

Deep Learning Methods for Time Series Forecasting: A Comprehensive Review

Durgesh Dere¹, Maheshwaree Talekar², Prof. Dhanashree Tamhane³

¹Department of Information Technology. Vidyalankar Institute of Technology Mumbai, India

²Department of Information Technology. Vidyalankar Institute of Technology Mumbai, India

³Department of Information Technology. Vidyalankar Institute of Technology Mumbai, India

Abstract - Forecasting using time series is the key to decision-making in areas such as energy systems, finance, transport, healthcare, climate change studies, and manufacturing automation. In recent years, the field of forecasting has been revolutionized by deep learning which has replaced traditional statistical methods by introducing end-to-end learnable architectures capable of capturing non-linear temporal dynamics. The current paper surveys the state-of-the-art in applying deep learning to forecasting by classifying existing techniques into five groups of models: (i) recurrent (RNN, LSTM, GRU, DA-RNN, TFT, P-sLSTM); (ii) convolutional (1-D CNN, TCN, MTCN, STCTN, STGCN, SCINet); (iii) Transformer-based (Informer, Autoformer, FEDformer, PatchTST, iTransformer, Crossformer, Fredformer, Pyraformer); (iv) specialized/hybrid (TimesNet, Times2D, MultiResFormer, N-BEATS, N-HiTS, TimeMixer, LSTM-mTrans-MLP, SpecAR-Net, ArrivalNet); and (v) foundation models (TimeGPT, Chronos, Moirai, TimesFM, Sundial, Timer-XL, UniTS, ChatTime, Lag-Llama, TimeLLM, GPT4TS, TEMPO). A unified problem formulation is provided together with a summary of evaluation metrics and benchmarks and a comparative analysis including the computational complexity, inductive biases, and empirical performance of the surveyed forecasting models. We particularly emphasize the evolution from task-specific supervised models to foundation models which promise zero-shot forecasting and easy deployment. Some key open challenges include data availability, domain shift, interpretability, calibration, multivariate heterogeneous data forecasting, and benchmarking issues.

Keywords: Time series forecasting, deep learning, recurrent neural networks, LSTM, Transformers, foundation models, zero-shot forecasting.

1. INTRODUCTION

A. Motivation and Importance

This paper reviews the state-of-the-art deep learning techniques for time series forecasting, which is a pervasive problem in data science with practical relevance to areas like electricity load [1], traffic [2], weather and climate [3], [4], finance [5], healthcare, supply chain management and industrial automation [6]. Time series consist of observations ordered according to time, and forecasting refers to predictions of future values based on past values within a window. Even small improvements in accuracy translate to great economic value like 1% improvement in load forecast accuracy is worth millions of dollars, while better epidemic outbreak forecasts could save

lives. Time series differ from cross-sectional data in that they have temporal dependencies, seasonality, trend and regime changes that violate the i.i.d. assumption of machine learning techniques. Real-life time series data can be multi-dimensional, noisy, unevenly spaced and have missing values, making forecasting uniquely challenging.

B. Transition from Classical to Deep Learning Techniques

Classical statistical techniques (ARIMA, exponential smoothing, Theta, state space models) have been prevalent in forecasting for decades. They are interpretable and work well for structured univariate time series. However, classical forecasting models require manual specification of differencing orders and seasonal lengths and do not scale to large sets of multivariate time series. In the 2010s, gradient boosting trees (XGBoost, LightGBM [7]) in combination with lag features became a dominant approach for forecasting platforms. Their performance in M4 competition [8] proved that no paradigm had absolute advantage. Deep learning introduced the third paradigm – neural networks that learn high-level temporal features in an end-to-end manner. Recurrent networks (LSTM [9], GRU [10]) showed great sequence modeling capabilities while convolutional networks [3], [11], [12] offered parallelizable alternatives. The Transformer [13] architecture spurred the rise of specialized forecasting architectures (Informer [14], Autoformer [15], FEDformer [16], PatchTST [17], iTransformer [2]) that achieved new SOTA on long-term forecasting benchmarks. Finally, foundation model era has begun: TimeGPT [18], Chronos [19], Moirai [20], TimesFM [21], Sundial [22], and Timer-XL [23] are capable of zero-shot forecasting across domains. Figure 1 illustrates the transition of paradigms in forecasting over the last decade.

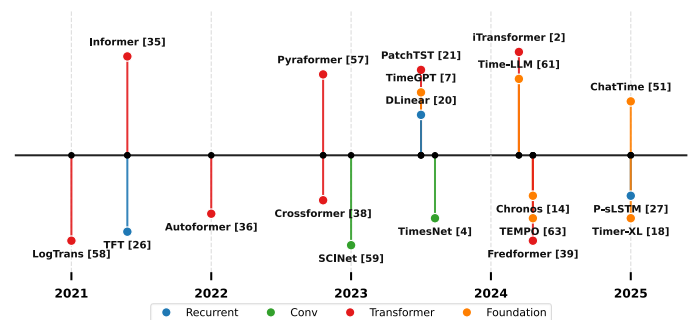


Figure 1. Chronological taxonomy of contemporary deep learning architectures for time series forecasting (2021–2025), categorized by structural paradigm.

C. Scope and Contributions

This paper delivers a methodologically structured survey, with the following contributions:

- A unified taxonomy of deep learning forecasters based on five dimensions: recurrent, convolutional, Transformer-based, specialized/hybrid, and foundation models, including newly introduced types (N-BEATS [24], N-HITS [25], TimeMixer [26], Lag-Llama [27], GPT4TS [28], Time-LLM [29], TEMPO [30]).
- Detailed technical description of selected models with their algorithms and empirical results derived directly from their original papers, focusing on real experimental setups.
- Comparison in terms of complexity, inductive biases, benchmark performance (ETT, ECL, Traffic, Weather), and methodological aspects (channel-wise vs. channel-mixing, RevIN [31], lookback sensitivity, seed variance).
- Foundation models: Tokenization, pre-training tasks, fine-tuning (LoRA [32]), zero-shot transfer, as well as LLM-reprogramming methods.
- Presentation of open problems related to benchmarks, distribution shift, interpretability, calibration, multivariate heterogeneity, contamination, edge computing. Section II contains the background, Section III – taxonomy, Sections IV–VIII – the five groups of forecasters, Section IX – comparison, Section X – open problems, and Section XI – conclusions.

2. BACKGROUND AND PRELIMINARIES

A. Problem Formulation

A univariate time series $x_{1:T} = (x_1, \dots, x_T)$, where x_t is an element of $\mathbb{R}$, is observed discretely. With a lookback window of length L , the problem lies in predicting the following H values of $x_{T+1:T+H}$ given $x_{T-L+1:T}$ and possibly covariates z_t . The modeler learns:

$$p_\theta(x_{T+1:T+H} | x_{T-L+1:T}, z_{T-L+1:T+H}) \quad (1)$$

$H = 1$ corresponds to one-step ahead prediction, while $H > 1$ is called multi-horizon prediction. LTSF usually assumes $H \geq 96$. In multivariate time series, when x_t is an element of $\mathbb{R}^N$, two possible strategies are channel independent (where each variable is modeled independently) and channel mixing (where the dependency between variables is explicitly modeled). Channel independence usually performs better than complicated multivariate strategies [17], leading to new strategies such as iTransformer [2], where cross-variable dependency is modeled through different embeddings. Figure 5 shows the difference between these strategies. Probabilistic forecasting considers not only point prediction but also the entire distribution of $p(x_{T+1:T+H})$, which is important for risk-sensitive decisions.

