A Comprehensive Survey of Time Series Forecasting: Architectural Diversity and Open Challenges

Jongseon Kim[1,3†], Hyungjoon Kim[1,4†], HyunGi Kim[2], Dongjun Lee[1], Sungroh Yoon[1, 2*]

[1]Interdisciplinary Program in Artificial Intelligence, Seoul National University.
[2]Department of Electrical and Computer Engineering, Seoul National University.
[3]R&D Department, LG Chem.
[4]R&D Department, Samsung SDI.

*Corresponding author(s). E-mail(s): sryoon@snu.ac.kr;
Contributing authors: hallooawooye@snu.ac.kr; khjn81@snu.ac.kr;
rlagusrl0128@snu.ac.kr; elite1717@snu.ac.kr;
†These authors contributed equally to this work.

Abstract

Time series forecasting is a critical task that provides key information for decision-making across various fields, such as economic planning, supply chain management, and medical diagnosis. After the use of traditional statistical methodologies and machine learning in the past, various fundamental deep learning architectures such as MLPs, CNNs, RNNs, and GNNs have been developed and applied to solve time series forecasting problems. However, the structural limitations caused by the inductive biases of each deep learning architecture constrained their performance. Transformer models, which excel at handling long-term dependencies, have become significant architectural components for time series forecasting. However, recent research has shown that alternatives such as simple linear layers can outperform Transformers. These findings have opened up new possibilities for using diverse architectures. In this context of exploration into various models, the architectural modeling of time series forecasting has now entered a renaissance. This survey not only provides a historical context for time series forecasting but also offers comprehensive and timely analysis of the movement toward architectural diversification. By comparing and re-examining various deep learning models, we uncover new perspectives and presents the latest trends in time series forecasting, including the emergence of hybrid models, diffusion models, Mamba models, and foundation models. By focusing on the inherent characteristics of time series data, we also address open challenges that have gained attention in time series forecasting, such as channel dependency, distribution shift, causality, and feature extraction. This survey explores vital elements that can enhance forecasting performance through diverse approaches. These contributions lead to lowering the entry barriers for newcomers to the field of time series forecasting, while also offering seasoned researchers broad perspectives, new opportunities, and deep insights.

Keywords: Time series forecasting, Deep learning, Foundation model, Distribution shift, Causality

1 Introduction

Time series forecasting (TSF) is a task that predicts future values based on sequential historical data over time (Cryer, 1986). It is utilized as a key decision-making tool in various fields, such as economics and finance, supply chain management, transportation, energy, weather and healthcare (Danese and Kalchschmidt, 2011; Abu-Mostafa and Atiya, 1996; Alghamdi et al, 2019; Nti et al, 2020; Dimri et al, 2020; Soyiri and Reidpath, 2013). Such applications offer various opportunities, including cost

A Comprehensive Survey of Time Series Forecasting: Architectural Diversity and Open Challenges

Fig. 1: Overview of This Survey

reduction, increased efficiency, and enhanced competitiveness (Danese and Kalchschmidt, 2011). The inherent diversity and complexity of time series data unfortunately make forecasting challenging. In addition to the apparent information, various hidden patterns make it challenging to learn temporal dependencies, and irregular values at times further complicate the problem. In multivariate problems, additional factors such as channel correlation make the task even more difficult (Section 2.1). Furthermore, time series data exhibits different characteristics depending on the domain, and the various times and environments in which the series is collected result in significantly different patterns (Section 2.3). Because of this, TSF problems exhibit limited model generalizability, requiring diverse architectures and approaches. The increasingly complicated TSF problems are presenting researchers with growing challenges, which has recently led to the active development of new methodologies and algorithms to address these issues (Lim and Zohren, 2021).

The explosive increase in the number of papers accepted at major AI and Machine Learning conferences (Fig. 2) demonstrates that TSF research is becoming increasingly important in the AI and Machine Learning fields. As various studies addressing time series forecasting problems are being actively conducted, survey papers are also being frequently published (Table 1). Over time, numerous survey papers have systematically organized the vast landscape of TSF, offering in-depth research that has provided valuable guidance and direction for researchers. However, existing survey papers still have room for improvement, particularly in addressing the inevitable increase in model diversity and the open challenges in the field.

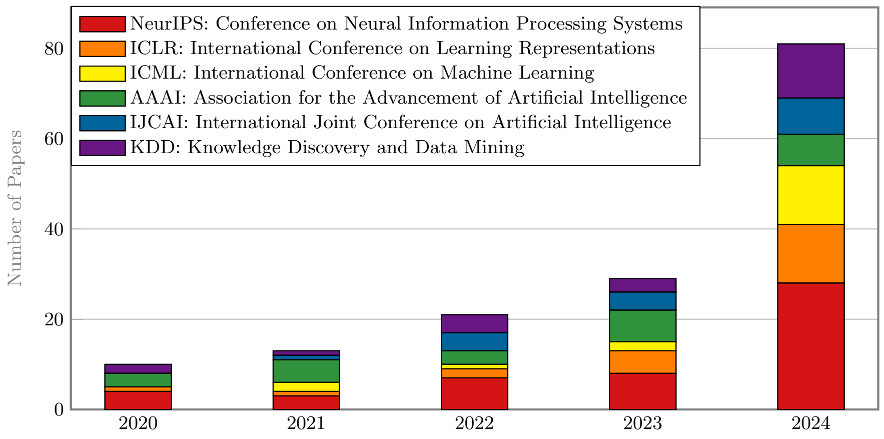


Fig. 2: Number of Top-tier AI and ML Conference Papers on Time Series Forecasting

Models for TSF have undergone various stages of development over an extended period. In the past, statistical methods based on moving averages were predominantly used, which later evolved into traditional approaches such as Exponential Smoothing and ARIMA (Bartholomew, 1971). Machine learning techniques such as Tree Models (Quinlan, 1986) and Support Vector Machines (SVM) (Cortes, 1995) have also been frequently used, but they had limitations in learning complex nonlinear patterns (Section 3.1). With the increase in available data and advancements in hardware computing power, various deep learning architectures such as MLPs (Rumelhart et al, 1986), RNNs (Hopfield, 1982), CNNs (LeCun et al, 1998), and GNNs (Scarselli et al, 2008) were developed, enabling the learning of more complex patterns. However, the performance of these early deep learning architectures was constrained by their intrinsic designs. To overcome these structural limitations, variants such as Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997a) and Temporal Convolutional Networks (TCN) (Bai et al, 2018) have been widely utilized (Section 3.2). Transformers (Vaswani, 2017), known for their ability to handle long-term dependencies, have demonstrated excellent performance in natural language processing and have been naturally extended to time series data as well. While Transformers have shown good performance in TSF and become widely popular, recent cases have shown that simple linear models can outperform Transformer models (Section 3.3). As a result, there has been a significant increase in reconsidering traditional deep learning methodologies, along with growing interest in various architectures such as foundation models, diffusion models and Mamba models (Section 4.2, 4.3, 4.4, 4.5). The Transformer model continues to improve in performance and still plays a significant role (Section 4.1). In this way, TSF has entered a renaissance of modeling, with various methodologies actively competing without being dominated by any single approach (Fig. 3). In this context, this survey offers two major strengths that set it apart from previous TSF survey papers.

First, we focus on the inevitable diversification of architectures, providing a timely and comprehensive view to understand the current trend of architecture diversification. Existing TSF survey papers, such as (Wen et al, 2022a; Zhang et al, 2024c; Lin et al, 2024a; Meijer and Chen, 2024; Patro and Agneeswaran, 2024a), focus on providing detailed analyses of specific architectures but have limitations when it comes to broadly comparing diversified architectures, including newly emerging ones. This paper systematically compares the developmental progress of various architectures (MLPs, CNNs, RNNs, GNNs, Transformer, Diffusion, foundation models, Mamba) and analyzes the strengths, weaknesses, and contributions of each. In addition, it addresses the performance of hybrid models that combine the strengths of multiple architectures, clearly highlighting key trends in TSF. With these contributions, readers can effectively understand the continuously evolving trends and directions of advancement in the field. Through this, it lowers the entry barrier for newcomers to TSF and provides a comprehensive roadmap that opens up new research opportunities for established researchers.

Second, we explore from the perspective of open challenges. Although numerous advanced architectures have resolved many issues, the core challenges in TSF continue to persist. In particular, issues such as channel correlation, distribution shifts, causality, and feature extraction (Section 5) remain significant challenges that need to be addressed (Fig. 4). This survey explores the latest methodologies aimed at addressing these challenges and provides readers with valuable insights for problem-solving. While previous surveys have provided useful perspectives on open challenges, they have not explored these issues in sufficient depth. This survey aims to bridge that gap by offering a more comprehensive analysis and proposing new solutions.

This survey covers the fundamental concepts of time series data and the problem definition of forecasting in Section 2, followed by an examination of the evolution of past methodologies in Section 3. In Section 4, it analyzes the key features of the latest models, and finally, in Section 5, it explores the open challenges in TSF and their solutions. Through this, readers will gain a broad understanding of the past and present of TSF research and acquire new ideas for future studies.

2 Background

In this section, before exploring time series forecasting models, we explain the definition and key characteristics of time series data to provide the necessary background. We also define the problem of time series forecasting tasks, discussing related datasets and evaluation metrics. This establishes the basic concepts of time series forecasting and provides preliminary information to help understand the models discussed in the following sections.

2.1 Time-Series Data

Time series data is a collection of sequential data points gathered at regular time intervals, representing a series of observations of phenomena that change over time. This temporal continuity allows for the understanding and analysis of phenomena that evolve according to time order. Each data point represents the state or value at a specific moment, and through the observation of these data points, various patterns such as long-term trends, seasonality, cyclicality, and irregularities can be recognized. These patterns provide valuable information for predicting future values or detecting changes at critical moments. By extracting and studying the meaningful information provided by time series data, it can be applied to practical applications, helping to address many challenges in various disciplines.

Characteristics of Time Series Data

Time series data encapsulates various characteristics that play a critical role in explaining the diverse patterns and fluctuations within time series. Understanding these characteristic elements is essential for analyzing and predicting data. The key properties are explained in Fig. 5.

In time series data, the above features frequently appear in a mixed form. Therefore, decomposition is commonly used to separate the components for detailed analysis, or distribution shift alleviation methods are widely applied. Many time series datasets provide information on multivariate variables. Sometimes, these data provide additional information that univariate data cannot, and it is important to understand this for many problems. The main properties are explained in Fig. 6.

Fig. 3: Evolution of Time Series Forecasting Models

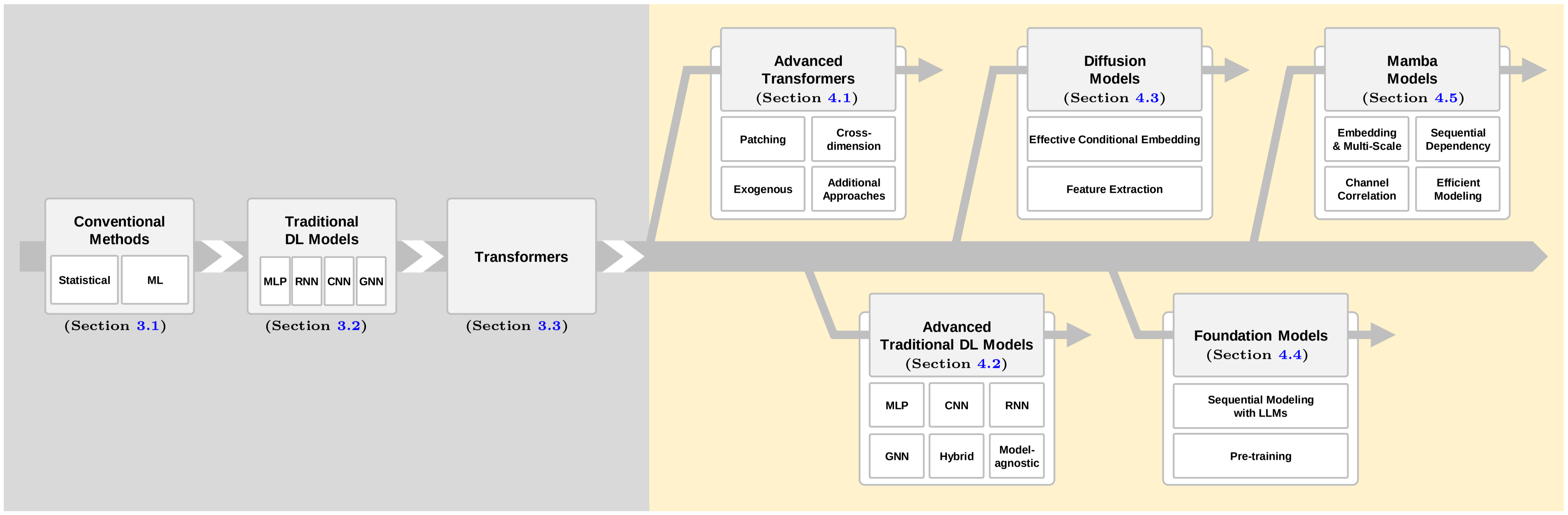


Table 1: Summary of Survey Papers on Time Series Forecasting

Articles	Focus	Broad Architecture Review	Forecasting Task-Intensive	Recent Work	Reference
Time-series forecasting with deep learning: a survey	· Encoder-Decoder structures and Hybrid models · Interpretability and causal inference for decision support	✓	✓		Lim and Zohren (2021)
Forecast Methods for Time Series Data: A Survey	· Traditional forecasting methods: statistical, ML, and DL · Challenges in preprocessing, modeling, and parallel computation	✓	✓		Liu et al (2021b)
Deep Learning for Time Series Forecasting: Tutorial and Literature Survey	· Key components of DL TSF · Practical applicability	✓	✓		Benidis et al (2022)
A Review on Deep Sequential Models for Forecasting Time Series Data	· Common deep sequential models · Guidelines on implementation, application, and optimization	✓	✓		Ahmed et al (2022)
Transformers in Time Series: A Survey	· Transformer structure variations and architectural improvements		✓		Wen et al (2022a)
Long Sequence Time-Series Forecasting with Deep Learning: A Survey	· Definition of long sequence forecasting challenges from various perspectives · New classification system and performance evaluation methods	✓			Chen et al (2023)
Machine Learning Advances for Time Series Forecasting	· Economic/financial TSF · Categorized into linear and nonlinear	✓			Masini et al (2023)
Diffusion Models for Time-Series Applications: A Survey	· Diffusion models in time series analysis		✓		Lin et al (2024a)
The Rise of Diffusion Models in Time-Series Forecasting	· Diffusion-based forecasting models with case studies		✓	✓	Meijer and Chen (2024)
Foundation Models for Time Series Analysis: A Tutorial and Survey	· Application of foundation models in time series analysis		✓	✓	Liang et al (2024c)
Large Language Models for Time Series: A Survey	· Categorization of methodologies for LLMs in time series · Explanation of bridging modality gaps		✓	✓	Zhang et al (2024c)
A Survey of Time Series Foundation Models	· Approaches to Foundation Models and LLMs in time series analysis		✓	✓	Ye et al (2024)
Mamba-360: Survey of State Space Models	· Emphasis on SSM advantages in long time series · Comparison of domain-specific applications and performance		✓	✓	Patro and Agneeswaran (2024a)
This Survey	· A comprehensive overview of the evolution of TSF models, introducing innovative architectures · Discussion of critical open challenges in forecasting tasks, along with advanced solutions to address them	✓	✓	✓	

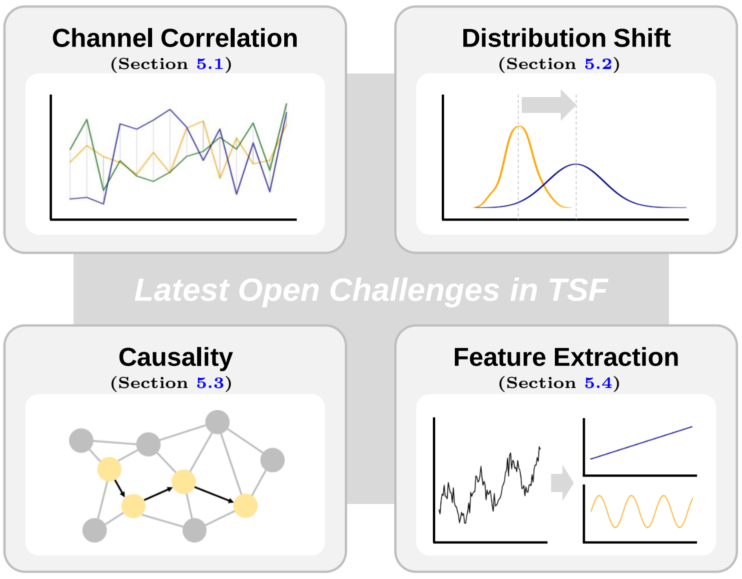


Fig. 4: Overview of Latest Open Challenges in Time Series Forecasting

These additional features convey the complexity of the real world in greater detail and sometimes provide reasons for significant pattern changes. Considering this information can lead to more precise time series analysis and forecasting.

2.2 Time-Series Forecasting

Time series forecasting involves analyzing historical time-series data to predict future values. This process entails analyzing inherent temporal patterns in the data, such as trends and seasonality, using past observations to estimate future values. The goal of time series forecasting is to support decision-making by providing information about an uncertain future, such as predicting future demand changes or fluctuations in prices. Time series forecasting can employ various methodologies, including statistical methods, machine learning, and deep learning. These methodologies focus on capturing the characteristics of data as it changes over time.

Univariate Time Series Forecasting (UTSF). Univariate forecasting refers to making predictions using only one variable. For example, predicting the next day's temperature based solely on past temperature data from a weather station is a univariate forecast. The advantage of univariate forecasting is that the models are simple and computationally efficient. Since the data consists of a single variable, the model is relatively straightforward and easy to understand, making data collection and management easier. However, it may only utilize limited information as it cannot account for important external factors or interactions between different variables:

$$\hat{y}_{t+1} = f(y_t, y_{t-1}, \dots, y_{t-p}) \tag{1}$$

where $\hat{y}_{t+1}$ represents the predicted value at time $t+1$. The terms $y_t, y_{t-1}, \dots, y_{t-p}$ are the past p observations, which include values at time t, $t-1$, continuing down to $t-p$. This approach uses historical data to forecast future values, relying on patterns within the observed variable itself.

Multivariate Time Series Forecasting (MTSF). Multivariate forecasting involves making predictions using multiple variables simultaneously. For instance, in weather forecasting, predicting the next day's temperature by considering various variables such as temperature, humidity, and wind speed is an example of multivariate forecasting. By incorporating interactions and correlations between multiple variables, multivariate forecasting can capture complex relationships, offering higher predictive accuracy. However, these models tend to be more complex, require more data, and can be more challenging to handle, increasing the risk of overfitting.

