Mingyue Cheng[1], Zhiding Liu[1], Xiaoyu Tao[1], Qi Liu[1], Jintao Zhang[1], Tingyue Pan[1], Shilong Zhang[1], Panjing He[1], Xiaohan Zhang[1], Daoyu Wang[1], Jiahao Wang[1], and Enhong Chen[1]

[1]University of Science and Technology of China, State Key Laboratory of Cognitive Intelligence

April 10, 2025

Posted on 10 Apr 2025 — CC-BY-NC-SA 4 — https://doi.org/10.36227/techrxiv.174430535.53879341/v1 — e-Prints posted on TechRxiv are preliminary reports that are not peer reviewed. They should not b...

A Comprehensive Survey of Time Series Forecasting: Concepts, Challenges, and Future Directions

Mingyue Cheng, Zhiding Liu, Xiaoyu Tao, Qi Liu*, Jintao Zhang, Tingyue Pan, Shilong Zhang, Panjing He, Xiaohan Zhang, Daoyu Wang, Jiahao Wang, Enhong chen

Abstract—Time series forecasting (TSF) has become an increasingly vital tool in various decision-making applications, including business intelligence and scientific discovery, in today's rapidly evolving digital landscape. Over the years, a wide range of methods for TSF has been proposed, spanning from traditional static-based models to more recent machine learning-driven, data-intensive approaches. Despite the extensive body of research, there is still no universally accepted, unified problem statement or systematic elaboration of the core challenges and characteristics of TSF. The extent to which deep TSF models can address fundamental issues—such as data sparsity and non-stationarity—remains unclear, and the broader TSF research landscape continues to evolve, shaped by diverse methodological trends. This comprehensive survey aims to address these gaps by examining the key entities in TSF (e.g., covariates) and their characteristics (e.g., frequency, length, missing values). We introduce a general problem formulation and challenge analysis for TSF, propose a taxonomy that classifies representative methodologies from the preprocessing and forecasting perspectives, and highlight emerging topics like transfer learning and trustworthy forecasting. Finally, we discuss promising research directions that are poised to drive innovation in this dynamic and rapidly advancing field. The related paper list is available at https://github.com/USTCAGI/Awesome-Papers-Time-Series-Forecasting.

Index Terms—Time Series, Time Series Forecasting, Statistics Model, Deep Learning

1 Introduction

Time Series Forecasting (TSF) has become a pivotal tool for making informed predictions and decisions across various fields such as healthcare [1], finance [2], manufacturing [3], and environment [4], [5]. In an era dominated by vast data generation, the need to predict future trends from time series data becomes increasingly critical. To this end, numerous efforts have been devoted to capturing both temporal and cross-channel correlations for more accurate time series forecasting.

Traditional time series forecasting models, such as Autoregressive Integrated Moving Average (ARIMA) [6] and Exponential Smoothing methods [7], have played a key role in time series analysis. However, as the complexity and scale of real-world time series data continue to grow, these models often find it challenging to capture underlying patterns effectively. The advent of machine learning (ML) and deep learning (DL) techniques has significantly transformed the landscape of time series forecasting [8], [9]. These data-driven approaches can learn from large, complex datasets and automatically discover patterns, offering substantial improvements over traditional methods. However, while ML and DL methods present promising results in many domains, challenges persist in addressing specific issues such as long-term and multivariate dependence capture, as well as modeling of exogenous variables. Furthermore, time series data is often characterized by noise, irregularity, and non-stationarity, making preprocessing and feature extraction critical steps for achieving reliable predictions. Methods such as imputation [10], denoising [11], and normalization [12] have been developed to handle these data challenges, yet further research is needed to refine these techniques for more robust time series forecasting.

With the development of the various forecasting approaches, several surveys have been published over the years, providing valuable insights into different methodologies. For example, [13], [14], [15] reviews the progress of specific deep-learning architecture (Transformers and Graph Neural Networks) or learning strategy (Self-Supervised Learning) for the time series analysis tasks, and [16] further provides a systematic overview of deep models of diverse architectures for different downstream tasks, with a benchmark analysis. More recently, with the advancement and success of foundation models, [17], [18] presents a survey about Large-Language-Model-based and time-series-foundation-model-based approaches for time series analysis. However, these works primarily focus on the deep-learning-based models and explain how different architectures are well suited for time series modeling, therefore lack a unified perspective that covers the diverse research directions, and bridges the historical development with the latest trends for the forecasting task.

To address the abovementioned issue, this paper provides a comprehensive overview focusing on time series forecasting, encompassing both traditional statistical models and advanced data-driven approaches. We delve into the fundamental concepts of time series data, investigate various modeling techniques, and analyze the challenges

- *All authors of this paper are affiliated with the University of Science and Technology of China, State Key Laboratory of Cognitive Intelligence.*
- *Email: {mycheng, qiliuql}@ustc.edu.cn. The corresponding author is Qi Liu.*

presented by real-world time series data. We also highlight the importance of time series preprocessing, offering a detailed examination of techniques such as tokenization and graph transformations, which have gained attention in recent literature for their effectiveness in improving model performance. Additionally, we discuss the progress in time series forecasting, including statistical models, data-driven approaches, transfer learning methods, and trustworthy forecasting techniques. By synthesizing diverse TSF methods under a single unifying framework, our survey aims to clarify the rich methodological landscape while exposing unresolved challenges and open questions.

The remainder of this article is organized as follows. Section 2 formally defines the fundamental concepts of TSF and introduces the widely used evaluation protocols. Following this, Section 3 provides an analysis of key data characteristics and challenges. In Section 4, we describe several mainstream time series preprocessing strategies. We then present the classical statistical-based approaches in Sections 5, before transitioning to data-driven methods in Section 6, which covers both machine-learning-based and deep-learning-based approaches. Besides, Section 7 focuses on recent advances in transfer learning models, and Section 8 discusses the trustworthy TSF. Additionally, We introduce some prevalent benchmark datasets and representative applications of TSF in various domains in Section 9, and highlight open research directions in Section 10. Finally, we conclude the survey in Section 11.

2 Foundation Concept Description

2.1 Introduction to Time Series Data

A time series is commonly defined as a sequence of data points indexed in chronological order, where each observation is obtained at a specific (often uniformly spaced) time interval. Although the data are frequently recorded at discrete intervals (e.g., hourly, daily, monthly), many real-world phenomena underlying these observations can be treated as continuous and, in principle, unbounded in both time and value. As a result, time series modeling must address both the discrete nature of the measurements and the potential continuity of the process itself.

A typical time series can often be decomposed into several key components:

- **Trend:** A long-term progression of increasing or decreasing values in the series over extended periods.
- **Seasonality:** Regular, repeating patterns that recur at fixed intervals (e.g., daily, weekly, annually), driven by predictable factors such as calendar effects or environmental cycles.
- **Cyclicity:** Fluctuations that are not strictly tied to a fixed calendar period but exhibit recognizable rises and falls over time, often linked to economic or other systemic influences.
- **Irregular or Random Variations:** Unpredictable changes in the series that are not explained by the trend, seasonality, or cyclical behavior, often treated as residual noise.

Because time series data are recorded in chronological order, statistical dependencies exist among observations, making it crucial to consider temporal structures when designing forecasting or analysis techniques. These properties—including continuous, unbounded states and typically uniform sampling intervals—distinguish time series data from cross-sectional data, where observations lack inherent temporal ordering.

2.2 Basic Concepts of Time Series Forecasting

Time series forecasting aims to predict future values of a target series based on past observations and, potentially, additional explanatory variables. The core formulation typically involves two key components: the *look-back window* (i.e., a set of recent past observations) as input, and the *predicted window* (i.e., one or more future time steps) as output. Below, we outline several fundamental concepts:

- **Look-back Window of the Target Series:** This is a sequence of consecutive time steps (e.g., the previous L observations) from the target series itself. It acts as the primary source of historical context, capturing trends, seasonal patterns, and other temporal dependencies.
- **Covariates (Exogenous Variables):** Beyond the target series, many applications leverage auxiliary factors such as weather conditions, economic indicators, or demographic information. These additional inputs, known as *covariates* or *exogenous variables*, can offer valuable insights when the target series is influenced by external drivers.
- **Predicted Window of the Target Series:** In forecasting tasks, the output is typically a set of future time steps to be estimated. This window could be as short as a single time step (e.g., predicting the next hour) or span multiple future periods (e.g., predicting the next week or month).
- **One-step vs. Multi-step Prediction:***One-step Prediction* aims to forecast a single future time step at a time (e.g., $t+1$ given observations up to t). It is often simpler to implement, but may require repeated application to forecast multiple steps ahead. *Multi-step Prediction* directly generates forecasts for several consecutive future steps (e.g., $t+1$ to $t+H$) in one shot. This approach can capture long-range dependencies but often entails greater complexity and potential error accumulation.
- **Univariate vs. Multivariate Forecasting:** *Univariate Forecasting* considers only a single target series without additional external information. *Multivariate Forecasting* involves multiple interrelated series or auxiliary inputs to exploit cross-dependencies and potentially improve accuracy.
- **Iterative vs. Direct Strategies:** *Iterative Forecasting* forecasts one step ahead repeatedly, feeding each predicted value back as input for the next step. *Direct Forecasting* trains separate models (or a single model with multi-output) to predict each future step or window of interest directly, avoiding error propagation at the cost of additional modeling complexity.

In summary, the essential building blocks of time series forecasting encompass which segments of past data (i.e., the look-back window) and what external factors (i.e.,

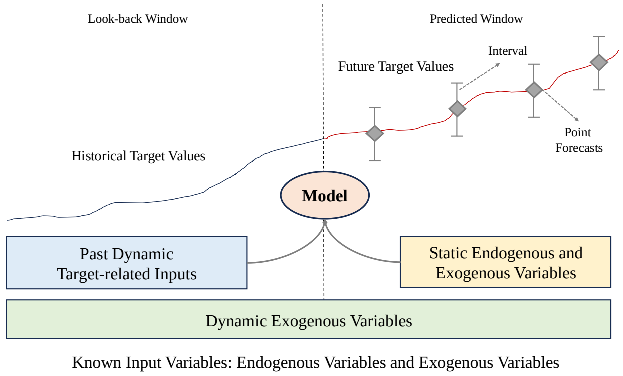

Fig. 1: An illustration of time series forecasting.

covariates) are used to model or learn patterns that best predict a specified time horizon (i.e., the predicted window). Understanding these basic concepts is pivotal for designing and implementing effective forecasting solutions.

2.3 Problem Definition of Time Series Forecasting

Time series forecasting aims to predict future values based on historical data. Depending on the nature of the output, forecasting problem can be categorized as **Non-Probabilistic (Point-based)** or **Probabilistic** ones.

2.3.1 Point-based Time Series Forecasting

Point-based forecasting focuses on generating a single deterministic prediction for each time step. These methods aim to minimize a specific error metric, such as Mean Absolute Error (MAE) or Root Mean Square Error (RMSE), without explicitly accounting for uncertainty in the predictions. Formally, given a time series $\mathbf{y} = \{y_1, y_2, \ldots, y_T\}$, the objective is to predict the future values $\hat{\mathbf{y}} = \{\hat{y}_{T+1}, \hat{y}_{T+2}, \ldots, \hat{y}_{T+\tau}\}$, where $\hat{y}_t$ represents a point estimate, such as the conditional expectation:

$$\hat{y}_t = \mathbb{E}[y_t \mid \mathbf{y}],$$

assuming that the model aims to minimize the squared error. Common methods include ARIMA, Support Vector Regression (SVR), and Neural Networks.

2.3.2 Probabilistic Time Series Forecasting

In contrast, probabilistic forecasting generates a predictive distribution for future values, capturing the inherent uncertainty in the data. Instead of producing a single point estimate, the model provides a conditional probability density function (PDF) $p(y_{T+1:T+\tau} \mid \mathbf{y})$. This allows for more nuanced predictions, such as the ability to compute prediction intervals or quantify risk. For example, the predictive distribution can be represented as:

$$y_t \sim p(y_t \mid \mathbf{y}),$$

where $p(y_t \mid \mathbf{y})$ is typically parameterized by the model (e.g., Gaussian distribution $\mathcal{N}(\mu_t, \sigma_t^2)$ with mean μ_t and variance σ_t) or modeled using advanced techniques like DeepAR [19], Variational Autoencoders [20], or Flow-based generative models [21].

2.4 Evaluation Protocols

Given the strong temporal dependency and sequential nature of time series data, evaluation protocols have been developed to objectively and fairly assess model performance in various time series forecasting tasks. The primary aim of these protocols is to accurately evaluate the effectiveness of different time series analysis techniques with diverse architectures. Specifically, evaluation protocols standardize methods for dividing data into training, validation, and test sets and define appropriate metrics to assess both short-term and long-term predictive performance. These protocols also reduce the risk of data leakage and ensure accurate evaluation of model generalization capabilities in time series contexts. By establishing a standardized framework, evaluation protocols promote fair model comparisons and optimization, ensuring that prediction results are interpretable and reliable, ultimately making model evaluation more rigorous and robust [16].

For time series data, a 7:2:1 ratio is commonly applied to divide the data into training, validation, and test sets. This distribution enables the model to effectively learn features, adjust parameters, and evaluate its generalization performance. Keeping the split in chronological order also helps prevent the data leakage issue, ensuring more reliable forecast outcomes.

Table 1 summarizes the calculation formulas, advantages and disadvantages of these indicators. Time series forecasting methods are typically assessed using two broad categories of metrics: *point-based* and *probabilistic*. Point-based metrics measure the accuracy of single-valued forecasts, while probabilistic metrics quantify the quality of an entire predictive distribution.

2.4.1 Point-Based Metrics

The point-based metrics compare the forecast $\hat{y}_t$ to the observed ground truth y_t at each time step. There are many commonly used point prediction indicators(e.g., MAE and MSE). Table 1 presents a compilation of commonly used metrics for point prediction, including MAE, MSE, SMAPE, MAPE, MASE, and OWA, along with a discussion of their respective advantages and limitations.

Briefly speaking, MAE and MSE are the most commonly used point-based metrics, comparing the error between the true values and predictions in the original space, while SMAPE and MAPE assess model performance based on percentage errors. In contrast, MASE and OWA evaluate the model through scaling and weighting approaches.

2.4.2 Probabilistic Metrics

In settings where models generate a predictive distribution instead of a single point estimate, probabilistic metrics evaluate distributional coverage, which measures the proportion of times the observed value falls within a predicted interval [22]. The choice of metric should reflect the practical objectives of the forecasting task, whether that involves minimizing average error, capturing uncertainty, or targeting specific quantiles. Table 1 provides an overview of widely used indicators for probabilistic forecasting, namely CRPS, ρ-Quantile Loss, NLL, and VG, accompanied with a discussion of their respective strengths and weaknesses.

TABLE 1: Overview of the metrics widely evaluated in point-based forecasting and probabilistic forecasting tasks.

Task	Metric	Formula	Advantages	Disadvantages
Point-based Forecasting	MAE	$\frac{1}{n}\sum_{t=1}^{n}\lvert y_t-\hat{y}_t\rvert$	Intuitive and robust to outliers	Insensitive to large errors
	MSE	$\frac{1}{n}\sum_{t=1}^{n}(y_t-\hat{y}_t)^2$	Penalizes deviations in predictions	Sensitive to outliers
	SMAPE	$\frac{1}{n}\sum_{t=1}^{n}\frac{2\lvert y_t-\hat{y}_t\rvert}{\lvert y_t\rvert+\lvert\hat{y}_t\rvert}\times 100\%$	Standardized and works across scales	Sensitive to zeros and large range
	MAPE	$\frac{1}{n}\sum_{t=1}^{n}\left\lvert\frac{y_t-\hat{y}_t}{y_t}\right\rvert\times 100\%$	Intuitive and easily compute	Sensitive to small values
	MASE	$\frac{\frac{1}{n}\sum_{t=1}^{n}\lvert y_t-\hat{y}_t\rvert}{\frac{1}{n}\sum_{t=1}^{n}\lvert y_t-y_{t-1}\rvert}$	Scaled and comparable across datasets	Seasonality-sensitive reference-dependent
	OWA	$\sum_{t=1}^{n}w_t\cdot\lvert y_t-\hat{y}_t\rvert$	Flexible and weighted error handling	Requires careful weight selection and complex
Probabilistic Forecasting	CRPS	$\int_{-\infty}^{\infty}[F(z)-\mathbf{1}(z\geq y)]^2\,dz$	Interpretable and widely applicable	Single-variable only; Sensitive to outliers
	ρ-QL	$2(\hat{y}-y)\left(\rho I_{\hat{y}>y}-(1-\rho)I_{\hat{y}\leq y}\right)$	Ideal for quantile forecasting	Overfitting to asymmetric Data
	NLL	$-\log p_D^f(y)$	Rich gradient info and solid theoretical basis	Lack of intuitive interpretation
	VG	$\sum_{a,b}\left(\lvert y_a-y_b\rvert^p-\mathop{\mathbb{E}}_{x\sim\mathcal{D}^f}[\lvert x_a-x_b\rvert^p]\right)^2$	Describe spatial correlation	High computational complexity

Continuous Ranked Probability Score (CRPS) is used to evaluate the accuracy of probabilistic predictions, where $F(z)$ represents the cumulative distribution function and y is the value to be predicted. The $\rho-$Quantile Loss, with $\hat{y}$ as the predicted value for the quantile level ρ, and $I_{\hat{y}>y}$ and $I_{\hat{y}\leq y}$ as indicator functions, computes the quantile-level prediction accuracy. In the context of Negative Log Likelihood (NLL), $\mathcal{D}^f$ is related to the data distribution or prediction model, and $p_D^f(y)$ is the predictive probability density function, with NLL mainly aiming to assess the model's goodness of fit to the data, where a smaller value implies a better fit. Besides, the Variogram (VG) describes the spatial variability of the data, helping to understand its spatial structure and correlation. It involves observed values y_a and y_b potentially related to x_a and x_b, a given parameter ρ and the data distribution $\mathcal{D}^f$.

3 Challenges Analysis

Sequence signals collected in chronological order can be classified as time series data, a distinct data modality derived from various sensors or real-world observations. Time series data captures the dynamic evolution of systems over time, reflecting both short-term fluctuations and long-term trends. While rich in information, time series data exhibits several key characteristics that pose specific challenges for accurate forecasting. In this survey, we provide a comprehensive discussion of these characteristics and challenges as follows:

3.1 Noise, Anomalies, and Outliers

Time series data, which reflects a system through multiple sensors or evaluations, inevitably raises concerns related to data quality. This can result in noise, anomalies, and outliers within the collected series, ultimately undermining the performance of downstream forecasting tasks. Despite significant efforts focused on individual imputation and anomaly detection [23], [24], mainstream research still often assumes that forecasting datasets are well-filtered and of high quality. As a result, there remains a challenge in exploring effective methods to mitigate the impact of low-quality data on forecasting models.

3.2 Irregular Time Series

Although continuous formulations are widely adopted in the field of time series forecasting, many real-world datasets are irregularly sampled due to technical constraints or limitations inherent in the data collection process. As a result, observations are recorded at variable time intervals. The time gaps between these observations themselves contain crucial information about the underlying time series, thereby presenting significant challenges in handling the irregular sampling intervals and effectively capturing the evolving latent dynamics [25], [26].