B. Evaluation Metrics

Mean Squared Error (MSE) and Mean Absolute Error (MAE) form the main point-wise measures:

$$MSE = (1/H) \sum_{t=1}^H (x_t - \hat{x}_t)^2,$$

$$MAE = (1/H) \sum_{t=1}^H |x_t - \hat{x}_t| \quad (2)$$

The error is measured as percentages using MAPE and sMAPE measures:

$$MAPE = \text{MAPE} = \frac{100\%}{H} \sum_{t=1}^H \frac{|x_t - \hat{x}_t|}{|x_t|},$$

$$sMAPE = (200\%/H) \sum_{t=1}^H |x_t - \hat{x}_t| / (|x_t| + |\hat{x}_t|) \quad (3)$$

The Continuous Ranked Probability Score (CRPS) forms the standard probabilistic measure:

$$CRPS(F, x) = \int_{-\infty}^{\infty} (F(y) - 1[y \geq x])^2 dy \quad (4)$$

where F refers to the cumulative distribution function (CDF). The CRPS is a generalization of the MAE and has been used by Chronos [19]. Weighted Quantile Loss (WQL) evaluates specific quantiles and is the principal metric of M5-style competitions.

C. Common Datasets and Benchmarks

There are several standardized benchmarks, most of which are provided through Time Series Library (TSLib) [23], see Table I.

Table I. Standard time series forecasting benchmarks and their characteristics

Benchmark	Description	Applications
ETT	ETTh1/h2, ETTm1/m2; hourly & 15-min	LTSF; oil temperature
ECL	321 clients, hourly, 2-year	Multivariate LTSF
Traffic	862 sensors, hourly, SF Bay Area	Spatio-temporal
Weather	21 indicators, 10-min	Multivariate LTSF
Exchange	Daily, 8 countries	Univariate, noisy
ILI	Weekly, CDC	Epidemiology LTSF
Solar-Energy	PV, 10-min	Multivariate LTSF
PEMS	PEMS03/04/07/08	Traffic, large N
M4 / M5	Heterogeneous domains	Foundation zero-shot [18], [19]
Monash	>30 datasets	Pre-training [19], [20]
GIFT-Eval	Held-out cross-domain	Zero-shot FM eval [22], [51]

Foundation models have motivated the employment of increasingly large training datasets, such as LOTSA for Moirai [20], TSMixup+KernelSynth for Chronos [19], TimeBench (about 1 trillion samples) for Sundial [22], and a private dataset

for TimeGPT [18]. Such datasets are an order of magnitude larger than existing public datasets, which justifies the scaling property witnessed in foundation forecasters (see Figure 6).

3. LITERATURE SURVEY

The forecasting algorithms are categorized into a hierarchy based on architectural family, training methodology and adaptation methodology. The top level of this hierarchy identifies five different families, as shown in Figure 2 and Table II.

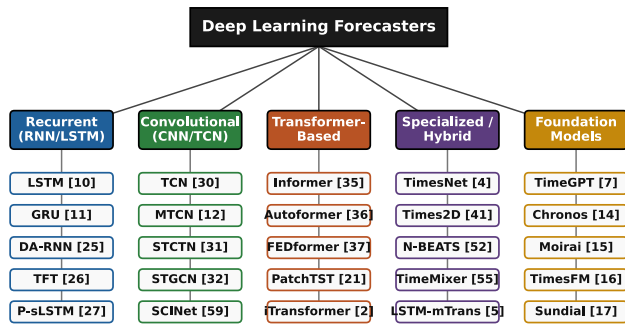


Figure 2. Hierarchical taxonomy of deep learning approaches for time series forecasting. Five top-level families cover the principal architectural paradigms; each is subdivided into representative methods surveyed.

This Literature Survey is navigational in nature as opposed to hierarchical, as there is an integration of concepts within different models. An example of this is the LSTM-mTrans-MLP hybrid model [5], which is a combination of recurrent neural network and attention architecture. TimesNet [4] uses inception blocks in two dimensions that are layered above a Transformer-inspired structure.

Table II. Five-family taxonomy of deep learning approaches for time series forecasting.

Family	Representative Models	Defining Char.	Refs
Recurrent	RNN, LSTM, GRU, DA-RNN, TFT, P-sLSTM	Sequential; gated memory	[9]-[10], [35]-[37]
Convolutional	1-D CNN, TCN, MTCN, STCTN, STGCN, SCINet	Sliding window; parallel	[3], [11], [12], [39]-[41]
Transformer	Informer, Autoformer, FEDformer, PatchTST, iTransformer,	Self-attention; long-range	[2], [13], [14]-[17],

	Crossformer, Fredformer		[42]-[43]
Specialized / Hybrid	TimesNet, Times2D, MultiResFormer, N-BEATS, N-HiTS, TimeMixer, LSTM-mTrans, SpecAR-Net, ArrivalNet	Domain-specific; multi-task	[4]-[5], [24]-[26], [46]-[49]
Foundation	TimeGPT, Chronos, Moirai, TimesFM, Sundial, Timer-XL, UniTS, ChatTime, Lag-Llama, Time-LLM, GPT4TS, TEMPO	Pre-trained; zero-shot	[18]-[23], [27]-[30], [52]-[53]

4. RECURRENT NEURAL NETWORKS AND VARIANTS

A. Vanilla RNN, LSTM, and GRU

Recurrent Neural Network (RNN) algorithms process time-series data sequentially, whereby the hidden state h_t holds all information of the previous inputs:

$$h_t = \sigma(W_h h_{t-1} + W_x x_t + b) \quad (5)$$

Vanilla RNNs suffer from vanishing and exploding gradients in case of long sequences which resulted in the development of gated models. In Long Short-Term Memory (LSTM) network [9], the memory cell c_t is controlled by input gate i_t , forget gate f_t , and output gate o_t as follows:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (6)$$

$$c_t = f_t * c_{t-1} + i_t * \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (7)$$

$$h_t = o_t * \tanh(c_t) \quad (8)$$

Forget gate helps in maintaining gradient flow along the long sequences whereas the cell state gives a stable memory path for information to flow. Therefore, LSTM became a prevalent choice for modeling long sequences for almost a decade. A compact version of LSTM was presented in the form of Gated Recurrent Unit (GRU) [10] by combining cell and hidden states along with the input and forget gates into one update gate z_t as:

$$z_t = \sigma(W_z[h_{t-1}, x_t]), r_t = \sigma(W_r[h_{t-1}, x_t]) \quad (9)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tanh(W_h[r_t * h_{t-1}, x_t]) \quad (10)$$

Both of the methods offer similar performance with reduced parameter in case of GRUs. Hence, they become a preferred

option in resource limited settings. Depending upon the application, either LSTM or GRU is selected and hybrid architectures outperform both of them in benchmarks.

