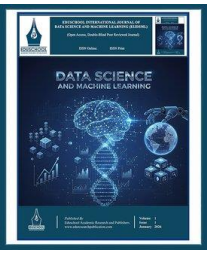
EDUSCHOOL INTERNATIONAL JOURNAL OF DATA SCIENCE AND MACHINE LEARNING (EIJDSML)

(Open Access, Double-Blind Peer Reviewed Journal)

ISSN Online:



Time-Series Foundation Models for Zero-Shot Forecasting

Win Mathew John

Associate professor, PG Department of Computer Applications, Marian College Kuttikkanam (Autonomous), Kerala, India

Article information

Received: 1st April 2026
Received in revised form: 4th May 2026
Accepted: 9th June 2026
Available online: 18th July 2026

Volume: 1
Issue: 2
DOI: https://doi.org/10.5281/zenodo.21411600

Abstract

Forecasting pipelines have long relied on a model trained separately for every dataset, an approach that is expensive to maintain and slow to adapt when new series arrive. Time-series foundation models change this picture. A single network is pretrained once on a very large and varied collection of series, after which it produces forecasts on previously unseen data without any further fitting. This paper surveys the design and evaluation of such models. The series is first split into fixed-length patches that act as tokens, and a Transformer backbone, either decoder-only or encoder-based, predicts future patches. The pretraining objective, tokenization scheme, and probabilistic output head together determine zero-shot quality. Five representative systems are examined, namely TimesFM, Chronos, Moirai, Lag-Llama, and TimeGPT, and their inductive choices are contrasted. Using illustrative metrics aligned with reported behaviour on the Monash archive and GIFT-Eval, the study compares these models against classical baselines such as ARIMA and exponential smoothing and against trained deep networks including N-BEATS, PatchTST, and DeepAR. The evidence indicates that a frozen foundation model often matches a per-dataset deep model while removing training cost at deployment, though a tuned specialist still leads on some series. Open problems remain around external covariates, very long horizons, and distribution shift, and the paper outlines directions that address them. The intent is descriptive synthesis rather than a single deployed benchmark.

Keywords:- Time-Series Forecasting, Foundation Models, Zero-Shot Learning, Transformers, Tokenization, Probabilistic Forecasting, Benchmarking.

I. INTRODUCTION

Demand planning, energy load management, traffic control, and financial monitoring all depend on forecasts of quantities that vary over time. For decades the standard recipe has been to choose a model family, fit its parameters to one dataset, and then refit whenever the data changes. Classical statistical methods such as autoregressive integrated moving average models [1] and exponential smoothing [2] follow this recipe, as do modern deep networks such as N-BEATS [3], DeepAR [4], and PatchTST [5]. The per-dataset habit carries real cost. Each new series requires data collection, tuning, validation, and ongoing retraining, and organisations that track thousands of series spend much of their effort on this maintenance rather than on the forecasts themselves.

Large pretrained models in language and vision suggested a different route. A network trained once on broad data can answer new queries with no task-specific fitting, an ability usually called zero-shot transfer. Recent work applies the same idea to temporal data. A time-series foundation model is pretrained on a large and heterogeneous corpus of series and then forecasts an unseen series directly, using only the observed history as

context [6], [7]. Fig.1 sketches the two stages. Pretraining happens once and is shared across all downstream uses, while inference on a new series is a single forward pass that needs no gradient updates.