3.3 Long-term Sequence Forecasting

Time series data essentially consists of a sequence of continuous numerical values and can thus be considered an intermediary modality between the extensively studied fields of image and language data, with further inherent characteristics. Modeling the temporal dependencies within time series data remains a prominent research focus in time series forecasting [27]. From a sequence perspective, time series data often spans longer time intervals, resulting in extended sequences. Additionally, from a numerical standpoint, time series data typically lacks well-defined upper and lower bounds, further complicating accurate forecasting. More specifically, many real-world applications are framed as long-term time series forecasting tasks [28], [29], requiring forecasts over extended horizons and windows, which presents substantial challenges in capturing long-term dependencies within the series. Furthermore, a critical challenge lies in mitigating the impact of error accumulation during the numerical sequence modeling process [30].

3.4 Multivariate Dependence Modeling

Collected time series datasets typically exhibit multivariate characteristics, primarily due to the complex cross-channel dependencies embedded within the data. On the one hand, causal or leading effects may exist in certain scenarios, including traffic load prediction and temperature analysis [31]. However, clear prior assumptions or experiential insights about these dependencies are often lacking, especially in complex and high-dimensional systems. This absence of guidance presents a significant challenge in accurately identifying and modeling cross-channel dependencies. On the other hand, in many complex systems, such as the human body or weather systems, pre-existing knowledge often makes it difficult to identify the relevant variables. This necessitates the collection of data from multiple perspectives to enable a precise and holistic analysis of the entire system [4]. Consequently, sophisticated approaches are required to effectively capture the hidden patterns and dependencies across multiple time series. The current channel-dependent [32] and channel-independent [33] solutions still warrant further exploration [34].

3.5 Exogenous Variables Modeling

Real-world systems often exhibit a partially observed nature due to incomplete prior knowledge, which can lead to suboptimal forecasting outcomes when relying solely on endogenous variables [35], [36], [37]. In particular, certain scenarios necessitate the incorporation of external variables, such as weather, policies, holidays, and macroeconomic indicators. The impact of these external variables on time series objectives may be nonlinear and time-varying. However, it remains unclear how to effectively identify the key exogenous variables and integrate this multimodal information into a unified forecasting framework.

3.6 Distribution Shift Modeling

Time series data typically exhibit non-stationary characteristics, meaning that the distribution of the data may change rapidly over time. This results in discrepancies between different time spans, ultimately hindering the generalization capabilities of deep learning models, as the distribution shift contradicts their fundamental assumption of independent and identically distributed (I.I.D.) data [38], [39]. Furthermore, traditional methods for addressing distribution shift, such as domain adaptation and domain generalization, may not be well-suited to this task, as defining a domain for time series data is not straightforward [12]. As a result, it remains an open challenge to develop models or frameworks that can effectively mitigate the impact of non-stationarity.

3.7 Trend-Seasonal Pattern Recognition

Time series data are usually numerical responses to natural indicators or human behavior, making them highly influenced by natural rhythms and exhibiting distinct, structured numerical patterns. It is widely accepted that time series data can be decomposed into several independent components, and considerable efforts have been dedicated to developing advanced decomposition techniques over the years [40], [41]. In forecasting tasks, a common approach involves a simple neural decomposition layer that separates the original series into trend and seasonal components. This decomposition captures the long-term progression and seasonal patterns inherent in the data, thereby facilitating more accurate forecasting [42]. However, precise modeling of structural characteristics, multi-periodicity, and the effects of holidays in time series data remains a challenge that requires further exploration.

3.8 Multi-scale Representations

The multi-scale characteristics of time series data are primarily exhibited from two perspectives: multi-granularity and hierarchical [43]. On one hand, time series data consists of discrete records sampled from a continuous time-space. Depending on the sampling frequency, it inherently exhibits multi-granularity characteristics, where high-frequency series tend to be more informative but noisier, while low-frequency series offer smoother trends but less detail. On the other hand, both local disturbances and global trends significantly influence the time series data. Consequently, determining an appropriate modeling granularity, fully exploiting multi-scale features, and achieving effective feature fusion present key research challenges [44], [45].

3.9 Computational Efficiency

In the forecasting of time series, data often consists of series from multiple channels, and both the forecast horizon and prediction lengths are typically long. This easily lead to a significant computational efficiency challenge for applying traditional neural networks for time series forecasting, particularly problematic in tasks requiring real-time performance, such as stock price forecasting [46] and human physiological signal prediction [47]. Consequently, developing prediction methods that effectively balance computational efficiency with model representation capability remains a key research challenge.

3.10 Generalizability and Transferability

Unlike the pixels used in computer vision (CV) and word tokens in natural language processing (NLP), there is no standardized definition of unified semantic units in time series analysis. Although these series share a similar data format, the values within them may have inconsistent meanings across different contexts due to the distinct physical significance in various scenarios. As a result, most forecasting methodologies are typically trained and evaluated on a single dataset, limiting their generalizability and transferability. Therefore, the development of a robust and scalable forecasting foundation model, built on extensive multi-source pretraining data, represents a significant challenge [48], [49]. Furthermore, the operational mechanisms, such as scaling laws and the influence of data distribution on model performance, remain unclear [50].

4 Time Series Preprocessing

Effective preprocessing of time series data is a critical step in building robust forecasting models. This section highlights key preprocessing techniques designed to address data quality issues and transform raw time series into more informative and structured forms. These techniques are essential for preparing time series data for analysis and ensuring its readiness for forecasting tasks.

4.1 Time Series Imputation

Time series data, often collected from complex sensor systems in real-world environments, is frequently affected by quality issues. First and foremost is the problem of missing data caused by factors such as localized sensor failures, which may lead to the loss of critical temporal and cross-channel dependency information. To this end, time series imputation aims to address missing values in the data by restoring them based on the observed segments. Considerable research has been dedicated to this area, with methods typically falling into two categories: predictive and generative approaches [10]. The predictive approach relies on deterministic prediction techniques, using traditional neural networks such as RNNs and attention-based models [23], [51], as well as reconstruction-based learning objectives, often without accounting for the uncertainty of the imputed values. In contrast, the generative approach employs methods such as GANs and diffusion models [52], [53], which facilitate the quantification of imputation uncertainty.

4.2 Time Series Denoising

More generally, time series data often suffers from significant noise contamination, in addition to the issue of missing data. Notably, noise in the data not only impacts the accuracy of model predictions but also affects the evaluation process of the models. Consequently, the denoising process is also not negligible for accurate forecasting results. Denoising approaches primarily aim to mitigate noise in time series data, thereby enhancing its quality and suitability for downstream tasks. Prominent techniques in this area include traditional decomposition-based methods [54] and filtering-based approaches [55], alongside more recent learning-based models [11], [56].

4.3 Standardization and Stationarization

Due to the dynamic nature of time series data, it inherently exhibits diverse data scales and complex distributions. To address this, standardization and stationarization are critical preprocessing steps, ensuring that time series data is well-prepared for reliable forecasting.

While time series data may share a similar format, the values within them often have inconsistent meanings across different contexts, given the distinct physical significance in various scenarios. As a result, collected time series data may exhibit significant scale differences, sometimes spanning orders of magnitude. To handle this, standardization is employed to convert the data into a standard normal distribution from a global dataset perspective. This eliminates the impact of different channel units and ensures a stabilized learning process of the forecasting model.

TABLE 2: Method types and features of decomposition approaches.

Type	Method	Feature
Multiplicative Methods	VMD Methods	Decomposed into specific frequency components, suitable for economic and financial time series forecasting, with strong stability.
	Wavelet and Filter-Based Methods	Captures frequency characteristics at different levels, demonstrating strong adaptability when combined.
	SSA Methods	Suitable for nonlinear and non-stationary time series, effectively avoiding mode mixing.
Additive Methods	EMD Methods	Suitable for non-stationary and nonlinear time series.
	STL Methods	Flexible adjustment of trend and smoothness, suitable for long-term series analysis.
	LMD Methods	Using average mutual information, particularly effective for capturing market data volatility.

On the other hand, stationarization approaches are proposed to address non-stationarity due to the distribution shifts between time series instances. DAIN [57] introduces a non-linear network that adaptively learns how to normalize each input instance, while RevIN [12] proposes a symmetric normalization method, first normalizing the input sequences and then denormalizing the model output sequences via instance normalization. Dish-TS [58] identifies both intra- and inter-space distribution shifts in time series and alleviates these issues by learning distribution coefficients. SAN [39] extends the normalization concept from the instance level to the time series slice level to enhance performance. Lastly, SIN [59] identifies key statistics for normalization and automatically learns the corresponding normalization transformations.

4.4 Time Series Decomposition Techniques

Time series decomposition is a key technique in time series analysis and forecasting. As shown in Figure 2, the decomposition process typically involves separating the original time series into three components: the trend (T_t), which captures long-term changes; the seasonal component (C_t), which represents periodic patterns; and the residual (S_t), which accounts for random noise.

As shown in table 2, decomposition can be performed using two common models: the additive methods, where the additive methods are expressed as:

$$Y_t = T_t + C_t + S_t, \tag{1}$$

and the multiplicative methods, where these methods are represented as:

$$Y_t = T_t \times C_t \times S_t. \tag{2}$$

By isolating these components, decomposition provides insights into the underlying structure of time series, improving the accuracy and interpretability of forecasting models.

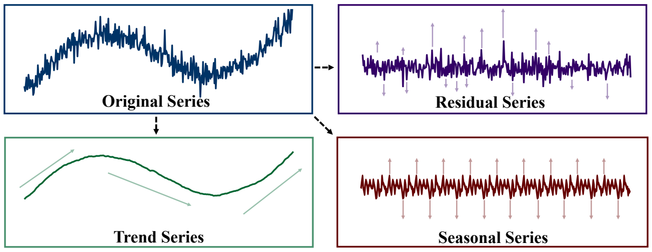

Fig. 2: An illustrative example of time series decomposition into its components: trend, seasonality, and residual.

4.4.1 VMD Methods

Variational Mode Decomposition (VMD) methods decompose time series into band-limited modes with specific central frequencies, using variational principles to identify the optimal modal bases. These methods allow for simultaneous estimation of both the modes and their corresponding frequencies, making it particularly effective for signals with multiple frequency components. VMD is ideal for financial forecasting, where accurate decomposition enhances prediction stability [60]. MVMD extends VMD to decompose multiple related time series, resolving frequency mismatches and capturing inter-variable relationships, which is useful for forecasting in meteorology and wind power [61]. SVMD integrates VMD with Sample Entropy (SampEn) to decompose industrial electricity load data into trend and fluctuation components, improving performance by addressing non-stationarity [62].

4.4.2 Wavelet and Filter-Based Methods

Wavelet and filter-based decomposition methods apply multi-scale analysis to extract frequency components at various resolutions. By isolating different frequency bands, these methods are particularly effective in modeling complex time series. Empirical Wavelet Transform (EWT) decomposes time series into sub-levels, extracting frequency-specific features and enhancing adaptability when combined with models like LSTM and RELM for error correction [63]. Time-Varying Filtered Empirical Mode Decomposition (TVFEMD) effectively handles high-noise data, such as bridge monitoring signals, by decomposing them into stable IMFs, improving interpretability and accuracy [64].

4.4.3 SSA Methods

Singular Spectrum Analysis (SSA) methods leverage embedding dimension and principal component selection to achieve adaptive decomposition of time series, ensuring the extraction of physically meaningful components while effectively suppressing noise. SSA performs exceptionally well when handling non-linear and non-stationary time series, avoiding mode-mixing issues and achieving high precision in complex time series forecasts [65].

4.4.4 EMD Methods

Empirical Mode Decomposition (EMD) methods decompose non-stationary and nonlinear time series into Intrinsic Mode Functions (IMFs), each representing oscillatory modes at different scales. This decomposition reveals the signal's multi-scale characteristics and improves forecasting by capturing distinct frequency components. EMD is effective in hydrological analysis due to its adaptability to complex nonlinear features [66]. Thresholding irrelevant IMFs refines decomposition, enhancing drought forecasting [67]. EEMD, by adding noise, mitigates mode-mixing in EMD, improving decomposition of annual runoff data [68]. CEEMD further reduces noise and reconstruction error with paired opposite white noise, strengthening financial data analysis by better capturing trends and fluctuations.

4.4.5 STL Methods

Seasonal and Trend decomposition using Loess (STL) methods utilizes LOESS to smoothly estimate trend, sesaonal and residual components. Its flexibility in adjusting trend and smoothing levels makes it ideal for long-term series analysis and handling missing data. STL enhances long-term forecasting accuracy and shows consistent performance across various time series characteristics [41], [69].

4.4.6 LMD Methods

Local Mean Decomposition (LMD) methods involve removing the local mean of a signal to extract its IMFs. This approach enhances forecasting accuracy by using average mutual information to capture underlying patterns. LMD is particularly effective for capturing price volatility in market data, such as oil prices, enhancing short- and long-term forecasting by mitigating endpoint effects [70].

4.5 Tokenization

Time series tokenization is a technique that converts time series data into a format that models can process. It primarily comes in two types: continuous tokenization and discrete tokenization.

4.5.1 Continuous Tokenization

Continuous tokenization methods directly process raw time series data by breaking it down into continuous segments or points, primarily categorized into point-wise and patch-wise approaches.

For point-wise methods, such as Informer [30] and NHits [44], data at each time point is processed independently. This approach is relatively simple to implement, but it can lead to significant computational overhead when dealing with long time series [71] and does not provide sufficient semantic information [72].

Subsequently, many methods opt to grouping specific number of adjacent time points into one block for unified modeling at the block level. This approach, called patch-wise tokenization, enhances the model's ability to understand local patterns and dynamic changes within the time series. The earliest work to apply the patch concept within the time series domain is PatchTST [73], which divides the time series into equal-sized segments and feeds them into a Transformer Encoder, achieving outstanding forecasting results. As the field of time series analysis evolves, researchers have discovered that the patching technique not only naturally adapts to transformer architectures [48], [74] but also plays an indispensable role in other architectures such as CNNs [75], [76] and MLPs [77].

However, a rigid patching operation may lead to the loss of temporal information, such as complete peaks and cycles. Recently, some studies have focused on improving the patching process, aiming to enable individual patches to retain more complete and rich temporal segment information. Examples of such advancements include deformable patchify [78], [79] and multi-scale patchify [80], [81]. The former allows the model to automatically find the optimal partition boundaries for each patch in a data-driven manner, while the latter enables the model to handle patch features of different scales, thereby mitigating the issue of local feature damage. In addition, some studies like Lag-llama [82] have also explored using lag variables instead of continuous time historical variables to construct a single patch, which allows the single patch to carry more comprehensive periodic information.

4.5.2 Discrete Tokenization

As discrete tokenization has demonstrated excellent results in image and audio generation [83], [84], researchers begin to explore how to represent time series data as discrete tokens. Let $X = \{x_1, x_2, \ldots, x_T\}$ be a continuous time series, where $x_t \in R$ for $t = 1, 2, \ldots, T$. Discretization involves mapping X to a sequence of tokens $S = \{s_1, s_2, \ldots, s_N\}$, where $s_i \in V$ and V is a finite vocabulary.

Although tokenize time series into discrete space may lead to some information loss [85], [86], this approach also has significant advantages. Compared to the continuous tokenization, the discrete tokenization can better capture the structure and patterns of data through discretized representations, while also exhibiting greater robustness to noise [87]. Additionally, discretized time series feature can integrate more effectively with other discrete data (such as text), making them suitable for multimodal learning tasks, thereby providing improved performance and efficiency in specific application scenarios [84], [88].

Currently, the most common discretization method in time series forecasting approaches is the vector quantization networks (VQN) [87]. VQN-based method is a data-driven discretization technique, which utilizing deep learning techniques to learn discrete representations of time series [86], [89], [90]. This scheme typically includes a vector quantization network that, through a learning paradigm focused on reconstruction, can automatically learn how to map time series data into a discrete symbolic space, and is capable of obtaining a high-density codebook that serves as a dedicated embedding vector table for the time series. The advantage of this method is its ability to handle complex data and provide rich information while reducing the length of time series [91], but it comes with higher computational costs and requires sufficient data and training time.

In addition, there are some statistical-based time series discretization methods that are used in models related to time series. These methods (such as those employed in Chronus [22] and WaveNet [88]) typically utilize statistical characteristics of the data, such as mean, variance, and quantiles, as bucketing thresholds, assigning individual time series data points to the corresponding buckets. The advantage of this approach is its simplicity and intuitiveness, as it requires no training; however, it does not reduce the length of the series [90].

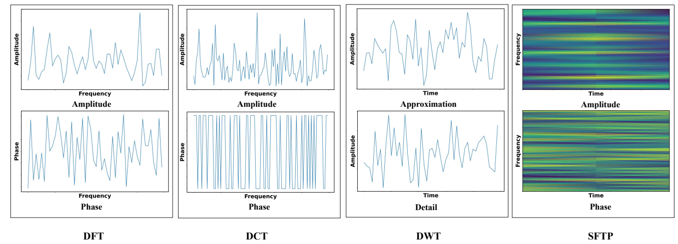

Fig. 3: An example of the outcomes of applying four time-frequency transformation methods, namely DFT, DCT, DWT, and STFT, to a time series.

4.6 Frequency Transformation

Frequency transformation offers distinct advantages over traditional time-domain methods by providing localized time-frequency representations. It effectively captures instantaneous frequency, enabling precise tracking of frequency variations in time series with abrupt changes. It is essential for analyzing non-stationary time series, where frequency dynamics evolve over time, yielding deeper insights. Proper selection of transformations enhances the representational power of deep learning models, improving predictive accuracy.

Time-frequency transformation methods in time series analysis include DFT, DCT, DWT, and STFT, which are briefly summarized in the table3. DFT decomposes a time series into sine and cosine waves, representing it in the frequency domain [92]. DCT expresses the series as a weighted sum of cosine functions with varying frequencies [93]. DWT applies multi-scale analysis and wavelet functions to capture both global trends and local details [94]. STFT segments the series and applies Fourier Transform to each segment, extracting time-frequency information [95].

DFT and DCT are effective for stationary time series, revealing frequency components and periodic characteristics [96]. However, strictly stationary series are rare, as most data are influenced by external factors, with non-stationary series like financial and meteorological data being prevalent. STFT is ideal for time series with localized frequency stability but variability across the series, extracting time-frequency information via a sliding window approach [97]. In contrast, DWT excels in analyzing non-stationary series with instantaneous frequency changes, such as EEG and financial data, using multi-scale analysis to capture frequency variations [98]. Moreover, time series often contain noise from measurement errors, system randomness, or external interference. DWT and DCT are commonly used for denoising, with DWT effectively removing high-frequency noise and DCT primarily for compression and feature extraction [99].

4.7 Graph Transformation

One of the most significant characteristics and challenges in multivariate time series forecasting is the complex, often non-linear correlations between variables. A viable solution is to construct a spatial-temporal graph from the original time series data, where each node represents a variable and the adjacency matrix may evolve over time. These graphs are processed using Graph Neural Networks (GNNs) to

TABLE 3: Time-frequency transformation methods.