B. Attention-Augmented and Recent LSTM Variants

Dual-Stage Attention-Based Recurrent Neural Network (DA-RNN) [35] employs two types of attention mechanisms which complement each other: input attention mechanism which focuses on the relevant driving series, and temporal attention mechanism which assigns weights to historical hidden states. DA-RNN increases the quality of prediction on the benchmarks that have multivariate time series with noise such as SML2010 and NASDAQ 100. The Temporal Fusion Transformer (TFT) [36] introduces multi-head attention, gated residual networks, and variable selection networks into recurrent neural network architectures and thus allows achieving feature-level interpretability of predictions for multi-horizon forecasting under static covariates, future known inputs, and observed past inputs. TFT has been used extensively for retail, energy, and financial applications where interpretability is crucial. Recent research shows that properly regularized LSTMs can still be competitive. Kong et al. [37] propose P-sLSTM model which extends sLSTM (exponential gating and memory mixing), applies patching as PatchTST does, and makes channels independent like iTransformer. P-sLSTM outperforms LSTM on five multivariate benchmarks – Weather, Electricity, Solar, ETTm1, and PEMS03 – up to horizon 720 by about 20% on average and reaches the same level of performance as PatchTST and iTransformer. This work supports the criticism of Zeng et al. [38] of "simple baselines" who show that advanced architectures do not necessarily mean better accuracy.

C. Limitations

Three major drawbacks are presented by recurrent architectures:

- (i) the sequential computation is in turn leading to the fact that the computation cannot be performed in parallel, which causes delay in training longer sequences;
- (ii) despite the gating, LSTMs and GRUs have difficulties with dependencies that are extended beyond a few hundred steps;
- (iii) the hidden state serves as a bottleneck for information as the hidden state compresses every past information into a vector of fixed dimension. Using this information, convolutional and attention-based alternatives were created.

5. CONVOLUTIONAL NEURAL NETWORKS AND TCNS

A. 1-D CNNs for Sequence Modeling

One-dimensional CNNs utilize a sliding filter w in R^K to process the input follows:

$$y_t = \sigma(\sum_{k=0}^{K-1} w_k \cdot x_{t+k} + b) \quad (11)$$

The usefulness of CNN in speeding up training is due to the fact that separate convolutions are processed at the same time. Additionally, with stacked causal dilated convolutions, different layers of CNN learn time information at different levels. According to Bai et al. [12], convolutional networks have advantages over RNN, namely in their speed of training and stability of gradients, and this leads to the formation of TCNs in its modern development.

An important empirical work conducted by Bai et al. [30] demonstrated that convolutions perform better than recurrent networks for a wide variety of sequence modeling tasks and possess advantages in parallelism, gradient stability and receptive fields flexibility. Those works served as the motivation for using Temporal Convolutional Networks as one of the most popular forecasting baselines.

$$y_t^{(l)} = \sum_{k=0}^{K-1} w_k^{(l)} \cdot x_t - d_l \cdot k^{(l-1)} \quad (12)$$

With the kernel size K and dilations $d_l = 2^l$ for $l = 0, 1, \dots, L-1$, the receptive field of L -layer TCN equals:

$$R = 1 + (K - 1) \cdot (2^L - 1) \quad (13)$$

For $K = 3$ and $L = 5$, R becomes 63 while for $K = 3$ and $L = 6$, R equals 127. Both of them can successfully identify the daily and weekly signals from the hourly data. In Padmasiri et al. [3], it was shown that the TCN-based algorithm significantly surpasses LSTM and traditional (SVR with linear or RBF kernels, linear regression) methods on multi-step forecast.

B. Multivariate and Spatio-Temporal Convolutional Variants

The M-TCN (Multivariate TCN) [11] utilizes one-dimensional convolutional layers in a multi-head structure to build independent sub-modules for each input feature, permitting the system to take into account both intra-variable temporal behavior and inter-variable relationships. M-TCN's performance was evaluated on the PM2.5 dataset from Beijing and the ISO-NE electric energy datasets, achieving results comparable to LSTM, ConvLSTM, TCN, and MALSTM-FCN methods. STCTN (Spatial-Temporal Convolutional Transformer Network) [39] employs a unique convolutional attention for both temporal and spatial domains, achieving impressive results on 6 multivariate benchmarks such as PEMS traffic datasets. The STGCN (Spatio-Temporal Graph Convolutional Network) [40] model leverages graph convolutions interleaved with temporally gated convolutions, becoming popular in traffic and ride-hailing demand forecasting. Lastly, SCINet [41] takes advantage of the recursive down-sample-convolve-interact architecture with sample convolution and interaction (SCI) blocks organized in a

binary tree that permits the aggregation of multi-level features with no explicit dilation.

C. Strengths and Limitations

The advantages of TCNs include:

- (i) concurrent computation;
- (ii) the ability to control receptive field through dilation;
- (iii) steady gradient flow. Drawbacks: inherent limitations of the receptive field (risk of not fully covering information or overparameterization);

the inability of causal convolutions to look ahead (correct for prediction but limit imputation/classification).

6. TRANSFORMER-BASED MODELS

A. From NLP to Time Series

The Transformer [13] substituted recurrent methods with self-attention, leading to the capability for parallel processing of long sequences and cutting-edge performance in natural language processing and computer vision. The main scaled dot-product attention is expressed in the following terms:

$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (14)$$

In this equation, Q , K , and V are the matrices of query, key, and value, which are created from the linear projections of the input. In order to adjust the Transformers for time series, one needs to solve the problems posed by:

- (i) absence of semantics in numeric series;
 - (ii) $O(L^2)$ complexity of attention in case of long sequences;
 - (iii) permutation invariance that contradicts temporal order.
- The complexity-of-computation trade-offs for different Transformer architectures and other competing methods are demonstrated in Figure 3.

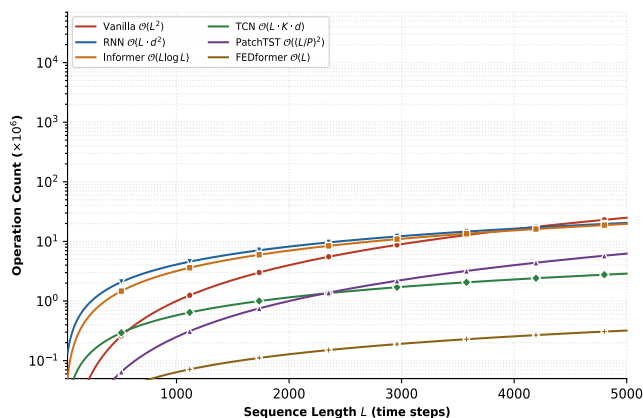


Figure 3. Computational complexity scaling of deep learning forecaster families with sequence length L (hidden dim $d=64$, kernel $K=3$, patch $P=16$). Vanilla Transformer attention scales quadratically; sparse-attention variants (Informer, LogTrans) and patching-based methods (PatchTST, iTransformer) achieve near-linear scaling

B. Informer, Autoformer, and FEDformer

The paper "Informer" [14] received the Best Paper Award at AAAI 2021 and introduced the idea of ProbSparse self-attention, in which attention is restricted to queries with the highest levels of sparsity (i.e., KL-divergence measure). This innovation leads to a decrease in complexity from $O(L^2)$ to $O(L \log L)$. Moreover, with the use of a generative decoder that produces all future values in a single forward pass, Informer sets up the LTSF benchmark protocol. The paper "Autoformer" [15] presented at NeurIPS 2021 comes up with the novel solution of replacing standard attention with auto-correlation in order to discover period-based dependencies by means of FFT (fast Fourier transform) and aggregating similar periods by means of time-delay operations. In addition, the network is built using the trend/seasonal decomposition principle. The research paper "FEDformer" [16] created at ICML 2022 makes use of frequency-enhanced decomposition, specifically, in the form of Fourier and wavelet transformations to achieve a final complexity of $O(L)$ and reports the reduction of multivariate prediction error by 14.8% and the decrease of univariate prediction error by 22.6% as compared to the preceding models of transformers.