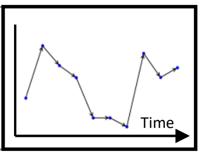
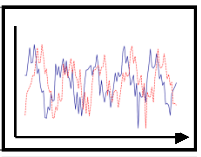
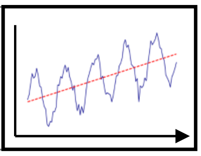
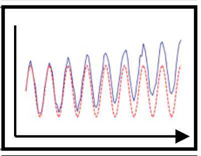
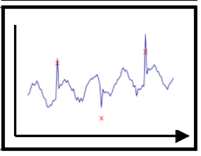
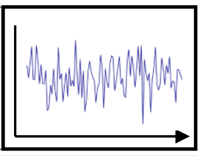
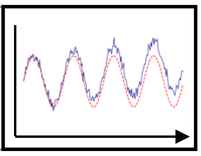
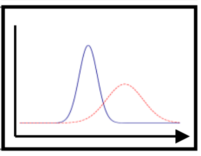
Properties of Time Series Data		
Temporal Order	Data points are arranged at consistent intervals over time, and the chronological order plays an important role in analysis.	
Autocorrelation	The current value of the data may have a correlation with past values, which indicates patterns or trends over time.	
Trend	This attribute represents long-term changes, providing a series of patterns where data increases, decreases, or remains constant over time.	
Seasonality	Represents short-term patterns that repeat over a fixed period, explaining fluctuations at specific times.	
Outliers or noise	Refers to abnormal data points that deviate from general patterns, representing random fluctuations that disrupt data consistency and complicate analysis, requiring removal or minimization during modeling.	
Irregularity	Random fluctuation elements that are difficult to predict in time series data, reflecting the inherent uncertainty of the data.	
Cycles	Involves longer period fluctuations than seasonality and exhibits patterns that repeat over somewhat irregular intervals, such as economic cycles with non-periodic regularity.	
Non-stationarity	Statistical elements of time series data (mean and variance) may change over time. This causes instability in models that assume stationarity, necessitating stabilization.	

Fig. 5: Properties of Time Series Data

To incorporate other influencing variables, we can extend the model to a multivariate approach, as shown in the following equation:

$$\hat{y}_{t+1} = f(y_t, y_{t-1}, \dots, y_{t-p}, x_t, x_{t-1}, \dots, x_{t-p}) \tag{2}$$

where $\hat{y}_{t+1}$ is the predicted value at time $t+1$. The terms $y_t, y_{t-1}, \dots, y_{t-p}$ represent the past p observations of the target variable, capturing its historical patterns. Additionally, $x_t, x_{t-1}, \dots, x_{t-p}$ are the past p observations of other related variables, which provide additional context and information that may influence the prediction. This multivariate approach allows for more comprehensive modeling by considering both the target variable's history and the effects of other relevant factors. Fig. 7 illustrates how a multivariate forecasting model uses a past lookback window to predict future intervals.

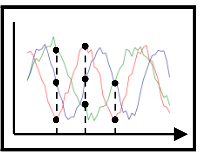
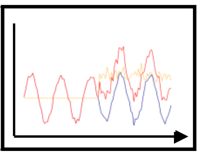
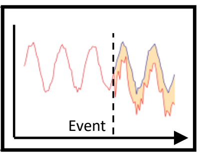
Properties of Multivariate Time Series Data		
Interdependency	In time series data collected simultaneously across multiple channels, these variables can be correlated. Understanding the interactions between variables is important as it can help comprehend complex patterns in time series data.	
Exogenous Variables	External factors or variables influence time series data. Although not included in the data itself, these variables can provide latent information, and considering them during modeling can lead to significant performance improvements.	
Contextual Information	Specific events, such as policy changes or natural disasters, occurring at the time of observation can affect time series data and create complex patterns.	

Fig. 6: Properties of Multivariate Time Series Data

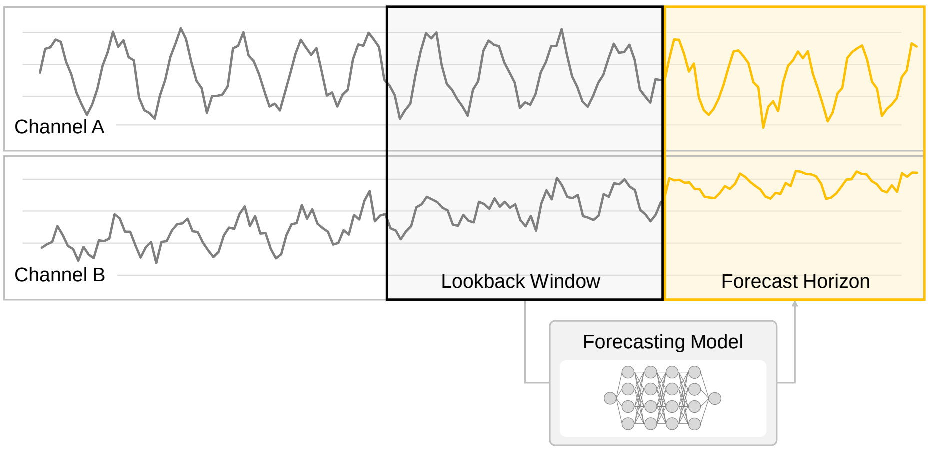


Fig. 7: Diagram of Multivariate Time Series Forecasting

Short-Term Time Series Forecasting (STSF). Short-term time series forecasting focuses on predictions for the near future, making it suitable for tasks that require quick responses, such as immediate operational planning or short-term decision-making. The models are simple, making them easy to train and implement, and they often demonstrate relatively high accuracy. However, because the forecast range is short, it cannot capture long-term trends or complex variations, which limits its applicability.

Long-Term Time Series Forecasting (LTSF). Long-term time series forecasting deals with predictions for the distant future, with forecast horizons increasingly extending to several months, years, or beyond. It is valuable for long-term strategy planning, investment decisions, and policy-making, addressing many real-world problems. By identifying long-term trends and cycles, organizations can prepare accordingly, highlighting its significance. However, predicting the distant future is challenging, and extensive research is being conducted to improve accuracy.

2.3 Time Series Forecasting Datasets

Models for Time Series Forecasting (TSF) are primarily trained and evaluated using open benchmark datasets. Open datasets offer data that has already been collected across various fields, allowing researchers to use them without going through separate data collection processes. Additionally, researchers can objectively compare the performance of different models and convincingly demonstrate the contributions of their developed models.

Datasets frequently used by time series forecasting models in various domains, such as energy consumption, traffic, weather, exchange rates, and disease outbreaks, are listed in Table 2. These time series datasets are typically collected in real-time via sensors, recorded as transaction logs in financial markets, accumulated as server logs, or gathered through periodic surveys.

Table 2: Datasets Frequently Used in Time Series Forecasting Models

Dataset	Channel	Length	Description	Frequency	Source
ETTm1, ETTm2	7	69680	Electricity Transformer Temperature data from two counties in China for 2 years	15 mins	ETDataset
ETTh1, ETTh2	7	17420	Electricity Transformer Temperature data from two counties in China for 2 years	Hourly	ETDataset
Electricity	321	26304	Hourly electricity consumption data of clients from 2012 to 2014	Hourly	UCI Archive
Traffic	862	17544	Hourly road occupancy rates on San Francisco Bay area freeways for 48 months (2015-2016)	Hourly, Weekly	PeMS
Weather	21	52696	Meteorological data (e.g., humidity, temperature) in Germany	10 mins	BGC Jena
Exchange	8	7588	Daily exchange rates of eight countries from 1990 to 2016	Daily	Multivariate Time Series Data
ILI	7	966	Weekly Influenza-Like Illness data from the U.S. CDC	Weekly	CDC FluView

Time series data exist across a wide range of domains, each exhibiting unique characteristics. This diversity is due to the nature of time series data, which addresses real-world problems in various fields and can display highly varied patterns depending on time and environmental factors. For example, stock price data often show long-term upward or downward trends. Stock market fluctuations tend to move in specific directions over the long term due to complex influences such as economic conditions, corporate performance, and policy changes. In contrast, electricity consumption data exhibit pronounced periodicity throughout the day, with higher consumption during daytime hours and lower consumption at night. There are also seasonal patterns, with different consumption rates during summer and winter.

Monash Time Series Forecasting Archive (Godahewa et al, 2021l) points out that there is no comprehensive time series dataset archive and thus provides a thorough time series repository. This repository offers 30 datasets for real-world data and competition purposes, covering various domains such as energy, economics, nature, and sales. Details are specified in Table 3.

Time series data exhibit a greater variety of domain-specific characteristics compared to Natural Language Processing (NLP). These differences in dataset characteristics make time series forecasting problems more diverse and complex, reducing the generalizability of time series forecasting models. Consequently, developing foundation models for TSF is more challenging compared to NLP. Time series data lack the well-designed vocabulary and grammar found in NLP, and obtaining vast amounts of data is more difficult than in fields like Computer Vision(CV) or NLP. In the CV field, there are general benchmark datasets such as MNIST and ImageNet, while in NLP, there are datasets like GLUE(General Language Understanding Evaluation) and SQuAD(Stanford Question Answering Dataset). In contrast, the time series analysis field lacks large-scale general benchmark datasets and large-scale challenges like those in image and language domains. The most famous time series forecasting competition is the M-Competitions, which started in 1982 and has only been held six times to date. The paper "Time Series Dataset Survey for Forecasting with Deep Learning" (Hahn et al, 2023) highlights the lack of general benchmark datasets for TSF and analyzes the statistical characteristics of datasets across various domains, clustering them to provide easily accessible information for researchers. Recently, the development of foundation models for time series has begun, leading to the gradual emergence of general datasets for TSF. Details about TSF foundation models will be discussed in Section 4.3.

Table 3: Summary of Datasets Across Domains from Monash Archive (Godahewa et al, 2021l)

Dataset	Domain	No. Series	Min. Length	Max. Length	Competition	Multivariate	Source
M1	Multiple	1001	15	150	Yes	No	Makridakis et al (1982)
M3	Multiple	3003	20	144	Yes	No	Makridakis and Hibon (2000)
M4	Multiple	100000	19	9933	Yes	No	Makridakis et al (2020)
Tourism	Tourism	1311	11	333	Yes	No	Athanasopoulos et al (2011)
CIF 2016	Banking	72	34	120	Yes	No	Štěpnička and Burda (2017)
London Smart Meters	Energy	5560	288	39648	No	No	Jean-Michel, 2019
Aus. Electricity Demand	Energy	5	230736	232272	No	No	Godahewa et al (2021e)
Wind Farms	Energy	339	6345	527040	No	No	Godahewa et al (2021b,c)
Dominick	Sales	115704	28	393	No	No	Center (2020)
Bitcoin	Economic	18	2659	4581	No	No	Godahewa et al (2021f,g)
Pedestrian Counts	Transport	66	576	96424	No	No	City of Melbourne, 2020
Vehicle Trips	Transport	329	70	243	No	No	fivethirtyeight, 2015
KDD Cup 2018	Nature	270	9504	10920	Yes	No	KDD Cup, 2018
Weather	Nature	3010	1332	65981	No	No	Sparks et al (2020)
NN5	Banking	111	791	791	Yes	Yes	Taieb et al (2012)
Web Traffic	Web	145063	803	803	Yes	Yes	Google, 2017
Solar	Energy	137	52560	52560	No	Yes	Solar, 2020
Electricity	Energy	321	26304	26304	No	Yes	UCI, 2020
Car Parts	Sales	2674	51	51	No	Yes	Hyndman, 2015
FRED-MD	Economic	107	728	728	No	Yes	McCracken and Ng (2016)
San Francisco Traffic	Transport	862	17544	17544	No	Yes	Caltrans, 2020
Rideshare	Transport	2304	541	541	No	Yes	Godahewa et al (2021h,i)
Hospital	Health	767	84	84	No	Yes	Hyndman, 2015
COVID Deaths	Nature	266	212	212	No	Yes	Johns Hopkins University, 2020
Temperature Rain	Nature	32072	725	725	No	Yes	Godahewa et al (2021j,k)
Sunspot	Nature	1	73931	73931	No	No	Sunspot, 2015
Saugeen River Flow	Nature	1	23741	23741	No	No	McLeod and Gweon, 2013
US Births	Nature	1	7305	7305	No	No	McLeod and Gweon (2013)
Solar Power	Energy	1	7397222	7397222	No	No	Godahewa et al (2021a)
Wind Power	Energy	1	7397147	7397147	No	No	Godahewa et al (2021d)

2.4 Evaluation Metrics

The evaluation metrics used in TSF play a crucial role in objectively comparing and assessing model performance. Common metrics for deterministic models include Mean Absolute Error (MAE) and Mean Squared Error (MSE), which are easy to use because they allow intuitive comparisons of prediction errors. On the other hand, probabilistic models that can reflect uncertainty typically use Continuous Ranked Probability Score (CRPS), which measures the accuracy of distribution forecasts by evaluating the difference between predicted and actual distributions. However, the limitations of these traditional evaluation metrics have been noted, leading to the proposal of various performance metrics.

The distinct characteristics of time series data necessitate the use of diverse metrics in TSF. It is essential to assess not only the magnitude of errors but also how well the model learns the temporal patterns and various features of time series data. Table 4 categorizes evaluation metrics by type and explains each metric. Each metric assesses different characteristics of the model, helping to comprehensively understand model performance in various scenarios.

For instance, each data point in time series data has temporal dependencies, and it is necessary to evaluate how well the model learns these dependencies. Therefore, beyond simply measuring the magnitude of prediction errors with metrics like MAE or MSE, it is essential to evaluate how these errors change over time. Additionally, assessing relative errors is necessary to determine whether the model accurately captures seasonal or periodic patterns, making metrics like Mean Absolute Percentage Error (MAPE) or Symmetric Mean Absolute Percentage Error (sMAPE) necessary. Sometimes, metrics like R^2, which explains the variability of the entire dataset, are useful for examining long-term trends of increase or decrease in time series data. The continuous and irregular nature of time series data can mean large errors may occur at certain points, and metrics like Root Mean Squared Error (RMSE) are sensitive to large errors, allowing evaluation of the model's ability to handle significant mistakes. To assess the tendency for prediction errors to increase over time, metrics such as Mean Forecast Error (MFE) or Cumulative Forecast Error (CFE) can be used to evaluate the accuracy of future predictions. Despite the availability of a wide range of metrics, MSE and MAE remain the most widely used due to their simplicity, intuitiveness, and widespread use in existing research, making them effective for objective performance comparisons.

3 Historical TSF Models

3.1 Conventional Methods (Before Deep Learning)

3.1.1 Statistical Models

Prior to machine learning and deep learning, traditional statistical models, which laid the foundation for analyzing sequential data, were commonly utilized for time series forecasting. Exponential smoothing was introduced by Brown (1959), Holt (1957), and Winters (1960) as a method to forecast future values using a weighted average of past data. This method operates by computing an exponential weight decay of historical data, assigning greater weight to more recent data. The Autoregressive Integrated Moving Average (ARIMA) model was formalized through the work of George Box and Gwilym Jenkins in their book "Time Series Analysis: Forecasting and Control (Bartholomew, 1971)." The ARIMA model predicts future values by leveraging the autocorrelation in data and is composed of three main components. AutoRegressive (AR) uses a linear combination of past values to predict the current value. Moving Average (MA) employs a linear combination of past error terms to predict the current value. Integrated (I) removes non-stationarity (the property where mean and variance change over time) by differencing the data to achieve stationarity. While exponential smoothing models are advantageous for real-time data analysis due to their simplicity, ARIMA models are better suited for capturing complex patterns, making them ideal for long-term forecasting. The SARIMA model extends ARIMA by incorporating seasonal differencing, along with seasonal autoregressive and moving average components, allowing it to effectively model and predict data with regular cyclical patterns. These statistical models are based on specific assumptions and are simple, intuitive, and useful for identifying basic patterns in data.

Table 4: Evaluation Metrics for Time Series Forecasting

Evaluation Type	Category	Metric	Formula	Explanation
Deterministic	Error-Based Metrics	Mean Absolute Error (MAE)	$\frac{1}{n}\sum_{t=1}^{n} \lvert y_t - \hat{y}_t \rvert$	Measures the average magnitude of the errors in a set of predictions, without considering their direction.
		Mean Squared Error (MSE)	$\frac{1}{n}\sum_{t=1}^{n} (y_t - \hat{y}_t)^2$	Measures the average of the squared errors.
		Root Mean Squared Error (RMSE)	$\sqrt{\frac{1}{n}\sum_{t=1}^{n} (y_t - \hat{y}_t)^2}$	Square root of the average of squared errors, giving higher weight to larger errors.
		Mean Absolute Percentage Error (MAPE)	$\frac{1}{n}\sum_{t=1}^{n} \left\lvert \frac{y_t - \hat{y}_t}{y_t} \right\rvert \times 100$	Measures the size of the error in percentage terms.
		Symmetric Mean Absolute Percentage Error (sMAPE)	$\frac{1}{n}\sum_{t=1}^{n} \frac{\lvert y_t - \hat{y}_t \rvert}{(\lvert y_t \rvert + \lvert \hat{y}_t \rvert)/2} \times 100$	Measures the accuracy based on relative error.
		Mean Forecast Error (MFE)	$\frac{1}{n}\sum_{t=1}^{n} (y_t - \hat{y}_t)$	Average of forecast errors, indicating bias.
		Cumulative Forecast Error (CFE)	$\sum_{t=1}^{n} (y_t - \hat{y}_t)$	Sum of all forecast errors, measures total bias over the forecast horizon.
	Scaled Error Metrics	Mean Absolute Scaled Error (MASE)	$\frac{\frac{1}{n}\sum_{t=1}^{n} \lvert y_t - \hat{y}_t \rvert}{\frac{1}{n-1}\sum_{t=2}^{n} \lvert y_t - y_{t-1} \rvert}$	MAE scaled by the MAE of a naive forecast.
	Explained Variance Metrics	Coefficient of Determination (R²)	$1 - \frac{\sum_{t=1}^{n} (y_t - \hat{y}_t)^2}{\sum_{t=1}^{n} (y_t - \bar{y})^2}$	Proportion of variance explained by the model.
		Adjusted Coefficient of Determination (Adjusted R²)	$1 - \frac{(1-R^2)(n-1)}{n-k-1}$	R² adjusted for the number of predictors.
		Explained Variance Score (EVS)	$1 - \frac{\mathrm{Var}(y_t - \hat{y}_t)}{\mathrm{Var}(y_t)}$	Measures the proportion of variance explained by the model.
	Model Selection Metrics	Akaike Information Criterion (AIC)	$2k - 2\ln(\hat{L})$	Trade-off between goodness of fit and model complexity.
		Bayesian Information Criterion (BIC)	$k\ln(n) - 2\ln(\hat{L})$	Similar to AIC with a stronger penalty for models with more parameters.
		Hannan-Quinn Criterion (HQC)	$\mathrm{AICc} = 2k - 2\ln(\hat{L}) + \frac{2k(k+1)}{n-k-1}$	Alternative to AIC and BIC with different penalty terms.
		Corrected Akaike Information Criterion (AICc)	$AIC + \frac{2k(k+1)}{n-k-1}$	AIC with correction for small sample sizes.
Probabilistic	Error-Based Metrics	Logarithmic Score (Log Score)	$-\frac{1}{n}\sum_{t=1}^{n} \log(p_t)$	Evaluates the difference between predicted probabilities and actual outcomes using a logarithmic function.
		Continuous Ranked Probability Score (CRPS)	$\int_{-\infty}^{\infty} (\hat{F}(z) - \mathbf{1}(z \geq y_t))^2 dz$	Evaluates the difference between the predicted probability distribution and the observed value using the cumulative distribution function.
	Interval Metrics	Prediction Interval Coverage Probability (PICP)	$\frac{1}{n}\sum_{t=1}^{n} I\left(y_t \in [\hat{y}_{\mathrm{lower},t}, \hat{y}_{\mathrm{upper},t}]\right)$	Measures the proportion of observed values that fall within the predicted intervals.
		Prediction Interval Width (PIW)	$\frac{1}{n}\sum_{t=1}^{n} \left(\hat{y}_{\mathrm{upper},t} - \hat{y}_{\mathrm{lower},t}\right)$	Evaluates precision by measuring the width of prediction intervals.
	Quantile Metrics	Quantile Loss (Pinball Loss)	$\frac{1}{n}\sum_{t=1}^{n} \begin{cases} \tau \cdot (y_t - \hat{y}_t) & \text{if } y_t \geq \hat{y}_t \\ (1-\tau) \cdot (\hat{y}_t - y_t) & \text{if } y_t < \hat{y}_t \end{cases}$	Penalizes over- and under-predictions based on quantile τ.
	Sharpness Metrics	Sharpness	$\frac{1}{n}\sum_{i=1}^{n} \mathrm{Var}(\hat{y}_i)$	Evaluates the concentration by the width of intervals or variance.