Method	DFT	DCT	DWT	STFT
Formulation	$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{N}kn}$	$X(k) = \sum_{n=0}^{N-1} x(n)\cos\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right)$	$W_{j,k} = \frac{1}{\sqrt{2^j}}\sum_{n=0}^{N-1} x(n)\psi\left(\frac{n-k}{2^j}\right)$	$X(t,f) = \int_{-\infty}^{\infty} x(\tau)w(t-\tau)e^{-j2\pi f\tau}\, d\tau$
Working Mechanism	Decomposes a signal into sine and cosine components to analyze frequency.	Uses cosine functions to represent the signal, focusing on low frequencies.	Analyzes the signal at different scales to capture both low and high frequencies.	Applies Fourier Transform to signal segments for localized frequency analysis.
Application Scenarios	Best for stationary signals with constant properties, such as periodic behaviors.	Used for compression and feature extraction in stationary signals, like images or videos.	Suitable for non-stationary signals with changing frequencies, such as financial or biomedical data.	Ideal for analyzing signals with localized frequency changes, like speech or audio.

effectively capture cross-channel relationships. Generally, spatial-temporal graphs can be generated through either heuristic-based or learning-based approaches [14], as illustrated in Figure 4.

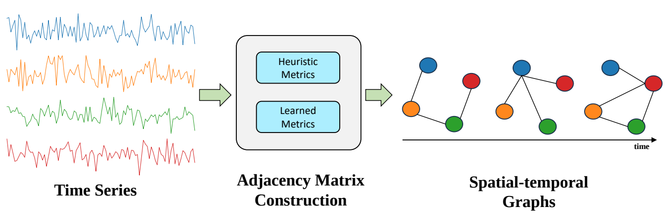

Fig. 4: Illustration of the process of constructing spatial-temporal graphs.

Heuristic-based methods construct the spatial-temporal graph based on the inherent characteristics of the original time series data. For instance, spatial distance and connectivity metrics between variables can provide valuable guidance in typical scenarios such as traffic load forecasting and weather prediction [100]. In addition, for forecasting tasks where prior knowledge of spatial correlations between variables is lacking, compositional methods based on data similarity metrics are widely used. These methods construct an approximate adjacency matrix based on the Pearson Correlation Coefficient (PCC) [101] or Dynamic Time Warping (DTW) distance [102] between the multivariate time series.

Learning-based approaches focus on learning the graph structure in an end-to-end manner, aligning the learning process with the forecasting task itself. This allows for the data-driven discovery of less obvious graph structures. Typically, these methods employ additional neural networks alongside the main forecasting model to generate the adjacency matrix by leveraging interactions between learned variable embeddings [103], [104] or the parametric distribution [105] based on the observed time series data.

5 Statistical Models

In time series forecasting, statistical models based on traditional methodologies occupy a prominent position due to their theoretical foundation and structured approach to model design. These models leverage inherent patterns within time series data, such as lagged values and error terms, to identify and predict future series, effectively capturing the underlying dynamics of the data. Classical time series models typically assume a linear relationship, relying on established statistical principles to produce accurate forecasts. In this section, we primarily introduce ARIMA [6] and its extensions, Exponential Smoothing methods, the State-Space Model (SSM), and the Gaussian Mixture Model (GMM), discussing their key characteristics.

5.1 Auto-Regressive Moving Average and Extensions

Autoregressive (AR) and moving average (MA) components are fundamental techniques in time series analysis. Several variants of the moving average (MA) model, such as the Simple Moving Average (SMA) and Exponential Moving Average (EMA), have been developed to improve forecast accuracy. SMA smooths short-term fluctuations by averaging over a fixed window [106], while EMA assigns exponentially decreasing weights to recent observations, making it particularly responsive to trends and seasonal changes [107]. These variants enhance flexibility in capturing different series patterns.

The Auto-Regressive Moving Average (ARMA) model combines AR and MA components and is effective for stationary time series, where statistical properties like mean, variance, and autocorrelation remain constant [27]. However, for non-stationary data, ARMA is insufficient. The Auto-Regressive Integrated Moving Average (ARIMA) model addresses this by introducing differencing (I) to stabilize the mean and make the series stationary, making it suitable for a broader range of data [6]. For data with seasonal fluctuations, the Seasonal ARIMA (SARIMA) model extends ARIMA by incorporating seasonal components to capture periodic patterns, though its complexity in model specification and parameter tuning can be challenging [6].

5.2 Exponential Smoothing Methods

Exponential smoothing methods are widely used in time series forecasting due to their simplicity, efficiency, and adaptability. The development of these methods focuses on advancements in model selection, parameter optimization, and techniques for adapting to structural changes to improve

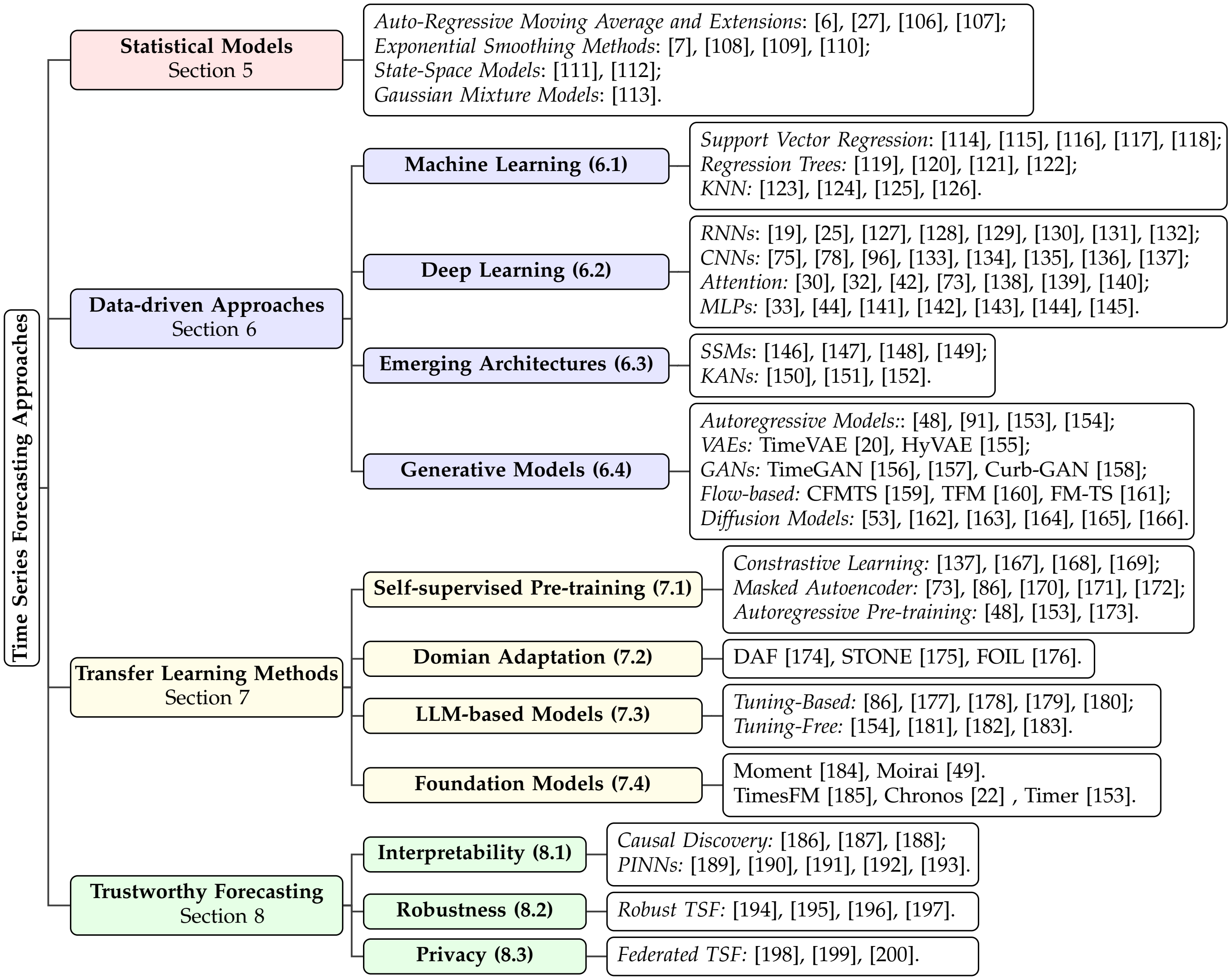

Fig. 5: A brief summary of the time series forecasting approaches.

forecasting accuracy [108]. Simple Exponential Smoothing (SES), which assigns weights to observations using a smoothing constant α, enables precise predictions [109]. Several studies compare different exponential smoothing models, highlighting the robustness and accuracy of state-space and dynamic nonlinear approaches, as well as strategies for managing uncertainties [110]. For seasonal data, the Holt-Winters method decomposes the time series into trend, seasonal, and random components, supporting both additive and multiplicative forms. This makes it particularly suitable for industries with predictable seasonal demand, such as retail and energy [7].

5.3 State-Space Models

The SSM, derived from control theory, provides a flexible framework for modeling dynamic systems. By incorporating state and observation equations, the State-Space Model (SSM) effectively captures both linear and nonlinear behaviors, as well as stationary and non-stationary time series. The Kalman Filter (KF), a core SSM algorithm, widely used in structural and nonlinear time series modeling [111]. Moreover, the Hidden Markov Model (HMM), a special case of SSM with discrete hidden states, is particularly well-suited for sequential data, offering a powerful tool for modeling systems where the underlying states evolve in a probabilistic manner. To address non-stationarity and nonlinearity in financial time series, an extended HMM integrates exponentially weighted and dual-weighted Expectation-Maximization (EM) algorithms, enhancing complex financial dynamics modeling and advancing high-dimensional nonlinear time series analysis [112].

5.4 Gaussian Mixture Models

The GMM assumes that data distribution can be represented as a linear combination of multiple Gaussian components and employs the Expectation-Maximization (EM) algorithm for parameter estimation. Due to its flexibility in modeling complex distributions and capturing multimodal characteristics, GMM has been widely applied in density estimation, clustering, and anomaly detection. In time series forecasting, GMM is particularly effective in modeling non-stationary characteristics. the GMM estimator [113] proposed a Generalized Method of Moments GMM-based approach to estimate MSM parameters. This method effectively captures structural changes in time series and improves volatility forecasting performance.

6 DATA-DRIVEN APPROACHES

In this section, we mainly cover the progress in data-driven time series forecasting approaches, which learn the temporal patterns directly from the time series data. We include methods ranging from traditional machine learning methods to recent advancements in deep neural networks, with an emphasis on the latest generative models.

6.1 Machine Learning Approaches

Machine learning methods are important tools for time series forecasting, offering advantages for capturing complex patterns. In this section, we will discuss three prominent approaches: Support Vector Regression (SVR), Regression Trees, and K-Nearest Neighbor (KNN) regression. Each of these approaches provides unique strengths in addressing forecasting challenges.

6.1.1 Support Vector Regression

Support Vector Regression (SVR) [201] is a regression form of Support Vector Machines (SVM) [202] designed to predict continuous values. The objective of SVR is to ensure that most of the data points $(\mathbf{x}_i, y_i)$ satisfy the condition that the prediction $f(\mathbf{x}_i)$ lies within an ϵ range of the actual value y_i. SVR can be solved by transforming the original optimization problem into its dual form with Lagrange multipliers α_i and α_i^* [203]. The final regression function is given by:

$$f(\mathbf{x}) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) K(\mathbf{x}_i, \mathbf{x}) + b. \tag{3}$$

The function $K(\mathbf{x}_i, \mathbf{x})$ is the kernel method. Common kernel functions include Linear kernel, Polynomial kernel, Gaussian (RBF) kernel, and so on.

In time series forecasting, SVR is widely used across various fields [114]. For example, Cao et al. [115] uses SVR to predict the relative price change trends of futures contracts, Lu et al. [116] uses SVR to forecast air quality parameters, Pai et al. [117] proposes a Seasonal Support Vector Regression (SSVR) model to address the challenges of seasonal time series forecasting. Zhang et al [118]. proposed a multiple support vector regression (MSVR) model to reduce error accumulation.

6.1.2 Regression Trees

Regression tree is a regression method based on decision tree. Its core concept is to iteratively partition the feature space into mutually exclusive regions, with each region associated with a predicted value.

The Classification and Regression Trees (CART) algorithm [204] is one of the most widely used methods. CART recursively splits data based on selected features and corresponding split points to minimize the sum of mean squared errors (MSE) for resulting subsets. This process continues until a predefined stopping criterion is met, and the target value is typically predicted by averaging the target values within the leaf nodes. Moreover, to reduce overfitting and improve prediction accuracy, ensemble learning methods like Random Forest [205], Gradient Boosted Decision Trees (GBDT) [206], Extreme Gradient Boosting (XGBoost) [207], LightGBM [208] are often used.

Tree-based methods have played an important role in time series forecasting competitions [119] and many fields such as traffic [120], healthcare [121], energy [122], and so on.

6.1.3 K-Nearest Neighbor Regression

K-Nearest Neighbor (KNN) regression is a non-parametric method and does not require explicit model construction [209]. It works by calculating the distance(e.g., Euclidean distance or Dynamic Time Warping) between the sample to be predicted and each sample in the training set, identifying the k most similar neighbors, and then producing the prediction by taking an average of their target values. Given a sample x, the prediction $\hat{y}$ can be formulated as:

$$\hat{y} = \frac{1}{K} \sum_{i \in N(x)} y_i, \tag{4}$$

where $N(x)$ denotes the K nearest neighbors to the sample x, and y_i are their corresponding target values in the training set. In early research, KNN regression was used to deal with properties like repetitive patterns [123] or seasonality [124]. Researchers have also explored its application in multivariate time series forecasting [125], [126].

6.2 Neural Networks and Deep Learning Models

With the development of deep learning, numerous neural forecasting models have been proposed that achieve superior forecasting accuracy, owing to their powerful capability to capture both temporal and cross-channel dependencies. In this section, we present recent advancements in deep learning-based forecasting approaches, focusing on mainstream architectures, as illustrated in Figure 6.

6.2.1 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) have gained a lot of attention due to their unique advantage in modeling sequential data. The basic recursive structure can be written as follows:

$$\begin{aligned} h_t &= \sigma(U h_{t-1} + V x_t + b_h), \\ \hat{y}_t &= W h_t + b_y. \end{aligned} \tag{5}$$

Where x_t, y_t and h_t are the input, output and hidden state at time step t, respectively. h_{t-1} is the hidden state at the previous time step $t-1$. U, V and W are the weight matrices. b_h and b_y are the bias terms. σ is a nonlinear activation function.

In recent years, many studies have used RNNs as backbone networks for time series forecasting [210]. DeepAR [19], MQRNN [211] and DF-Model [130] are probabilistic forecasting models designed for uncertainty quantification. DeepAR generates the probability distribution for future time steps by jointly learning historical patterns and seasonal features across multiple sequences. MQRNN employs an Encoder-Decoder structure, with LSTM as the Encoder and two MLP branches as the Decoder, to simultaneously predict quantiles for multiple future time steps. DF-Model decomposes time series into global and local parts, using RNN to extract complex non-linear patterns globally and capturing individual random effects for each time series locally with probabilistic models like Gaussian processes(GP).

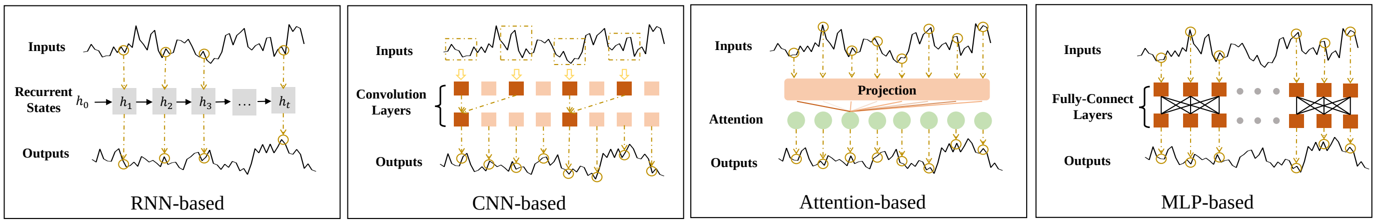

Fig. 6: Illustration of several types of deep neural networks for time series forecasting.

In the point-based forecasting area, DA-RNN [128] and MH-TAL [129] both apply the attention mechanism to enhance context capture and prediction accuracy in tasks. DA-RNN introduces input attention at the encoder stage and temporal attention at the decoder stage, while MH-TAL uses the attention mechanism to establish a connection between future and historical time step features. Besides, the CRU model [25] introduces a new RNN variant for modeling irregular time series, and SegRNN [132] proposes a GRU-based model that employs channel-independent strategy and replaces the original time point-wise iterations with sequence segment-wise iterations. Additionally, hybrid methods have also been studied, LSTNet [127] combines CNN and RNN architectures to capture both short-term and long-term dependencies, with an autoregressive component enhancing stability in multi-step forecasting. Its recurrent-skip component helps alleviate gradient vanishing in long-sequence modeling. ESLSTM [131] combines the exponential smoothing model with LSTM to create a hybrid hierarchical forecasting model.

In summary, RNNs have several strengths, including the ability to process time series of arbitrary length and model temporal dependencies. But they also have certain limitations [212], [213], such as the difficulty in capturing long-term dependencies due to vanishing or exploding gradients and challenges with parallelization due to their sequential nature. Recent advancements, such as RWKV [214] and xLSTM [215], have provided new insights into RNNs. The integration of these advancements into time series forecasting is still an open question and requires further exploration.

6.2.2 Convolutional Neural Networks

Convolutional Neural Network (CNN) is a prevalent architecture in deep learning, demonstrating exceptional performance in fields such as image processing, video analysis, and natural language processing. In the field of time series analysis, the convolution operation, a fundamental operation in the CNN, can be defined as follows:

$$Y(t) = \sum_{c=1}^{C} \sum_{k=0}^{K-1} X_c(t+k) \cdot h_c(k), \tag{6}$$

where X be the input time series sequence of length L with C channels, and h be the convolutional kernel of size K. Y is the output of the convolution operation at time t for a single output channel.

Convolutional Neural Networks (CNNs) exhibit three key characteristics when analyzing time series data: local connectivity, weight sharing, and translation invariance [216]. Local connectivity ensures that each neuron focuses on a local segment of the input data, weight sharing uses the same weights across different segments of the input, and translation invariance allows the network to recognize features learned at any position in the input. These characteristics are crucial for capturing local patterns in time series that may shift over time, while significantly reducing the number of parameters and enhancing the network's ability to efficiently analyze time series data. Given CNNs' strong performance in feature extraction and computational efficiency, researchers suggest considering convolutional networks as one of the primary candidate models for modeling sequence data [217].