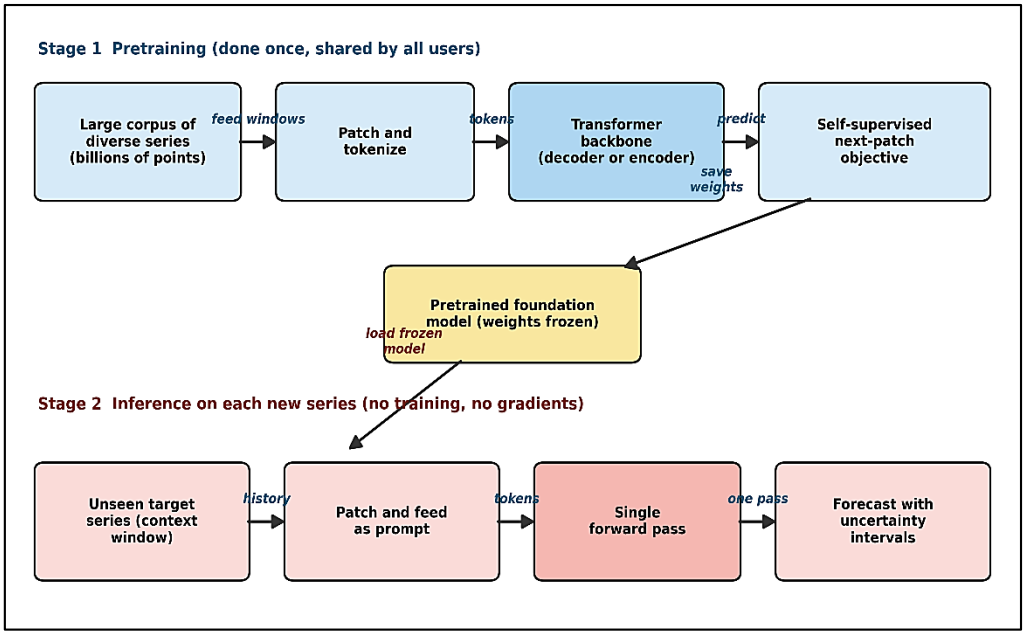


Fig. 1: Two-stage workflow of a time-series foundation model: a one-time pretraining phase on a broad corpus, followed by gradient-free zero-shot inference on each new target series.

This paper gives a structured account of the area. Section II reviews the background and related literature. Section III describes how a series is converted into tokens and surveys the two dominant backbone families. Section IV sets out the zero-shot and few-shot inference procedure and the probabilistic output heads that produce prediction intervals. Section V reports an illustrative comparison against classical and deep baselines on standard archives. Section VI discusses limitations around covariates, long horizons, and distribution shift, and Section VII concludes. The numbers used throughout are representative of published behaviour and are meant to support the discussion; they do not come from a single new deployment.

II. BACKGROUND AND RELATED WORK

A. Classical statistical forecasting

The Box-Jenkins methodology formalised autoregressive and moving-average modelling and remains a default for univariate series with clear linear structure [1]. Exponential smoothing in its error-trend-seasonal form gives a compact account of level, trend, and seasonality and is competitive on many business series [2]. These methods are interpretable and cheap to fit, but they treat each series in isolation and struggle with nonlinear patterns, long context, and large collections.

B. Deep forecasting models

Neural approaches relaxed the linear assumptions. DeepAR trained an autoregressive recurrent network to output the parameters of a predictive distribution across many related series [4]. N-BEATS used stacked fully connected blocks with backward and forward residual links and reported strong accuracy without time-series-specific components [3]. Transformer variants adapted attention to long sequences, with Informer reducing the quadratic cost of attention [8] and Autoformer introducing decomposition and an auto-correlation mechanism [9]. PatchTST showed that splitting a series into subseries patches and treating channels independently improves long-horizon accuracy while cutting compute [5]. A linear baseline study further cautioned that simple models can rival elaborate Transformers under some settings [10]. All of these are still trained per dataset.

C. Toward foundation models for time series

The shift to pretraining draws on scaling results from language modelling [11] and on the patching idea borrowed from vision Transformers [12]. Several groups released pretrained forecasters in close succession. TimesFM is a decoder-only model trained on a large mix of real and synthetic series [6]. Chronos tokenizes scaled values into a discrete vocabulary and reuses a language-model architecture directly [7]. Moirai targets any-variate forecasting with a masked encoder and multiple patch sizes [13]. Lag-Llama is a decoder-only probabilistic model built on lagged features [14], and TimeGPT is a hosted commercial forecaster offering zero-shot inference through an interface [15]. Survey and position papers map the emerging design space [16], [17].