3.1.2 Machine Learning Models

While statistical methods struggle to capture complex patterns in time series data due to their reliance on predefined linear relationships, machine learning models excel in learning nonlinear patterns directly from the data without relying on such assumptions. To address the limitations of statistical methods, traditional machine learning models have been increasingly applied to time series forecasting. Decision Trees (Quinlan, 1986) are machine learning models that use a tree structure for classification or prediction. The term "classification and regression tree (CART) analysis" was first introduced by Barlin et al (2013), and in time series forecasting, regression decision trees are used to split data into a tree structure to predict continuous values. While they are intuitive and easy to interpret, they are prone to overfitting. Support Vector Machines (SVM), introduced by Cortes and Vapnik (Cortes, 1995), are supervised learning models used for classification and regression analysis, characterized by finding the maximum margin. They handle high-dimensional data and non-linearities effectively, making them robust for classification and regression tasks. Support Vector Regression (SVR) applies SVM concepts to regression problems, predicting data points using an optimal hyperplane (Vapnik et al, 1996). This hyperplane is trained to minimize errors between data points, ensuring errors do not exceed a certain threshold. Gradient Boosting Machines (GBM) were developed in 1999 with an explicit regression gradient boosting algorithm. This method uses ensemble learning to combine multiple weak models into a single, strong predictive model (Friedman, 2001). This method iteratively improves the model, offering high predictive performance. XGBoost, an extension of the Gradient Boosting algorithm (Chen and Guestrin, 2016), incorporates various optimization techniques to enhance efficiency and performance. It gained significant attention for its outstanding performance in machine learning competitions, particularly in time series forecasting challenges. These machine learning models outperform traditional statistical models in capturing data structures and patterns, demonstrating high predictive power on large datasets (Kontopoulou et al, 2023). Their ability to automatically learn and optimize various data features and relationships has led to their widespread use in time series forecasting.

3.2 Traditional Deep Learning Models

3.2.1 MLPs: The Emergence and Constraints of Early Artificial Neural Networks

The development of the **Multi-layer Perceptron (MLP)** (Rumelhart et al, 1986) and the backpropagation algorithm established the foundation for artificial neural networks. Early artificial neural network models utilizing these MLPs demonstrated strong performance in modeling nonlinear patterns, prompting numerous attempts to apply them to time series data processing. However, several key limitations, as outlined below, restricted the training of deep neural networks using MLPs.

- **Limited in learning temporal dependencies**: MLPs are well-suited for processing fixed-length input vectors but struggle to adequately capture the temporal dependencies in time series data.
- **Vanishing gradient issue**: The vanishing gradient problem in deep networks made training difficult.
- **Lack of data and computing resources**: The scarcity of large-scale datasets and high-performance computing resources made it challenging to effectively train complex neural network models.

At that time, artificial neural network technology was still immature, and there was a lack of deep understanding and effective methodologies for dealing with time series data. Consequently, traditional statistical methods and machine learning models previously discussed continued to be widely used for time series analysis.

3.2.2 RNNs: The first neural network capable of processing sequential data and modeling temporal dependencies

The Emergence and Early Applications of RNNs

Recurrent Neural Networks (RNNs) (Hopfield, 1982) emerged, opening up new possibilities for processing time series data. RNNs are specialized models designed to process sequential data, such as time series data, and were utilized to overcome the limitations of MLPs. The structure of an RNN consists of an input layer, a hidden layer, and an output layer, and it uses the output from previous time steps(hidden layer output) as the input for the current time step. This structural feature allows RNNs to naturally model the dependencies in data over time. The hidden state preserves past information by storing data from the previous time step and passing it on to the next time step, thereby maintaining historical context. Additionally, it is parametrically efficient because the same weights are used for calculations across all time steps, regardless of the sequence length.

Due to these features, RNNs have been utilized in various fields where sequential data modeling is crucial, such as time series data analysis, natural language processing (NLP), and so on. However, early RNNs had the following issues.

- **Vanishing gradient problem**: When RNNs learn long sequences, the gradient gradually diminishes during the backpropagation process, causing information from earlier time steps to fail to reach later time steps. Thus, this made it difficult to learn long-term dependencies.
- **Exploding gradient problem**: In contrast to the vanishing gradient problem, the gradient values could become excessively large, making the training process unstable.
- **Difficulty in parallel processing**: Due to the sequential nature of RNNs, where the computation at each time step depends on the previous one, parallel processing becomes challenging. This leads to high computational costs and slow learning speeds.

Because of these critical drawbacks, RNNs were not widely used until the early 2000s.

Overcoming the Limitations of RNNs

To address the aforementioned issues with vanilla RNNs, researchers began developing various models. Notably, the **Long Short-Term Memory (LSTM)** (Hochreiter and Schmidhuber, 1997b) model was developed to address the long-term dependency issues of vanilla RNNs. With the cell state and gate mechanisms (input, output, and forget gates), LSTMs preserves crucial information over extended periods and the discards unnecessary information. The **Gated Recurrent Unit (GRU)** (Cho, 2014) emerged as a simpler alternative to LSTM, offering similar performance while using a more straightforward structure(update and reset gates) to manage information. These two models effectively addressed the vanishing gradient problem and sparked a boom in RNN-based models for time series analysis. Many subsequent RNN-based models used these two as their backbone, making them powerful tools for handling time series data until the advent of Transformers.

Notable RNN Variants

The **Dilated RNN** (Chang et al, 2017) proposed Dilated Recurrent Skip Connections as a solution to various issues inherent in traditional vanilla RNNs. This approach enhanced parameter efficiency in long sequences and enabled the network to effectively learn complex dependencies across diverse temporal scales. The **DA-RNN** (Qin et al, 2017) attempted to effectively address long-term dependencies, which were challenging for existing methods, by employing a dual-stage attention mechanism. Utilizing an encoder with Input Attention and a decoder with Temporal Attention, it adaptively selected salient information from input features and temporal steps, thereby enhancing the accuracy and performance of time series predictions. Additionally, the **MQ-RNN** (Wen et al, 2017) combined the Sequence-to-Sequence neural network framework with Quantile Regression to provide probabilistic forecasts, thereby reflecting uncertainties across various scenarios. This model simultaneously generated multiple forecast horizons at each time step and employed the Forking-Sequences training method to enhance the stability and performance of the RNN.

3.2.3 CNNs: Extracting key patterns in time series data beyond just images

The Emergence and Early Applications of CNNs

The **Neocognitron** (Fukushima, 1980) was designed for visual pattern recognition and served as an early form of Convolutional Neural Networks (CNNs), introducing the fundamental concepts of CNNs. The CNN architecture, which became more widely known with the development of **LeNet** (LeCun et al, 1998), learned spatial patterns in images through a combination of convolutional and pooling layers. These features made CNNs well-suited for handling image data, leading early CNN-based models to primarily focus on image data without being directly applied to time series data. However, over time, their potential for handling time series data has also been recognized.

Attempts to Apply CNNs to Time Series Data

1D CNNs use one-dimensional convolutional filters to learn local patterns in time series data, allowing them to extract features that consider the temporal structure of the data. A prominent example is **WaveNet** (Van Den Oord et al, 2016a), which utilized 1D convolutions and dilated convolutions to model speech signals, effectively capturing long-term dependencies in audio signals. WaveNet demonstrated the utility of CNN-based models in speech synthesis and time series prediction, leading to the widespread adoption of CNN-based models for time series analysis. The development of the **Temporal Convolutional Networks (TCNs)** (Bai et al, 2018) model further highlighted the potential of CNN-based models in the time series domain. TCN consists of multiple layers of 1D convolutional networks, using dilated convolutions at each layer to achieve a wide receptive field, enabling the construction of deeper networks. This allows TCNs to effectively learn from long sequences of data. TCNs have demonstrated excellent performance in various sequential domains, including time series prediction, signal processing, and natural language processing.

CNN and RNN Ensemble Models

Building on the popularity of traditional time series forecasting models like RNNs and the emerging interest in CNN models, hybrid CNN-RNN models began to emerge, combining the strengths of both architectures. Models that combined CNNs and LSTMs were particularly advantageous for simultaneously learning local patterns and long-term dependencies in time series data. The structure involved using CNNs to extract complex features from the time series data and LSTMs to learn temporal dependencies.

DeepAR (Salinas et al, 2020) combined LSTM and CNN to generate probabilistic forecasts of future distributions rather than single predicted values. Additionally, it improved generalization performance by simultaneously learning and predicting multiple time series within a single model. **DCRNN** (Li et al, 2017) modeled traffic flow as a diffusion process on a directed graph, capturing both temporal and spatial dependencies. The Diffusion Convolution leveraged bidirectional random walks on the graph to capture spatial dependencies within the traffic network, while the Gated Recurrent Units (GRU) modeled temporal dependencies. The introduction of the Diffusion Convolutional Gated Recurrent Unit (DCGRU) enabled the effective processing of temporal information. the **TPA-LSTM** (Shih et al, 2019) combined CNN and LSTM with an added attention mechanism, introducing the Temporal Pattern Attention(TPA) methodology to learn important patterns in time series data. Local patterns were extracted using CNN, while the combination of LSTM and the attention mechanism captured significant temporal patterns.

3.2.4 GNNs: Structurally modeling relationships between variables

The Emergence and Slow Growth of GNN-Based Models

Graph Neural Networks (GNNs) (Scarselli et al, 2008) were primarily developed to process graph-structured data because they can effectively model the complex structural relationships between nodes and edges in graph data. Over time, GNNs gradually began to be applied to time series data analysis, primarily because multivariate time-series data often encapsulates intricate structural relationships.

During their initial development, GNNs were less popular compared to widely used RNN or CNN-based models. The prevailing belief was that GNNs were not suitable for time series analysis compared to CNNs which are effective at extracting local patterns, and RNNs which are strong at learning long-term dependencies. However, as the range and complexity of time-series data increased, understanding their structural characteristics became more important. Consequently, GNNs began to gain prominence in fields such as traffic prediction and social networks. Research on applying GNNs to time-series data has primarily focused on capturing the dynamic characteristics of graph data, and the structural learning capabilities of GNNs have proven to be highly useful in capturing the complex patterns within time-series data.

Development and Expansion of GNN Applications

Until the development of Transformers, GNNs consistently garnered interest, showing consistent yet slow growth compared to RNNs and CNNs. The development of **Graph Convolutional Networks (GCNs)** (Kipf and Welling, 2016) marked another turning point for GNNs. GCNs learns node features through convolution operations on graph structures, and are powerful tools for processing graph data. This advancement positioned GCNs as a versatile solution for addressing a wide range of graph-based challenges.

GCNs were more effective at learning the relationships between variables in time series and temporal locality, and subsequently, GNNs began to be widely used for time-series data. **Spatial-Temporal Graph Convolutional Networks (ST-GCN)** (Yan et al, 2018) were developed as a model combining GCNs and RNNs to learn complex patterns in spatio-temporal data. By leveraging GCNs to learn spatial patterns and RNNs to capture temporal patterns effectively, ST-GCN demonstrated the potential of GNNs in time-series problems, such as traffic prediction. **Graph Attention Networks (GATs)** (Veličković et al, 2017) were introduced, learning relationships between nodes using an attention mechanism. This allowed GATs to capture the importance of nodes, making them advantageous for identifying significant events in time-series data. **Dynamic Graph Neural Networks (DyGNNs)** (Ma et al, 2020) were developed to learn graph structures that change over time. Designed to reflect the dynamic characteristics of time-series data, DyGNNs could effectively capture complex temporal dependencies and structural relationships by allowing the graph structure to evolve over time. **Temporal Graph Networks (TGNs)** (Rossi et al, 2020) were introduced as a model in which the nodes of a graph change over time, enabling the learning of temporal patterns in time-series data. By using a memory module to track the temporal state changes of nodes, TGNs learn the information of neighboring nodes whose importance varies over time, thus providing a basis for predicting future states.

After the era of traditional statistical techniques and machine learning methodologies, traditional deep learning approaches such as RNNs, CNNs, and GNNs were widely used for time-series analysis and prediction over a long period. However, the advent of Transformers brought a revolutionary change to the AI field. Upon their introduction, Transformers outperformed all existing methods across various domains and continue to dominate as the leading foundation model. In the time-series domain, a similar trend was observed. Beginning with the introduction of the **Logsparse Transformer** (Li et al, 2019), numerous transformer-based models emerged, achieving state-of-the-art performance. As a result, traditional deep learning methods experienced a period akin to the Ice Age. The remarkable historical TSF models mentioned above are organized chronologically in Fig. 8.

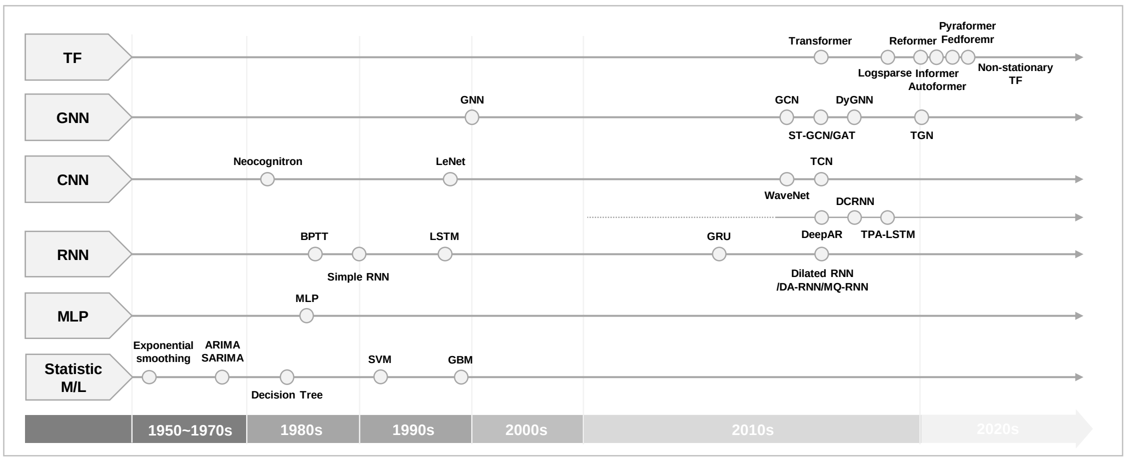


Fig. 8: Remarkable Historical TSF Models

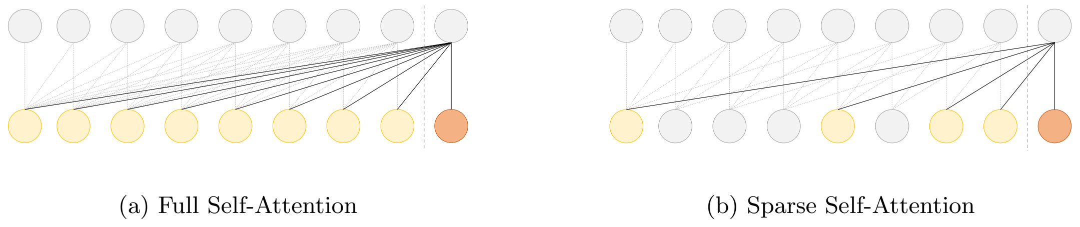
3.3 The Prominence of Transformer-based Models

Transformer (Vaswani et al, 2017), introduced by the Google Brain team, is an innovative model designed to perform complex sequence-to-sequence tasks in natural language processing. This model features an encoder-decoder structure and utilizes an attention mechanism to capture the relationships between tokens in the input sequence. As a result, Transformers have gained significant attention for replacing traditional RNN-based models, offering parallel processing capabilities and effectively addressing long-term dependency issues. Transformers' success with sequential data naturally led to their extension into time series applications.

This section explores various Transformer variants that address the limitations of the original Transformer for time series forecasting tasks. It discusses the background of these variants, how they improve upon the original Transformer, and the reasons for their enhanced performance in time series prediction. Additionally, it addresses the remaining limitations of these variant models.

3.3.1 Transformer Variants

In recent years, substantial research has focused on transformer-based time series analysis, particularly long-term time series forecasting (LTSF), which has led to the development of various models (Zeng et al, 2023a). The original Transformer model has several limitations when applied to long-term time series forecasting (TSF). These limitations include quadratic time and memory complexity, as well as error accumulation caused by the auto-regressive decoder design. Specifically, the time and memory complexity of self-attention increases quadratically with the length of the input sequence. This high complexity can become a computational bottleneck, especially for time series forecasting tasks that rely on long-term historical information. To address these issues, Transformer variants primarily focus on reducing time and memory complexity.