The initial convolutional-based neural network designed specifically for time series data was Temporal Convolutional Network(TCN) [217]. TCN employs dilated convolutions, allowing it to achieve a wider receptive field with the same number of model layers. Subsequently, researchers have continued to explore the potential of convolutional neural networks in time series analysis. MLCNN [133] utilizes a multi-layer CNN (more than 10 layers) to learn deep abstract features of time series, while DSANet [218] and MICN [134] employ convolutional kernels of two different scales, extracting both local and global features of time series simultaneously. Unlike the traditional organization of convolutional kernels, SCINet [135] uses sequence downsampling segmentation and organizes the convolutional networks according to binary trees, alleviating the issues of limited receptive fields in lower layers of TCN and the inability of a single convolutional filter to capture complex features. To fully leverage the frequency characteristics of time series, FTMixer [136] employs convolutional neural networks to extract both frequency domain and time domain features of time series simultaneously.

As the exploration in the field of time series continue to advance, researchers have started to focus on the study of generic frameworks for time series analysis. CNNs, with their powerful data understanding capabilities and high computational efficiency, have been widely used in the design of generic frameworks for time series. TimesNet [96] folds the time series according to its primary cycles, treats the folded time series as images, and uses 2D convolution to extract abstract features. ModernTCN [75] employs DepthWise Separable convolutions instead of traditional convolutions, achieving high effectiveness and computational efficiency of time series analysis. It also introduces the concept of reparameterization, which stabilizes the learning of large convolutional kernels. At the same time, ConvTimeNet [78] adopts a multi-scale deep convolutional neural network with large kernels to simultaneously learn global representations and deep representations. TS2Vec [137], which utilizes the TCN architecture to extract features, employs contrastive learning for the feature views at each layer, providing strong contextual support for each timestamp.

In summary, convolutional neural networks excel in computational efficiency and performance in time series forecasting. However, due to their parameter sharing and local perception characteristics, these networks relatively struggle to capture dependencies between different time points in long sequences.

6.2.3 Attention-based Neural Networks

Transformer [219] is one of the most successful architectures in the era of deep learning, which has brought significant advancements in various research areas. The most outstanding design of the Transformer is its attention mechanism, which is expressed as:

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{Softmax}(\frac{\mathbf{Q}\mathbf{K}^{\mathbf{T}}}{\sqrt{d}})\mathbf{V}, \tag{7}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are the query, key, and value vectors with dimension d respectively. Based on this, the Transformer is stacked by multiple blocks which function as:

$$\begin{aligned} \mathbf{H}_l^{'} &= \mathrm{LayerNorm}(\mathrm{SelfAttn}(\mathbf{H}_l) + \mathbf{H}_l), \\ \mathbf{H}_{l+1} &= \mathrm{LayerNorm}(\mathrm{FFN}(\mathbf{H}_l^{'}) + \mathbf{H}_l^{'}). \end{aligned} \tag{8}$$

Here $\mathrm{SelfAttn}$ is a special form of the attention mechanism where all three vectors are derived from the same input. Besides, $\mathbf{H}_l$ is the input of $l-$th block, FFN is the feed-forward network made up of multi-layer perception, and $\mathrm{LayerNorm}$ refers to the layer normalization operation. Through the iterative modeling approach, the Transformer has shown great modeling ability for long-range dependencies, and recent works have proposed many variants of Transformers tailored for time series forecasting tasks [13].

Early Transformer-based forecasting models generally adopt the traditional method of projecting multi-channel data at a single time step into a hidden state [30]. However, due to the inherent noise in time series data, a single time step lacks semantic meaning comparable to that of a word in a sentence. To address this limitation, PatchTST [73] introduces a patching technique that enhances locality and captures richer semantic information, a method now widely adopted by recent studies. Additionally, iTransformer [140] refines this approach by projecting individual time points into variate-level tokens, enabling a more effective representation of multivariate correlations.

On the other hand for the core attention mechanism, the original attention module operates on each time step, exhibiting quadratic time and memory complexity of $O(L^2)$, where L is the sequence length. Given that time series data are often formed in long sequences, the conventional attention mechanism may lead to a significant computational burden and be susceptible to disturbances from noise or outliers. To this end, LogTrans [138] suggests a sparse convolutional self-attention mechanism that generates queries and keys using causal convolution for lowering computational complexity. Besides, Autoformer [42] has developed the Auto-Correlation mechanism, discovers and represents dependencies at the sub-series level through the use of Fast Fourier Transform (FFT). The aforementioned representative methods have achieved a computational complexity of $O(L \log L)$, while some more recent methods have pushed the complexity to $O(L)$: Pyraformer [220] introduces the pyramidal attention module (PAM) in which the inter-scale tree structure summarizes features at different resolutions and the intra-scale neighboring connections model the temporal dependencies of different ranges. FEDformer [139] proposes frequency-enhanced blocks to capture important structures in time series through frequency domain mapping, which also achieves linear complexity By randomly selecting a fixed number of Fourier components. Moreover from the non-stationarity perspective, the Non-stationary Transformer [221] exploits the De-stationary Attention mechanism for boosting the forecasting performance of the mainstream transformers.

In terms of the overall structure, early research on Transformer variants for forecasting predominantly adopted the traditional encoder-decoder structure [30], [32], [42], [139], [221], [222]. The encoders process the long horizon series into hidden states, which are later decoded by the decoders to generate future forecasting results through one forward procedure. Later works point out that complex decoders are not necessary and adopt the encoder-only structure by replacing the decoder part with a linear prediction layer. The experimental results show that encoder-only transformers achieve more accurate forecasting results on the benchmark datasets [45], [73], [140], [223], [224]. Moreover, recent studies have focused on training a time series foundation model, the auto-regressive decoder-only transformer structure is widely utilized due to its ability to process and generate arbitrary lengths of series [48], [49], [153].

In summary, the Transformer architecture has been extensively studied for forecasting tasks due to its ability to effectively capture long-term dependencies and its scalability as a foundation model. However, when applied to small-scale domain-specific time series data, the substantial data requirements for training Transformers may result in overfitting issues.

6.2.4 Multi-layer Perceptrons

Time series forecasting task basically regresses future values based on the observation within a lookback window. The Multi-layer Perceptrons (MLP) based approaches assume that the major correlation exists in a linear form which linear models with high computational efficiency and interpretability can well capture.

From the model's architecture perspective, N-BEATS [141] is a groundbreaking work that constructs a deep neural architecture based on backward and forward residual links and an exceptionally deep stack of fully connected layers. Building on this stacking approach, Nhits [44] integrates innovative hierarchical interpolation and multi-rate data sampling techniques to reduce computational complexity and achieve forecast volatility, and Koopa [142] disentangles time-variant and time-invariant components from intricate non-stationary series by Fourier Filter and designs MLP-based Koopman Predictor to advance respective dynamics forward. Additionally, TimeMixer [145] notes that time series display unique patterns at various sampling scales. Consequently, it constructs a fully MLP-based architecture to fully exploit the disentangled multiscale series during both past extraction and future prediction phases.

Besides, recent work further leverages the advantages of MLP-based models from the perspective of reducing the number of model parameters for better computational efficiency. Initially, DLinear [33] first questions the necessity of complex deep neural networks for time series forecasting tasks, and proposes that the combination of simplest linear transformation with decomposition technique may achieve superior performance on the benchmark datasets. FITS [143] offers a pioneering approach to time series analysis by employing a complex-valued neural network to capture both amplitude and phase information simultaneously, further reducing the parameters to about 10k. Moreover, SparseTSF [144] proposes the Cross-Period Sparse Forecasting technique, simplifying the forecasting task by decoupling the periodicity and trend in time series data and utilizing only 1k parameters to conduct accurate forecasting.

In summary, MLP-based forecasting models have demonstrated their effectiveness on benchmark datasets, offering advantages in interpretability and computational efficiency. However, due to their inherently simple structure, their scalability to large-scale and complex scenarios requires further investigation.

6.3 Emerging Network Architectures

6.3.1 Deep State Space Models

State Space Models (SSMs) are a class of mathematical models that represent dynamic systems using a set of linear or nonlinear equations that describe the system's state evolution over time, which have emerged as promising alternatives for sequence modeling paradigms. Rangapuram et al. [146] originally proposes the deep state space model for probabilistic time series forecasting by parametrizing a per-time-series linear state space model with a jointly-learned recurrent neural network, combing the data efficiency and interpretability of SSMs with strong representation learning capability of deep learning. SpaceTime [147] further introduces a new SSM parameterization that is more suitable for autoregressive time series processes and performs better. Moreover, with the proposal of the advanced SSM structure, Mamba [225], many works have attempted to apply it to the field of time series forecasting and have achieved satisfactory results [148], [149].

6.3.2 Kolmogorov-Arnold Networks

Kolmogorov-Arnold Networks (KAN) [150] is a recently proposed architecture, which is believed to have the potential to serve as an alternative to MLPs whose activation functions are applied on the connections between nodes. Some pioneer works have been devoted to validating its effectiveness in many time series tasks including general time series analysis [151], forecasting [152], and so on.

6.4 Deep Time Series Generative Models

Generative models have emerged as powerful tools in time series forecasting due to their ability to capture the underlying data distribution and generate diverse samples. Discriminative models focus on estimating the conditional

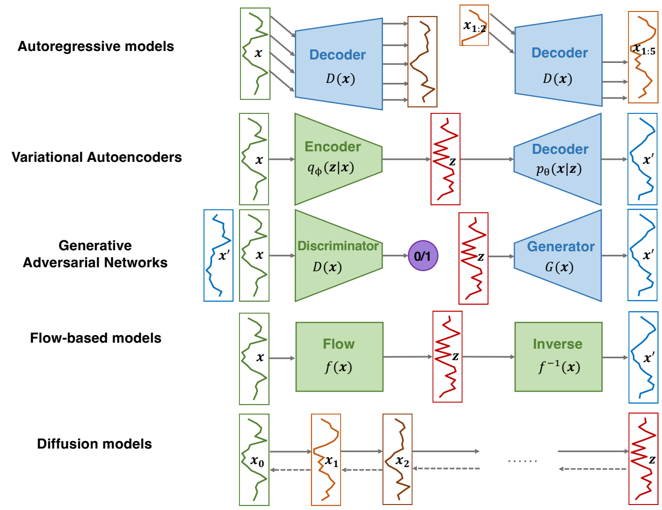

Fig. 7: Illustration of generative model.

probability $P(Y|X)$ to predict an output Y given an input X, whereas generative models aim to learn the joint probability distribution $P(X, Y)$. These two types of models differ in their objectives and learning approaches. This enables generative models not only to predict outcomes but also to synthesize new data that align with the observed distribution. For better illustration, we present a schematic of common generative models in Figure 7.

6.4.1 Autoregressive Models

Autoregression is one of the fundamental concepts in both language modeling and time series forecasting. However, due to the non-stationary nature of time series data and the high demand for precision [12], autoregressive methods often face challenges such as inherent error accumulation. As a result, autoregressive models have not been fully explored in time series modeling. The most well-known model, ARIMA [27], combines the autoregressive (AR) and moving average (MA) models by introducing differencing. Recently, with the rise of pre-trained large language models [226], the potential of autoregressive models in time series data has begun to be explored [91], [153], leveraging their generalization and task versatility. These models have even been adapted as autoregressive predictors [154] to achieve arbitrary input-output mappings.

6.4.2 Variational Autoencoders (VAEs)

VAEs are probabilistic generative models that encode input data into a latent space and then decode it to reconstruct the original data [227]. This structure allows VAEs to model complex data distributions and generate new samples by sampling from the latent space. VAEs are particularly advantageous for handling uncertainty and learning structured latent representations of the data, making them well-suited for time series tasks.

One notable work in this area is TimeVAE [20]. TimeVAE introduces a novel VAE-based architecture tailored for multivariate time series generation. It emphasizes interpretability, allows for encoding domain-specific knowledge, and significantly reduces training time. Through experiments on multiple datasets, TimeVAE demonstrates its ability to accurately represent temporal attributes, performing well in both similarity and next-step prediction tasks. Furthermore, it can integrate domain-specific patterns, such as polynomial trends and seasonalities, to generate interpretable outputs. This feature is especially beneficial for applications requiring transparency.

Another important contribution is HyVAE [155]. HyVAE combines the strengths of VAEs with diffusion models by employing a hybrid variational inference approach. This integration enhances the model's ability to capture temporal dependencies and uncertainty, leading to improved performance in time series forecasting tasks.

6.4.3 Generative Adversarial Networks (GANs)

GANs consist of two components: a generator, which learns to produce realistic samples, and a discriminator, which distinguishes between real and generated data [228]. This adversarial training framework enables GANs to generate highly realistic data, making them powerful tools for data synthesis. In the context of time series, GANs must address the challenge of modeling sequential dependencies.

A seminal work in this field is TimeGAN [156]. TimeGAN adapts the GAN framework specifically for time series data by incorporating a recurrent architecture into both the generator and the discriminator. This allows the model to capture temporal dependencies while maintaining statistical consistency with the original data. TimeGAN has been shown to effectively generate realistic time series data, which can be used for tasks such as simulation, data augmentation, and anomaly detection. Recent studies have extended this framework to specific domains. For example, the Context-aware Traffic Flow Forecasting in New Roads [157] presents a GAN-based model that predicts traffic flow on new roads by considering contextual factors like weather and day type, demonstrating effectiveness in scenarios with limited data. Another approach, Curb-GAN [158], addresses the urban traffic estimation problem by using a conditional GAN with dynamic convolutional layers and self-attention mechanisms to capture both spatial and temporal dependencies, providing accurate estimations even under unprecedented travel demand patterns.

6.4.4 Flow-based Models

Normalizing Flows-based models transform a simple base distribution into a more complex target distribution through a series of invertible and differentiable transformations [229]. This method facilitates exact likelihood estimation, making it particularly effective for modeling high-dimensional data with complex dependencies. In the context of time series, MAF [21] utilizes normalizing flows to model multivariate time series by conditioning on past observations, capturing intricate temporal dependencies.

Recent developments have further extended this technique in various ways. For instance, Conditional Flow Matching for Time Series (CFMTS) [159] improves the training of neural ODEs by regressing vector fields of conditional probability paths, outperforming traditional methods on long trajectory tasks. Trajectory Flow Matching (TFM) [160] introduces a simulation-free training method for Neural SDEs, enhancing stability and scalability, with promising results on clinical time series. Additionally, FM-TS [161] simplifies time series generation through a Flow Matching-based framework, offering efficient training and inference while outperforming diffusion models in both unconditional and conditional time series generation.

6.4.5 Diffusion Models

In time series forecasting, Diffusion Denoising Probabilistic Models (DDPMs) have gained prominence for capturing complex temporal patterns, achieving high predictive performance, and generating realistic data samples. DDPMs operate via a diffusion-denoising process: noise is incrementally added during forward diffusion, and subsequently removed in reverse diffusion to recover the original data. This allows the model to learn complex data distributions and produce high-quality forecasts.

Early models like TimeGrad [162] introduced autoregressive denoising with Langevin sampling. TSDiff [163] improved short-term accuracy and data generation through self-guidance, while ScoreGrad [164] utilized stochastic differential equations (SDEs) for continuous-time forecasting, addressing irregularly sampled data. Conditional diffusion models, such as TimeDiff [165] and DiffLoad [230], incorporated external information to improve accuracy, with applications ranging from power load forecasting to sparse ICU and ECG data [231], [232].

Recent advancements include Latent Diffusion Models (LDMs), with examples such as Latent Diffusion Transformers (LDT) [233], which have demonstrated notable improvements in both precipitation forecasting and scalability. Innovations such as DSPD and CSPD [53] extend diffusion to function space for anomaly detection and interpolation. Models like FDF [166] address challenges in trend modeling by integrating linear layers and conditional modules to capture trends and seasonal components, improving long-term forecasting.

Diffusion models have also been applied to diverse domains, including flood forecasting [234], stock market prediction [235], and electric vehicle load forecasting [236]. While they show strong potential, limitations remain in effectively modeling trends and long-term dependencies. Future advancements in denoising techniques and trend separation methods, combined with diffusion models' high parallelization and cross-domain applicability, offer promising opportunities for further enhancement and deployment across various fields.

7 Transfer Learning Methods

In this section, we introduce the transfer learning techniques for time series forecasting, including self-supervised pre-training, domain adaptation, and LLM-based methods.

7.1 Self-supervised Pre-training Methods

Self-supervised learning alleviates the dependence on large labeled datasets by enabling models to learn transferable representations through pre-training on unlabeled data. Prominent self-supervised pre-training techniques include contrastive learning, denoising masked autoencoders, and autoregressive pre-training models. The proposed taxonomy is illustrated in Figure 8, and the related works can be found in Table 4.

TABLE 4: Systematic elucidation and analysis of representative work categories.

Optimization	Model Name	Space	Cross-domain	GitHub Link Page	Year
Contrastive Learning	TNC [167]	Raw Space	No	https://github.com/sanatonek/TNC_representation_learning	2021
	TS-TCC [168]	Raw Space	No	https://github.com/emadeldeen24/TS-TCC	2021
	TS2Vec [137]	Raw Space	No	https://github.com/yuezhihan/ts2vec	2022
	TF-C [170]	Raw Space	No	https://github.com/mims-harvard/TFC-pretraining	2021
Denosiong Masking	TST	Raw Space	No	https://github.com/gzerveas/mvts_transformer	2022
	PatchTST [73]	Raw Space	No	https://github.com/yuqinie98/PatchTST	2023
	TimeMAE [171]	Discrete Space	No	https://github.com/Mingyue-Cheng/TimeMAE	2024
	CrossTimeNet [86]	Discrete Space	Yes	https://github.com/Mingyue-Cheng/CrossTimeNet	2024
	SimMTM [172]	Manifold space	No	https://github.com/thuml/SimMTM	2024
Auto-regressive	TimeDART [173]	Raw Space	No	https://github.com/melmaphother/timedart	2024
	GPHT [48]	Raw Space	Yes	https://github.com/icantnamemyself/GPHT	2024
	Timer [153]	Raw Space	Yes	https://github.com/thuml/Large-Time-Series-Model	2024

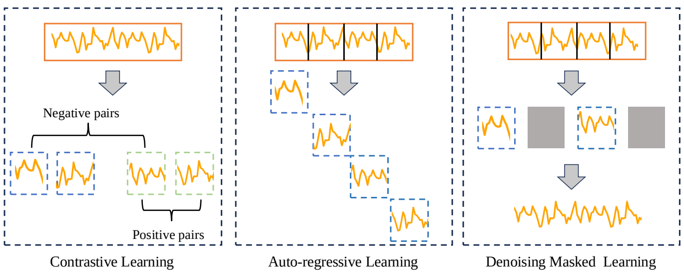

Fig. 8: Illustration of different pre-training learning.

7.1.1 Contrastive Learning Model

The fundamental principle of contrastive learning is to allow the model to differentiate between positive and negative samples. This is achieved by drawing positive samples closer together and pushing negative samples further apart, thereby enabling the model to develop a highly discriminative representation. For time series data, contrastive learning frequently employs data augmentation techniques such as jittering, temporal cropping, and masking to generate varied perspectives of the samples, thereby enhancing the model's generalization across diverse scenarios. TS2Vec [137] serves as a foundational contrastive learning framework for time series, introducing a universal approach to time series representation. Building upon this groundwork, TNC [167] and TS-TCC [168] further advance unsupervised representation techniques, refining the model's capacity for feature extraction in time series data. TF-C [169] extends this framework by incorporating frequency domain information and employing contrastive learning to enforce consistency between time and frequency domains, thereby enabling the model to capture stable and generalizable representations across diverse time series datasets.