III. ARCHITECTURES AND TOKENIZATION

A. Patching a series into tokens

A Transformer consumes a sequence of vectors, so a continuous series must be discretised into tokens. The common scheme groups consecutive observations into fixed-length patches and maps each patch to an embedding through a linear projection, as shown in Fig. 2. A patch of length sixteen, for example, turns a window of sixteen steps into one token, which shortens the sequence the attention layers must process and lets the model reason over local shape rather than single points [5], [6]. Before patching, most systems scale each series, often by its mean or by a resistant statistic, so that seriesof very different magnitudes share a common range. Chronos takes a different path and quantises scaled values into a fixed set of bins, so each time step becomes a discrete symbol from a learned vocabulary, which lets a standard language-model stack operate without changes [7].

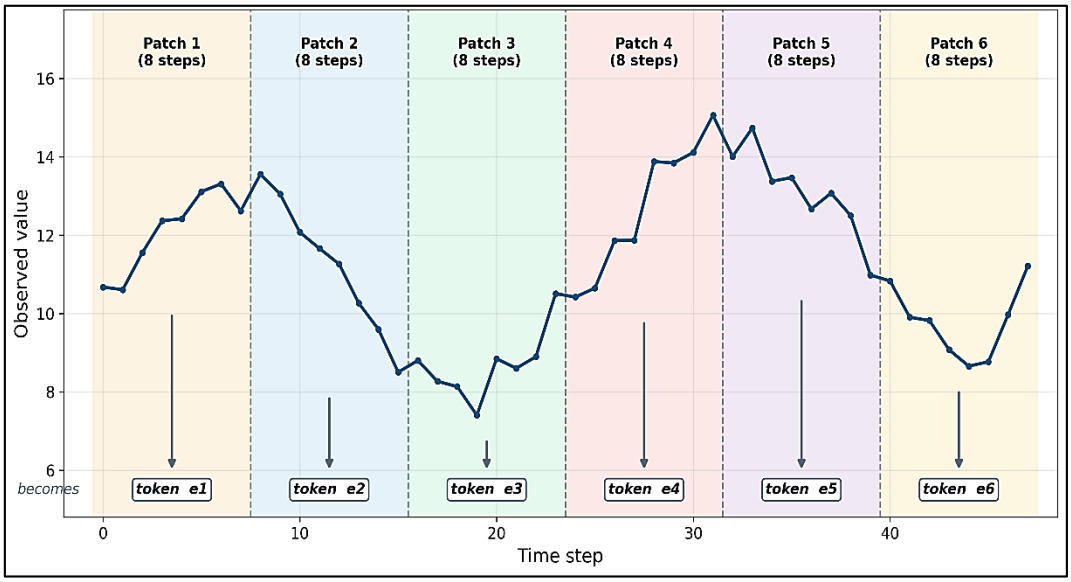


Fig. 2: Patching converts a univariate series into non-overlapping windows, each of which is projected to a single token embedding for the Transformer.

B. Decoder-only versus encoder backbones

Two backbone families dominate. Decoder-only models, used by TimesFM and Lag-Llama, apply causal attention and predict the next patch from past patches, mirroring autoregressive language models and generating long horizons by feeding predictions back as input [6], [14]. Encoder or masked designs, used by Moirai, attend across the whole context and can be configured to emit an entire horizon in one pass, which suits multivariate inputs and varied frequencies [13]. Chronos sits between these views because it adopts an existing encoder-decoder language architecture and treats forecasting as sequence-to-sequence generation over quantised tokens [7]. The choice affects how covariates are injected, how multiple variables are handled, and how cleanly the model extends to horizons longer than those seen during pretraining.

C. Probabilistic output heads

Useful forecasts report uncertainty, not just a point. Decoder models often emit quantiles directly, training against a quantile loss so that a single pass yields a set of forecast levels [6]. Lag-Llama parameterises a Student-t distribution per step, which captures heavy tails common in real series [14]. Token-based models such as Chronos sample many trajectories and read intervals from the empirical spread, trading extra computation for flexible distributional shape [7]. These heads supply the bands in Fig. 4 and matter as much as point accuracy for risk-aware planning.