(a) Full Self-Attention

(b) Sparse Self-Attention

Fig. 9: Comparison of Full and Sparse Attention Mechanisms in Transformer

LogTrans (Li et al, 2019) leverages the efficiency of Log Sparse attention mechanisms to reduce time and memory complexity to $O(LlogL)$, making it well-suited for long sequence forecasting tasks. This model assumes recent data is more relevant than older data, utilizing positional sparsity to focus attention only on relevant data. **Reformer** (Kitaev et al, 2020) enhances efficiency through locality-sensitive hashing and reversible residual layers, reducing memory and computational requirements to $O(LlogL)$ while maintaining performance on long sequences. Locality-sensitive hashing (LSH) approximates attention by clustering similar items and performing attention partially on these clusters. **Informer** (Zhou et al, 2021) improves computational and memory efficiency to $O(LlogL)$ by utilizing a ProbSparse self-attention mechanism and a generative-style decoder. It effectively handles long sequences and captures dependencies across various time scales. Informer addresses the limitations of Reformer, which performs well only on extremely long sequences, and LogTrans, which uses heuristic positional-based attention. ProbSparse self-attention features dynamic query selection based on query sparsity measurement. **Autoformer** (Wu et al, 2021) integrates time series decomposition into the internal blocks of the Transformer and utilizes an auto-correlation mechanism to aggregate similar sub-series. Autoformer improves upon previous methods by addressing point-wise dependency and aggregation issues. It employs a decomposition architecture to handle complex temporal patterns by separating seasonal and trend components. **Pyraformer** (Liu et al, 2021a) introduces a novel pyramid attention mechanism designed to efficiently model long-term dependencies in time series data. By using a hierarchical structure to progressively reduce sequence length, Pyraformer significantly lowers computational complexity compared to traditional Transformer models. **Fedformer** (Zhou et al, 2022) combines Fourier-enhanced decomposition blocks with frequency domain sparse representations, integrating seasonal-trend decomposition from Autoformer and offering a more effective representation of time series data. The sparse representation basis can be Fourier-based or Wavelet-based, encompassing a subset of frequency components, including both high and low frequencies. Additionally, attempts have been made to overcome the limitations of normalized Transformers, which often produce indistinguishable attention patterns and fail to capture significant temporal dependencies. **Non-stationary Transformer** (Liu et al, 2022a) identifies this issue as excessive normalization and addresses it with De-stationary Attention, which allows for the use of non-normalized data during QKV computations.

3.3.2 Limitation of Transformer-based Models

However, even the models designed specifically for time series forecasting based on the original Transformer still have the following limitations.

Efficiency Challenges

Efforts to overcome the primary drawback of quadratic scaling in computational and memory complexity with window length have not fully resolved the issues. Various approaches to reduce this complexity have been proposed, but the attention mechanism inherently incurs higher computational and memory costs compared to MLP-based or convolutional models with $O(L)$ complexity. Applying Transformers to the time domain does not easily mitigate this problem. Models such as **Fedformer** (Zhou et al, 2022), which uses Fourier or Wavelet transformations to operate in the frequency domain, have been made to address these issues. Although research into more efficient attention variants is ongoing (Tay et al, 2020), these solutions often sacrifice some of the effective characteristics of Transformers. Sparse attention mechanisms, used to reduce Self-Attention complexity, may result in the omission of important information. Models like LogTrans and Pyraformer introduce explicit sparsity biases in the attention mechanism but may suffer from significant performance degradation due to the loss of crucial information. None of these variants have yet been proven effective across diverse domains (Gu and Dao, 2023).

Finite Context Window

The efficiency of self-attention is attributed to its ability to densely route information within a context window, enabling the modeling of complex data. However, unlike models such as RNNs or State Space Models (SSMs), Transformers have a fundamental limitation in modeling data beyond a finite window.

Ineffectiveness of Expanding Input Window Length

Another significant issue is the minimal or no performance improvement observed when increasing the input window length. Strong time series forecasting (TSF) models are generally expected to achieve better results with larger look-back window sizes due to their robust temporal relationship extraction capabilities. However, research (Zeng et al, 2023b; Zhou et al, 2021; Wen et al, 2022b) indicates that the performance of Transformers either deteriorates or remains stable as the look-back window increases, in contrast to improvements seen with linear-based methods (Zeng et al, 2023b). This suggests that transformer-based models tend to overfit noise rather than extract long-term temporal information when provided with longer sequences.

In conclusion, while transformer-based models have made significant advancements in time series forecasting, they still face limitations. Therefore, future research will continue to address these limitations and will also serve as a catalyst for re-exploring various alternative architectures.

3.4 Uprising of Non-Transformer-based Models

As previously discussed, transformer-based models demonstrate strong performance in processing time series data. However, they have several limitations when dealing with long-term time series data.

- The point-wise operations of the self-attention mechanism have quadratic time complexity. Therefore, as sequence length increases, the computational load grows exponentially, making it challenging to handle long-term historical information.
- Because storing the relationship information for all input token pairs requires substantial memory usage, applying Transformers in environments with limited GPU memory becomes challenging.
- When the length of the look-back window exceeds the structural capacity, learning long-term dependencies becomes challenging.
- Due to the high complexity of the model, large-scale and high-quality datasets are required. In the absence of sufficient data, overfitting can occur leading to a drop in model performance.

Contemporary TSF tasks increasingly require the prediction of diverse multivariates and long sequences. Unlike earlier tasks that involved relatively short sequences, the limitations of transformer-based models have become more pronounced when dealing with long sequences. Researchers adhering to the Transformer's philosophy have begun focusing intensively on two aspects to address these issues: reducing computational complexity and enhancing long-term dependency learning. Conversely, some researchers have reverted to non-transformer methodologies, renewing interest in traditional deep learning models such as RNNs, CNNs, GNNs, and MLPs. Non-transformer-based models have the potential to overcome the limitations of Transformers, and research has begun to leverage these strengths to move beyond the hegemony of the Transformer backbone.

RNN-based models sequentially process input data, retaining the previous state at each time step to predict the next state. This enables the natural modeling of long-term dependencies and helps overcome the limitations of Transformers by leveraging time-ordering learning. They require relatively less data and memory, making them less restrictive in thier application. Additionally, their ability to handle real-time streaming data enhances their versatility across various applications.

Convolution operations in **CNN-based** models offer lower time complexity and reduced memory requirements compared to the self-attention mechanism in Transformers. Since time series data often exhibit periodic trends and seasonality, the ability of convolution to extract local patterns complements Transformers' specialization in learning global patterns, addressing their shortcomings.

GNN-based models can effectively represent and learn the complex relationships and structures of time series data through nodes and edges. They integrate localities efficiently through local operations, resulting in lower memory usage. Additionally, their ability to model both spatial and temporal relationships simultaneously makes them advantageous for handling multivariate datasets.

The primary advantage of **MLP-based** models lies in their simple structure, which facilitates efficient training, fast inference, and straightforward handling. Current AI models require lightweight designs suitable for mobile applications, and MLP-based models are well-suited for environments with many constraints.

Especially, the counterattack of simple linear models based on MLPs is one of the notable trend shifts. The emergence of **LTSF-Linear** (Zeng et al, 2023a) models which surpass transformer-based models using only simple linear layers has triggered this change. One of the key characteristics of time series data is that the time ordering of data points holds important meaning. Conversely, the permutation-invariant nature of the Transformer's attention mechanism, which emphasizes relationships between data points, is argued to be inherently unsuitable for time series data. Although positional embedding attempts to preserve time order, the multi-head attention process inevitably leads to information loss (Zhou et al, 2021).

LTSF-Linear models highlighted the critical limitations of Transformers in handling long sequences, demonstrating that simple linear models could better preserve time-ordering information and extract features related to trends and seasonality.

This claim gained further traction through subsequent studies, leading to a more diversified interest in backbone models across various architectures. The emergence of this model sparked skepticism regarding the performance of Transformer-based models and served as a catalyst for the exploration of various alternative architectures. The era of Transformer prevalence has come to an end, ushering in a renaissance in time series research, characterized by a reevaluation of traditional DNNs, the emergence of novel models like Mamba, and new approaches such as the utilization of LMMs.

4 New Exploration of TSF Models

4.1 Overcoming Limitations of Transformer

As previously mentioned in 3.4, the performance of simple linear models in **LTSF-Linear** (Zeng et al, 2023a) surpassing traditional transformer-based models has raised doubts about the effectiveness of Transformers. However, in the field of NLP, Transformers still demonstrate superior performance in handling long-term dependencies in sequential data compared to other models (Patwardhan et al, 2023). This observation suggests that while Transformers have great potential, researchers have not fully leveraged their capabilities in time series analysis.

Therefore, various methods have emerged in time series forecasting to overcome the limitations of existing Transformer-based models. This section categorizes and explains in detail the specific limitations of existing models and how these limitations are addressed. The structure begins with an introduction to the patching technique, followed by the use of cross-dimension and exogenous variables, and then provides a detailed explanation of other approaches, concluding with a summary of key points in Table 5.

4.1.1 Patching Technique

Transformers were originally developed for natural language processing (NLP), and applying them to time-series analysis requires appropriate adjustments to fit the domain of time series. Different from the rich semantic information carried by individual word tokens in NLP, individual data points in time series are often similar to their neighboring values, lacking substantial information (Nie et al, 2022). Therefore, the point-wise attention mechanism of traditional models fails to consider the broader context or patterns that may span multiple consecutive time steps and only calculates attention for individual time steps, which makes it difficult to capture the characteristics of time series data. For this reason, considering the surrounding context along with a single time point provides more information in time series data.

Patching refers to the technique of dividing input sequences into multiple patches, as illustrated in Fig. 10. This method preserves the information within each patch, thereby enhancing locality. By processing patches instead of individual points, the model processes fewer tokens, thereby reducing the computational complexity of the attention mechanism. This approach helps overcome the issue of prediction performance degradation, which can occur when sparse attention is used to make self-attention more efficient, potentially missing critical information.

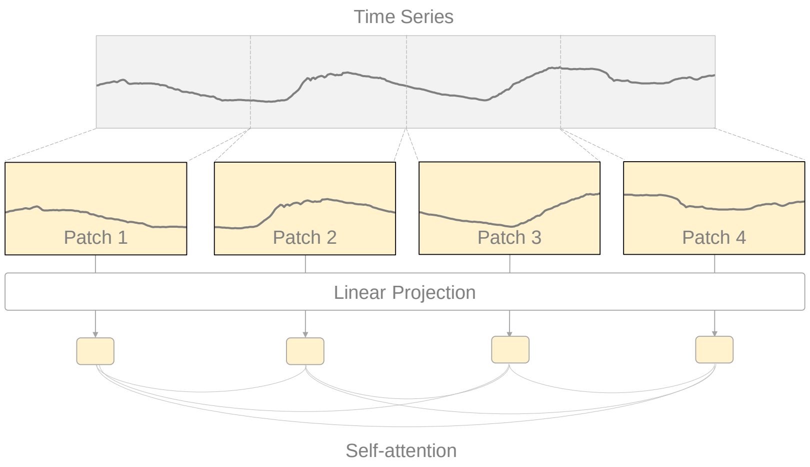


Fig. 10: Patching Technique in Self-Attention for Time Series Forecasting

PatchTST (Nie et al, 2022), inspired by Vision Transformer's (Dosovitskiy et al, 2020) division of images into 16×16 patches for capturing local semantic information, divides time series into 64 patches. This approach allows for utilizing a longer look-back window by grouping data into patches. By processing time-series data in patch units, PatchTST maximizes the advantages of Transformer models for time-series applications, achieving better performance than LTSF-Linear. PatchTST employs only the Transformer encoder, flattens the encoder output, and uses a linear layer for the final predictions. Additionally, the study shows that channel independence (CI) yields better performance, highlighting the limitations in learning channel correlations despite the intuitive consideration of channel dependencies (CD) in multivariate scenarios. **MTST** (Zhang et al, 2024d) addresses the limitation of PatchTST in learning patterns across different scales present in time-series data. By adopting a multi-scale approach, MTST proposes an effective model utilizing both shorter and longer patches for locality and long-term trend analysis. **PETformer** (Lin et al, 2023) critiques the flattening of Transformer encoder output in PatchTST, which results in a significant increase in parameters. PETformer introduces a placeholder-enhanced technique, modeling past and future data on the same time scale, thereby reducing the number of parameters by over 95% compared to PatchTST. This reduction enhances generalization performance while using less memory and computational resources. The model also leverages rich context information by allowing direct interaction between past and future data, maintaining the continuity of time-series data.

By applying the patching technique, it was possible to refute the notion that traditional linear models are superior to Transformers. This led to the patching technique becoming a widely adopted approach.

4.1.2 Cross-Dimension

In multivariate time-series forecasting, understanding the relationships between variables is crucial for improving prediction accuracy. Intuitively, higher temperatures lead to increased ice cream sales, indicating a relationship between variables. Despite this apparent correlation, surprising results have shown that models treating channels independently, such as LTSF-Linear, PatchTST, and PETformer, often outperform those considering inter-channel correlations. This result implies that current models fail to effectively capture the relationships between variables. Time series analysis differs significantly from natural language processing (NLP) and computer vision (CV) in terms of channel correlation. In NLP, there is no clear concept of channels. In CV, although channels exist, their relationships are tightly intertwined and well-defined, as seen in the RGB representation of images.

Conversely, in time series analysis, channel relationships can be either independent or interdependent and often hidden, adding complexity to the task. Therefore, time series analysis requires models capable of capturing these intricate correlations. While earlier Transformer-based models primarily focused on temporal attention, recent models have increasingly focused on explicitly modeling the correlations between variables.

Fig. 11 illustrates the explicit modeling of relationships between variables using the attention mechanism.

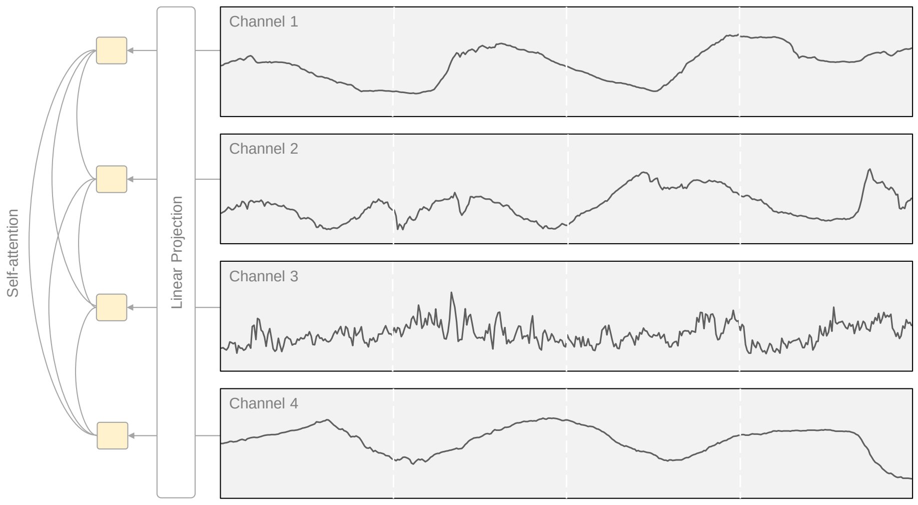


Fig. 11: Cross-Dimensional Self-Attention for Modeling Variable Relationships

The overall progression of advancements is as follows. It begins with modeling that explicitly accounts for correlations between variables. This progresses to directly modeling the relationships between both temporal and variable aspects. Additionally, the model incorporates the possibility of time lags in the relationships between variables, allowing it to flexibly learn dependencies.

Crossformer (Zhang and Yan, 2023) breaks away from solely temporal attention by employing a Two-Stage Attention mechanism that sequentially processes Cross-Time and Cross-Dimension stages. It divides each dimension's time series into patches, embeds them into 2D vectors, and performs attention. Crossformer incorporates a router mechanism in the Cross-Dimension stage to handle the increased complexity of dual attention. This hierarchical structure across multiple scales enables Crossformer to effectively model both temporal dependencies and inter-dimensional correlations, significantly enhancing multivariate time series forecasting performance. **DSformer** (Yu et al, 2023) criticizes Crossformer for emphasizing global interactions between variables in time series data, while overlooking the importance of local information, such as short-term variations and patterns. DSformer addresses this by using double sampling, obtaining global information via downsampling and local information through piecewise sampling. These samples undergo parallel temporal variable attention, allowing the model to integrate global, local, and inter-variable correlations in multivariate time series data through dual sampling and time-variable attention mechanisms. **CARD** (Wang et al, 2024b) points out the structural complexity and high computational cost of Crossformer's hierarchical attention mechanism. Simplifying the structure, CARD employs only the encoder and uses a lightweight token blend module instead of explicitly generating token hierarchies. This module merges adjacent tokens to represent larger ranges, efficiently leveraging multi-scale information and enhancing prediction accuracy. A robust loss function is introduced to prevent overfitting, adjusting weights to favor near-future data over far-future data, which shows less temporal correlation. **iTransformer** (Liu et al, 2023) innovatively reverses the time and variable dimensions in the standard Transformer model, learning inter-variable correlations. By embedding each variable into variate tokens via a Transformer encoder and performing temporal predictions through a feed-forward network, iTransformer surprisingly outperforms Crossformer, highlighting the limitations of merely adding explicit channel attention. Its advantage lies in not needing to alter the original Transformer structure. **VCformer** (Yang et al, 2024) addresses the issue of neglecting time lags in variable correlations. Its variable correlation attention module calculates cross-correlation scores between queries and keys across various lags using FFT (Fast Fourier Transform), efficiently exploring and adaptively integrating correlations for each lag. **GridTST** (Cheng et al, 2024c) transforms time series data into a grid format, applying horizontal and vertical attention mechanisms to model time and variable correlations effectively. Like PatchTST and iTransformer, GridTST leverages the original Transformer, distinguishing itself from the more complex Crossformer. It experiments with three configurations: time-first, channel-first, and cross-application, finding that prioritizing channel attention yields the best performance, aligning with human intuitive analysis of inter-variable relationships. **UniTST** (Liu et al, 2024b) critiques the inability of cross-time and cross-dimension attention methods to directly and explicitly learn complex inter-variable and intra-variable dependencies. It proposes a unified attention mechanism by flattening patch tokens. To mitigate the increased computational cost, UniTST incorporates a dispatcher module to reduce memory complexity and enhance model feasibility. **DeformTime** (Shu and Lampos, 2024) introduces deformable attention, dynamically adjusting to recognize and adapt to important data features instead of fixed attention areas. Parallel deformable attention mechanisms within the Transformer encoder capture correlations between variables and time, adaptively extracting critical information across various time intervals and variable combinations, significantly improving prediction accuracy and generalizability.