7.1.2 Denoising Masked Autoencoder Model

The masked autoencoder model strategically masks portions of the input data, allowing the model to learn robust data representations by reconstructing the masked segments. In the raw data space, TST [170] establishes the masked autoencoder framework, while PatchTST [73] further advances it by introducing channel independence and partitioning the time series into patches, thereby enhancing model flexibility in time series representation. Within the discrete latent space, TimeMAE [171] and CrossTimeNet [86] project time series into a discrete representation space, unifying the input format and facilitating cross-domain self-supervised pre-training. In the manifold space, SimMTM [172] reconstructs masked time points by weighted aggregation of adjacent time slices.

7.1.3 Auto-regressive Pre-training Model

Auto-regressive models capture time-dependent characteristics by leveraging historical data to predict values at the future time point. In time series data, auto-regressive pre-training models progressively learn sequence dynamics and temporal dependencies by modeling historical patterns. TimeDART [173] uses time series blocks as fundamental modeling units, employing a self-attention-based Transformer encoder to establish inter-block dependencies and incorporating diffusion and denoising mechanisms to capture fine-grained features within blocks, effectively applying diffusion models to time series data. GPHT [48] constructs a hybrid dataset for pre-training under the channel independence assumption, significantly expanding the data scale. Additionally, it employs an autoregressive prediction method to model the temporal dependencies of the output sequence without a custom prediction head, allowing for adaptation to various prediction lengths, and providing new insights for self-supervised pre-training. Timer [153] demonstrates strong scalability and generalization by unifying heterogeneous time series into a single sequence format (S3) and utilizing a GPT-style decoder structure.

7.2 Domain Adapation

Distribution shift is a common issue in time series data, leading to the out-of-distribution (OOD) challenge that hampers the performance of forecasting models. Moreover, time series data from different domains often exhibit distinct mapping relationships, further complicating the development of transferable forecasting models. Therefore to achieve transferable time series forecasting, the primary challenge lies in explicitly implementing domain adaptation.

SLARDA [237] introduces a self-supervised pretraining approach for the source domain, utilizing contrastive predictive loss to improve representation learning and enhance the transferability of learned features. This method also accounts for the temporal dynamics of data during both feature learning and domain alignment. Besides, DAF [174] employs an attention-based shared module paired with a domain discriminator to learn domain-invariant latent features, leveraging statistical strengths from the source domain to enhance target domain performance. STONE [175] focuses on learning invariant node dependencies for OOD spatial-temporal data, addressing both structural and temporal shifts. Moreover, FOIL [176] introduces an innovative surrogate loss function designed to alleviate the influence of unobserved variables, coupled with a joint optimization strategy. This facilitates the acquisition of invariant representations across the inferred environments, thereby enhancing the forecasting accuracy for OOD generalized time series data.

7.3 LM-based Models

The LM-based time series predictor is an innovative approach that utilizes advanced language models to forecast future values. These language models, pretrained on extensive textual data, possess the ability to capture rich semantic information and robust reasoning capabilities [238], [239]. This enables them to effectively analyze and reason about time series data, thereby enhancing their understanding and predictive abilities regarding time series patterns and trends [182]. Moreover, compared to traditional time series predictor, language models can further improve prediction accuracy by integrating relevant textual information such as individual characteristics and sampling backgrounds with corresponding numerical features through specific prompts or quantification processes [178], [180], [240]. We categorize the LM-based time series predictors into two types: tuning-based and tuning-free.

7.3.1 Tuning-Based

The tuning-based approach involves making precise adjustments to the backbone parameters to adapt to specific time series data. This usually employs pre-trained language models and involves additional training on specific time series data to adjust the model weights. The fine-tuning process helps the model to more accurately understand and predict specific patterns and trends within the time series. To better achieve this goal, researchers have explored multiple tuning perspectives. First of all, from the forecasting paradigm, the AutoTimes [154] utilizes an autoregressive generation approach, which is particularly suited for simulating left-to-right sequential relationships and incrementally constructing the target sequence [153], [173]. To avoid the potential error accumulation associated with autoregression, other models such as FPT [177] opt for a One-step Prediction approach, generating the entire forecast target sequence in one go. Besides, for the training paradigm, models like CrossTimeNet [86] introduce a pre-training phase of self-supervised representation learning, which deepen the model's perception and understanding of time series data through extensive representation learning. Additionally, models such as TEMPO [180] incorporate the input of related textual information, enabling the effective use of text features to assist in predictions. Moreover, regarding base model parameter updates, some models like LLM4TS [179] employ a LoRA [241] fine-tuning strategy to make low-rank adjustments to critical components, while other models [86], [177] may directly update certain key components or the entire base parameters. Finally, for the processing of the input layer, models like Chronus [22] and UniTime [178] might directly input time series data processed through embedding into the model, whereas models like TimeLLM [182] adopt a feature fusion approach, reprogramming temporal features with textual information to enhance the informativeness of the input data. The combined application of these diverse strategies and methods enables the models to exhibit exceptional performance in handling time series forecasting tasks. To better demonstrate the above content, we select several representative models and present their tuning perspectives in Table 5.

7.3.2 Tuning-Free

The tuning-free approach utilizes pre-trained language models for direct time series forecasting without the need for additional fine-tuning. This method offers the advantage of rapid deployment, significantly reducing computational costs and time. By leveraging the general linguistic features already learned by the model, effective predictions can be made across various types of time series data. In this context, LLMTime [181] incorporates the sampling background of instance as textual information, constructing it along with the input sequence into a prompt. Going further, LSTPrompt [242] introduces a TimeBreak design that simulates human thinking scenarios, allowing the model to take a break after every k predictions. When designing prompts for predicting patient survival probabilities [243], Zhu et al. make full use of the patients' Electronic Health Records (EHR) and adopt a context learning strategy consistent with the clinical environment. Besides, TimeRAF [244] adopts the RAG concept, using a K-nearest neighbors approach in the database to retrieve the k closest time series samples and constructing them into prompts for reference by the language model. Similarly, TimeRAG [245] integrates multiple retrieved neighboring samples with the original text into JSON-formatted strings, and then feed it into the language model. Meanwhile, TableTime [246] validates the capabilities of the large language model, Llama-3.1-405b-instruct, in understanding and classifying time series. In agent perspective, TESSA [247] designs a time series annotation scheme that leverages both general and specific domain annotation agents to generate corresponding textual annotations for time series, significantly enhancing the understanding and reasoning capabilities of language models (such as GPT-4o) regarding time series data. It is noteworthy that A. Merrill et al. [248] have found that tuning-free forecasting models perform only marginally better than random guessing in time series reasoning tasks, and the introduction of related context also offers only modest improvements in predictive ability. These weaknesses indicate that time series reasoning is an influential yet severely underdeveloped direction in tuning-free LM-based time series predictor.

7.4 Time Series Foundation Models

Recent years have witnessed the emergence of time series foundation models pretrained on large-scale temporal data, which drive innovations in time series forecasting tasks through their cross-domain transferable representations. Existing architectures of time series foundation models gener-

TABLE 5: The summary of tuning perspectives from several representative methods, corresponding to the definitions in section 7.3. In the column of "Parameter Updating", $\mathcal{D}$ and $\mathcal{L}$ represent "Directly Updating Parameter" and "Lora Fine-tuning" respectively, with the specific tuning components indicated in parentheses.

Model	Forecasting Paradigm	Training Paradigm	Textual Information	Parameter Updating	Input with Reprogramming	Backbone
AutoTimes	auto-regressive	w/o pretrain	✓	$\mathcal{D}$(projection)	✓	LLaMA-7B
FPT	one-step	w/o pretrain	✗	$\mathcal{D}$(add&norm)	✗	GPT-2
TimeLLM	one-step	w/o pretrain	✓	$\mathcal{D}$(projection)	✓	LLaMA-7B
CrossTimeNet	one-step	w/ pretrain	✓	$\mathcal{D}$(full-tuning)	✗	BERT
LLM4TS	one-step	w/ pretrain	✗	$\mathcal{L}$(attention Q&K)	✗	GPT-2
Chronus	auto-regressive	w/o pretrain	✗	$\mathcal{D}$(full-tuning)	✗	T5
UniTime	one-step	w/o pretrain	✓	$\mathcal{D}$(full-tuning)	✗	GPT-2
TEMPO	one-step	w/o pretrain	✓	$\mathcal{D}$(full-tuning)	✗	GPT-2

ally adopt two paradigms: encoder-based structures that extract universal temporal features, and decoder-based structures that focus on autoregressive generation capabilities.

The encoder-based approaches include MOMENT [184], which integrates multi-domain data from transportation and healthcare to establish a multi-task pretraining system supporting classification, forecasting, and anomaly detection. Following similar principles, Moirai [49] introduces attention mechanisms for arbitrary variates and achieves generalizable forecasting through masked pretraining on the LOTSA dataset, demonstrating effectiveness in out-of-distribution scenarios like electricity load forecasting. In contrast, decoder-based architectures exhibit distinct design philosophies: TimesFM [185] combines patch techniques with decoder structures and frequency-specific tokenization to enable zero-shot generalization, while Chronos [22] discretizes continuous time series into token buckets and leverages synthetic data with cross-entropy training for few-shot forecasting. Recently, Timer [153], unifies diverse downstream tasks through decoder-only architectures and autoregressive generative training strategies.

Despite architectural variations, these models share fundamental principles of simplicity and generality, avoiding over-specialized designs. Current foundation models focus primarily on constructing large-scale temporal datasets, developing cross-domain modeling techniques, and establishing unified frameworks for multivariate analysis. Notably, the data scale surpasses previous end-to-end methods by orders of magnitude (e.g., 27B time steps in LOTSA). Furthermore, synthetic time series data is emerging as a critical pretraining resource, with synthetic data constituting 20% of TimesFM's pretraining corpus. These developments signal a paradigm shift toward data-centric methodologies in both foundation model development and downstream forecasting tasks.

8 Trustworthy Time Series Forecasting

Recently, the demand for reliable and trustworthy time series forecasting has grown exponentially, driven by its wide applications across various domains, including finance, healthcare, and energy management. As these forecasting models become increasingly integrated into critical decision-making processes, ensuring their trustworthiness becomes paramount. Next, we primarily discuss the research advancements in trustworthy time series forecasting, focusing on interpretability, robustness, and privacy preserving.

8.1 Interpretability

Trustworthy time series forecasting involves not only delivering accurate predictions but also addressing key concerns such as model interpretability, which enables users to understand and trust the reasoning behind predictions. To achieve interpretability in forecasting models, mainstream research primarily focuses on two approaches: causal discovery and physics-informed neural networks (PINNs).

Causal discovery enhances time series forecasting by uncovering the underlying cause-and-effect relationships between variables [249]. This approach provides deeper insights into how variables influence one another over time, improving both accuracy and interpretability, particularly in complex systems where predictions rely on the causal structure [250], [251]. Statistical models such as VAR utilize Granger causality to enhance predictions [186], while dynamic Bayesian networks capture temporal dependencies and adapt to changing causal structures [187]. Deep learning models also benefit from integrating causal inference, enhancing their interpretability and ability to manage complex, non-stationary patterns [188].

On the other hand, integrating PINNs into time series forecasting has recently emerged as a promising direction for improving both prediction accuracy and interpretability [189], [190]. PINNs guide data-driven approaches using physical principles, such as conservation laws and differential equations, ensuring predictions are consistent with both data and physical realities—particularly in scenarios with limited or noisy data. By embedding physical knowledge directly into time series models, PINNs ensure that predictions align with physical laws [191], [192], [193]. This not only enhances the learning process but also improves the robustness and reliability of the models, making them better suited for practical applications where physical consistency is crucial.

8.2 Robustness

Despite significant advancements in recent forecasting models, they remain vulnerable to adversarial attacks, which raises important concerns about their trustworthy deployment in critical applications. Understanding the robustness of these models against malicious attacks and developing effective defense mechanisms is crucial. To address this, Dang-Nhu et al. [194] introduce an effective adversarial attack generation method through Monte Carlo estimation of statistics from the joint distribution of the target series, applied to probabilistic autoregressive forecasting. Liu et al. [195] identify a new attack pattern involving strategic, sparse modifications, and propose defense strategies using randomized smoothing techniques and adversarial training. Besides, RDAT [196] employs a reinforcement learning-based adversarial training approach with self-knowledge distillation regularization to enhance the adversarial robustness of spatiotemporal traffic forecasting models. Another notable work, BACKTIME [197], investigates backdoor attacks on time series forecasting tasks. Their findings show that mainstream forecasting models can be significantly degraded by stealthy, sparse, and highly effective GNN-based triggers.

8.3 Privacy

With the increasing demand for time series foundation models, a promising approach is to train using vast amounts of time series data from multiple sources. However, this raises significant concerns regarding data privacy. To address this issue, several efforts have focused on integrating the federated learning paradigm, enabling the use of multi-domain data while preventing the exposure of sensitive information. CNFGNN [198] explicitly encodes the graph structure under the constraints of cross-node federated learning to effectively model complex dependencies in spatio-temporal forecasting, ensuring that data privacy is maintained while still capturing intricate relationships across nodes. MetePFL [199] introduces a federated prompt learning mechanism, combined with a graph-based approach, to mitigate the impact of data heterogeneity while preserving data security. Moreover, Time-FFM [200] proposes a prompt adaption module and a personalized federated training strategy by learning global encoders and local prediction heads to construct a federated foundation model for time series forecasting by leveraging pretrained language models.

9 Benchmark Datasets and Applications

In this section, we introduce some prevalent benchmark datasets and representative time series forecasting methods applied in various domains.

9.1 Overview of Benchmark Datasets

Time series benchmark datasets are foundational to time series forecasting tasks. In table 6, we summarizes 36 publicly available real-world datasets frequently used for evaluating time series forecasting models. The table provides information on domian, channel,frequency, obs(total observations) and source. By utilizing those datasets, we can assess model performance across various scenarios, thereby evaluating its generalization ability and ensuring the robustness and reliability of the model in real-world applications.

9.2 Representative Applications

In this part, we briefly introduce some recent advancements of time series forecastor applied in various domains.

9.2.1 Healthcare

Healthcare is committed to maintaining and improving the health of individuals and groups through prevention, treatment, and rehabilitation. Time series forecasting plays a crucial role in aspects of health such as disease prevention, treatment plan formulation, and rehabilitation progress assessment. For example, forecasting medical bookings improves appointment scheduling and enhances hospital scheduling [271]. In chronic disease management, such as Type 1 diabetes, models like LSTM and TCN predict blood glucose levels to prevent hypo- or hyperglycemic events [272]. Additionally, forecasting intraoperative hypotension (IOH) using real-time biosignals and deep learning increases prediction accuracy [1], [273], [274], [275]. These applications demonstrate the essential role of time series forecasting in healthcare resource allocation and disease management.

9.2.2 Manufacturing

Manufacturing is a key driver of global economic growth, innovation, and job creation. Time series forecasting is essential for optimizing production processes, predicting equipment failures, and enhancing quality control. By analyzing historical data, these models uncover patterns that help manufacturers make informed decisions, boost efficiency, and cut costs. For instance, in aero-engine manufacturing, time-series forecasting with the multi-resolution transformer (MRT) model [3] improves component modeling, fault diagnosis, and performance prediction. Additionally, in semiconductor manufacturing, time-series fault detection and diagnosis with the multiple time-series convolutional neural network (MTS-CNN) model [276] enhances equipment monitoring, fault classification, and cause identification. These applications are crucial for developing smart manufacturing systems, increasing productivity and operational reliability across the sector.

9.2.3 Finance and Economics

The financial sector has been transformed by deep learning techniques for time series forecasting, particularly in stock price prediction and economic indicator forecasting. Advanced transformer-based models like the Market-Guided Stock Transformer [2] accurately predict stock trends, enabling data-driven investment strategies and portfolio management. Additionally, LSTM networks forecast energy prices and commodity trends, aiding financial institutions and investors in resource allocation [277]. These deep learning applications highlight their transformative impact on modern finance and the advancement of intelligent, adaptive financial systems.

9.2.4 Environment

Advancements in deep learning for spatio-temporal forecasting are transforming environmental conservation efforts. Models like DeepSTF [278] combine CNNs and attention mechanisms for accurate, efficient multi-site forecasts, essential for urban management and disaster warnings. Corrformer [279] captures global spatio-temporal patterns with a multi-correlation mechanism, enabling forecasts for thousands of stations. These models enhance renewable energy optimization, environmental monitoring, and smart

TABLE 6: The statistics of evaluation datasets.

Domain	Dataset	Channel	Frequency	Observations	Source
Energy	BDG-2-Panther	105	1 hour	919,800	[252]
Energy	SMART	5	1 hour	95709	[253]
Energy	Low Carbon London	713	1 hour	9,543,348	[254]
Energy	KDD Cup 2022	134	10 minutes	4,727,519	[255]
Energy	Solar	137	10 minutes	7,200,720	[127]
Energy	ETTh	7	1 hour	403,200	[30]
Energy	ETTm	7	1 hour	100,800	[30]
Energy	Electricity	321	1 hour	8,443,584	[42]
Transportation	Uber TLC Daily	262	1 day	47,087	[256]
Transportation	PEMS03	358	5 minutes	9,382,464	[257]
Transportation	Kaggle Web Traffic	4	1 day	145,000	[258]
Transportation	Traffic	862	1 hour	15,122,928	[42]
Transportation	Beijing Subway	276	30 minutes	248,400	[257]
Transportation	Los-Loop	207	5 minutes	7,094,304	[257]
Transportation	SHMetro	288	15 minutes	1,934,208	[257]
Transportation	HZMetro	80	15 minutes	146,000	[257]
Transportation	METR-LA	207	5 minutes	7094304	[259]
Environment	SubseasonalClimateUSA	862	1 day	14,097,148	[260]
Environment	Weather	21	10 minutes	1,106,616	[42]
Environment	AQWan	11	1 hour	385,704	[261]
Environment	AQShunyi	11	1 hour	385,704	[261]
Environment	China Air Quality	437	1 hour	5,739,234	[262]
Environment	Beijing Air Quality	12	1 hour	420,768	[263]
Economic	M1 Monthly	1	1 month	44,892	[264]
Economic	M3 Monthly	1	1 month	141,858	[265]
Economic	M4 Monthly	1	1 month	709522	[266]
Economic	M5	1,947	1 day	58,327,370	[267]
Economic	NN5 Daily	111	1 day	81,585	[268]
Economic	Tourism Yearly	1311	1 month	11,198	[269]
Economic	Exchange	8	1 day	60,704	[42]
Economic	NASDAQ	5	1 day	6,220	[261]
Economic	NYSE	5	1 day	6,215	[261]
Healthcare	CDC Fluview ILINet	75	1 week	63,903	[49]
Healthcare	CDC Fluview WHO NREVSS	74	1 week	41,760	[49]
Healthcare	Project Tycho	1258	1 week	1,377,707	[270]
Healthcare	ILI	7	1 week	6,762	[42]

city development by improving energy scheduling, smart grid management, and air quality tracking, supporting intelligent and sustainable environmental management.