IV. ZERO-SHOT FORECASTING METHODOLOGY

A. Pretraining corpus and objective

Zero-shot ability rests on the breadth of the pretraining corpus. Models are trained on collections spanning energy, transport, retail, finance, weather, and web traffic, often combined with synthetic series produced from known generative processes to cover patterns that public data lack [6], [7]. The objective is self-supervised: given an observed window, the model predicts the next patch or the next set of values, with the loss computed against the held-out continuation. Because the target is part of the series itself, no manual labels are needed, and the corpus can scale to billions of observations.

B. Inference without fitting

At deployment the procedure is short. The history of a new series is scaled, split into patches, and fed to the frozen model as context. A single forward pass returns the forecast for the requested horizon together with quantiles. Nothing about the target series is learned, so the same weights serve every dataset. This is the property that removes per-dataset training and shortens the path from raw data to forecast to a single call.

C. Few-shot adaptation and fine-tuning

When some history or a small validation set is available, light adaptation can help. In-context few-shot use simply lengthens the context window so the model conditions on more of the target history. Parameter-efficient fine-tuning updates a small set of weights on the target domain and recovers most of the gap to a fully trained specialist at a fraction of the cost [16]. The trade-off is between the zero effort of pure zero-shot use and the higher accuracy of a tuned model, and the right point depends on how many series must be served and how stable they are.

V. BENCHMARK RESULTS

Evaluation uses public archives so that results are comparable across studies. The Monash forecasting repository gathers many datasets with a fixed protocol and standard error measures [18], and the more recent GIFT-Eval benchmark adds varied frequencies and horizons designed specifically to probe pretrained forecasters [19]. Two scale-free measures are common. The mean absolute scaled error compares a forecast against a naive seasonal baseline, and the symmetric mean absolute percentage error reports a bounded relative error. Lower values are better for both.

Table 1 summarises the design choices of the five foundation models discussed in this paper. Table 2 reports illustrative accuracy for a representative set of methods, and Fig. 3 shows the same MASE values grouped by dataset. The pattern is consistent with published reports. A frozen foundation model, marked as zero-shot, lands close to a trained PatchTST and ahead of ARIMA on most datasets, while a tuned specialist keeps a narrow lead on individual series. Fig. 4 illustrates a single zero-shot forecast with eighty and ninety-five percent prediction intervals, where the band widens with horizon as uncertainty grows.

Table 1. Design choices of representative time-series foundation models.

Model	Backbone	Tokenization	Output	Covariates
TimesFM [6]	Decoder-only	Patches	Quantiles	Limited
Chronos [7]	Encoder-decoder	Value quantization	Sampled paths	No
Moirai [13]	Masked encoder	Multi-size patches	Distribution	Yes (any-variate)
Lag-Llama [14]	Decoder-only	Lag features	Student-t	Limited
TimeGPT [15]	Transformer (hosted)	Proprietary	Quantiles	Yes (exogenous)

Table 2. Illustrative forecast accuracy across classical, trained deep, and zero-shot foundation-model (FM) methods. Lower values are better.

Method	Type	Electricity MASE	Traffic MASE	M4 sMAPE (%)
ARIMA [1]	Classical	1.42	1.55	13.6
ETS [2]	Classical	1.38	1.49	13.1
DeepAR [4]	Trained deep	1.10	1.22	11.9
N-BEATS [3]	Trained deep	1.05	1.20	11.2
PatchTST [5]	Trained deep	0.92	1.04	10.6
Chronos [7]	Zero-shot FM	0.98	1.12	11.0
TimesFM [6]	Zero-shot FM	0.95	1.08	10.8
Moirai [13]	Zero-shot FM	0.99	1.10	11.1