4.1.3 Exogenous Variable

Most existing research primarily utilizes endogenous variables for prediction. However, in real-world scenarios, relying solely on endogenous variables can be insufficient due to the complexity of various influencing factors. For instance, stock price predictions are significantly affected by external factors such as economic indicators, political changes, and technological advancements. Ignoring these external factors and relying only on past data of endogenous variables can lead to failures in accurately predicting market volatility. Therefore, incorporating exogenous variables as supplementary information has emerged as a method to improve prediction performance.

TimeXer (Wang et al, 2024c) proposes a method to integrate exogenous variables into the canonical Transformer model without structural changes. It operates by dividing the time series data of endogenous variables into patches, learning the temporal dependencies of each patch through self-attention. Then, it generates variate tokens summarizing the entire series of endogenous and exogenous variables and learns their interactions using a cross-attention mechanism. Through this process, TimeXer simultaneously considers the temporal patterns of endogenous variables and the impact of exogenous variables, enabling more precise and in-depth time-series predictions. However, TimeXer requires manual identification and input of appropriate exogenous variables. If unsuitable data is provided, it can hinder the model's predictive accuracy. **TGTSF** (Xu et al, 2024b) integrates text data from channel descriptions and news messages to enhance prediction accuracy. It embeds channel descriptions and news messages using a pre-trained text model, transforming them into a sequence of vectors over time. The cross-attention layer then calculates the relevance of the text to each channel. By incorporating text data into the time series prediction model, TGTSF not only improves prediction accuracy but also allows for direct comparison of the impact of textual information on predictive performance.

4.1.4 Additional Approaches

Beyond the topics discussed above, there are various other approaches to time series forecasting.

Generalization

Research has been conducted to improve model generalization, avoid overfitting, and achieve consistent performance across diverse datasets. **SAMformer** (Ilbert et al, 2024) addresses the issue of self-attention mechanisms converging to sharp local minima, causing entropy collapse during training, and demonstrates that applying SAM (Sharpness-Aware Minimization) can significantly enhance performance. **Minusformer** (Liang et al, 2024b) highlights the redundancy and overfitting caused by a large number of parameters in Transformers. To combat this, it employs a boosting ensemble method, where each subsequent model predicts the residuals of the previous model's outputs, thus reducing redundancy and improving generalization.

Multi-scale

The multi-scale approach extracts more information from time series data across various scales, offering distinct advantages. **Scaleformer** (Shabani et al, 2022) proposes a general framework by stacking existing models across different scales, resulting in improved performance. **Pathformer** (Chen et al, 2024b), on the other hand, allows the model to learn adaptive scales independently, rather than relying on fixed scales.

Decoder-only

Large-scale language models (LLMs) like LLaMA3 (Dubey et al, 2024) have been successfully implemented using only a decoder without the need for an encoder. The decoder-only architecture is simpler and involves less complex computations, resulting in faster training and inference. Additionally, the decoder-only structure helps avoid the temporal information loss often associated with the self-attention mechanism in encoders. This has led to a research proposal aimed at improving performance in time series forecasting using a decoder-only structure. **CATS** (Kim et al, 2024) addresses the high time and memory complexity of self-attention in Transformer encoders and the loss of temporal order information. It proposes a structure using only cross-attention, focusing solely on the relationship between future and past data with a decoder-only architecture, which reduces parameter count and enhances efficiency. In contrast to the encoder-only structure of most models discussed CATS demonstrates the effectiveness of using only the decoder in achieving superior performance.

Feature Enhancement

Fredformer (Piao et al, 2024) identifies the issue of frequency bias in time series data, where learning tends to focus disproportionately on either low or high frequencies. It addresses this by normalizing the frequencies to eliminate bias. **Basisformer** (Ni et al, 2024) proposes a method to construct flexible relationships with each time series by leveraging biases learned through contrastive learning.

4.2 Growth of Traditional Deep Learning Models

Since the advent of simple linear models, there has been a surge in research focused on non-transformer-based models. Attention has shifted to various architectures such as MLP, RNN, CNN, and GNN, with many models surpassing Transformers and achieving remarkable performance improvements. Although transformer-based models exhibit excellent performance across numerous fields, they have structural limitations in learning temporal order information, which is crucial for time series problems. While past tasks were simple and general enough to overlook these limitations, current real-world tasks involve many constraints and data-specific issues with diverse variables, necessitating approaches from various perspectives. Each architecture has its own strengths, and these characteristics provide valuable solutions for addressing diverse contemporary time series forecasting challenges. In this section, we will investigate the latest major models for each architecture and analyze their technical features. The key characteristics of each backbone architecture have been briefly summarized in comparison to Transformers in Table 6.

4.2.1 MLP-Based Models

MLP-based models have recently emerged as a key methodology for replacing Transformers in time series forecasting tasks. The simple structure of MLPs makes them easy to handle and interpret. Additionally, they perform well even in constrained environments with limited computational resources

Table 5: Taxonomy and Methodologies of Transformer Models for Time Series Forecasting

Main Improvement	Model Name	Main Methodology	Channel Correlation	Enc/Dec Usage	Publication
Patching Technique	PatchTST	· Patching · Channel Independence	CI	Enc	2023
	MTST	· Multiple Patch-based Tokenizations	CI	Enc	2024
	PETformer	· Placeholder-enhanced Technique	CI	Enc	2022
Cross-Dimension	Crossformer	· Dual Attention: Cross-time & Cross-dimension	CD	Enc & Dec	2023
	DSformer	· Dual Sampling & Dual Attention	CD	Enc	2023
	CARD	· Dual Attention · Token Blend Module for multi-scale	CD	Enc	2024
	iTransformer	· Attention on Inverted Dimension	CD	Enc	2024
	VCformer	· Variable Correlation Attention Considering Time Lag · Koopman Temporal Detector for Non-stationarity	CD	Enc	2024
	GridTST	· Dual Attention with original Transformer	CD	Enc	2024
	UniTST	· Unified Attention by Flattening	CD	Enc	2024
	DeformTime	· Deformable Attention Blocks	CD	Enc	2024
Exogenous Variable	TimeXer	· Integration of Endogenous and Exogenous Information	CD	Enc	2024
	TGTSF	· Exogenous Variable with Description, News	CD	Enc & Dec	2024
Generalization	SAMformer	· SAM (sharpness-aware minimization)	CD	Enc	2024
	Minusformer	· Dual-stream and Subtraction mechanism	CD	Enc	2024
Multi-scale	Scaleformer	· Multi-scale framework	Any	Enc & Dec	2023
	Pathformer	· Adaptive Multi-scale Blocks	CD	Enc & Dec	2024
Decoder-only	CATS	· Cross-Attention-Only Transformer	CI	Dec	2024
Feature Enhancement	Fredformer	· Frequency Debias	CD	Enc	2024
	BasisFormer	· Automatic Learning of a Self-adjusting Basis	CD	Dec	2023

Table 6: Comparison of Other Deep Learning Models with Transformers in Terms of Criteria

Criteria	Transformer-based models	MLP-based models	CNN-based models	RNN-based models	GNN/GCN-based models
Structure	Complex self-attention mechanism	Simple layer, relatively easy to implement and interpret	Utilizes convolutional layers, effectively capturing specific local patterns	Specialized in sequential data processing, effectively handling temporal dependencies but may struggle with long sequences	Learns relationships between nodes using graph structures, effectively capturing complex relationships
Data Requirements	Requires large datasets	Can train on smaller datasets	Can train on smaller datasets	Can train on smaller datasets, suitable for quick training with limited data	Can achieve high performance with relatively small datasets
Training Time	Relatively slow due to global attention mechanisms	Relatively fast	Generally faster due to localized computations	Trains effectively on smaller datasets but can be slow for long sequences	Varies depending on graph complexity
Model Size	Comparatively larger and more parameter-intensive	Comparatively small, efficient use of resources	Typically smaller and more parameter-efficient, making it resource-efficient and scalable	Comparatively small	Depends on graph size and complexity, achieving high performance with fewer parameters in specific problems
Interpretability	Difficult to interpret	Relatively high interpretability	More interpretable through visualizations of filters and feature maps	Moderately interpretable, easier to understand and explain model behavior	Moderately interpretable depending on graph structure and model complexity
Performance	Suitable for learning long-term dependencies	Suitable for short-term predictions with sufficient performance in many cases	Excels at capturing local temporal dependencies, superior for problems with strong local patterns	Suitable for short-term and sequential dependencies, providing sufficient performance in specific time series problems	Excels at learning complex dependencies between nodes, offering high performance in learning specific relationship patterns
Flexibility	Requires complex adjustments	Easily adjustable for specific problems	Easily customizable with various types of convolutions	Extensible with various RNN architectures	Can handle various graph structures and formats, adaptable to different data types and structures
Application	Suitable for complex time series problems or NLP-related tasks	Versatile for various general forecasting problems	Well-suited for applications requiring spatial and temporal locality, effective for a wide range of time series problems	Effective for sequential data and time series forecasting, but struggles with long-term dependencies without modifications	Suitable for complex graph structures in tasks like social networks, recommendation systems, and time series graphs
Hardware Requirements	High due to their complex structure and computationally intensive self-attention mechanisms	Lower due to their simpler structure and fewer computational demands	Lower computational and memory requirements	Low but inefficient on parallel hardware	Generally low but depends on graph size
Memory Usage	Higher memory usage due to full sequence attention	Lower memory usage due to their simple structure and fewer parameters	Lower memory usage due to localized operations	Low but can increase with sequence length	Generally low but depends on graph size
Parallel Processing	Highly parallelizable but requires synchronization due to attention mechanisms	Highly parallelizable due to independent computations	Highly parallelizable due to independent convolution operations	Difficult due to sequence dependencies	Difficult due to graph structure dependencies

and limited data. Their lightweight architectures enable fast training and inference, making them increasingly important and widely used in contemporary industries. Previously, interest in MLPs diminished due to their structural limitations, such as the lack of sequential dependency, challenges with long-term dependency, difficulties in processing high-dimensional data, and limitations in capturing periodic patterns. However, recent advancements have enabled long-sequence learning and various technical enhancements, leading to remarkable performance improvements.

Koopa (Liu et al, 2024c) is a model designed to effectively handle non-stationary time series data by utilizing Koopman theory to predict the dynamic changes of components. It uses a Fourier filter to separate time-invariant and time-variant elements, which are then fed into respective Koopman predictors(KP). In KP, nonlinear time series data are linearly transformed, making them easier to manage. To effectively capture the characteristics of each component, the time-invariant KP uses a globally learned Koopman operator, while the time-variant KP computes a local operator for predictions. **TSMixer** (Ekambaram et al, 2023) is a lightweight patch-based model that introduces the MLP-mixer from the computer vision field to the time-series domain. It efficiently learns long-term and short-term dependencies through inter-patch and intra-patch mixers respectively, and explicitly learns channel relationships through an inter-channel mixer. To enhance the understanding of inter-channel correlations, it adjusts the original forecast results using reconciliation heads and employs gated attention to filter important features. By introducing these techniques, TSMixer upgrades the simple MLP structure, resulting in improved performance that outperforms complex Transformer models. **FreTS** (Yi et al, 2024) is a model that leverages two key characteristics of the frequency domain—global view and energy compaction—to directly train an MLP model in the frequency domain. While existing models primarily use frequency transformation to verify the periodicity of the model, this model distinguishes between channel and temporal dependencies in the frequency domain as real and imaginary parts, respectively, and learns them directly to better extract hidden features. **TSP** (Wang et al, 2024e) utilizes a PrecMLP block with a precurrent mechanism to effectively model temporal locality and channel dependency. The precurrent mechanism combines previous information with current information to create hybrid features at each time point, serving as a computationally-free method to effectively recognize and process temporal locality. This lightweight MLP model comprises an encoder, consisting of tMLP operating in the time dimension and vMLP operating in the variable dimension, and a decoder using a linear model. **FITS** (Xu et al, 2024c) is a very lightweight model with fewer than 10,000 parameters, yet it demonstrates competitive performance comparable to larger models. It performs interpolation through a complex linear layer that learns amplitude scaling and phase shifts in the frequency domain, thereby expanding the frequency representation of the input time series. By using a low-pass filter to remove high-frequency components, which are often noise, the model efficiently represents the data and reduces the number of learnable parameters. **U-Mixer** (Ma et al, 2024) utilizes a hierarchical structure of U-Net's encoder-decoder composed of MLPs to extract and combine low-dimensional and high-dimensional features. Each MLP block processes temporal dependencies and channel interactions separately, enabling stable feature extraction. Additionally, the model employs a stationary correction method to limit the differences in data distribution before and after processing, thereby restoring the non-stationary information of the data. By calculating the transformation matrix and mean difference between the input and output data, the model adjusts the output to maintain temporal dependency, thereby enhancing its prediction performance. **TTMs** (Ekambaram et al, 2024) is a time series foundation model that uses TSMixer as its backbone and relies solely on time series data for rapid pre-training. This model is trained on the extensive collection of time series datasets with diverse channels and resolutions from the Monash archive (Table 3). Despite the data-specific characteristics of the time series datasets, TTMs demonstrate efficient transfer learning and high generalization performance. **TimeMixer** (Wang et al, 2024a) leverages the observation that time series data exhibit distinct patterns at different sampling scales. To extract important information from past data, multi-scale time series are generated through downsampling. In the Past-Decomposable-Mixing (PDM) block, the seasonal and trend components are decomposed and mixed separately. The Future-Multipredictor-Mixing (FMM) block then integrates predictions by ensembling multiple predictors, which utilize the past information extracted at various scales. **CATS** (Lu et al, 2024) enhances the performance of multivariate time series forecasting by generating Auxiliary Time Series (ATS) from the Original Time Series (OTS) and integrating them into the prediction process. The model proposes various types of ATS constructors, each with different roles and objectives, combining them to maximize predictive performance. To effectively generate and utilize ATS, the model structure and loss function incorporate three key principles: continuity, sparsity, and variability. Despite using a simple two-layer

MLP predictor as the foundation, CATS effectively predicts multivariate information through the use of ATS. **HDMixer** (Huang et al, 2024a) is a model that applies a Length-Extendable Patcher to overcome the limitations of fixed-length patching. It measures the point-wise similarity with fixed patches and compares the patch entropy loss, updating parameters to increase the complexity of the extendable patches. Through the Hierarchical Dependency Explorer mixer, it learns long-term, short-term dependencies and cross-variable dependencies. **SOFTS** (Han et al, 2024) proposes the STar Aggregate-Redistribute (STAR) Module to effectively learn the relationships between channels. It converts the data from each channel into high-dimensional vectors and then aggregates these vectors to extract comprehensive information, referred to as the core. The generated core representation is then combined with the time series representation of each channel and converted back to the time series dimension, reducing complexity and enhancing robustness. **SparseTSF** (Lin et al, 2024b) is an extremely lightweight model that uses fewer than 1,000 parameters while demonstrating excellent generalization performance. It downsamples the data into multiple periods, performs predictions on each generated sub-sequence, and then upsamples to create the overall prediction sequence. While it shows limited performance only on data with clear periodic patterns, it efficiently handles complex temporal dependencies with minimal computational resources. **TEFN** (Zhan et al, 2024) proposes a novel backbone model from the perspective of information fusion to extract latent distributions from simple data structures. Based on evidence theory (Shafer, 1976), the Basic Probability Assignment Module maps each time and channel dimension of the time series data to a mass function in the event space. This mass function assigns probabilities to multiple possibilities for each data point, allowing for explicit modeling of uncertainty. The generated mass functions are fused using their expected values to derive the final predictions. This model achieves high predictive performance with few parameters and fast training times. **PDMLP** (Tang and Zhang, 2024) surpasses the performance of complex transformer-based models using only simple linear layers and patching. It divides the data into patches of various scales and embeds each patch with a single linear layer. The embedding vectors are then decomposed into smooth components and residual components containing noise, allowing the model to process the sequences in two ways: intra-variable and inter-variable. **AMD** (Hu et al, 2024b) introduces the AMS block, which assesses the importance of various temporal patterns and generates appropriate weights for each pattern. By simultaneously modeling the temporal and channel dependencies of the input information decomposed into multiple scales, it effectively captures interactions at different scales. Instead of merely integrating the results from each scale, it improves prediction performance by reflecting the importance of each pattern through the weights.

4.2.2 CNN-Based Models

CNNs were primarily developed for image recognition and processing tasks because they are highly effective at identifying 2D spatial patterns. Some studies have begun utilizing Temporal Convolutional Networks (TCNs), which are 1D CNNs that move along the time axis to recognize local patterns, effectively extracting key features of time series data. Additionally, CNNs can use filters of various sizes to capture patterns at multiple scales, allowing them to effectively learn both short-term and long-term patterns in time series data. CNN's deep network technology and parallelism, which have been developed over a long period of time, have provided high performance and reliability for time series forecasting. However, the fixed filter size of CNNs lacks flexibility in learning complex patterns of long sequences and poses challenges in capturing long-term dependencies. Furthermore, as the network depth increases, there are limitations such as information loss, increased computational cost, and higher memory usage. For these reasons, the relatively flexible Transformer, which is advantageous for learning long-term dependencies, has gained more attention in time series data processing. However, due to their superior ability to capture diverse local patterns within long sequences, CNNs are once again receiving attention as a backbone architecture.

TimesNet (Wu et al, 2023) introduces the TimesBlock, which embeds 1D time series into 2D tensors for analysis to effectively capture multi-periodicity. After extracting key frequencies via Fast Fourier Transformation (FFT), it expands the data into a 2D tensor, representing intra-period variation in columns and inter-period variation in rows. It uses a parameter-efficient Inception module with 2D kernels to capture temporal variations across various scales. After converting back to a 1D tensor, it adaptively aggregates the data based on the importance of frequency amplitude values. **PatchMixer** (Gong et al, 2023) introduces a patch-mixing design to efficiently replace the permutation invariant property of the Transformer's attention mechanism. It learns intra-patch correlations using single-scale depth-wise separable convolutions and inter-patch correlations using point-wise convolutions on the patched input sequence. **ModernTCN** (Luo and Wang, 2024) enhances the traditional TCNs to make it more suitable for time series analysis. By using larger kernels than conventional TCNs, it expands the effective receptive field, reducing the number of layers and effectively capturing long-term dependencies. It separates depth-wise convolution and point-wise convolution to independently learn temporal dependencies and feature dependencies, thereby enhancing computational efficiency. Additionally, it explicitly handles cross-variable dependency in multivariate time series data, resulting in improved performance. **ConvTimeNet** (Cheng et al, 2024b) follows the framework of a Transformer encoder, replacing the role of self-attention with depthwise convolution and the role of multi-head attention with convolutions of various kernel sizes to increase computational efficiency. It uses Deformable Patch Embedding, which adaptively adjusts each patch's size and position based on the data. Depthwise convolutions capture temporal features for each channel, while point-wise convolutions model channel dependencies. **ACNet** (Zhang and Wang, 2024) aims to effectively model temporal dependencies and nonlinear features. It starts by removing unnecessary noise from the data using Wavelet Denoising. Then, it extracts temporal features through Multi-Resolution Dilated Convolution and Global Adaptive Average Pooling. To more accurately capture nonlinear features, Gated Deformable Convolution is employed, which adaptively adjusts the sampling positions. Additionally, ACNet employs dynamic prediction, retraining the model with new data if prediction performance declines. **FTMixer** (Li et al, 2024c) leverages the observation that the time domain effectively captures local dependencies while the frequency domain excels at learning global dependencies. It applies convolution in the frequency domain to capture global dependencies. Meanwhile, it converts multi-scale patches using Discrete Cosine Transformation (DCT), which is computationally simpler than Discrete Fourier Transformation (DFT), and then applies convolution to capture local dependencies.