9.2.5 Transportation

Time series forecasting is crucial in transportation for accurate traffic flow predictions and smart system support. These models capture complex spatiotemporal dependencies [100], [280] to optimize route planning, congestion management, and resource allocation. PDFormer [223] enhances prediction accuracy by incorporating propagation delay awareness and modeling long-range dependencies. Besides, RGDAN [281] integrates graph diffusion attention and temporal attention modules to better capture spatial and temporal dependencies. Together, these models advance smart transportation systems by improving route planning, congestion control, and resource allocation.

10 Prospects and Future Direction

In this section, we discuss the future directions of time series forecasting from various perspectives.

10.1 Time Series Foundation Models

Drawing inspiration from the great progress of the foundation models in relevant research areas including computer vision and natural language processing, constructing a foundation model for time series analysis becomes a promising direction, and many recent efforts have been devoted as pioneering explorations. Despite the effectiveness, there remain multiple unresolved issues worth exploring.

On the one hand, constructing a context-aware foundation model for more accurate general time series forecasting is the principal direction. As mentioned in the above sections, existing foundational model approaches often mix massive amounts of time series data from diverse sources to construct pretraining datasets. Through pretraining, these methods aim to enhance the model's generalization ability and investigate the potential emergence within the time series field. However, these approaches often overlook the contextual characteristics of time series data, which can lead to suboptimal forecasting results. Firstly on the data point level, different time series datasets often have varying sampling frequencies, noise levels, and, more critically, distinct and well-defined physical meanings across domains. Fully accounting for these properties is essential to establish a unified semantic representation for the time series domain. Secondly on the data instances level, even within similar domains, time series data can exhibit significant differences due to the individuality of observed subjects. For example, vital sign data from different patient groups or traffic flow data from various regions may display unique patterns. Finally, on the task objective level, different application scenarios impose diverse requirements on the forecasting outcomes. Models must adapt to specific instructions to optimally generate predictions that meet the demands of distinct tasks. Furthermore, to better model the contextual characteristics mentioned above, an urgent research challenge lies in effectively integrating multiple modalities, such as text, images, and graphs, into the forecasting process.

On the other hand, it remains inconclusive whether training a universal foundation model is the optimal solution for time series forecasting. This uncertainty, however, opens the door to several potentially valuable research directions. First, it is worth exploring the operational mechanisms of general foundation models in more depth, such as scaling laws and emergent phenomena. Additionally, investigating the impact of different data ratios on model performance to guide the construction of pretraining datasets is a promising avenue. Lastly, training domain-specific time series forecasting foundation models, which fully integrate the inductive biases and prior knowledge of the respective domain for more precise predictions, is also a direction worth exploring.

10.2 Trustworthy Time Series Forecasting

Most of the current mainstream time series forecasting methods rely on black-box neural network models, which, while offering high accuracy, are difficult to apply in many sensitive real-world applications such as healthcare analysis and decision-making. Therefore, a promising research direction is to leverage advanced techniques such as explainable AI and causal inference, in combination with existing powerful forecasting models, to construct interpretable and trustworthy forecasting systems that can facilitate practical, real-world applications.

Moreover, with the increasing demand for training time series forecasting models using multi-source cross-domain data, ensuring data privacy and model security has become an important research topic. Specifically, multi-source cross-domain training often relies on federated learning frameworks. This raises critical issues such as how to protect the privacy of client-side time series data from leakage using techniques like differential privacy, and how to improve the robustness of models to guard against data poisoning attacks from malicious clients. These are pressing research challenges that need to be addressed.

10.3 Emerging Modeling Paradigms

Time series forecasting has always been an open research field, encouraging the validation of various diverse techniques, including different neural network architectures, modeling perspectives, and model training paradigms. These methods, in turn, provide valuable insights that propel the further development of the entire field. Therefore, combining emerging technologies from the broader machine learning domain with the unique characteristics of time series data is a promising future direction for research. For example, exploring how to integrate automated machine learning methods to adjust network architectures, balance computational costs with forecasting accuracy, and how to incorporate physical knowledge of the time series data through Physics-Informed Neural Networks (PINNs) are potential areas for future investigation.

10.4 Comprehensive Benchmark Evaluation

In addition to model design, model performance evaluation and the design of appropriate benchmarks are also important future research directions in the field of time series forecasting. Existing benchmarks often lack a sufficiently broad range of data distributions and fail to clearly differentiate the difficulty levels of tasks. Moreover, the evaluation metrics used are typically too simplistic, making it difficult to assess the strengths and weaknesses of each model from multiple perspectives. This limitation may even encourage an arms race-style search for hyperparameters of each model for better testing performance. Therefore, there is a need to develop more generalized evaluation datasets and diverse evaluation metrics to ensure the healthy and well-rounded development of the entire research field.

11 Conclusion

In this survey, we have presented a holistic and structured examination of recent advancements in time series forecasting, encompassing foundational concepts, methodological evolutions, and critical challenges. We established a unified perspective that bridges classical statistical modeling and cutting-edge deep learning paradigms, highlighting how shifts in data availability, computational power, and algorithmic sophistication are reshaping the field. Through careful attention to key hurdles—ranging from handling non-stationarity and uncertainty quantification to managing high-dimensionality and interpretability—we have illustrated the complexity and dynamism that define modern time series forecasting tasks. We further surveyed prominent benchmark datasets and evaluation metrics, underscoring the importance of robust and fair performance comparisons to drive meaningful progress. Crucially, we identified emerging trends, such as leveraging generative models, explainable AI techniques, and integrative frameworks that combine domain knowledge with data-driven insights. The work presented here not only consolidates the state of the art but also illuminates avenues for future innovation, offering researchers and practitioners a coherent reference point as they navigate the ever-evolving landscape of time series forecasting research.

REFERENCES

[1] F. Hatib, Z. Jian, S. Buddi, C. Lee, J. Settels, K. Sibert, J. Rinehart, and M. Cannesson, “Machine-learning algorithm to predict hypotension based on high-fidelity arterial pressure waveform analysis,” *Anesthesiology*, vol. 129, no. 4, pp. 663–674, 2018.

[2] T. Li, Z. Liu, Y. Shen, X. Wang, H. Chen, and S. Huang, “Master: Market-guided stock transformer for stock price forecasting,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 1, 2024, pp. 162–170.

[3] H.-J. Jin, Y.-P. Zhao, and M.-N. Pan, “A novel method for aero-engine time-series forecasting based on multi-resolution transformer,” *Expert Systems with Applications*, vol. 255, p. 124597, 2024.

[4] K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian, “Accurate medium-range global weather forecasting with 3d neural networks,” *Nature*, vol. 619, no. 7970, pp. 533–538, 2023.

[5] R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, F. Alet, S. Ravuri, T. Ewalds, Z. Eaton-Rosen, W. Hu *et al.*, “Learning skillful medium-range global weather forecasting,” *Science*, vol. 382, no. 6677, pp. 1416–1421, 2023.

[6] R. J. Hyndman and Y. Khandakar, “Automatic time series forecasting: the forecast package for r,” *Journal of statistical software*, vol. 27, pp. 1–22, 2008.

[7] P. S. Kalekar *et al.*, “Time series forecasting using holt-winters exponential smoothing,” *Kanwal Rekhi school of information Technology*, vol. 4329008, no. 13, pp. 1–13, 2004.

[8] B. Lim and S. Zohren, “Time-series forecasting with deep learning: a survey,” *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, p. 20200209, 2021.

[9] K. Benidis, S. S. Rangapuram, V. Flunkert, Y. Wang, D. Maddix, C. Turkmen, J. Gasthaus, M. Bohlke-Schneider, D. Salinas, L. Stella *et al.*, “Deep learning for time series forecasting: Tutorial and literature survey,” *ACM Computing Surveys*, vol. 55, no. 6, pp. 1–36, 2022.

[10] J. Wang, W. Du, W. Cao, K. Zhang, W. Wang, Y. Liang, and Q. Wen, “Deep learning for multivariate time series imputation: A survey,” *arXiv preprint arXiv:2402.04059*, 2024.

[11] S. Zhou, D. Zha, X. Shen, X. Huang, R. Zhang, and F.-L. Chung, “Denoising-aware contrastive learning for noisy time series,” *arXiv preprint arXiv:2406.04627*, 2024.

[12] 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 *International Conference on Learning Representations*, 2022.

[13] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, “Transformers in time series: a survey,” in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 2023, pp. 6778–6786.

[14] M. Jin, H. Y. Koh, Q. Wen, D. Zambon, C. Alippi, G. I. Webb, I. King, and S. Pan, “A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[15] K. Zhang, Q. Wen, C. Zhang, R. Cai, M. Jin, Y. Liu, J. Y. Zhang, Y. Liang, G. Pang, D. Song *et al.*, “Self-supervised learning for time series analysis: Taxonomy, progress, and prospects,” *IEEE transactions on pattern analysis and machine intelligence*, 2024.

[16] Y. Wang, H. Wu, J. Dong, Y. Liu, M. Long, and J. Wang, “Deep time series models: A comprehensive survey and benchmark,” *arXiv preprint arXiv:2407.13278*, 2024.

[17] J. Su, C. Jiang, X. Jin, Y. Qiao, T. Xiao, H. Ma, R. Wei, Z. Jing, J. Xu, and J. Lin, “Large language models for forecasting and anomaly detection: A systematic literature review,” *arXiv preprint arXiv:2402.10350*, 2024.

[18] 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 *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, 2024, pp. 6555–6565.

[19] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, “Deepar: Probabilistic forecasting with autoregressive recurrent networks,” *International journal of forecasting*, vol. 36, no. 3, pp. 1181–1191, 2020.

[20] A. Desai, C. Freeman, Z. Wang, and I. Beaver, “Timevae: A variational auto-encoder for multivariate time series generation,” *arXiv preprint arXiv:2111.08095*, 2021.

[21] K. Rasul, A.-S. Sheikh, I. Schuster, U. M. Bergmann, and R. Vollgraf, “Multivariate probabilistic time series forecasting via conditioned normalizing flows,” in *International Conference on Learning Representations*, 2021.

[22] A. F. Ansari, L. Stella, C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. P. Arango, S. Kapoor *et al.*, “Chronos: Learning the language of time series,” *arXiv preprint arXiv:2403.07815*, 2024.

[23] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, “Brits: Bidirectional recurrent imputation for time series,” *Advances in neural information processing systems*, vol. 31, 2018.

[24] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, “A review on outlier/anomaly detection in time series data,” *ACM computing surveys (CSUR)*, vol. 54, no. 3, pp. 1–33, 2021.

[25] M. Schirmer, M. Eltayeb, S. Lessmann, and M. Rudolph, “Modeling irregular time series with continuous recurrent units,” in *International conference on machine learning*. PMLR, 2022, pp. 19 388–19 405.

[26] Y. Chen, K. Ren, Y. Wang, Y. Fang, W. Sun, and D. Li, “Contiformer: Continuous-time transformer for irregular time series modeling,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[27] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[28] A. K. Singh, S. K. Ibraheem, M. Muazzam, and D. Chaturvedi, “An overview of electricity demand forecasting techniques,” *Network and complex systems*, vol. 3, no. 3, pp. 38–48, 2013.

[29] S. Kaushik, A. Choudhury, P. K. Sheron, N. Dasgupta, S. Natarajan, L. A. Pickett, and V. Dutt, “Ai in healthcare: time-series forecasting using statistical, neural, and ensemble architectures,” *Frontiers in big data*, vol. 3, p. 4, 2020.

[30] 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 *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[31] L. Zhao and Y. Shen, “Rethinking channel dependence for multivariate time series forecasting: Learning from leading indicators,” in *The Twelfth International Conference on Learning Representations*, 2024.

[32] Y. Zhang and J. Yan, “Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting,” in *The Eleventh International Conference on Learning Representations*, 2023.

[33] A. Zeng, M. Chen, L. Zhang, and Q. Xu, “Are transformers effective for time series forecasting?” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 9, 2023, pp. 11 121–11 128.

[34] L. Han, H.-J. Ye, and D.-C. Zhan, “The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting,” *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[35] K. G. Olivares, C. Challu, G. Marcjasz, R. Weron, and A. Dubrawski, “Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with nbeatsx,” *International Journal of Forecasting*, vol. 39, no. 2, pp. 884–900, 2023.

[36] A. Das, W. Kong, A. Leach, S. K. Mathur, R. Sen, and R. Yu, “Long-term forecasting with tiDE: Time-series dense encoder,” *Transactions on Machine Learning Research*, 2023.

[37] Y. Wang, H. Wu, J. Dong, G. Qin, H. Zhang, Y. Liu, Y. Qiu, J. Wang, and M. Long, “Timexer: Empowering transformers for time series forecasting with exogenous variables,” *arXiv preprint arXiv:2402.19072*, 2024.

[38] Y. Du, J. Wang, W. Feng, S. Pan, T. Qin, R. Xu, and C. Wang, “Adarnn: Adaptive learning and forecasting of time series,” in *Proceedings of the 30th ACM international conference on information & knowledge management*, 2021, pp. 402–411.

[39] Z. Liu, M. Cheng, Z. Li, Z. Huang, Q. Liu, Y. Xie, and E. Chen, “Adaptive normalization for non-stationary time series forecasting: A temporal slice perspective,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[40] A. C. Harvey and S. Peters, “Estimation procedures for structural time series models,” *Journal of forecasting*, vol. 9, no. 2, pp. 89–108, 1990.

[41] R. B. Cleveland, W. S. Cleveland, J. E. McRae, I. Terpenning *et al.*, “Stl: A seasonal-trend decomposition,” *J. off. Stat*, vol. 6, no. 1, pp. 3–73, 1990.

[42] H. Wu, J. Xu, J. Wang, and M. Long, “Autoformer: Decomposition transformers with auto-correlation for long-term series forecast-

ing," *Advances in neural information processing systems*, vol. 34, pp. 22 419–22 430, 2021.

[43] M. C. Mozer, "Induction of multiscale temporal structure," *Advances in neural information processing systems*, vol. 4, 1991.

[44] C. Challu, K. G. Olivares, B. N. Oreshkin, F. G. Ramirez, M. M. Canseco, and A. Dubrawski, "Nhits: Neural hierarchical interpolation for time series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 6, 2023, pp. 6989–6997.

[45] M. A. Shabani, A. H. Abdi, L. Meng, and T. Sylvain, "Scaleformer: Iterative multi-scale refining transformers for time series forecasting," in *The Eleventh International Conference on Learning Representations*, 2023.

[46] M. Hou, C. Xu, Y. Liu, W. Liu, J. Bian, L. Wu, Z. Li, E. Chen, and T.-Y. Liu, "Stock trend prediction with multi-granularity data: A contrastive learning approach with adaptive fusion," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 700–709.

[47] B. Rim, N.-J. Sung, S. Min, and M. Hong, "Deep learning in physiological signal data: A survey," *Sensors*, vol. 20, no. 4, p. 969, 2020.

[48] Z. Liu, J. Yang, M. Cheng, Y. Luo, and Z. Li, "Generative pretrained hierarchical transformer for time series forecasting," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 2003–2013.

[49] G. Woo, C. Liu, A. Kumar, C. Xiong, S. Savarese, and D. Sahoo, "Unified training of universal time series forecasting transformers," in *Forty-first International Conference on Machine Learning*, 2024.

[50] J. Shi, Q. Ma, H. Ma, and L. Li, "Scaling law for time series forecasting," *arXiv preprint arXiv:2405.15124*, 2024.

[51] P. Bansal, P. Deshpande, and S. Sarawagi, "Missing value imputation on multidimensional time series," *Proceedings of the VLDB Endowment*, vol. 14, no. 11, pp. 2533–2545, 2021.

[52] Y. Luo, X. Cai, Y. Zhang, J. Xu *et al.*, "Multivariate time series imputation with generative adversarial networks," *Advances in neural information processing systems*, vol. 31, 2018.

[53] M. Biloš, K. Rasul, A. Schneider, Y. Nevmyvaka, and S. Günnemann, "Modeling temporal data as continuous functions with stochastic process diffusion," in *International Conference on Machine Learning*. PMLR, 2023, pp. 2452–2470.

[54] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences*, vol. 454, no. 1971, pp. 903–995, 1998.

[55] P. Chaovalit, A. Gangopadhyay, G. Karabatis, and Z. Chen, "Discrete wavelet transform-based time series analysis and mining," *ACM Computing Surveys (CSUR)*, vol. 43, no. 2, pp. 1–37, 2011.

[56] T. Yoon, Y. Park, E. K. Ryu, and Y. Wang, "Robust probabilistic time series forecasting," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 1336–1358.

[57] N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Deep adaptive input normalization for time series forecasting," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 9, pp. 3760–3765, 2019.

[58] W. Fan, P. Wang, D. Wang, D. Wang, Y. Zhou, and Y. Fu, "Dish-ts: a general paradigm for alleviating distribution shift in time series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 6, 2023, pp. 7522–7529.

[59] L. Han, H.-J. Ye, and D.-C. Zhan, "SIN: Selective and interpretable normalization for long-term time series forecasting," in *Forty-first International Conference on Machine Learning*, 2024.

[60] S. Lahmiri, "A variational mode decompoisition approach for analysis and forecasting of economic and financial time series," *Expert Systems with Applications*, vol. 55, pp. 268–273, 2016.

[61] E. Ghanbari and A. Avar, "Short-term wind power forecasting using the hybrid model of multivariate variational mode decomposition (mvmd) and long short-term memory (lstm) neural networks," *Electrical Engineering*, pp. 1–31, 2024.

[62] Y. Wang, S. Sun, X. Chen, X. Zeng, Y. Kong, J. Chen, Y. Guo, and T. Wang, "Short-term load forecasting of industrial customers based on svmd and xgboost," *International Journal of Electrical Power & Energy Systems*, vol. 129, p. 106830, 2021.

[63] Y. Li, H. Wu, and H. Liu, "Multi-step wind speed forecasting using ewt decomposition, lstm principal computing, relm subordinate computing and iewt reconstruction," *Energy Conversion and Management*, vol. 167, pp. 203–219, 2018.

[64] J. Xin, C. Zhou, Y. Jiang, Q. Tang, X. Yang, and J. Zhou, "A signal recovery method for bridge monitoring system using tvfemd and encoder-decoder aided lstm," *Measurement*, vol. 214, p. 112797, 2023.

[65] P. Bonizzi, J. M. Karel, O. Meste, and R. L. Peeters, "Singular spectrum decomposition: A new method for time series decomposition," *Advances in Adaptive Data Analysis*, vol. 6, no. 04, p. 1450011, 2014.

[66] L. Karthikeyan and D. N. Kumar, "Predictability of nonstationary time series using wavelet and emd based arma models," *Journal of hydrology*, vol. 502, pp. 103–119, 2013.

[67] N. A. Agana and A. Homaifar, "Emd-based predictive deep belief network for time series prediction: An application to drought forecasting," *Hydrology*, vol. 5, no. 1, p. 18, 2018.

[68] W.-c. Wang, K.-w. Chau, D.-m. Xu, and X.-Y. Chen, "Improving forecasting accuracy of annual runoff time series using arima based on eemd decomposition," *Water Resources Management*, vol. 29, pp. 2655–2675, 2015.