C. PatchTST and the Patching Paradigm

The PatchTST algorithm utilizes the patch method from Vision Transformers in time series: the time series is split into patches with dimension P , and every individual patch is mapped to an embedding token. The usage of patches reduces L to L/P in magnitude while increasing efficiency by attention costs. The channel-independent structure of PatchTST shows that the patch method is sufficient to achieve breakthroughs in LTSF. The initial scheme of this application is demonstrated in Figure 4.

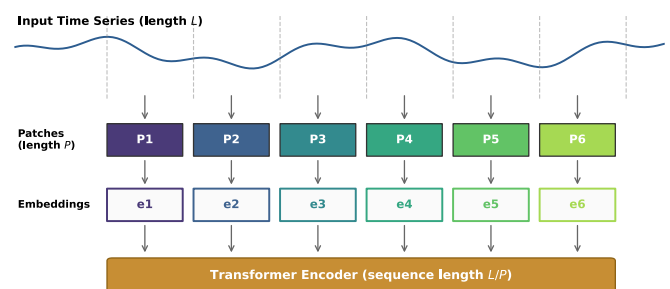


Figure 4. Patching approach. The time series of length L is split into L/P patches with dimension P and then transformed into the embedding tokens. Next, the Transformer architecture

processes the time series with L/P patches, which reduces the attention costs by P^2 times.

D. iTransformer: Inverted Transformers

iTransformer [2], ICLR 2024, revolutionizes the handling of time and variates in the Transformer model. Unlike the original Transformer, which uses time steps as input tokens (and mixes differences inside each token), iTransformer has a different approach where an entire set of the variate in question (the complete historical series of values for a single variate) is taken as an input token, allowing the model to calculate attention scores across the variates and pass inputs through the feedforward network across various time steps. This helps to maintain independence between variates within the tokens while cross-variates dependence is calculated via attention, and temporal embedding is learned in the FFN layer. The benchmark results show that iTransformer exceeds other models in terms of predictive power on the following datasets: ETT, ECL, Traffic, Weather, Solar, PEMS, and others (See fig.5 below for comparison).

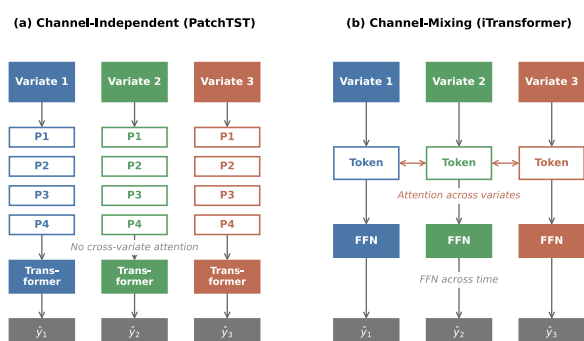


Figure 5. Channel-independent vs. channel-mixing embedding strategies. (a) PatchTST processes each variate independently with separate Transformer blocks. (b) iTransformer embeds each variate's full lookback as a single token, applies attention across variates and FFN across time, restoring cross-variate modeling without architectural modification.

E. Crossformer, Fredformer, and Other Variants

Crossformer [42], featured at ICLR 2023, employs a two-axis attention model that takes into consideration temporal dependencies as well as cross-variate dependencies, therefore using a Dual Self-Attention component. Although promising in theory, evidence in later studies indicates cross-variate attention is detrimental whenever the variates are independent or have noise [17]. Meanwhile, Fredformer [43], presented at KDD 2024, solves frequency bias issue in the context of standard Transformers by introducing a frequency-debiased attention approach that equally contributes to all frequency bands. The validity of this approach was confirmed through validation across eight different datasets (Weather, ETTh1/h2, ETTm1/m2, ECL, Traffic, Solar-Energy).

Alternative solutions include LogTrans ([44] providing a LogSparse approach and $O(L(\log L)^2)$ complexity) and Pyraformer [45], an ICLR 2022 Oral solution that presents a pyramidal hierarchical attention suitable for $O(L)$ complexity.

F. The Linear Baseline Critique

A significant event was demonstrated in Zeng et al. [38], AAAI 2023, where it was established that a single layer linear model (LTSF-Linear) outperforms complex Transformers in long-term forecasting by a margin of 20-50% on nine datasets. The argument is built around permutation invariance: despite the introduction of positional encoding, the temporal order is already diluted and the model ends up functioning as a smoothing operator failing to capture abrupt changes. This led to a number of architectural innovations-PatchTST [17] and iTransformer [2]-and discussions regarding inductive bias. The current view [2], [37] is that Transformers perform well if the tokenization preserves local temporal patterns and the embedding is consistent with variate independence principles.

7. SPECIALIZED AND HYBRID ARCHITECTURES

A. TimesNet: Temporal 2-D Variation Modeling

TimesNet [4], ICLR 2023, finds that real-world time series demonstrate attribute of multi-periodicity (exhibit daily, weekly, yearly cycles), and takes advantage of it by changing the representation of 1-D series into a 2-D tensor where rows correspond to periods and columns relate to phases intra-periodically. Variations within the periods can be interpreted as channel-wise variations, whereas variations within the periods are represented as row-wise analogies, both being addressed through 2-D kernel convolutions. TimesNet correlates FFT to find the k most significant periods, converting the input into a number of 2-D tensors and further applying a few but effective Inception blocks. It is worth noting that the architecture is universal for many applications, with TimesNet achieving the state of the art in different forecasting, imputation, classification, and anomaly detection tasks being the default model for further analysis.

B. Times2D, MultiResFormer, and Pure-MLP Architectures

Times2D [46] enhances the 2-D decomposition by introducing the First/Second derivative heat map (FSDH) module that is engaged with capturing serious changes and turning points. The structure consists of three components: a Periodic Decomposition Block (PDB) built on FFT; FSDH; and Aggregation Forecasting Block (AFB). MultiResFormer [47] addresses shortcomings of traditional Transformers in a single aspect resolution by implementing multilayer blocks, where different resolutions of branches correspond to the salient periodicity detected through FFT and the applied multifaceted architecture tackling inter-period variations by employing multifaceted architecture such as MHA and FFNN that

occupies the same purpose of tackling intra-period variations. N-BEATS [24] in ICLR 2020 introduces a deeply built neural network with backward/forward connectivity showing SOTA results in M4, TOURISM, and M3 competitions. N-HITS [25] contributes to N-BEATS by creating hierarchical interpolation and multi-rate sampling that reduces the model footprint by approximately 50 percent while providing better results in long-horizon prediction - about 25 percent of improvement.

TimeMixer [26] in ICLR 2024 Spotlight presents a fully MLP model that provides past decomposable mixing and future multiprediction mixing procedures that allow disentangling the variations. The final results match the claim about the effectiveness of well-designed MLPs which show that it is capable of working on the same level as transformer-based forecasters in major competitions.