4.2.3 RNN-Based Models

RNNs were fundamentally developed to process sequential data. Their recurrent structure can effectively model the temporal dependencies of time series data. Gated RNN models, such as LSTM and GRU, have partially alleviated the long-term dependency issues of vanilla RNNs. However, RNNs are difficult to parallelize because they process data sequentially. This slow speed is particularly detrimental when handling long sequences. RNNs also have inherent limitations in learning long-term dependencies and global information. Transformers have effectively addressed these issues through self-attention mechanisms. However, unlike Transformers, which rely on large datasets, RNN-based models can learn effectively with smaller amounts of data. They also use memory efficiently and are well-suited for real-time data processing due to their sequential nature, making them advantageous for online streaming applications. Additionally, because time ordering is naturally learned, RNNs still have inherent advantages over Transformers for time series processing.

PA-RNN (Zhang et al, 2024b) aims to address prediction Uncertainty Quantification and Model Compression issues for dependent time series data by proposing an optimization methodology for Sparse Deep Learning. It applies a Mixture of Gaussian Prior to introduce sparsity to the parameters and uses a Prior Annealing technique to gradually increase the strength of the prior distribution during the training process. Properties such as Posterior Consistency, Structure Selection Consistency, and Asymptotic Normality ensure the model's theoretical validity. **WITRAN** (Jia et al, 2024) proposes a new paradigm by implementing information transmission in bi-granular flows to capture long-term and short-term repetitive patterns. It models global and local correlations through the Horizontal Vertical Gated Selective Unit (HVGSU), which uses bidirectional Gated Selective Cells (GSC). To enhance the efficiency of information transmission through parallel processing, it applies the Recurrent Acceleration Network (RAN). This approach results in excellent prediction performance while reducing time complexity and maintaining memory complexity. **SutraNets** (Bergsma et al, 2023) transforms long sequences into multiple lower-frequency sub-series in order to effectively predict long sequence time series data. It performs autoregressive predictions among the sub-series, with each sub-series being conditionally predicted based on the values of other sub-series. This approach reduces the signal path length and allows for the generation of long sequences in fewer steps. Consequently, it mitigates the common RNN problem of error accumulation and models long-distance dependencies better. **CrossWaveNet** (Huang et al, 2024b) uses a dual-channel network to separate and process the seasonality and trend-cyclic components of data. The input data are initially decomposed into individual elements through series decomposition, and features are extracted using enriched LSTM and GRU structures. These extracted features undergo a cross-decomposition process to further refine the elements, which are then sent to each channel for integration. Series decomposition is performed at each RNN step, progressively filtering and reintegrating the seasonality and trend components to learn precise information. **DAN** (Li et al, 2024b) proposes a new model that utilizes Polar Representation learning to predict long-term time series data with high volatility. The Distance-weighted Auto-regularized Neural network (DAN) distinguishes and learns the polar representations of distant and nearby data, combining them to enhance prediction performance. It addresses the imbalance problem of extreme data, which occur infrequently but are critical for accurate predictions, using Kruskal-Wallis Sampling and employs a Multi-Loss Regularization Method to effectively learn both extreme and normal values simultaneously. **RWKV-TS** (Hou and Yu, 2024) proposes an efficient Recurrent Weighted Key-Value (RWKV) backbone characterized by linear time complexity and memory usage. It uses a Multi-head Weighted Key-Value (WKV) Operator, which is similar to the self-attention mechanism but maintains linear time and space complexity. Each head captures different temporal patterns, increasing the representational power of the information and enabling parallel computation, thus providing high computational efficiency. This allows RWKV-TS to capture longer sequence information more effectively than traditional RNNs. **CONTIME** (Jhin et al, 2024) introduces Continuous GRU to minimize the prediction delay issue in time series forecasting. This approach applies Neural ODE to the existing GRU, modeling data changes continuously over time. The bi-directional structure integrates forward and backward through the time series data, capturing more accurate temporal dependencies. By employing Time-Derivative Regularization, the model is guided to learn the rate of change in predictions over time, enhancing both the accuracy and speed of predictions.

4.2.4 GNN-Based Models

GNNs or GCNs are specialized in processing graph-structured data, being suitable for modeling complex interactions and learning local patterns in time series data. Multivariate time series data often involve interactions between variables, which can be effectively represented as relationships between nodes and edges. By assigning various attributes to nodes and edges, richer representations can be learned, and the structural characteristics of specific domains can be better reflected. Traditionally, various approaches such as GCN, ST-GCN, GAT, and TGN have been proposed, contributing to the advancement of the time series forecasting field.

However, these models primarily focus on learning local information from adjacent nodes, leading to a lack of capability in learning global information and difficulty in capturing long-term dependencies with distant nodes. The sequential calculation of nodes and edges makes parallel processing challenging, and the models' specificity to particular graph structures reduces their generalization ability. These drawbacks have diminished interest in GNN-based models. However, with the recent advancements in social networks, much of current data naturally follows a graph structure. Various optimization techniques and hardware acceleration technologies have been developed, making it possible to process large-scale graph data. Additionally, the increased practical relevance due to the flexibility in dynamically handling time-varying data has once again drawn researchers' attention to this architecture.

MSGNet (Cai et al, 2024a) effectively learns the complex correlations of MTS data by combining frequency domain analysis with adaptive graph convolution. It uses Fast Fourier Transformation to identify key periods and transform time series data into various time scales. Then, it employs Mix-hop graph convolution to learn inter-channel correlations and uses Multi-head Attention and Scale Aggregation to capture intra-series correlations. **TMP-Nets** (Coskunuzer et al, 2024) is a model proposed to learn features extracted from Temporal MultiPersistence (TMP) vectorization and capture key topological patterns. It combines two advanced topological data analysis techniques, Multipersistence and Zigzag Persistence, to effectively capture topological shape changes in the data. Spatial Graph Convolution is used to model dependencies between nodes in the graph, and the topological features extracted with TMP vectors are learned using CNN. **HD-TTS** (Marisca et al, 2024) is a model that demonstrates high performance even with missing data through masking and hierarchical downsampling. It incrementally reduces data in the temporal and spatial processing modules, learning various temporal and spatial dependencies. The model uses an attention mechanism to learn the patterns of missing values and combines them with weights to generate the final prediction. **ForecastGrapher** (Cai et al, 2024b) is based on the finding that the attention mechanism is more suitable for modeling inter-channel correlations than temporal correlations. Utilizing the similarity between GNN's neighborhood aggregation and attention mechanisms, it linearly embeds each channel as a node and learns the relationships between channels using GNN. Each GNN layer employs a self-learnable adjacency matrix to independently learn the interactions between nodes. It uses learnable scalers to divide node features into multiple groups and applies 1D convolution to each group.

4.2.5 Hybrid Models

Time series data often involve not only temporal dependencies but also important interactions between variables. Therefore, many models explicitly learn the interdependencies between variables in addition to temporal dependencies. However, incorporating all this information into a single architecture can lead to overfitting, which may limit the learning process. Recent models often combine two or more architectures to efficiently learn by distinguishing the roles based on the nature of the information.

WaveForM (Yang et al, 2023) improves prediction performance by transforming time series data into the wavelet domain and combining it with GNN. The Discrete Wavelet Transformation (DWT) uses High-Pass and Low-Pass Filters to decompose the time series data into small wavelets. This process removes noise and allows CNN to efficiently capture temporal and frequency characteristics. The GNN is then used to model the interactions between these components. **TSLANet** (Eldele et al, 2024) is a model that replaces the computationally expensive attention mechanism with CNN in the Transformer framework. It incorporates frequency analysis and a learnable threshold to selectively attenuate high-frequency noise, effectively learning long-term and short-term interactions. Additionally, it uses parallel convolution layers with different kernel sizes to capture both local patterns and long-term dependencies. This approach effectively compensates for the typical CNN's limitations in modeling long-term dependencies. **DERITS** (Fan et al, 2024a) enhances the prediction performance of non-stationary time series data by combining convolution operations in the frequency domain with MLP in the time domain. The Frequency Derivative Transformation (FDT) converts time series signals into the frequency domain and then differentiates the frequency components to represent them in a static form that is easier to predict. The Order-adaptive Fourier Convolution Network (OFCN) is responsible for frequency filtering and learning dependencies in the frequency domain. To improve prediction performance, it processes and fuses derivative information of various orders in parallel, utilizing a Parallel-stacked Architecture. **BiTGraph** (Chen et al, 2024c) focuses on improving performance in situations with missing data by capturing both temporal dependencies and spatial structures. The Multi-Scale Instance Partial TCN module learns short-term and long-term dependencies through kernels of various sizes and can also compensate for missing values through partial TCN. The Biased GCN module for inter-channel relationship learning represents the relationships between data points as a graph structure and adjusts the strength of information propagation between nodes, considering the missing patterns.

4.2.6 Model-Agnostic Frameworks

Some studies propose model-agnostic methodologies that can be universally applied without being limited to a specific model backbone. These studies focus on the intrinsic characteristics of time series data or specific problems, improving model robustness by addressing these issues. Model-agnostic architectures, not being tied to a particular model, allow for flexible selection of the optimal model through various comparisons. They are also highly reusable and make it easier to interpret and understand the model's predictions, making them applicable to a wide range of domains and scenarios.

These models primarily address the distribution shift problem, which is one of the core issues in time series forecasting, and propose various plug-and-play alleviation methods. The details related to this will be discussed in Section 5, while here, we will focus on other methodologies, excluding distribution shift alleviation methods.

RobustTSF (Cheng et al, 2024a) introduces a selective sampling algorithm to train a resilient prediction model in Time Series Forecasting with Anomalies (TSFA). It calculates anomaly scores based on the difference between the trend and the actual data, selecting samples with scores below a certain threshold to train the model, thereby minimizing the impact of anomalies. By analyzing three types of anomalies (Constant, Missing, Gaussian), it identifies the most robust loss function as MAE. This approach addresses the discontinuity issues of the traditional detection-imputation-retraining pipeline, enabling the model to deliver more stable and consistent performance. **PDLS** (Hounie et al, 2024) proposes a constrained learning approach by setting an upper bound on the loss at each time step. It controls the loss by setting a constant upper bound(ϵ) for all time steps and introduces a variable(ζ) that relaxes the constraints, allowing automatic adjustment during training. This model overcomes the problem of existing methodologies that focus on averaging prediction performance, which often leads to uneven error distributions. **Leddam** (Yu et al, 2024) proposes two independently usable modules, Learnable Decomposition and Dual Attention, to effectively capture the complex patterns of inter-series dependencies and intra-series variations in MTS. The Learnable Decomposition module replaces the traditional moving average kernel with a learnable 1D convolution kernel initialized with a Gaussian distribution for decomposition, making it more sensitive to critical information. The Dual Attention module simultaneously models inter-data dependencies and intra-series variability using channel-wise self-attention and auto-regressive self-attention. **InfoTime** (Qi et al, 2024) proposes a method to effectively utilize inter-channel and temporal correlations. For channel mixing, it employs Cross-variable Decorrelation Aware feature Modeling (CDAM) to minimize redundant information between channels and enhance useful mutual information. Additionally, it uses TAM(Temporal correlation Aware Modeling) to maximize the temporal correlations between the predicted future sequence and the target sequence. **CCM** (Chen et al, 2024a) combines the strengths of the traditional channel-independent and channel-dependent strategies to overcome their limitations. It dynamically clusters channels based on their inherent similarities and learns representative patterns for each cluster to generate prototype embeddings. Separate FFNs are used for each cluster to perform predictions for the channels within the cluster, and the prototype embeddings enable zero-shot predictions. **HCAN** (Sun et al, 2024) addresses the issue of excessive flattening caused by using the MSE loss function and proposes reconfiguring the prediction problem as a classification problem to better learn high-entropy features. It extracts important features from the backbone model, divides them into multiple layers and categories, and uses an Uncertainty-Aware Classifier at each layer to reduce uncertainty. It learns relative prediction values within each classified category and integrates the prediction results from multiple layers through the Hierarchy-Aware Attention module. This approach effectively captures the complex patterns and variability in time series data. **TDT Loss** (Xiong et al, 2024) points out the error accumulation issue in the traditional auto-regressive approach which recursively models Temporal Dependencies among Targets (TDT). To effectively capture changes between consecutive time points, it represents TDT using first-order differencing. The final TDT loss combines the target prediction loss, TDT values prediction loss, and an adaptive weight, allowing the model to dynamically balance between target prediction and TDT learning. Thus, TDT loss replaces the optimization objective of non-autoregressive models.

The different deep learning models and their methodologies discussed so far are summarized in Table 7.

4.3 Emergence of Foundation Models

In recent years, foundation models have demonstrated remarkable performance and strong zero-shot capabilities across various tasks and domains, leading to increased focus (Touvron et al, 2023; Liu et al, 2024a; Li et al, 2023; Achiam et al, 2023; Radford et al, 2021). While significant progress has been made in domains like vision and language, developing foundation models for time series data has faced several challenges. Firstly, time series data exhibit distinct characteristics depending on the dataset. For instance, electrocardiogram (ECG), weather, and sensor data from industrial processes have unique properties in terms of variables, frequency, periodicity, and noise, often requiring domain-specific knowledge for effective modeling. Additionally, unlike the vision and language domains, where large-scale pre-training corpora can be relatively easily constructed from publicly available sources like the web, collecting time series data is more difficult due to high acquisition costs and security concerns. Despite these obstacles, research on time series foundation models is essential for improving model scalability and generality. This necessity has led to active exploration in several key directions, particularly in the field of time series forecasting.

4.3.1 Sequence Modeling with LLMs

One prominent approach leverages the sequence modeling capabilities of LLMs. LLMs have revolutionized deep learning with their groundbreaking generalizable sequence modeling. Given the sequential nature of both text and time series data, extending the sequential capabilities of language models to time series is a natural progression. Early research includes **GPT4TS** (Zhou et al, 2023), which demonstrates that by freezing most of the language model's backbone and fine-tuning only the layer normalization parameters and positional embeddings on time series data, GPT4TS can serve as a general time series task solver for forecasting, anomaly detection, and classification. Additionally,

Table 7: Taxonomy and Methodologies of Traditional Deep Learning Architectures for Time Series Forecasting

Model Name	Base	Main Methodology	Channel Correlation	Publication
Koopa	MLP	· Koopa Block with Koopman Predictor(KP)	CD	2023
TSMixer		· MLP Mixer layer architecture · Gated attention (GA) block · Online hierarchical patch reconciliation head	CI/CD	2023
FreTS		· Frequency-domain MLPs	CD	2023
TSP		· PrecMLP block with precurrent mechanism	CD	2024
FITS		· Complex Frequency Linear Interpolation · Low Pass Filter(LPF)	CI	2024
U-Mixer		· Unet Encoder-decoder with MLPs · Stationarity Correction	CD	2024
TTMs		· Multi-Resolution Pre-training via TTM Backbone (TSMixer blocks) · Exogenous mixer	CD	2024
TimeMixer		· Past-Decomposable-Mixing (PDM) block · Future-Multipredictor-Mixing (FMM) block	CD	2024
CATS		· Auxiliary Time Series(ATS) Constructor	CD	2024
HDMixer		· Length-Extendable Patcher · Patch Entropy Loss · Hierarchical Dependency Explorer	CD	2024
SOFTS		· STar Aggregate-Redistribute Module	CD	2024
SparseTSF		· Cross-Period Sparse Forecasting	CI	2024
TEFN		· Basic Probability Assignment(BPA) Module	CD	2024
PDMLP		· Multi-Scale Patch Embedding & Feature Decomposition · Intra-, Inter-Variable MLP	CD	2024
AMD		· Multi-Scale Decomposable Mixing(MDM) Block · Dual Dependency Interaction(DDI) Block · Adaptive Multi-predictor Synthesis(AMS) Block	CD	2024
TimesNet	CNN	· Transform 1D-variatios into 2D-variations · Timesblock	CD	2023
PatchMixer		· Patch Embedding · PatchMixer layer with Patch-mixing Design	CI	2023
ModernTCN		· ModernTCN block with DWConv & ConvFFN	CD	2024
ConvTimeNet		· Deformable Patch Embedding · Fully Convolutional Blocks	CD	2024
ACNet		· Temporal Feature Extraction Module · Nonlinear Feature Adaptive Extraction Module	CD	2024
FTMixer		· Frequency Channel Convolution · Windowing Frequency Convolution	CD	2024
PA-RNN	RNN	· Mixture Gaussian Prior · Prior Annealing Algorithm	CI	2023
WITRAN		· Horizontal Vertical Gated Selective Unit · Recurrent Acceleration Network	CI	2023
SutraNets		· Sub-series autoregressive networks	CI	2023
CrossWaveNet		· Deep cross-decomposition · Dual-channel network	CD	2024
DAN		· RepGen & RepMerg with Polar Representation Learning · Distance-weighted Multi-loss Mechanism · Kruskal-Wallis Sampling	CI	2024
RWKV-TS		· RWKV Blocks with Multi-head WKV Operator	CD	2024
CONTIME		· Bi-directional Continuous GRU with Neural ODE	CI	2024
MSGNet	GNN/GCN	· ScaleGraph block with Scale Identification · Multi-scale Adaptive Graph Convolution. · Multi-head Attention and Scale Aggregation	CD	2024
TMP-Nets		· Temporal MultiPersistence(TMP) Vectorization	CD	2024
HD-TTS		· Temporal processing module with Temporal message passing · Spatial processing module with Spatial message passing	CD	2024
ForecastGrapher		· Group Feature Convolution GNN (GFC-GNN)	CD	2024
WaveForM	Hybrid (CNN+GCN)	· Discrete Wavelet Transform(DWT) Module · Graph-Enhanced Prediction Module	CD	2023
BiTGraph		· Multi-Scale Instance PartialTCN(MSIPT) Module · Biased GCN Module	CD	2024
TSLANet	Hybrid (TF+CNN)	· Adaptive Spectral Block · Interactive Convolutional Block	CI	2024
DERITS	Hybrid (CNN+MLP)	· Frequency Derivative Transformation(FDT) · Order-adaptive Fourier Convolution Network(OFCN)	CD	2024
RobustTSF	Model-agnostic	· RobustTSF Algorithm	-	2024
PDLS		· Loss Shaping Constraints · Empirical Dual Resilient and Constrained Learning	-	2024
Leddam		· Learnable Decomposition Module · Dual Attention Module	CD	2024
InfoTime		· Cross-Variable Decorrelation Aware Feature Modeling (CDAM) · Temporal Aware Modeling (TAM)	CD	2024
CCM		· Channel Clustering & Cluster Loss · Cluster-aware Feed Forward	CD	2024
HCAN		· Uncertainty-Aware Classifier(UAC) · Hierarchical Consistency Loss(HCL) · Hierarchy-Aware Attention(HAA)	-	2024
TDT Loss		· Temporal Dependencies among Targets(TDT) Loss	-	2024

- : Indicates that the feature is model-agnostic and depends on which backbone model is applied.