[69] M. Theodosiou, "Forecasting monthly and quarterly time series using stl decomposition," *International Journal of Forecasting*, vol. 27, no. 4, pp. 1178–1195, 2011.

[70] J. Nasir, M. Aamir, Z. U. Haq, S. Khan, M. Y. Amin, and M. Naeem, "A new approach for forecasting crude oil prices based on stochastic and deterministic influences of lmd using arima and lstm models," *IEEE Access*, vol. 11, pp. 14 322–14 339, 2023.

[71] M. Wang, J. Yang, B. Yang, H. Li, T. Gong, B. Yang, and J. Cui, "Towards lightweight time series forecasting: a patch-wise transformer with weak data enriching," *arXiv preprint arXiv:2501.10448*, 2025.

[72] C. Ying and J. Lu, "Tfeformer: Temporal feature enhanced transformer for multivariate time series forecasting," *IEEE Access*, 2024.

[73] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *The Eleventh International Conference on Learning Representations*, 2023.

[74] P. Chen, Y. Zhang, Y. Cheng, Y. Shu, Y. Wang, Q. Wen, B. Yang, and C. Guo, "Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting," *arXiv preprint arXiv:2402.05956*, 2024.

[75] D. Luo and X. Wang, "Moderntcn: A modern pure convolution structure for general time series analysis," in *The Twelfth International Conference on Learning Representations*, 2024.

[76] Z. Gong, Y. Tang, and J. Liang, "Patchmixer: A patch-mixing architecture for long-term time series forecasting," *arXiv preprint arXiv:2310.00655*, 2023.

[77] V. Ekambaram, A. Jati, N. Nguyen, P. Sinthong, and J. Kalagnanam, "Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 459–469.

[78] M. Cheng, J. Yang, T. Pan, Q. Liu, and Z. Li, "Convtimenet: A deep hierarchical fully convolutional model for multivariate time series analysis," *arXiv preprint arXiv:2403.01493*, 2024.

[79] Q. Huang, L. Shen, R. Zhang, J. Cheng, S. Ding, Z. Zhou, and Y. Wang, "Hdmixer: Hierarchical dependency with extendable patch for multivariate time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 11, 2024, pp. 12 608–12 616.

[80] S. Zhong, S. Song, W. Zhuo, G. Li, Y. Liu, and S.-H. G. Chan, "A multi-scale decomposition mlp-mixer for time series analysis," *arXiv preprint arXiv:2310.11959*, 2023.

[81] Y. Zhang, L. Ma, S. Pal, Y. Zhang, and M. Coates, "Multi-resolution time-series transformer for long-term forecasting," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2024, pp. 4222–4230.

[82] K. Rasul, A. Ashok, A. R. Williams, A. Khorasani, G. Adamopoulos, R. Bhagwatkar, M. Biloš, H. Ghonia, N. Hassen, A. Schneider *et al.*, "Lag-llama: Towards foundation models for time series forecasting," in *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023.

[83] J. Xie, W. Mao, Z. Bai, D. J. Zhang, W. Wang, K. Q. Lin, Y. Gu, Z. Chen, Z. Yang, and M. Z. Shou, "Show-o: One single

transformer to unify multimodal understanding and generation," *arXiv preprint arXiv:2408.12528*, 2024.

[84] P. Anastassiou, J. Chen, J. Chen, Y. Chen, Z. Chen, Z. Chen, J. Cong, L. Deng, C. Ding, L. Gao *et al.*, "Seed-tts: A family of high-quality versatile speech generation models," *arXiv preprint arXiv:2406.02430*, 2024.

[85] P. Schäfer and M. Högqvist, "Sfa: a symbolic fourier approximation and index for similarity search in high dimensional datasets," in *Proceedings of the 15th international conference on extending database technology*, 2012, pp. 516–527.

[86] M. Cheng, X. Tao, Q. Liu, H. Zhang, Y. Chen, and C. Lei, "Learning transferable time series classifier with cross-domain pre-training from language model," *arXiv preprint arXiv:2403.12372*, 2024.

[87] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[88] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu *et al.*, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, vol. 12, 2016.

[89] M. Łajszczak, G. Cámbara, Y. Li, F. Beyhan, A. van Korlaar, F. Yang, A. Joly, Á. Martín-Cortinas, A. Abbas, A. Michalski *et al.*, "Base tts: Lessons from building a billion-parameter text-to-speech model on 100k hours of data," *arXiv preprint arXiv:2402.08093*, 2024.

[90] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023.

[91] M. Cheng, Y. Chen, Q. Liu, Z. Liu, Y. Luo, and E. Chen, "Instructime: Advancing time series classification with multimodal language modeling," in *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, 2025, pp. 792–800.

[92] S. Winograd, "On computing the discrete fourier transform," *Mathematics of computation*, vol. 32, no. 141, pp. 175–199, 1978.

[93] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE transactions on Computers*, vol. 100, no. 1, pp. 90–93, 1974.

[94] M. J. Shensa *et al.*, "The discrete wavelet transform: wedding the a trous and mallat algorithms," *IEEE Transactions on signal processing*, vol. 40, no. 10, pp. 2464–2482, 1992.

[95] A. V. Oppenheim, *Discrete-time signal processing*. Pearson Education India, 1999.

[96] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," in *The Eleventh International Conference on Learning Representations*, 2023.

[97] X. Zhu, D. Shen, H. Wang, and Y. Hao, "Fcnet: Fully complex network for time series forecasting," *IEEE Internet of Things Journal*, 2024.

[98] P. Liu, B. Wu, N. Li, T. Dai, F. Lei, J. Bao, Y. Jiang, and S.-T. Xia, "Wftnet: Exploiting global and local periodicity in long-term time series forecasting," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 5960–5964.

[99] A. Ma, D. Luo, and M. Sha, "Mmfnet: Multi-scale frequency masking neural network for multivariate time series forecasting," *arXiv preprint arXiv:2410.02070*, 2024.

[100] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3634–3640.

[101] X. Zhang, R. Cao, Z. Zhang, and Y. Xia, "Crowd flow forecasting with multi-graph neural networks," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–7.

[102] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 5, 2021, pp. 4189–4196.

[103] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 753–763.

[104] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," *Advances in neural information processing systems*, vol. 33, pp. 17 804–17 815, 2020.

[105] C. Shang, J. Chen, and J. Bi, "Discrete graph structure learning for forecasting multiple time series," in *International Conference on Learning Representations*, 2021.

[106] F. Johnston, J. E. Boyland, M. Meadows, and E. Shale, "Some properties of a simple moving average when applied to forecasting a time series," *Journal of the Operational Research Society*, vol. 50, no. 12, pp. 1267–1271, 1999.

[107] C. C. Holt, "Forecasting seasonals and trends by exponentially weighted moving averages," *International journal of forecasting*, vol. 20, no. 1, pp. 5–10, 2004.

[108] E. S. Gardner Jr, "Exponential smoothing: The state of the art," *Journal of forecasting*, vol. 4, no. 1, pp. 1–28, 1985.

[109] E. Ostertagova and O. Ostertag, "Forecasting using simple exponential smoothing method," *Acta Electrotechnica et Informatica*, vol. 12, no. 3, p. 62, 2012.

[110] C. Chatfield, A. B. Koehler, J. K. Ord, and R. D. Snyder, "A new look at models for exponential smoothing," *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 50, no. 2, pp. 147–159, 2001.

[111] A. C. Harvey, "Forecasting, structural time series models and the kalman filter," 1990.

[112] Y. Zhang, "Prediction of financial time series with hidden markov models," 2004.

[113] T. Lux, "The markov-switching multifractal model of asset returns: Gmm estimation and linear forecasting of volatility," *Journal of business & economic statistics*, vol. 26, no. 2, pp. 194–210, 2008.

[114] N. I. Sapankevych and R. Sankar, "Time series prediction using support vector machines: a survey," *IEEE computational intelligence magazine*, vol. 4, no. 2, pp. 24–38, 2009.

[115] L.-J. Cao and F. E. H. Tay, "Support vector machine with adaptive parameters in financial time series forecasting," *IEEE Transactions on neural networks*, vol. 14, no. 6, pp. 1506–1518, 2003.

[116] W. Lu, W. Wang, A. Y. Leung, S.-M. Lo, R. K. Yuen, Z. Xu, and H. Fan, "Air pollutant parameter forecasting using support vector machines," in *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, vol. 1. IEEE, 2002, pp. 630–635.

[117] P.-F. Pai, K.-P. Lin, C.-S. Lin, and P.-T. Chang, "Time series forecasting by a seasonal support vector regression model," *Expert Systems with Applications*, vol. 37, no. 6, pp. 4261–4265, 2010.

[118] L. Zhang, W.-D. Zhou, P.-C. Chang, J.-W. Yang, and F.-Z. Li, "Iterated time series prediction with multiple support vector regression models," *Neurocomputing*, vol. 99, pp. 411–422, 2013.

[119] T. Januschowski, Y. Wang, K. Torkkola, T. Erkkilä, H. Hasson, and J. Gasthaus, "Forecasting with trees," *International Journal of Forecasting*, vol. 38, no. 4, pp. 1473–1481, 2022.

[120] B. Wang, P. Wu, Q. Chen, and S. Ni, "Prediction and analysis of train passenger load factor of high-speed railway based on lightgbm algorithm," *Journal of Advanced Transportation*, vol. 2021, no. 1, p. 9963394, 2021.

[121] H. Wu, Y. Cai, Y. Wu, R. Zhong, Q. Li, J. Zheng, D. Lin, and Y. Li, "Time series analysis of weekly influenza-like illness rate using a one-year period of factors in random forest regression," *Bioscience trends*, vol. 11, no. 3, pp. 292–296, 2017.

[122] V. Mayrink and H. S. Hippert, "A hybrid method using exponential smoothing and gradient boosting for electrical short-term load forecasting," in *2016 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*. IEEE, 2016, pp. 1–6.

[123] F. Martínez, M. P. Frías, M. D. Pérez, and A. J. Rivera, "A methodology for applying k-nearest neighbor to time series forecasting," *Artificial Intelligence Review*, vol. 52, no. 3, pp. 2019–2037, 2019.

[124] F. Martínez, M. P. Frías, M. D. Pérez-Godoy, and A. J. Rivera, "Dealing with seasonality by narrowing the training set in time series forecasting with knn," *Expert systems with applications*, vol. 103, pp. 38–48, 2018.

[125] B. Rajagopalan and U. Lall, "A k-nearest-neighbor simulator for daily precipitation and other weather variables," *Water resources research*, vol. 35, no. 10, pp. 3089–3101, 1999.

[126] F. H. Al-Qahtani and S. F. Crone, "Multivariate k-nearest neighbour regression for time series data—a novel algorithm for forecasting uk electricity demand," in *The 2013 international joint conference on neural networks (IJCNN)*. IEEE, 2013, pp. 1–8.

[127] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *The*

41st international ACM SIGIR conference on research & development in information retrieval, 2018, pp. 95–104.
[128] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," *arXiv preprint arXiv:1704.02971*, 2017.
[129] C. Fan, Y. Zhang, Y. Pan, X. Li, C. Zhang, R. Yuan, D. Wu, W. Wang, J. Pei, and H. Huang, "Multi-horizon time series forecasting with temporal attention learning," in *Proceedings of the 25th ACM SIGKDD International conference on knowledge discovery & data mining*, 2019, pp. 2527–2535.
[130] Y. Wang, A. Smola, D. Maddix, J. Gasthaus, D. Foster, and T. Januschowski, "Deep factors for forecasting," in *International conference on machine learning*. PMLR, 2019, pp. 6607–6617.
[131] S. Smyl, "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting," *International journal of forecasting*, vol. 36, no. 1, pp. 75–85, 2020.
[132] S. Lin, W. Lin, W. Wu, F. Zhao, R. Mo, and H. Zhang, "Segrnn: Segment recurrent neural network for long-term time series forecasting," *arXiv preprint arXiv:2308.11200*, 2023.
[133] J. Cheng, K. Huang, and Z. Zheng, "Towards better forecasting by fusing near and distant future visions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 3593–3600.
[134] H. Wang, J. Peng, F. Huang, J. Wang, J. Chen, and Y. Xiao, "Micn: Multi-scale local and global context modeling for long-term series forecasting," in *The eleventh international conference on learning representations*, 2023.
[135] 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," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5816–5828, 2022.
[136] Z. Li, Y. Qin, X. Cheng, and Y. Tan, "Ftmixer: Frequency and time domain representations fusion for time series modeling," *arXiv preprint arXiv:2405.15256*, 2024.
[137] Z. Yue, Y. Wang, J. Duan, T. Yang, C. Huang, Y. Tong, and B. Xu, "Ts2vec: Towards universal representation of time series," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, 2022, pp. 8980–8987.
[138] 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," *Advances in neural information processing systems*, vol. 32, 2019.
[139] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *International conference on machine learning*. PMLR, 2022, pp. 27 268–27 286.
[140] 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 *The Twelfth International Conference on Learning Representations*, 2024.
[141] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, "N-beats: Neural basis expansion analysis for interpretable time series forecasting," in *International Conference on Learning Representations*, 2020.
[142] Y. Liu, C. Li, J. Wang, and M. Long, "Koopa: Learning non-stationary time series dynamics with koopman predictors," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
[143] Z. Xu, A. Zeng, and Q. Xu, "FITS: Modeling time series with $10k$ parameters," in *The Twelfth International Conference on Learning Representations*, 2024.
[144] S. Lin, W. Lin, W. Wu, H. Chen, and J. Yang, "SparseTSF: Modeling long-term time series forecasting with *1k* parameters," in *Forty-first International Conference on Machine Learning*, 2024.
[145] 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 *The Twelfth International Conference on Learning Representations*, 2024.
[146] S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski, "Deep state space models for time series forecasting," *Advances in neural information processing systems*, vol. 31, 2018.
[147] M. Zhang, K. K. Saab, M. Poli, T. Dao, K. Goel, and C. Re, "Effectively modeling time series with simple discrete state spaces," in *The Eleventh International Conference on Learning Representations*, 2023.
[148] M. A. Ahamed and Q. Cheng, "Timemachine: A time series is worth 4 mambas for long-term forecasting," *arXiv preprint arXiv:2403.09898*, 2024.
[149] Z. Wang, F. Kong, S. Feng, M. Wang, X. Yang, H. Zhao, D. Wang, and Y. Zhang, "Is mamba effective for time series forecasting?" *arXiv preprint arXiv:2403.11144*, 2024.
[150] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "Kan: Kolmogorov-arnold networks," *arXiv preprint arXiv:2404.19756*, 2024.
[151] C. J. Vaca-Rubio, L. Blanco, R. Pereira, and M. Caus, "Kolmogorov-arnold networks (kans) for time series analysis," *arXiv preprint arXiv:2405.08790*, 2024.
[152] R. Genet and H. Inzirillo, "A temporal kolmogorov-arnold transformer for time series forecasting," *arXiv preprint arXiv:2406.02486*, 2024.
[153] Y. Liu, H. Zhang, C. Li, X. Huang, J. Wang, and M. Long, "Timer: generative pre-trained transformers are large time series models," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 32 369–32 399.
[154] Y. Liu, G. Qin, X. Huang, J. Wang, and M. Long, "Autotimes: Autoregressive time series forecasters via large language models," *Advances in Neural Information Processing Systems*, vol. 37, pp. 122 154–122 184, 2025.
[155] B. Cai, S. Yang, L. Gao, and Y. Xiang, "Hybrid variational autoencoder for time series forecasting," *Knowledge-Based Systems*, vol. 281, p. 111079, 2023.
[156] J. Yoon, D. Jarrett, and M. Van der Schaar, "Time-series generative adversarial networks," *Advances in neural information processing systems*, vol. 32, 2019.
[157] N. Kim, D.-K. Chae, J. A. Shin, S.-W. Kim, D. H. Chau, and S. Park, "Context-aware traffic flow forecasting in new roads," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 4133–4137.
[158] Y. Zhang, Y. Li, X. Zhou, X. Kong, and J. Luo, "Curb-gan: Conditional urban traffic estimation through spatio-temporal generative adversarial networks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 842–852.
[159] E. Tamir, N. Laabid, M. Heinonen, V. Garg, and A. Solin, "Conditional flow matching for time series modelling," *ICML 2024 Workshop on Structured Probabilistic Inference and Generative Modeling*, 2023.
[160] X. N. Zhang, Y. Pu, Y. Kawamura, A. Loza, Y. Bengio, D. Shung, and A. Tong, "Trajectory flow matching with applications to clinical time series modelling," *Advances in Neural Information Processing Systems*, vol. 37, pp. 107 198–107 224, 2025.
[161] Y. Hu, X. Wang, L. Wu, H. Zhang, S. Z. Li, S. Wang, and T. Chen, "Fm-ts: Flow matching for time series generation," *arXiv preprint arXiv:2411.07506*, 2024.
[162] K. Rasul, C. Seward, I. Schuster, and R. Vollgraf, "Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8857–8868.
[163] M. Kollovieh, A. F. Ansari, M. Bohlke-Schneider, J. Zschiegner, H. Wang, and Y. B. Wang, "Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
[164] T. Yan, H. Zhang, T. Zhou, Y. Zhan, and Y. Xia, "Scoregrad: Multivariate probabilistic time series forecasting with continuous energy-based generative models," *arXiv preprint arXiv:2106.10121*, 2021.
[165] L. Shen and J. T. Kwok, "Non-autoregressive conditional diffusion models for time series prediction," in *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023, pp. 31 016–31 029.
[166] J. Zhang, M. Cheng, X. Tao, Z. Liu, and D. Wang, "Fdf: Flexible decoupled framework for time series forecasting with conditional denoising and polynomial modeling," *arXiv preprint arXiv:2410.13253*, 2024.
[167] S. Tonekaboni, D. Eytan, and A. Goldenberg, "Unsupervised representation learning for time series with temporal neighborhood coding," in *International Conference on Learning Representations*, 2021.
[168] E. Eldele, M. Ragab, Z. Chen, M. Wu, C. K. Kwoh, X. Li, and C. Guan, "Time-series representation learning via temporal and contextual contrasting," *arXiv preprint arXiv:2106.14112*, 2021.