C. LSTM-Transformer Hybrids and Domain-Specific Models

An LSTM-mTrans-MLP combination is suggested by Kabir et al. [5] for forecasting applications, comprising three building blocks: an LSTM to learn long-duration dependencies in noisy environments, a modified Transformer (mTrans) with self-attention with normalization before attention mechanism, deletion of input embeddings, and MLP-based decoder, and finally an MLP that learns dependencies between attributes. The model was tested on seven financial data sets-Bitcoin, SSEC, China Unicom, CSI300, CSI100, Amazon, Google-including ARMA, MLP, LSTM, CNN, BiLSTM, CNN-LSTM, CNN-BiLSTM, BO-LightGBM, and ensembles as baselines.

SpecAR-Net [48] (IJCAI 2024) uses STFT transforms to represent series as spectrograms and employs multi-scale 2-D complex convolutions to extract time-frequency relationships; it is general-purpose, highly capable for classification, anomaly detection, imputation, and forecasting. ArrivalNet [49] forecasts city-wide bus/tram arrival times using 2-D modeling of temporal variation inspired by period detection in TimesNet through FFTs, reshaping of 1-D data into 2-D form, and Inception-inspired 2-D blocks-lowering RMSE, MAE, MAPE by $\geq 6.1\%$, $\geq 14.7\%$, $\geq 34.2\%$ compared to SOTA on Dresden transportation data. Voice2Series [50] (ICML 2021 Spotlight) repurposes pre-trained acoustic networks for time-series classification through input and output mapping.

Such architecture examples prove that, while general-purpose models take the limelight in headline benchmarks, domain-specific models are still effective where the domain exhibits structure (spectrograms, transport periodicity, acoustic patterns).

8. FOUNDATION MODELS AND LARGE PRE-TRAINED MODELS

TSF has been revolutionized by the paradigm of foundation models, which has been adopted in NLP by BERT and GPT and in vision by CLIP and ViT. TSFM is a pre-trained model

from a huge heterogeneous pre-training corpus that can be used for forecasting zero-shot, few-shot, or through full fine-tuning [6]. The advantages of TSFM lie in one pre-trained model being sufficient for different tasks. The development of pre-training corpus size of recent TSFMs is shown in Fig. 6.

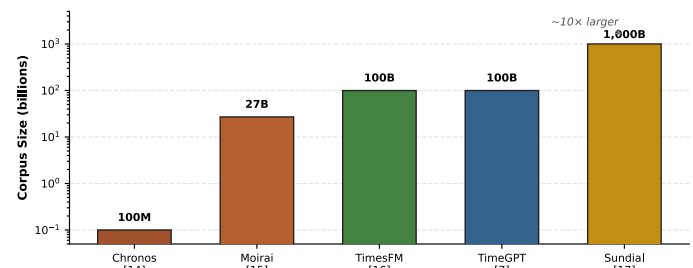


Figure 6. Pre-training corpus sizes of major time series foundation models, in billions of time points. Sundial's TimeBench is ~1 trillion points, roughly 10x larger than the next-largest public corpus

A. TimeGPT and Chronos

TimeGPT-1 [18], invented by Nixtla in 2023, is known to be the first foundation model for time series forecasting, capable of performing zero-shot tasks in diverse application domains. Trained on a proprietary dataset of more than 100 billion of observations of time series in finance, economics, demography, healthcare, weather, IoT, energy, website visits, sales, transportation, and banking, TimeGPT uses the encoder-decoder Transformer architecture with input embeddings with local positional encoding. Zero-shot probabilistic forecasts are produced with conformal prediction instead of quantile regression, providing calibrated prediction intervals without re-training. In practice, TimeGPT shows better results in comparison with classical statistical approaches, gradient-boosting baselines, and task-specific deep learning models on M4, M5, traffic, electricity, and weather benchmarks-zero-shot. Liao et al. [1] apply TimeGPT to campus building load forecasting, reporting 35-76% RMSE decrease in comparison with eight benchmarks in case of low data availability, while in high data availability case, TimeGPT underperforms, which shows that foundation model paradigm is especially useful in case of limited data from target domain. Training on target data is crucial for achieving best results.

Chronos [19], invented by AWS AI Labs (TMLR 2024) follows the principle of minimalism and tokenizes series values by mean scaling and quantization to integer bins from a fixed vocabulary, then training a standard T5-family language model on these tokens using cross-entropy loss. No time series-specific architecture changes are performed. The key idea is that once time series are tokenized, the language modeling objective is able to capture sequence information automatically. Chronos is pre-trained using TSMixup augmented with synthetic data generation using Gaussian processes with random kernels (KernelSynth). Models vary in

size from Mini (20M) to Large (710M). Chronos shows superior results compared to local methods

and task-specific deep learning models on in-domain datasets and shows competitive zero-shot performance on out-of-domain datasets. The combination of simplicity and repurposing of language model infrastructure makes Chronos a popular baseline.

B. Moirai, TimesFM, Sundial, and Timer-XL

Moirai [20], Salesforce Research (ICML 2024) is a universal forecasting Transformer pre-trained on the LOTSA corpus consisting of more than 27 billion observations from nine domains. Moirai handles heterogeneity in frequencies, number of variates, and horizon by means of multi-resolution patches and any-variate attention that allows to support multivariate series without fixed input channels. The output head generates mixture distribution forecasts (Student-t components) for calibrated uncertainty. Moirai matches or surpasses in-domain and zero-shot baselines on LTSF and Monash. Released in sizes Small (14M), Base (91M), Large (311M).

TimesFM [21], Google Research (ICML 2024) is a decoder-only Transformer mirroring the architecture of autoregressive language models such as GPT. It uses patch-based tokenization with large patch lengths (32 input, 128 output) and is pre-trained on a proprietary corpus of around 100 billion time points. The decoder-only architecture generates forecasts in an autoregressive way, thus making it possible to generate forecasts for variable horizons. TimesFM showed excellent zero-shot performance on Monash benchmarks and proved the viability of decoder-only paradigm as an alternative to encoder-decoder. The release along with Moirai helped to understand that both paradigms can be used as effective TSFM backbones.

Sundial [22], ICML 2025 is a family of native, flexible and scalable TSFMs created for trillion-scale pre-training. Sundial introduces TimeFlow Loss, based on flow matching, allowing native pre-training on continuous-valued series without tokenization. It is pre-trained on TimeBench (~1 trillion time points; ~0.05% synthetic supplement), released in sizes Small (32M), Base (128M), Large (444M). Sundial reports SOTA zero-shot results on TSLib, GIFT-Eval [51] and FEV. Timer-XL [23], ICLR 2025 is a long-context Transformer that extends the concept of next-token prediction to the multivariate setting, thus making it possible to work with thousands of patch tokens (tens of thousands of time points), allowing yearly contexts (ERA5-S, 40 years, 117k points per station).

C. UniTS, ChatTime, Lag-Llama, and LLM Reprogramming

UniTS [52], NeurIPS 2024, is a unified multitask model covering forecasting, classification, imputation, and anomaly detection. It relies on universal task tokenization that allows to work with variable-size input/output sequences without modifications of the model network and employs task-specific

prompts. ChatTime [53], AAAI 2025, is a multimodal TSFM connecting numerical and NLP modalities through expanding the vocabulary of pre-trained LLMs and allows zero-shot forecasting, context-driven forecasting, and time-series question answering. Lag-Llama [27] is a general univariate probabilistic forecasting foundation model relying on a decoder-only transformer with lags as covariates and competitive zero-shot performance.