PromptCast (Xue and Salim, 2023) introduces a framework where numerical time series sequences are input as text prompts to a large language model. Then the model outputs the forecasting results in a question-answering format. This approach integrates domain-specific knowledge through text and provides forecasting results in a human-friendly format.

Both GPT4TS and PromptCast rely on fine-tuning to achieve their results. In contrast, without fine-tuning, **LLMTime** (Gruver et al, 2023) has demonstrated impressive zero-shot forecasting capabilities. The main idea of LLMTime is encoding time series data as a string of numerical digits, which emphasizes the importance of preprocessing and its inherent dependencies. To align the time series modality with the language modality, **Time-LLM** (Jin et al, 2024) introduces the concept of patch reprogramming. This concept aims to mitigate the challenges of a large reprogramming space and attempts to connect time series local characteristics with language semantics, such as “short up”. While the patch reprogramming offers more flexibility by reducing the reliance on large-scale time series corpora, it also presents the challenge of adapting time series data to fit the characteristics of large language models.

4.3.2 Pre-training

Another approach that focuses on building large-scale time series corpora to pre-train time series foundation models from scratch has emerged. An example is **Lag-LLaMA** (Rasul et al, 2024), which uses a decoder-only Transformer to generate forecasting results, enabling probabilistic forecasting. Lag-LLaMA also consolidates 27 publicly available forecasting datasets from various domains to create a comprehensive pre-training corpus. On the other hand, **TimesFM** (Das et al, 2024) extends beyond publicly available forecasting datasets by incorporating additional pre-training corpora based on Google Trends and Wiki Pageview statistics. It also utilizes synthetic datasets generated from piece-wise linear trends, autoregressive processes, and sine and cosine functions to capture universal characteristics. The entire pre-training corpus spans hundreds of billions of time steps. While most of the time series foundation models rely on temporal embedding, **CHRONOS** (Ansari et al, 2024) takes a different idea by learning a patch-based tokenizer, similar to traditional language models, to capture the intrinsic “language” of time series data.

Traditional foundation models often overlook the relationships between variables in multivariate time series forecasting, typically extending to multivariate forecasting by independently combining univariate forecasts based on channel independence. **Uni2TS** (Woo et al, 2024) addresses this limitation by explicitly considering the expansion to arbitrary multivariate time series using variate IDs. Additionally, it leverages a large-scale time series dataset called LOTSA, which accounts for multi-domain and multi-frequency characteristics.

4.4 Advance of Diffusion Models

Diffusion models are renowned generative models that have gained prominence for their ability to produce high-quality images, as demonstrated by **DALL-E2** (Ramesh et al, 2022), **Stable Diffusion** (Rombach et al, 2022a), and **Imagen** (Saharia et al, 2022). In addition to their success in image generation, diffusion models have shown excellent performance in various fields, such as audio generation, natural language processing, and video generation (Cao et al, 2024). Consequently, there has been a growing number of research papers exploring their application in time series forecasting.

A diffusion model operates through two primary processes: the forward process and the reverse process. In the forward process, noise is gradually added to the data until it transforms into complete noise. Conversely, the reverse process reconstructs meaningful data from random noise. By training a denoising network, the model generates data by injecting random values and processing them through the reverse process.

For forecasting tasks, it is crucial to employ conditional generation, where conditions are incorporated into the model to produce data that aligns with the given conditions. In this context, historical data is used as a condition and injected into the model to enable it to learn and predict future values. Fig. 12 illustrates the diffusion model process, visualizing the transition from data to noise and back to data. Here, the addition of conditions explains the concept of the conditional diffusion model.

Diffusion-based models for time series forecasting have achieved significant performance improvements through the introduction of effective conditional embedding, the integration of time series feature extraction, and advancements in the diffusion models themselves. The following sections explain the key characteristics of diffusion-based models according to these classification criteria, and Table 9 provides a clear summary of this information.

Table 8: Taxonomy and Methodologies of Foundation Models for Time Series Forecasting

Approach	Model Name	Main Improvement & Methodology	Publication
Sequential modeling with LLM	GPT4TS	· Demonstrate the effectiveness of LLM for time series modeling · Fine-tune the layer normalization and positional embedding parameters	2023
	PromptCast	· Enable text-level domain-specific knowledge for TSF · Cast TSF problem into question and answering format	2023
	LLMTime	· Zero-shot TSF with pre-trained LLMs · Covert time series input into a string of digits	2023
	Time-LLM	· Align time series modality into text modality · Introduce patch reprogramming to mitigate a large reprogramming space	2024
Pre-training	Lag-Llama	· First pre-training based time series foundation model · Pre-train a decoder-only model with autoregressive loss	2024
	TimesFM	· Pre-trained with hundreds of billions time steps · Autoregressive decoding with arbitrary forecasting length	2024
	CHRONOS	· Learning the language of time series · Utilize tokenizer to capture the intrinsic language of time series	2024
	UniTS	· Explicit consideration of multivariate TSF · Provide variate IDs to directly consider multiple variables	2024

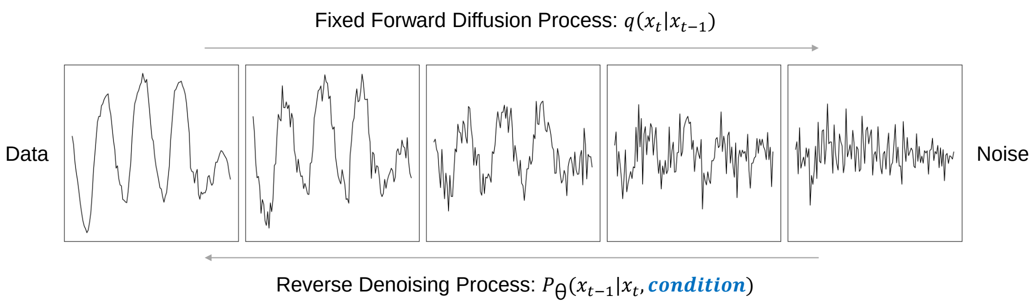

Fixed Forward Diffusion Process: $q(x_t|x_{t-1})$
Data
Noise
Reverse Denoising Process: $P_\theta(x_{t-1}|x_t, \boldsymbol{condition})$

Fig. 12: Conditional Diffusion Process for Time Series Data

4.4.1 Effective Conditional Embedding

Early diffusion-based models for time series forecasting focused on effective conditional embedding to guide the reverse process (Li et al, 2024a). Typically, for forecasting tasks, conditional diffusion models use past data as a condition to predict the future. Therefore, the meticulous construction of the condition is paramount for the denoising model to effectively learn the data and enhance prediction performance. This highlights the importance of data preparation in the forecasting process. This preparation allows the model to effectively utilize past information, thereby enhancing the accuracy of future time-series predictions and improving the model's ability to learn from historical data.

TimeGrad (Rasul et al, 2021) is the first notable diffusion model that operates using an autoregressive Denoising Diffusion Probabilistic Model (DDPM) (Ho et al, 2020) approach. It encodes past data with an RNN, using the hidden state in a conditional diffusion model for forecasting. The hidden state encapsulates information from past sequential data, capturing temporal dependencies effectively. The denoising network employs dilated ConvNets with residual connections, adopted from WaveNet (Van Den Oord et al, 2016b) and DiffWave (Kong et al, 2020), which are designed for audio generation. Unlike the commonly used U-net for image synthesis in diffusion models, TimeGrad utilizes a broader receptive field suitable for time series data, similar to audio. **CSDI** (Tashiro et al, 2021) captures temporal and feature dependencies of time series using a two-dimensional attention mechanism. Designed for both forecasting and imputation tasks, CSDI replaces RNNs with Transformers for imputation since RNNs are not suitable. The attention mechanism learns the relationships across all positions in the input sequence, capturing long-term dependencies effectively. Therefore, for long-term time series forecasting, using attention is more advantageous than RNNs and dilated convolution, as seen in TimeGrad. **SSSD** (Alcaraz and Strodthoff, 2022) demonstrates that replacing the Transformer used as the denoising network in CSDI with the S4 model (Gu et al, 2021a) yields superior performance. This is because S4 is more efficient and better at capturing long-term dependencies compared to the high computational complexity required for Transformer attention. **TimeDiff** (Shen and Kwok, 2023) enhances prediction accuracy by using future mixups during training, mixing past data with actual future values to generate conditional signals. Including some actual future values helps the model create effective conditional signals for more accurate predictions. Additionally, it addresses the boundary disharmony issue in non-autoregressive models like CSDI and SSSD. TimeDiff uses a simple statistical model, the Linear Autoregressive model, to provide an initial guess, alleviating the boundary disharmony problem. **TMDM** (Li et al, 2024a) offers an extreme form of effective conditional embedding. It states that the best condition is the prediction itself, using prediction values from transformer-based models like Informer (Zhou et al, 2021) or Autoformer (Wu et al, 2021), which have shown good performance in time series forecasting tasks, as the condition. This allows the Transformer to handle the estimated mean while the diffusion model focuses on uncertainty estimation. Unlike previous researches that utilize conditional embeddings only in the reverse process, TMDM uses conditional information as prior knowledge for both the forward and reverse processes.

4.4.2 Time Series Feature Extraction

Time series data hides various features, and there are techniques to extract these unique characteristics effectively. Thus, many works have emerged that combine time series feature extraction methods with diffusion models to understand the complex patterns in time series data and improve prediction performance.

Decomposition

Decomposition techniques involve breaking down time series data into components such as trend, seasonality, and irregularity, analyzing the unique patterns of each component. **Diffusion-TS** (Yuan and Qiao, 2024) points out that traditional methods cannot properly learn each component because the forward process causes the components to collapse. Therefore, it models the decomposed time series data individually during the diffusion process, learning each component, such as trend, seasonality, and residuals, independently and then recombining them to restore the entire time series data.

Frequency Domain

Fourier analysis, a type of decomposition technique, converts time series data into the frequency domain to analyze periodic components. This method helps identify periodic patterns and remove noise. Crabbé et al (2024) explore the idea that the representation of time series data in the frequency domain can provide useful inductive biases for score-based diffusion models. This paper demonstrates that the components of time series data are more clearly represented in the frequency domain, and diffusion models in frequency domain outperform those in the time domain.

Multi-Scale

Multi-scale techniques analyze time series data at various time scales, effectively extracting long-term trends and diverse features. This approach plays a crucial role in understanding the complex patterns of time series data to improve prediction performance. **MG-TSD** (Fan et al, 2024b) observes that the forward process of diffusion models aligns with gradually smoothing fine data into coarser representations. It suggests that coarse-granularity data can serve as effective guides in the intermediate stages of diffusion models. In other words, the initial stages of the reverse process involve coarse-granularity data, guiding the process to intermediate-stage targets. This multi-granularity level approach helps learn various levels of information, enhancing prediction stability and accuracy. **mr-Diff** (Shen et al, 2024) constructs time series data at multiple resolutions, performing sequential predictions at each resolution. In the initial stages, it predicts coarse data, and in subsequent stages, it uses these predictions as conditions to generate finer data gradually. This structure incrementally adds finer patterns and noise at each stage, ultimately reconstructing high-resolution time series data. This allows for effective prediction of both long-term trends and short-term fluctuations in time series data.

4.4.3 Additional Approaches

Beyond the two main approaches mentioned earlier, there are numerous examples where techniques evolving from the diffusion framework itself are applied to models for time series forecasting.

Score-Based Generative Modeling through SDEs

Score-Based Generative Modeling through Stochastic Differential Equations (SDEs) (Song et al, 2020) serves as continuous energy-based generative models, handling data in a continuous time domain, thus reflecting natural and precise changes and noise in the real world. The previously discussed methods are based on **DDPM** (Ho et al, 2020), which add and remove noise incrementally using fixed-time steps. However, the DDPM approach faces challenges such as specific functional constraints and sensitivity to hyperparameters. Some research applying SDEs overcomes these limitations and explores diverse approaches. **ScoreGrad** (Yan et al, 2021) is the first work to apply SDE, overcoming the constraints of DDPM-based models and offering a more flexible and powerful generative model. **D^{3}M** (Yan et al, 2024) addresses the limitations of traditional SDEs, such as the complexity of determining drift and diffusion coefficients and slow generation speed, by utilizing a decomposable noise-removal diffusion model based on explicit solutions. This method reduces computational complexity through clear SDE formulations and separates the signal decay and noise injection processes in model design. As a result, it enhances model efficiency and accuracy while accelerating generation speed. Crabbé et al (2024) improve denoising score matching in the frequency domain by using mirrored Brownian motions instead of standard Brownian motion, emphasizing the symmetry between components when applying SDEs.

Latent Diffusion Model

Latent Diffusion Model (Rombach et al, 2022b) is a generative model that operates not on the original data directly but within a latent space where the diffusion process takes place. Transforming data into the latent space reduces complexity and stabilizes the training process, resulting in high-quality outputs. **LDT** (Feng et al, 2024) applies the concept of the latent diffusion model to time series data, addressing the non-stationarity issues that often arise during the compression into the latent space. Dynamically updating statistical properties during the autoencoder training process effectively overcomes these challenges and enhances model performance.

Guidance

Some works use guidance instead of explicitly feeding conditions into the denoising network for forecasting. **Diffusion-TS** (Yuan and Qiao, 2024) employs classifier guidance, using a separate classifier to guide the sampling process through the gradients of classifier. This method maintains the basic unconditional diffusion model while performing conditional generation tasks through various classifiers. It generates samples that are better aligned with specific conditions, resulting in higher-quality outputs. However, classifier guidance requires a classifier for each time step, which necessitates training a new classifier instead of using pre-trained ones. **LDT** (Feng et al, 2024) uses classifier-free guidance, learning both conditional and unconditional models within a single model to perform conditional sampling. This work eliminates the need for an additional classifier and implicitly obtains a classifier, making implementation simpler and more efficient. **TSDiff** (Kollovieh et al, 2024) proposes observation self-guidance, allowing the use of an unconditional diffusion model for conditional generation without separate networks or modifications to the training process.

4.5 Debut of the Mamba

4.5.1 History of the SSM(State Space Model)

One of the notable recent developments is the emergence of a new architecture called **Mamba** (Gu and Dao, 2023). In an atmosphere previously influenced by Transformers, Mamba has garnered significant interest from researchers and is rapidly establishing its own ecosystem.

RNNs, which were once the mainstream in sequence modeling, lost their dominance after the advent of Transformers. This was due to the limitations in information encapsulation of a “context” (single vector) in RNN-based encoder-decoder models and the slow training speed inherent in their recurrent nature. In contrast, the parallelism of the attention mechanism and its ability to focus on all individual pieces of information overcame the limitations of RNNs and demonstrated superior performance. However, new challenges have emerged with Transformers, such as the quadratic computational complexity, which limits the window length, and the increased memory requirements for processing long sequences. Subsequently, many efforts have been made to overcome the limitations of both approaches while preserving their advantages. In this context, some research that continues the philosophy of RNNs has turned its attention to state space models (SSMs) (Kalman, 1960).

SSM is a mathematical framework used to model the dynamic state of a system that changes over time, compressing only the essential information for effective sequential modeling. SSM describes the relationship between the internal state of the system and external observations and is used in various fields such as control theory, signal processing, and time series analysis. It comprises a ‘State equation’ that explains how the internal state changes over time and an ‘Observation equation’ that explains how the internal state is transformed into external observations. Although it is a continuous model that performs linear transformations on the current hidden state and input, it can handle discrete sequences like time series through discretization. It closely resembles RNNs in that it combines observation data and hidden state data.

Table 9: Taxonomy and Methodologies of Diffusion Models for Time Series Forecasting

Main Improvement	Model Name	Main Methodology	Diffusion Type	Conditional Type	Publication
Effective Conditional Embedding	TimeGrad	· Autoregressive DDPM using RNN & Dilated Convolution	DDPM	Explicit	2021
	CSDI	· 2D Attention for Temporal & Feature Dependency · Self-supervised Training for Imputation	DDPM	Explicit	2021
	SSSD	· Combination of S4 model	DDPM	Explicit	2023
	TimeDiff	· Future Mixup · Autoregressive Initialization	DDPM	Explicit	2023
	TMDM	· Integration of Diffusion and Transformer-based Models	DDPM	Explicit	2024
Time-series Feature Extraction	Diffusion-TS	· Decomposition techniques · Instance-aware Guidance Strategy	DDPM	Guidance	2024
	Diffusion in Frequency	· Diffusing in the Frequency Domain	SDE	Explicit	2024
	MG-TSD	· Multi-granularity Data Generator · Temporal Process Module · Guided Diffusion Process Module	DDPM	Explicit	2024
	mr-Diff	· Integration of Decomposition and Multiple Temporal Resolutions	DDPM	Explicit	2024
SDE	ScoreGrad	· Continuous Energy-based Generative Model	SDE	Explicit	2021
	D^3 M	· Decomposable Denoising Diffusion Model based on Explicit Solutions	SDE	Explicit	2024
Latent Diffusion Model	LDT	· Symmetric Time Series Compression · Latent Diffusion Transformer	DDPM	Guidance	2024
Guidance	TSDiff	· Observation Self-guidance	DDPM	Guidance	2023

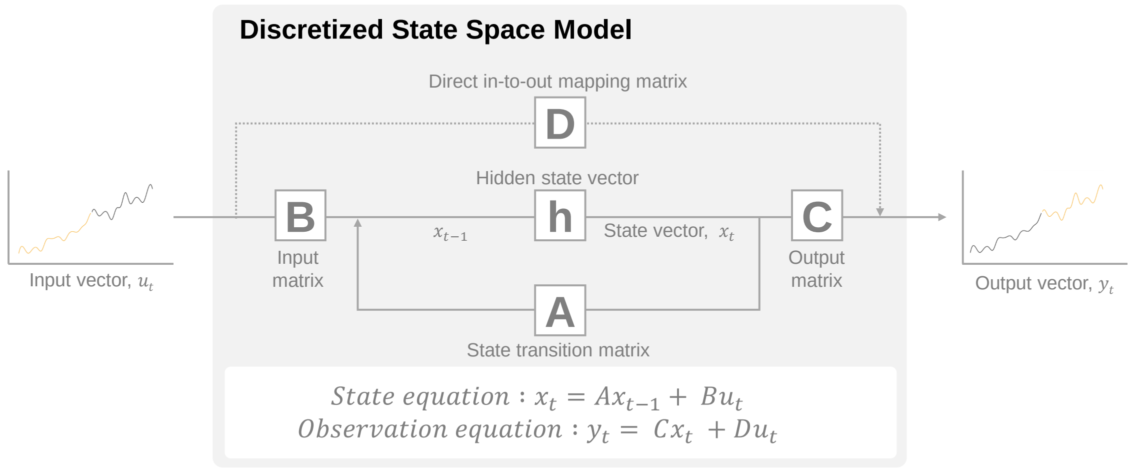

Discretized State Space Model
Direct in-to-out mapping matrix
D
Hidden state vector
B
h
C
x_{t-1}
State vector, x_t
Input vector, u_t
Input matrix
Output matrix
Output vector, y_t
A
State transition matrix
State equation : $x_t = Ax_{t-1} + Bu_t$
Observation equation : $y_t = Cx_t + Du_t$

Fig. 13: Diagram of Discretized State Space Model

However, SSMs have different characteristics from RNNs in that they are linear and time-invariant (LTI). This means that the operations performed on each token do not vary, allowing for the pre-computation of global kernels. In other words, the parameter matrices of the system do not change over time and operate consistently across time. Therefore, the ability to precompute globally applicable kernels enables parallel processing, which can overcome the limitations of RNNs.