[169] X. Zhang, Z. Zhao, T. Tsiligkaridis, and M. Zitnik, "Self-supervised contrastive pre-training for time series via time-frequency consistency," *Advances in Neural Information Processing Systems*, vol. 35, pp. 3988–4003, 2022.
[170] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 2114–2124.
[171] M. Cheng, Q. Liu, Z. Liu, H. Zhang, R. Zhang, and E. Chen, "Timemae: Self-supervised representations of time series with decoupled masked autoencoders," *arXiv preprint arXiv:2303.00320*, 2023.
[172] J. Dong, H. Wu, H. Zhang, L. Zhang, J. Wang, and M. Long, "Simmtm: A simple pre-training framework for masked time-series modeling," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
[173] D. Wang, M. Cheng, Z. Liu, Q. Liu, and E. Chen, "Timedart: A diffusion autoregressive transformer for self-supervised time series representation," *arXiv preprint arXiv:2410.05711*, 2024.
[174] X. Jin, Y. Park, D. Maddix, H. Wang, and Y. Wang, "Domain adaptation for time series forecasting via attention sharing," in *International Conference on Machine Learning*. PMLR, 2022, pp. 10 280–10 297.
[175] B. Wang, J. Ma, P. Wang, X. Wang, Y. Zhang, Z. Zhou, and Y. Wang, "Stone: A spatio-temporal ood learning framework kills both spatial and temporal shifts," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 2948–2959.
[176] H. Liu, H. Kamarthi, L. Kong, Z. Zhao, C. Zhang, and B. A. Prakash, "Time-series forecasting for out-of-distribution generalization using invariant learning," in *Forty-first International Conference on Machine Learning*, 2024.
[177] T. Zhou, P. Niu, L. Sun, R. Jin *et al.*, "One fits all: Power general time series analysis by pretrained lm," *Advances in neural information processing systems*, vol. 36, pp. 43 322–43 355, 2023.
[178] X. Liu, J. Hu, Y. Li, S. Diao, Y. Liang, B. Hooi, and R. Zimmermann, "Unitime: A language-empowered unified model for cross-domain time series forecasting," in *Proceedings of the ACM on Web Conference 2024*, 2024, pp. 4095–4106.
[179] C. Chang, W.-C. Peng, and T.-F. Chen, "Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms," *arXiv preprint arXiv:2308.08469*, 2023.
[180] 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 *The Twelfth International Conference on Learning Representations*, 2024.
[181] N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson, "Large language models are zero-shot time series forecasters," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
[182] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan *et al.*, "Time-llm: Time series forecasting by reprogramming large language models," *arXiv preprint arXiv:2310.01728*, 2023.
[183] C. Sun, H. Li, Y. Li, and S. Hong, "TEST: Text prototype aligned embedding to activate LLM's ability for time series," in *The Twelfth International Conference on Learning Representations*, 2024.
[184] M. Goswami, K. Szafer, A. Choudhry, Y. Cai, S. Li, and A. Dubrawski, "Moment: A family of open time-series foundation models," 2024. [Online]. Available: https://arxiv.org/abs/2402.03885
[185] A. Das, W. Kong, R. Sen, and Y. Zhou, "A decoder-only foundation model for time-series forecasting," *arXiv preprint arXiv:2310.10688*, 2023.
[186] S. Johansen, "Estimation and hypothesis testing of cointegration vectors in gaussian vector autoregressive models," *Econometrica: journal of the Econometric Society*, pp. 1551–1580, 1991.
[187] L. Song, M. Kolar, and E. Xing, "Time-varying dynamic bayesian networks," *Advances in neural information processing systems*, vol. 22, 2009.
[188] P. Cui, Z. Shen, S. Li, L. Yao, Y. Li, Z. Chu, and J. Gao, "Causal inference meets machine learning," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 3527–3528.
[189] K. Sel, A. Mohammadi, R. I. Pettigrew, and R. Jafari, "Physics-informed neural networks for modeling physiological time series for cuffless blood pressure estimation," *npj Digital Medicine*, vol. 6, no. 1, p. 110, 2023.
[190] F. M. Abushaqra, H. Xue, Y. Ren, and F. D. Salim, "Seqlink: A robust neural-ode architecture for modelling partially observed time series," *Transactions on Machine Learning Research*, 2024.
[191] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.
[192] B. Huang and J. Wang, "Applications of physics-informed neural networks in power systems-a review," *IEEE Transactions on Power Systems*, vol. 38, no. 1, pp. 572–588, 2022.
[193] A. Bracco, J. Brajard, H. A. Dijkstra, P. Hassanzadeh, C. Lessig, and C. Monteleoni, "Machine learning for the physics of climate," *Nature Reviews Physics*, pp. 1–15, 2024.
[194] R. Dang-Nhu, G. Singh, P. Bielik, and M. Vechev, "Adversarial attacks on probabilistic autoregressive forecasting models," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2356–2365.
[195] L. Liu, Y. Park, T. N. Hoang, H. Hasson, and L. Huan, "Robust multivariate time-series forecasting: Adversarial attacks and defense mechanisms," in *The Eleventh International Conference on Learning Representations*, 2023.
[196] F. Liu, W. Zhang, and H. Liu, "Robust spatiotemporal traffic forecasting with reinforced dynamic adversarial training," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 1417–1428.
[197] X. Lin, Z. Liu, D. Fu, R. Qiu, and H. Tong, "Backtime: Backdoor attacks on multivariate time series forecasting," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
[198] C. Meng, S. Rambhatla, and Y. Liu, "Cross-node federated graph neural network for spatio-temporal data modeling," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 1202–1211.
[199] S. Chen, G. Long, T. Shen, and J. Jiang, "Prompt federated learning for weather forecasting: toward foundation models on meteorological data," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 2023, pp. 3532–3540.
[200] Q. Liu, X. Liu, C. Liu, Q. Wen, and Y. Liang, "Time-ffm: Towards lm-empowered federated foundation model for time series forecasting," *arXiv preprint arXiv:2405.14252*, 2024.
[201] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Advances in neural information processing systems*, vol. 9, 1996.
[202] C. Cortes, "Support-vector networks," *Machine Learning*, 1995.
[203] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, pp. 199–222, 2004.
[204] L. BREIMAN, "Classification and regression trees," *Monterey, CA: Wadsworth and Brools*, 1984.
[205] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.
[206] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.
[207] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.
[208] 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," *Advances in neural information processing systems*, vol. 30, 2017.
[209] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.
[210] H. Hewamalage, C. Bergmeir, and K. Bandara, "Recurrent neural networks for time series forecasting: Current status and future directions," *International Journal of Forecasting*, vol. 37, no. 1, pp. 388–427, 2021.
[211] R. Wen, K. Torkkola, B. Narayanaswamy, and D. Madeka, "A multi-horizon quantile recurrent forecaster," *arXiv preprint arXiv:1711.11053*, 2017.
[212] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.
[213] R. Pascanu, "On the difficulty of training recurrent neural networks," *arXiv preprint arXiv:1211.5063*, 2013.

[214] B. Peng, E. Alcaide, Q. Anthony, A. Albalak, S. Arcadinho, S. Biderman, H. Cao, X. Cheng, M. Chung, M. Grella *et al.*, "Rwkv: Reinventing rnns for the transformer era," *arXiv preprint arXiv:2305.13048*, 2023.

[215] M. Beck, K. Pöppel, M. Spanring, A. Auer, O. Prudnikova, M. Kopp, G. Klambauer, J. Brandstetter, and S. Hochreiter, "xlstm: Extended long short-term memory," *arXiv preprint arXiv:2405.04517*, 2024.

[216] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[217] 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.

[218] S. Huang, D. Wang, X. Wu, and A. Tang, "Dsanet: Dual self-attention network for multivariate time series forecasting," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 2129–2132.

[219] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[220] 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 *International Conference on Learning Representations*, 2022.

[221] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary transformers: Exploring the stationarity in time series forecasting," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9881–9893, 2022.

[222] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi, "Etsformer: Exponential smoothing transformers for time-series forecasting," *arXiv preprint arXiv:2202.01381*, 2022.

[223] J. Jiang, C. Han, W. X. Zhao, and J. Wang, "Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 4, 2023, pp. 4365–4373.

[224] R. Ilbert, A. Odonnat, V. Feofanov, A. Virmaux, G. Paolo, T. Palpanas, and I. Redko, "Samformer: unlocking the potential of transformers in time series forecasting with sharpness-aware minimization and channel-wise attention," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 20 924–20 954.

[225] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[226] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[227] D. P. Kingma, M. Welling *et al.*, "Auto-encoding variational bayes," 2013.

[228] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[229] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *International conference on machine learning*. PMLR, 2015, pp. 1530–1538.

[230] Z. Wang, Q. Wen, C. Zhang, L. Sun, and Y. Wang, "Diffload: uncertainty quantification in load forecasting with diffusion model," *arXiv preprint arXiv:2306.01001*, 2023.

[231] P. Chang, H. Li, S. F. Quan, S. Lu, S.-F. Wung, J. Roveda, and A. Li, "Tdstf: Transformer-based diffusion probabilistic model for sparse time series forecasting," *arXiv preprint arXiv:2301.06625*, 2023.

[232] N. Neifar, A. Ben-Hamadou, A. Mdhaffar, and M. Jmaiel, "Diffecg: A versatile probabilistic diffusion model for ecg signals synthesis," *arXiv preprint arXiv:2306.01875*, 2023.

[233] S. Feng, C. Miao, Z. Zhang, and P. Zhao, "Latent diffusion transformer for probabilistic time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 11, 2024, pp. 11 979–11 987.

[234] P. Shao, J. Feng, J. Lu, P. Zhang, and C. Zou, "Data-driven and knowledge-guided denoising diffusion model for flood forecasting," *Expert Systems with Applications*, vol. 244, p. 122908, 2024.

[235] D. Daiya, M. Yadav, and H. S. Rao, "Diffstock: Probabilistic relational stock market predictions using diffusion models," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 7335–7339.

[236] S. Li, H. Xiong, and Y. Chen, "Diffplf: A conditional diffusion model for probabilistic forecasting of ev charging load," *arXiv preprint arXiv:2402.13548*, 2024.

[237] M. Ragab, E. Eldele, Z. Chen, M. Wu, C.-K. Kwoh, and X. Li, "Self-supervised autoregressive domain adaptation for time series data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 1, pp. 1341–1351, 2022.

[238] S. Mirchandani, F. Xia, P. Florence, B. Ichter, D. Driess, M. G. Arenas, K. Rao, D. Sadigh, and A. Zeng, "Large language models as general pattern machines," *arXiv preprint arXiv:2307.04721*, 2023.

[239] Y. Wang, Z. Chu, X. Ouyang, S. Wang, H. Hao, Y. Shen, J. Gu, S. Xue, J. Y. Zhang, Q. Cui *et al.*, "Enhancing recommender systems with large language model reasoning graphs," *arXiv preprint arXiv:2308.10835*, 2023.

[240] F. Jia, K. Wang, Y. Zheng, D. Cao, and Y. Liu, "Gpt4mts: Prompt-based large language model for multimodal time-series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 21, 2024, pp. 23 343–23 351.

[241] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022.

[242] H. Liu, Z. Zhao, J. Wang, H. Kamarthi, and B. A. Prakash, "Lstprompt: Large language models as zero-shot time series forecasters by long-short-term prompting," *arXiv preprint arXiv:2402.16132*, 2024.

[243] Y. Zhu, Z. Wang, J. Gao, Y. Tong, J. An, W. Liao, E. M. Harrison, L. Ma, and C. Pan, "Prompting large language models for zero-shot clinical prediction with structured longitudinal electronic health record data," *arXiv preprint arXiv:2402.01713*, 2024.

[244] H. Zhang, C. Xu, Y.-F. Zhang, Z. Zhang, L. Wang, J. Bian, and T. Tan, "Timeraf: Retrieval-augmented foundation model for zero-shot time series forecasting," *arXiv preprint arXiv:2412.20810*, 2024.

[245] M. Xiao, Z. Jiang, Z. Chen, D. Li, S. Chen, S. Ananiadou, J. Huang, M. Peng, and Q. Xie, "Timerag: It's time for retrieval-augmented generation in time-series forecasting."

[246] J. Wang, M. Cheng, Q. Mao, Q. Liu, F. Xu, X. Li, and E. Chen, "Tabletime: Reformulating time series classification as zero-shot table understanding via large language models," *arXiv preprint arXiv:2411.15737*, 2024.

[247] M. Lin, Z. Chen, Y. Liu, X. Zhao, Z. Wu, J. Wang, X. Zhang, S. Wang, and H. Chen, "Decoding time series with llms: A multi-agent framework for cross-domain annotation," *arXiv preprint arXiv:2410.17462*, 2024.

[248] M. A. Merrill, M. Tan, V. Gupta, T. Hartvigsen, and T. Althoff, "Language models still struggle to zero-shot reason about time series," *arXiv preprint arXiv:2404.11757*, 2024.

[249] C. Glymour, K. Zhang, and P. Spirtes, "Review of causal discovery methods based on graphical models," *Frontiers in genetics*, vol. 10, p. 524, 2019.

[250] J. Tian and J. Pearl, "Causal discovery from changes," *arXiv preprint arXiv:1301.2312*, 2013.

[251] C. K. Assaad, E. Devijver, and E. Gaussier, "Survey and evaluation of causal discovery methods for time series," *Journal of Artificial Intelligence Research*, vol. 73, pp. 767–819, 2022.

[252] C. Miller, A. Kathirgamanathan, B. Picchetti, P. Arjunan, J. Y. Park, Z. Nagy, P. Raftery, B. W. Hobson, Z. Shi, and F. Meggers, "The building data genome project 2, energy meter data from the ashrae great energy predictor iii competition," *Scientific data*, vol. 7, no. 1, p. 368, 2020.

[253] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, J. Albrecht *et al.*, "Smart*: An open data set and tools for enabling research in sustainable homes," *SustKDD, August*, vol. 111, no. 112, p. 108, 2012.

[254] D. A. Bashawyah and S. M. Qaisar, "Machine learning based short-term load forecasting for smart meter energy consumption data in london households," in *2021 IEEE 12th International Conference on Electronics and Information Technologies (ELIT)*. IEEE, 2021, pp. 99–102.

[255] J. Zhou, X. Lu, Y. Xiao, J. Su, J. Lyu, Y. Ma, and D. Dou, "Sdwpf: A dataset for spatial dynamic wind power forecasting challenge at kdd cup 2022," *arXiv preprint arXiv:2208.04360*, 2022.

[256] A. Alexandrov, K. Benidis, M. Bohlke-Schneider, V. Flunkert, J. Gasthaus, T. Januschowski, D. C. Maddix, S. Rangapuram, D. Salinas, J. Schulz *et al.*, "Gluonts: Probabilistic and neural time series modeling in python," *Journal of Machine Learning Research*, vol. 21, no. 116, pp. 1–6, 2020.

[257] J. Wang, J. Jiang, W. Jiang, C. Li, and W. X. Zhao, "Libcity: An open library for traffic prediction," in *Proceedings of the 29th international conference on advances in geographic information systems*, 2021, pp. 145–148.

[258] Google, "Web traffic time series forecasting," https://www.kaggle.com/c/web-traffic-time-series-forecasting, 2017.

[259] H. V. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi, "Big data and its technical challenges," *Communications of the ACM*, vol. 57, no. 7, pp. 86–94, 2014.

[260] S. Mouatadid, P. Orenstein, G. Flaspohler, M. Oprescu, J. Cohen, F. Wang, S. Knight, M. Geogdzhayeva, S. Levang, E. Fraenkel *et al.*, "Subseasonalclimateusa: a dataset for subseasonal forecasting and benchmarking," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[261] X. Qiu, J. Hu, L. Zhou, X. Wu, J. Du, B. Zhang, C. Guo, A. Zhou, C. S. Jensen, Z. Sheng *et al.*, "Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods," *Proceedings of the VLDB Endowment*, vol. 17, no. 9, pp. 2363–2377, 2024.

[262] Y. Zheng, X. Yi, M. Li, R. Li, Z. Shan, E. Chang, and T. Li, "Forecasting fine-grained air quality based on big data," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 2267–2276.

[263] S. Chen, "Beijing Multi-Site Air Quality," UCI Machine Learning Repository, 2017, DOI: https://doi.org/10.24432/C5RK5G.

[264] S. Makridakis, A. Andersen, R. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, E. Parzen, and R. Winkler, "The accuracy of extrapolation (time series) methods: Results of a forecasting competition," *Journal of forecasting*, vol. 1, no. 2, pp. 111–153, 1982.

[265] S. Makridakis and M. Hibon, "The m3-competition: results, conclusions and implications," *International journal of forecasting*, vol. 16, no. 4, pp. 451–476, 2000.

[266] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The m4 competition: Results, findings, conclusion and way forward," *International Journal of forecasting*, vol. 34, no. 4, pp. 802–808, 2018.

[267] ——, "M5 accuracy competition: Results, findings, and conclusions," *International Journal of Forecasting*, vol. 38, no. 4, pp. 1346–1364, 2022.

[268] S. B. Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition," *Expert systems with applications*, vol. 39, no. 8, pp. 7067–7083, 2012.

[269] G. Athanasopoulos, R. J. Hyndman, H. Song, and D. C. Wu, "The tourism forecasting competition," *International Journal of Forecasting*, vol. 27, no. 3, pp. 822–844, 2011.

[270] W. G. van Panhuis, A. Cross, and D. S. Burke, "Project tycho 2.0: a repository to improve the integration and reuse of data for global population health," *Journal of the American Medical Informatics Association*, vol. 25, no. 12, pp. 1608–1617, 2018.

[271] F. Piccialli, F. Giampaolo, E. Prezioso, D. Camacho, and G. Acampora, "Artificial intelligence and healthcare: Forecasting of medical bookings through multi-source time-series fusion," *Information Fusion*, vol. 74, pp. 1–16, 2021.

[272] J. Xie and Q. Wang, "Benchmarking machine learning algorithms on blood glucose prediction for type i diabetes in comparison with classical time-series models," *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 11, pp. 3101–3124, 2020.

[273] E. Hwang, Y.-S. Park, J.-Y. Kim, S.-H. Park, J. Kim, and S.-H. Kim, "Intraoperative hypotension prediction based on features automatically generated within an interpretable deep learning model," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[274] F. Lu, W. Li, Z. Zhou, C. Song, Y. Sun, Y. Zhang, Y. Ren, X. Liao, H. Jin, A. Luo *et al.*, "A composite multi-attention framework for intraoperative hypotension early warning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 12, 2023, pp. 14 374–14 381.

[275] M. Cheng, J. Zhang, Z. Liu, C. Liu, and Y. Xie, "Hmf: A hybrid multi-factor framework for dynamic intraoperative hypotension prediction," *arXiv preprint arXiv:2409.11064*, 2024.

[276] C.-Y. Hsu and W.-C. Liu, "Multiple time-series convolutional neural network for fault detection and diagnosis and empirical study in semiconductor manufacturing," *Journal of Intelligent Manufacturing*, vol. 32, no. 3, pp. 823–836, 2021.

[277] H. Ben Ameur, S. Boubaker, Z. Ftiti, W. Louhichi, and K. Tissaoui, "Forecasting commodity prices: empirical evidence using deep learning tools," *Annals of Operations Research*, vol. 339, no. 1, pp. 349–367, 2024.

[278] W. Kong, H. Li, C. Yu, J. Xia, Y. Kang, and P. Zhang, "A deep spatio-temporal forecasting model for multi-site weather prediction post-processing," *Communications in Computational Physics*, vol. 31, no. 1, pp. 131–153, 2022.

[279] H. Wu, H. Zhou, M. Long, and J. Wang, "Interpretable weather forecasting for worldwide stations with a unified deep model," *Nature Machine Intelligence*, vol. 5, no. 6, pp. 602–611, 2023.

[280] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 922–929.

[281] J. Fan, W. Weng, H. Tian, H. Wu, F. Zhu, and J. Wu, "Rgdan: A random graph diffusion attention network for traffic prediction," *Neural networks*, vol. 172, p. 106093, 2024.