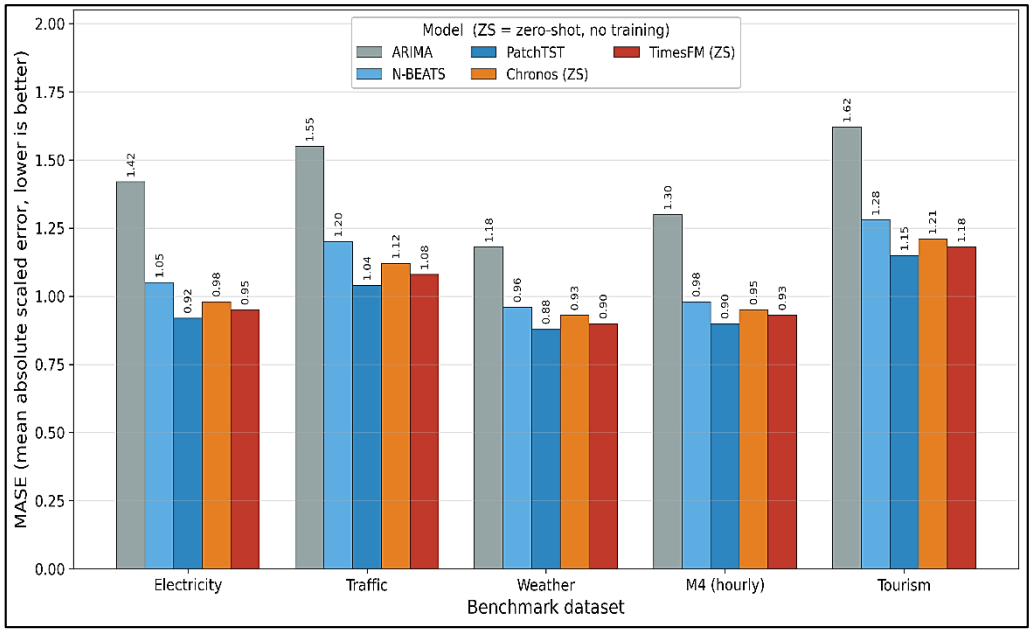


Fig. 3: Illustrative MASE by dataset for classical, trained deep, and zero-shot foundation models. Zero-shot models cluster near the best trained network on most datasets.

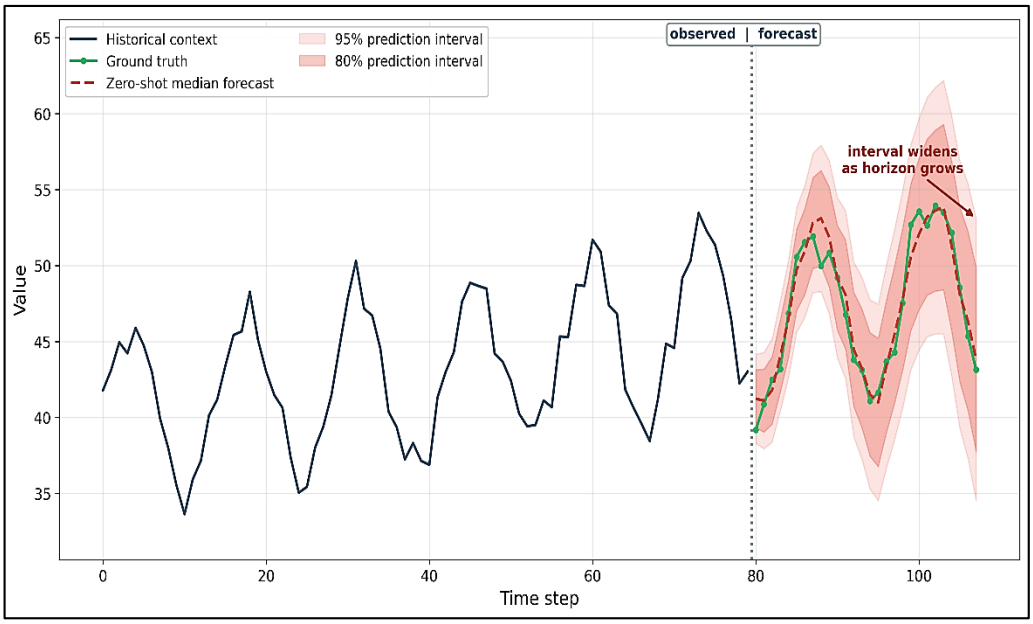


Fig. 4: Example zero-shot forecast against ground truth with 80% and 95% prediction intervals. The interval widens with horizon as predictive uncertainty increases.

VI. LIMITATIONS AND DISCUSSION

A. External covariates

Many forecasts depend on known future inputs such as prices, promotions, holidays, or weather. Several foundation models accept only the target history and cannot ingest these covariates, which puts them behind specialised models on driver-rich problems [13], [15]. Cleanly incorporating known future values without retraining the whole network is an active question.