Another approach is reprogramming pre-trained LLMs. Time-LLM [29], ICLR 2024, repurposes frozen LLMs for forecasting by using text prototype reprogramming and prompt-as-prefix fusion and achieves state-of-the-art results with minimal parameter updates. GPT4TS [28] ('One Fits All'), NeurIPS 2023 Spotlight, makes use of GPT-2 backbones with minimal adaptations, reaching state-of-the-art results in forecasting, classification, anomaly detection, and imputation tasks. TEMPO [30], ICLR 2024, is an extension of GPT-style frozen backbones with learnable prompts that help to handle dynamic temporal patterns. The reprogramming approach offers parameter efficiency, yet leaves the question open whether linguistic knowledge transfers to numerical sequences.

D. Adaptation Approaches and Trends

When it comes to domain-specific applications, fine-tuning is computationally expensive and entails catastrophic forgetting. Low-Rank Adaptation (LoRA) [32], ICLR 2022, freezes the pre-trained weights and inserts trainable low-rank decompositions into the attention weight matrices (W_q, W_v), which reduces the number of trainable parameters up to four orders of magnitude. There are also prompt tuning, prefix tuning, and adapter modules. There is empirical evidence [19, 25, 48] that even zero-shot usage achieves parity with the supervised baseline on many benchmarks. The distribution of parameter counts in popular TSFM families is shown in Figure 7.

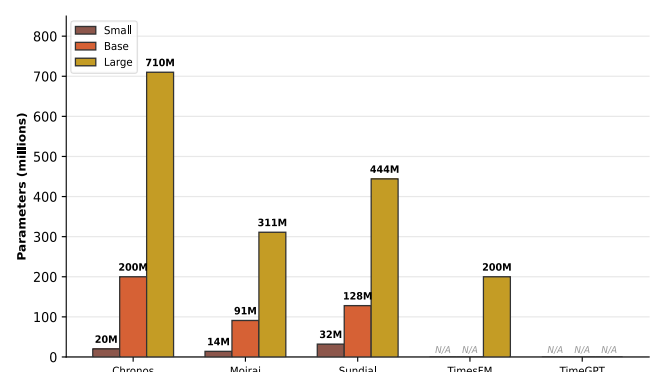


Figure 7. Parameter counts of time series foundation model families by size variant. Sundial-Large is 444M; Chronos-Large is 710M; Moirai-Large is 311M. TimeGPT is undisclosed (proprietary); TimesFM reports only the Large variant (200M).

The TSFM panorama reveals three converging trends:

(i) architectural convergence – most TSFMs use Transformer backbone and patch-based tokenization;

(ii) data-centric convergence – pre-training datasets are rapidly growing from 27B in LOTSA to ~1T in TimeBench and include synthetic data generation approaches like TSMixup and KernelSynth;

(iii) deployment convergence – model families range from <100M to >1B parameters in terms of parameter count.

9. COMPARATIVE ANALYSIS

A. Strengths and Weaknesses of Each Family

Table III highlights the pros and cons, level of difficulty, and the applications of each group of models. The LTSF benchmarks show that there has been consistent progress in this area, as the Informer model (2021) proved to be better than LSTMs; the Autoformer and FEDformer (2021-2022) used the ideas of decomposition and frequency-domain attention; the critique for DLinear written by Zeng et al. (2022) [38] triggered plenty of architectural reactions; PatchTST (2023) was the first Transformer system to surpass the linear benchmark on a steady basis; TimesNet (2023) achieved SOTA in all tasks, and the iTransformer (2024) improved by inverting the embedding dimension. Among the models based on the idea of foundation models, Chronos-Large, Moirai-Large, and TimesFM show decent zero-shot scores when compared to PatchTST and iTransformer, while Sundial also reports similar improvement. Figure 8 presents the results of MSE comparison for ETTh1.

Table III. Comparative analysis of deep learning model families for time series forecasting.

Family	Strengths	Weaknesses	Complexity
Recurrent	Short/medium horizon; interpretable; handles covariates (TFT)	Sequential; long-range limit; info bottleneck	$O(L * d^2)$
Convolutional	Parallel; controllable receptive field; stable gradients	Fixed receptive field; no native cross-variate	$O(L * K * d)$
Transformer	Long-range; multi-variate; flexible; strong LTSF	Quadratic cost; permutation-invariance; tokenization-sensitive	$O(L^2)$ or $O(L \log L)$

Specialized / Hybrid	Domain-specific; multi-task (TimesNet); multi-scale	Implementation complexity; harder to scale	Varies; $O(L * d^2)$ typical
Foundation	Zero-shot; simplified deployment; probabilistic;	Large pre-train cost; opaque pre-data; contamination risk	Inference $O(L^2)$; pre-train very high

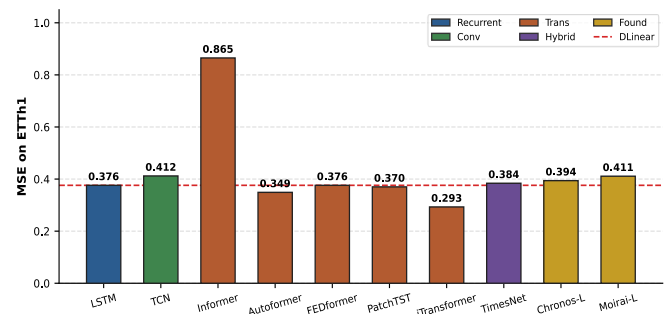


Figure 8. MSE on the ETTh1 benchmark (horizon = 96) across representative models from each family, color-coded by architectural family. The dashed red line marks the DLinear baseline [38]. iTransformer achieves the lowest MSE among supervised models, while foundation models (Chronos-Large, Moirai-Large) approach supervised performance in zero-shot settings

B. Reproducibility and Benchmark Contamination

Benchmark contamination is becoming concerning, since the pre-training data might unintentionally contain pieces of public benchmarks thereby creating a false image of the zero-shot results. Chronos [19] made sure to exclude this kind of data in their analysis, and in turn showed results both at the level of the domain and zero-shot level qualifications. Moirai [20] employed reasonable practices similar to Chronos, and TimesF [21] printed the results from the held-out Monash evaluation. The community now uses held-out Cross-Domain benchmarks in their analysis like GIFT-Eval [51] or reproducibility checklists. In addition to contamination, we should pay attention to three methodological issues, such as: (i) non-comparability of channel-independent and channel-mixing results; (ii) length of a lookback window L ; (iii) the presence or absence of Reversible Instance Normalization (RevIN) [31]. The criteria of distinguishing between the in-domain, zero-shot and few-shot qualifications, disclosure of corpus composition and description of the computational costs of evaluation must be established in future evaluations.

10. CONCLUSION

This Survey has consolidated deep learning methods for time series forecasting into five architectural families: recurrent, convolutional, Transformer-based, specialized/hybrid and foundation models. From vanilla RNNs to gated recurrent architectures (LSTM, GRU), attention-augmented (DA-RNN, TFT), convolutional alternatives (TCN), to the Transformer revolution triggered by Informer, Autoformer and FEDformer. PatchTST and iTransformer. In addition, specialized architectures such as TimesNet introduced 2-D tensors into the design space and pure-MLP architectures (N-BEATS, N-HiTS, TimeMixer) demonstrated that simple baselines continue to be very competitive.

A key recent development that has had the biggest impact is the emergence of time series foundation models. TimeGPT, Chronos, Moirai, TimesFM, Sundial, Timer-XL, UniTS, ChatTime and Lag-Llama are instances of a paradigm shift from task-specific supervised models to general-purpose pre-trained models that can make zero-shot forecasts. Chronos's finding-that a vanilla T5 trained on tokenized series can match or exceed supervised suggests a consensus on the language model paradigm, where innovations specific to time-series are mostly related to tokenization, pre-training data, adaptation, and not the underlying architecture. Linguistic knowledge is transferred to numerical series with few adaptations, as shown by LLM-reprogramming approaches (Time-LLM, GPT4TS, TEMPO) and by Sundial's flow-matching TimeFlow Loss towards native continuous-valued pre-training.

Incorporating open challenges, such as data scarcity, distribution shift, interpretability, calibration, multivariate heterogeneity, benchmarking inconsistencies, and edge deployment, necessitates concerted efforts in data curation, model architecture, model evaluation, and deployment infrastructure. Time series forecasting is now a commodity service available in nearly all scientific and industrial fields, as the foundation models evolve, just as text generation and image recognition did before it.

11. CHALLENGES AND FUTURE SCOPE

Although there has been great progress, there are still open challenges:

- (i) Data scarcity: public repositories have $\sim 10^{10}$ observations, while LLMs have $\sim 10^{12}$ tokens, but synthetic data (KernelSynth, TSMixup) partially resolves this, with a question of realism. Sundial's TimeBench at Open community-curated corpora at this scale (~ 1 trillion points) are needed.
- (ii) Distribution shift: real-world series experience regime shifts, sensor drift and policy changes. TSFMs exhibit a trade-off between stability and plasticity, online adaptation holds promise but has not been explored at foundation model scale, and continual learning and RevIN are also promising.

(iii) Interpretability: Interpretability is required in forecasting in healthcare, finance, and energy. TFT offers feature-level interpretability by the selection of variables, while most Transformer and foundation models are opaque. Mechanistic interpretability methods that have been applied to LLMs (e.g., probing, sparse autoencoders, circuit analysis) have not been systematically used on TSFMs.

(iv) Probabilistic calibration: Quantile regression or token sampling (Chronos) based TSFMs are usually miscalibrated on OOD data. Post-hoc calibration (conformal prediction, isotonic regression) is helpful, but needs to be used with validation data. Consideration should be given to hybrid parametric + sample based approaches.

(v) Multivariate heterogeneity: standard benchmarks are based on fixed variate counts, but in reality, when it is put to use in the field, it will be a multivariate heterogeneous variate that changes over time. This is partially solved by any-variate attention (Moirai) and by using prompt-based variate specification (ChatTime), but streaming multivariate heterogeneity is still an open question.

(vi) Benchmarking inconsistencies: varying lookback periods, normalization methods and train/test splits in the different papers. We call for a common evaluation framework, a separation between in-domain/zero-shot/few-shot settings, disclosure of the corpora, a uniform mention of computational costs, and held-out cross-domain benchmarks such as GIFT-Eval [51].

(vii) Edge Deployment: Often TSFMs are large, and deployment on IoT/mobile devices demands aggressive compression. To achieve edge efficient TSFMs, distillation, quantization, sparse MoE and Fredformer's Nystrom approximation are steps taken. In particular, opportunities for multimodal forecasting (combining TSFMs with LLM-based reasoning agents and applications across different domains, like climate science, as in FourCastNet for data-driven weather prediction) are extremely promising.

ACKNOWLEDGEMENT

It is our pleasure to extend our thanks to Prof. Dhanashree Tamhane, Department of Information Technology, for their valuable guidance, inspiration, and consistent support offered during the preparation of this survey paper, titled "Deep Learning Methods for Time Series Forecasting: A Comprehensive Review" Their constructive criticisms have helped a lot in enhancing the quality of this work. We would also like to thank the Department of Information Technology, Vidyalankar Institute of Technology for providing us with an enabling academic atmosphere and all the required facilities for the conduct of this research work. We appreciate the contributions of researchers and authors who had worked on deep learning for time series and their publications paved way for the preparation of this comprehensive review. Lastly, we

would like to thank our peers, colleagues, and other people who helped us indirectly or directly in completing this survey paper.

REFERENCES

- W. Liao, S. Wang, D. Yang, Z. Yang, J. Fang, C. Rehtanz, and F. Porte-Agel, "TimeGPT in Load Forecasting: A Large Time Series Model Perspective," arXiv preprint arXiv:2404.04885, 2024.
- Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "iTransformer: Inverted Transformers Are Effective for Time Series Forecasting," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2024.
- P. Hewage, A. Behera, M. Trovati, E. Pereira, M. Ghahremani, F. Palmieri, and Y. Liu, "Temporal Convolutional Neural (TCN) Network for an Effective Weather Forecasting Using Time-Series Data From the Local Weather Station," *Soft Comput.*, vol. 24, pp. 16453–16482, 2020.
- H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2023.
- M. R. Kabir, D. Bhadra, M. Ridoy, and M. Milanova, "LSTM–Transformer-Based Robust Hybrid Deep Learning Model for Financial Time Series Forecasting," *Sci*, vol. 7, no. 1, Art. no. 7, 2025, doi: 10.3390/sci7010007.
- Y. Liang, H. Wen, Y. Nie, Y. Jiang, M. Jin, D. Song, S. Pan, and Q. Wen, "Foundation Models for Time Series Analysis: A Tutorial and Survey," in *Proc. 30th ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD)*, pp. 6555–6565, 2024.
- G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems*, vol. 30, pp. 3146–3154, 2017.
- S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M4 Competition: Results, Findings, Conclusion and Way Forward," *Int. J. Forecasting*, vol. 34, no. 4, pp. 802–808, 2018.
- S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- K. Cho, B. van Merriënboer, C. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation," in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014.
- R. Wan, S. Mei, J. Wang, M. Liu, and F. Yang, "Multivariate Temporal Convolutional Network: A Deep Neural Networks Approach for Multivariate Time Series Forecasting," *Electronics*, vol. 8, no. 8, Art. 876, 2019.
- S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," arXiv preprint arXiv:1803.01271, 2018.
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.
- H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 12, pp. 11106–11115, 2021.
- H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 22419–22430, 2021.
- T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting," in *Proc. 39th Int. Conf. Mach. Learn. (ICML)*, PMLR, vol. 162, pp. 27268–27286, 2022.
- Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A Time Series Is Worth 64 Words: Long-Term Forecasting With Transformers," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2023.
- A. Garza, C. Challu, and M. Mergenthaler-Canseco, "TimeGPT-1," arXiv preprint arXiv:2310.03589, 2023.
- A. F. Ansari, L. Stella, C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. P. Arango, S. Kapoor, J. Zschiegner, D. C. Maddix, H. Wang, M. W. Mahoney, K. Torkkola, A. G. Wilson, M. Bohlke-Schneider, and Y. Wang, "Chronos: Learning the Language of Time Series," *Trans. Mach. Learn. Res.*, 2024.
- G. Woo, C. Liu, A. Kumar, C. Xiong, S. Savarese, and D. Sahoo, "Unified Training of Universal Time Series Forecasting Transformers," in *Proc. 41st Int. Conf. Mach. Learn. (ICML)*, PMLR, vol. 235, pp. 53140–53164, 2024.
- A. Das, W. Kong, R. Sen, and Y. Zhou, "A Decoder-Only Foundation Model for Time-Series Forecasting," in *Proc. 41st Int. Conf. Mach. Learn. (ICML)*, PMLR, vol. 235, pp. 10148–10167, 2024.
- Y. Liu, G. Qin, Z. Shi, Z. Chen, C. Yang, X. Huang, J. Wang, and M. Long, "Sundial: A Family of Highly Capable Time Series Foundation Models," in *Proc. 42nd Int. Conf. Mach. Learn. (ICML)*, PMLR, vol. 267, pp. 39295–39317, 2025.
- Y. Liu, G. Qin, X. Huang, J. Wang, and M. Long, "Timer-XL: Long-Context Transformers for Unified Time Series Forecasting," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2025.
- B. N. Oreshkin, D. Carpo, N. Chapados, and Y. Bengio, "N-BEATS: Neural Basis Expansion Analysis for Interpretable Time Series Forecasting," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2020.
- C. Challu, K. G. Olivares, B. N. Oreshkin, F. G. Ramirez, M. M. Canseco, and A. Dubrawski, "N-HiTS: Neural Hierarchical Interpolation for Time Series Forecasting," *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 6, pp. 6989–6997, 2023.
- S. Wang, H. Wu, X. Shi, T. Hu, H. Luo, L. Ma, J. Y. Zhang, and J. Zhou, "TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2024.
- K. Rasul, A. Ashok, A. R. Williams, H. Ghonia, R. Bhagwatkar, A. Khorasani, M. J. D. Bayazi, G. Adamopoulos, R. Riachi, N. Hassen, M. Bilos, S. Garg, A. Schneider, N. Chapados, A. Drouin, V. Zantedeschi, Y. Nevmyvaka, and I. Rish, "Lag-Llama: Towards Foundation Models for Probabilistic Time Series Forecasting," arXiv preprint arXiv:2310.08278, 2023.
- T. Zhou, P. Niu, X. Wang, L. Sun, and R. Jin, "One Fits All: Power General Time Series Analysis by Pretrained LM," in *Advances in Neural Information Processing Systems*, vol. 36, pp. 43322–43355, 2023.
- M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, and Q. Wen, "Time-LLM: Time Series Forecasting by Reprogramming Large Language Models," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2024.