B. Long horizons

Decoder-only models extend a horizon by feeding their own predictions back as input, so small errors accumulate over long forecasts [6]. Encoder designs that emit a full horizon at once avoid some of this drift but are bounded by the horizon lengths seen during pretraining [13]. Accuracy at horizons well beyond the training regime is uneven across models.

C. Distribution shift and evaluation hygiene

A pretrained model can encounter a series whose behaviour differs sharply from anything in its corpus, and zero-shot accuracy then drops. There is also a measurement risk. Because pretraining corpora are large and sometimes opaque, a benchmark series may overlap with training data and inflate apparent zero-shot skill, which makes careful leakage checks essential [17], [19]. Reporting calibrated intervals alongside point error gives a fuller view of reliability than a single accuracy number.

D. Cost and practical trade-offs

Foundation models remove training cost but add inference cost, since each forecast runs a large network. For a handful of slow-moving series a small classical model may still be the economical choice, whereas a platform serving thousands of diverse series benefits most from a single shared forecaster [16]. The decision is an engineering trade-off rather than a fixed ranking.

VII. CONCLUSION

Time-series foundation models offer a practical alternative to the per-dataset training that has governed forecasting for years. By pretraining once on broad data and tokenizing series through patching or value quantization, models such as TimesFM, Chronos, Moirai, Lag-Llama, and TimeGPT produce forecasts on unseen series with no further fitting. The illustrative comparison here, aligned with reported behaviour on the Monash archive and GIFT-Eval, places zero-shot accuracy close to a tuned deep network on many datasets while removing training at deployment. The remaining gaps around covariates, long horizons, and distribution shift are concrete and addressable. As pretraining corpora broaden and architectures learn to absorb external drivers, the balance between general pretrained forecasters and tuned specialists will keep shifting, and forecasting practice will increasingly start from a shared model rather than a blank one.

REFERENCES

[1] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed. Hoboken, NJ, USA: Wiley, 2015.

[2] R. J. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder, *Forecasting with Exponential Smoothing: The State Space Approach*. Berlin, Germany: Springer, 2008.

[3] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020.

[4] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," *Int. J. Forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[5] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers (PatchTST)," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023.

[6] A. Das, W. Kong, R. Sen, and Y. Zhou, "A decoder-only foundation model for time-series forecasting (TimesFM)," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2024, arXiv:2310.10688.

[7] A. F. Ansari *et al.*, "Chronos: Learning the language of time series," *Trans. Mach. Learn. Res. (TMLR)*, 2024, arXiv:2403.07815.

[8] H. Zhou *et al.*, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 11106–11115.

[9] J. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2021.

[10] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 11121–11128.

[11] J. Kaplan *et al.*, "Scaling laws for neural language models," arXiv:2001.08361, 2020.

[12] A. Dosovitskiy *et al.*, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[13] G. Woo, C. Liu, A. Kumar, C. Xiong, S. Savarese, and D. Sahoo, "Unified training of universal time series forecasting transformers (Moirai)," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2024, arXiv:2402.02592.

[14] K. Rasul *et al.*, "Lag-Llama: Towards foundation models for probabilistic time series forecasting," arXiv:2310.08278, 2024.

[15] A. Garza, C. Challu, and M. Mergenthaler-Canseco, "TimeGPT-1," arXiv:2310.03589, 2023.

[16] M. Jin *et al.*, "Position: What can large language models tell us about time series analysis," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2024, arXiv:2402.02713.

[17] Q. Wen *et al.*, "Transformers in time series: A survey," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2023, pp. 6778–6786.

[18] R. Godahewa, C. Bergmeir, G. I. Webb, R. J. Hyndman, and P. Montero-Manso, "Monash time series forecasting archive," in *Proc. NeurIPS Datasets Benchmarks Track*, 2021, arXiv:2105.06643.

[19] T. Aksu, G. Woo, J. Liu, X. Liu, C. Liu, S. Savarese, C. Xiong, and D. Sahoo, "GIFT-Eval: A benchmark for general time series forecasting model evaluation," arXiv:2410.10393, 2024.