30. D. Cao, F. Jia, S. O. Arik, T. Pfister, Y. Zheng, W. Ye, and Y. Liu, "TEMPO: Prompt-Based Generative Pre-Trained Transformer for Time Series Forecasting," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2024.
31. T. Kim, J. Kim, Y. Tae, C. Park, J.-H. Choi, and J. Choo, "Reversible Instance Normalization for Accurate Time-Series Forecasting Against Distribution Shift," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2022.
32. [32] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2022.
33. 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.
34. H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Time-Series-Library (TSLib)," GitHub repository, 2023. [Online]. Available: <https://github.com/thuml/Time-Series-Library>
35. Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. W. Cottrell, "A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction," in *Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI)*, pp. 2627–2633, 2017.
36. B. Lim, S. O. Arik, N. Loeff, and T. Pfister, "Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting," *Int. J. Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.
37. Y. Kong, Z. Wang, Y. Nie, T. Zhou, S. Zohren, Y. Liang, P. Sun, and Q. Wen, "Unlocking the Power of LSTM for Long Term Time Series Forecasting," *Proc. AAAI Conf. Artif. Intell.*, vol. 39, no. 11, pp. 11968–11976, 2025.
38. A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are Transformers Effective for Time Series Forecasting?," *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 9, pp. 11121–11128, 2023.
39. L. Huang, F. Mao, K. Zhang, and Z. Li, "Spatial-Temporal Convolutional Transformer Network for Multivariate Time Series Forecasting," *Sensors*, vol. 22, no. 3, Art. no. 841, 2022.
40. B. Yu, H. Yin, and Z. Zhu, "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, pp. 3634–3640, 2018.
41. M. Liu, A. Zeng, M. Chen, Z. Xu, Q. Lai, L. Ma, and Q. Xu, "SCINet: Time Series Modeling and Forecasting with Sample Convolution and Interaction," in *Advances in Neural Information Processing Systems*, vol. 35, pp. 5816–5828, 2022.
42. Y. Zhang and J. Yan, "Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2023.
43. X. Piao, Z. Chen, T. Murayama, Y. Matsubara, and Y. Sakurai, "Fredformer: Frequency Debaised Transformer for Time Series Forecasting," in *Proc. 30th ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD)*, pp. 2400–2410, 2024.
44. S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting," in *Advances in Neural Information Processing Systems*, vol. 32, pp. 5244–5254, 2019.
45. S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, and S. Dustdar, "Pyraformer: Low-Complexity Pyramidal Attention for Long-Range Time Series Modeling and Forecasting," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2022.
46. R. Nematirad, A. Pahwa, and B. Natarajan, "Times2D: Multi-Period Decomposition and Derivative Mapping for General Time Series Forecasting," arXiv preprint arXiv:2504.00118, 2025.
47. L. Du, J. Xin, A. Labach, S. Zuberi, M. Volkovs, and R. G. Krishnan, "MultiResFormer: Transformer with Adaptive Multi-Resolution Modeling for General Time Series Forecasting," arXiv preprint arXiv:2311.18780, 2023.
48. Y. Dong, L. Zhang, Y. Zhang, S. Peng, W. Chen, and Z. Ma, "SpecAR-Net: Spectrogram Analysis and Representation Network for Time Series," in *Proc. 33rd Int. Joint Conf. Artif. Intell. (IJCAI)*, pp. 3917–3925, 2024.
49. Z. Li, P. Wolf, and M. Wang, "ArrivalNet: Predicting City-wide Bus/Tram Arrival Time with Two-dimensional Temporal Variation Modeling," arXiv preprint arXiv:2410.14742, 2024.
50. C.-H. H. Yang, Y.-Y. Tsai, and P.-Y. Chen, "Voice2Series: Reprogramming Acoustic Models for Time Series Classification," in *Proc. 38th Int. Conf. Mach. Learn. (ICML)*, PMLR, vol. 139, pp. 11808–11819, 2021.
51. 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 preprint arXiv:2410.10393, 2024.
52. S. Gao, T. Koker, O. Queen, T. Hartvigsen, T. Tsiligkaridis, and M. Zitnik, "UniTS: A unified multi-task time series model," *Advances in Neural Information Processing Systems*, vol. 37, pp. 140589–140631, 2024.
53. C. Wang, Q. Qi, J. Wang, H. Sun, Z. Zhuang, J. Wu, L. Zhang, and J. Liao, "ChatTime: A Unified Multimodal Time Series Foundation Model Bridging Numerical and Textual Data," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 12, pp. 12694–12702, 2025.
54. J. Pathak, S. Subramanian, P. Harrington, S. Raja, A. Chattopadhyay, M. Mardani, T. Kurth, D. Hall, Z. Li, K. Azizzadenesheli, P. Hassanzadeh, K. Kashinath, and A. Anandkumar, "FourCastNet: A global data-driven high-resolution weather model using adaptive Fourier neural operators," arXiv preprint arXiv:2202.11214, 2022.