Early applications of SSMs, such as the S4 (Gu et al, 2021b) model, utilized diagonalization of the transition matrix to effectively model long-term dependencies in long and complex sequence data. These models were able to achieve high performance when combined with deep learning architectures like Transformers, RNNs, and CNNs. Subsequently, based on S4, advanced blocks like H3 (Fu et al, 2023) were developed, which hierarchically structured convolution, gating, and MLP to provide more efficient and powerful sequence modeling.

4.5.2 Introduction of the Mamba

SSMs have fixed state transition and observation equations, which limits their ability to flexibly handle input data. Furthermore, due to their inherent continuous system origins, they are weaker in information-dense and discrete domains such as text.

Mamba addresses these limitations of SSMs by introducing a advanced deep structured SSM model with selective SSM. It is designed so that the parameters of the SSM dynamically change depending on the input, allowing for selective memory or disregard of specific parts of the sequence, enabling efficient data processing. However, this approach, while enhancing the system's flexibility to better learn complex patterns, sacrifices the parallel processing advantages of SSMs. To compensate, traditional techniques such as kernel fusion, parallel scan, and recomputation are applied to efficiently implement selective SSMs on GPUs. Mamba adopts a simplified architecture centered around selective SSM, replacing the first multiplicative gate in the traditional H3 block with an activation function and incorporating an SSM into the gated MLP block. Due to the absence of attention mechanisms or separate MLP blocks, computational complexity is reduced, resulting in efficient training, fast inference, and high scalability and versatility.

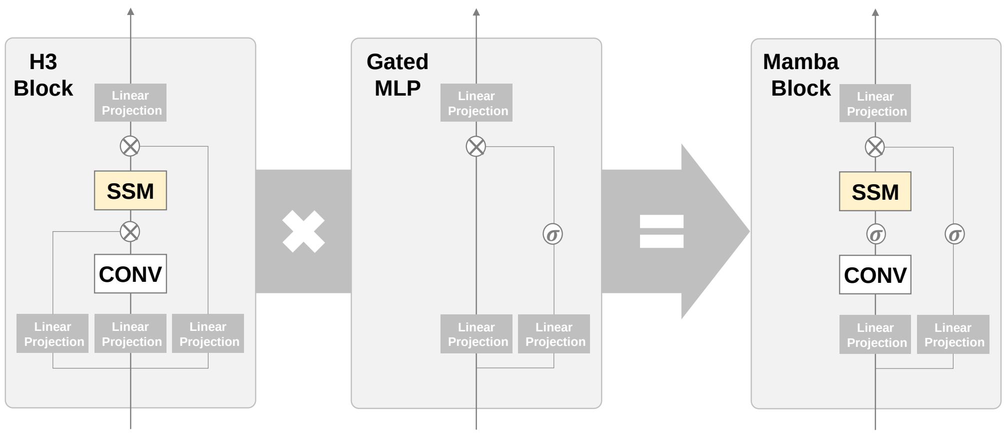


Fig. 14: Structure of the Mamba Block

4.5.3 Applications of the Mamba

In the field of time-series forecasting, there is a growing trend to apply Mamba. A lot of variants based on Mamba have been proposed to address TSF tasks, and these models are reported to exceed the performance of Transformer SOTA models.

Most of these models typically share the following common features:

- They aim to ensure stability in large networks. Mamba, based on SSM, heavily relies on the eigenvalues of the system's dynamic matrix. Since the general solution of the system is expressed as an exponential function of the eigenvalues, if the real part of the eigenvalues is positive, the system cannot converge and becomes unstable. Therefore, a key feature is the ideation aimed at improving this instability.
- They explicitly incorporate the learning of channel correlations. Recent research has demonstrated the superiority of channel dependence (CD) learning strategies, and the adoption of the CD strategy by the iTransformer, which shows SOTA performance, has further heightened this trend. This is a common feature observed in other foundational models as well.
- They incorporate various techniques from transformer-based models. Mamba-based models, which integrate various techniques from Transformers such as patching, frequency domain application, bidirectional processing, and FFN incorporation, are demonstrating significant performance improvements. The diverse techniques developed over many years of research on Transformer models provide valuable ideation for Mamba variants.

The following are representative examples of Mamba-based models for handling TSF tasks and their key technical features.

Embedding and Multi-Scale Learning

In this section, we introduce methodologies for embedding and learning from data at various scales. These approaches primarily focus on richly extracting information to capture long-term dependencies and context.

TimeMachine (Ahamed and Cheng, 2024) aims to capture long-term dependencies and context at different scales through two-stage embeddings. It is divided into internal and external embeddings based on embedding dimensions, and each section consists of two parallel Mamba modules that explore temporal dependencies and channel dependencies, respectively. The internal Mamba module operates at a low resolution to capture both global and local contexts, while the external Mamba module operates at a high resolution to capture the global context.

Channel Correlation Learning

Next, we examine models that focus explicitly on learning the inter-channel correlations in time series data. These models are centered around techniques that effectively integrate information from each channel and model their interdependencies.

S-Mamba (Wang et al, 2024d) features bi-directional Mamba blocks to consider both past and future information, enabling the learning of inter-channel correlations. The role of learning temporal dependencies is assigned to the Feed Forward Network (FFN). By clearly separating the roles of Mamba blocks and FFN, computational efficiency is improved, ensuring the stability of the Mamba architecture in large networks. In **SiMBA** (Patro and Agneeswaran, 2024b), channel modeling is achieved through Einstein FFT (EinFFT), while sequence modeling is handled by Mamba modules. After applying the Fast Fourier Transform (FFT), the real and imaginary parts are separated, and their respective weights are learned. Channel mixing is performed through Einstein Matrix Multiplication (EMM), creating new data that reflects the relational information between channels, thus internalizing channel relationships. Additionally, to ensure system stability, eigenvalues of the state matrix are adjusted to be negative through Fourier transformation and nonlinear layers. **MambaTS** (Cai et al, 2024c) reconstructs sequences by integrating past information of all variables to learn channel correlations. Since variable information is integrated in advance, unnecessary convolution in Mamba blocks is removed to enhance computational efficiency, and dropout is applied to Temporal Mamba Blocks (TMB) to reduce overfitting. Additionally, Variable Permutation Training (VPT) is introduced to dynamically determine the optimal order of integrated variable information, enabling predictions based on the optimal sequence of variables. **C-Mamba** (Zeng et al, 2024) generates new virtual sample data by linearly combining different channels (channel mixup), which is expected to enhance generalization performance. It uses a main block that combines patched Mamba modules with attention modules for learning channel relationships.

Sequence Information and Dependency Learning

This category emphasizes methods for learning the sequential information and dependencies in time series data. It proposes various techniques for modeling both long-term and short-term dependencies within the series.

Mambaformer (Xu et al, 2024a) is a hybrid model that combines Mamba with a Transformer decoder framework. Since the Mamba block naturally internalizes the sequence order information, positional embedding is unnecessary. Long-term dependencies are learned through the Mamba block, while short-term dependencies are captured through the self-attention layer, effectively capturing the overall dependencies. This approach overcomes the computational efficiency limitations of attention mechanisms. **Bi-Mamba+** (Liang et al, 2024a) integrates patching techniques to finely learn inter-dependencies in the data. To preserve long-term information, it introduces Mamba+ blocks, which add a forget gate to the Mamba block. It also employs a Bi-Mamba+ encoder to process input sequences bidirectionally. Using the Spearman correlation coefficient, the Series-Relation-Aware (SRA) decider is designed to automatically select channel tokenization strategies (CI or CD). **DTMamba** (Wu et al, 2024) is composed of Dual Twin Mamba blocks, effectively learning long-term dependencies in time series data channel independently. Each Twin Mamba block consists of two parallel Mamba structures that process the same input data to capture different patterns effectively. One Mamba structure learns detailed patterns and short-term variations, while the other learns overall patterns and long-term trends.

Theoretical Frameworks and Efficient Modeling

Lastly, we explore models that introduce new theoretical frameworks or propose efficient modeling techniques. These models focus on effectively capturing the dynamic characteristics of time series data and employ theoretical approaches and methodologies that enhance computational efficiency.

Time-SSM (Hu et al, 2024a) proposes a theoretical framework called the Dynamic Spectral Operator, which extends the Generalized Orthogonal Basis Projection (GOBP) theory for efficient use of SSM. The Dynamic Spectral Operator explores the changing spectral space over time to effectively capture the dynamic characteristics of time series data. To implement this, a novel variant of the SSM basis called Hippo-LegP is proposed, enabling more precise modeling of time series data and achieving optimal performance through S4D-real initialization. This allows it to demonstrate excellent performance with only about one-seventh of the parameters required by Mamba models. **Chimera** (Behrouz et al, 2024) features the use of 2D SSM to capture dependencies independently along the time and variable axes. By updating states in parallel along both axes, it achieves efficient computation.

The emergence of a new deep learning architecture, Mamba, is causing a shift in the long-standing hegemony of deep learning architectures. In current TSF tasks, which increasingly deal with long-term sequences, Mamba's strengths—efficient sequence processing, selective information retention, simplified architecture, and hardware optimization—prove to be highly valuable. These features allow Mamba-based models to overcome the limitations of Transformers, showing rapid performance improvements in a short period and suggesting a new direction for deep learning modeling. The growth trajectory of Mamba raises attention to whether it will become the new dominance in this field.

5 TSF Latest Open Challenges & Handling Methods

5.1 Channel Dependency Comprehension

Spread of Channel Independent Strategy

Multi-variate time series (MTS) forecasting primarily hinges on how well it can learn the short-term and long-term temporal dependencies. However, many recent real-world datasets predominantly deal with multivariate data, where the relationships between variables carry significant semantic information. As the relationships between variables become more complex, models can provide more information and thus improve predictive performance by leveraging this complexity.

Traditionally, it has been vaguely assumed that understanding the relationships between variables in time series forecasting problems would make better performance. Despite being obtained from different instruments, the data observe the same phenomenon, which intuitively suggests that they offer rich interpretations from various perspectives. However, with the PatchTST (Nie et al, 2022) model adopting a channel-independent (CI) strategy and achieving state-of-the-art (SOTA) performance, research has begun to question the previously assumed channel-dependent (CD) strategy. These studies have shown that high performance can be attained without learning the interactions between variables.

The CI strategy simplifies the model by excluding inter-variable modeling, allowing it to focus solely on learning the temporal dependencies of each channel. This approach reduces model complexity and enables faster inference, while also mitigating the risk of overfitting due to noise from inter-variable interactions. Additionally, since the addition of new channels does not affect the model, it can adapt flexibly to changes in data. These advantages have led many studies to adopt the CI strategy, resulting in improved performance. However, the CI strategy did not consistently show superior performance across all studies. The CD strategy still demonstrated high performance in many studies, such as iTransformer (Liu et al, 2023). Additionally, both strategies showed inconsistent performance depending on the datasets used. Consequently, without clear justification, both CI and CD strategies were employed according to the researchers' preferences for a period of time.

Importance of Learning Channel Correlations

Learning the correlations between variables remains important. In multivariate time series data, each variable does not change independently but is interdependently related to others. Even if these relationships sometimes introduce noise or fail to provide critical information, the relationships themselves are not devoid of meaning. The information from multivariate variables often intertwines to create complex patterns that cannot be captured by a single variable. Understanding the correlations between multiple variables helps interpret these complex patterns.

Modeling the correlations between variables is also crucial for improving prediction accuracy. Especially when dealing with long sequence patterns, it is essential to understand the numerous local patterns within them. During this process, important causal relationships are often derived from other variables or exogenous factors. In the case of noisy data, learning the correlations between variables can help extract key information effectively. By comprehensively observing the information from multiple variables, it is possible to emphasize important features and complement missing information.

What Makes CI Look Better?

If the CD strategy is important and has the potential for significant performance improvements, why do many studies show that the CI strategy performs better? To answer this question, some research has been conducted, and the following summary can be provided based on the **"The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting"** (Lu Han, 2024) paper.

According to the study, an examination of the Auto-correlation Function (ACF) values between the training set and test set of the benchmark data, which have been used in various experiments,

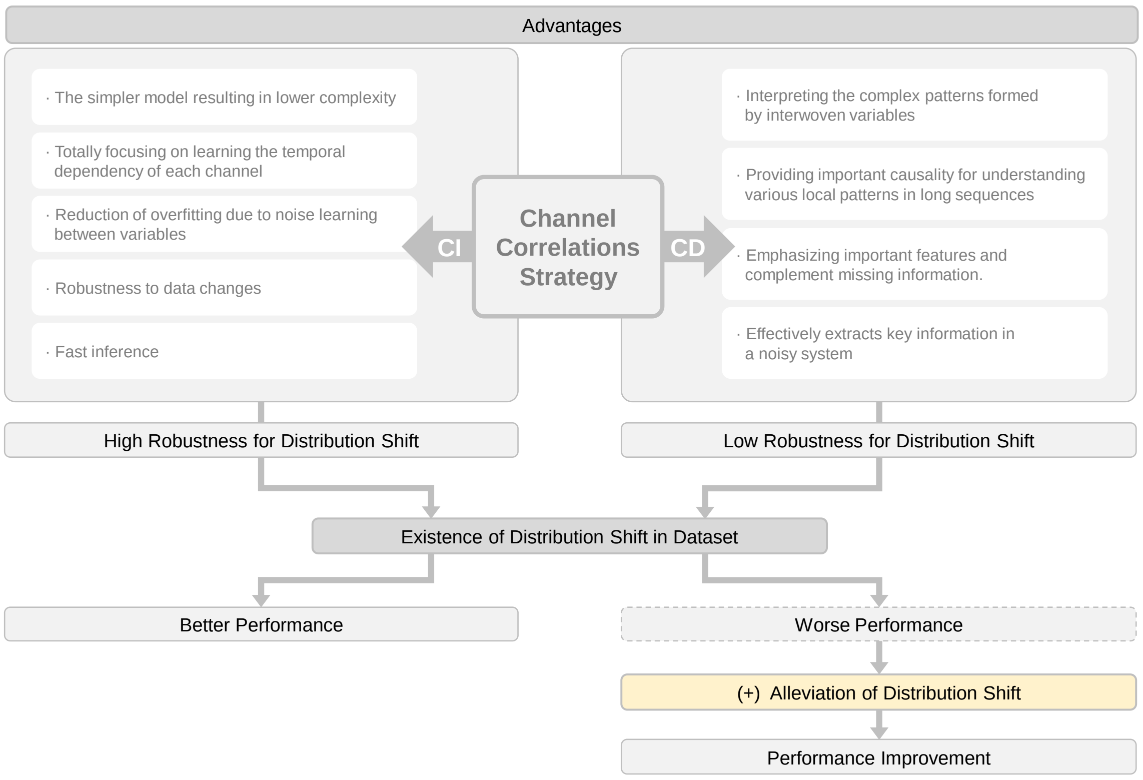
Table 10: Taxonomy and Methodologies of Mamba Models for Time Series Forecasting

Main Improvement	Model Name	Main Methodology	Channel Correlation	Base	Publication
Embedding and Multi-Scale Learning	TimeMachine	· Integrated Quadruple Mambas	CD	Mamba	2024
Channel Correlation Learning	S-Mamba	· Channel Mixing: Mamba VC Encoding Layer · Sequence Modeling: FFN TD Encoding Layer	CD	Mamba MLP	2024
	SiMBA	· Channel Mixing: Einstein FFT (EinFFT) · Sequence Modeling: Mamba	CD	Mamba	2024
	MambaTS	· Temporal Mamba Block (TMB) · Variable Permutation Training (VPT)	CD	Mamba	2024
	C-Mamba	· Channel Mixup · C-Mamba Block (PatchMamba + Channel Attention)	CD	Mamba	2024
Sequence Information and Dependency Learning	Mambaformer	· Mambaformer (Attention + Mamba) Layer	CI	Mamba Transformer	2024
	Bi-Mamba+	· Series-Relation-Aware (SRA) Decider · Mamba+ Block · Bidirectional Mamba+ Encoder	CI/CD	Mamba	2024
	DTMamba	· Dual Twin Mamba Blocks	CI	Mamba	2024
Theoretical Frameworks and Efficient Modeling	Time-SSM	· Dynamic Spectral Operator with Hippo-LegP	CD	Mamba	2024
	Chimera	· 2-Dimensional State Space Model	CD	Mamba	2024

revealed a distribution shift. The ACF can identify the correlation between data values at specific time lags in time series data.

A distribution shift refers to the difference in statistical properties between the training data and test data extracted from the same dataset. Distribution shifts can occur for various reasons, often due to anomalies in the training series or variations in the trend, but sometimes the exact cause cannot be clearly defined. An important finding from the study is that the CI strategy demonstrates relatively greater robustness to distribution shifts. Since the CI strategy relies on the average ACF of all channels, it is less sensitive to distribution shifts compared to the CD strategy, which depends on the ACF variations of individual channels. Additionally, by excluding inter-variable relationships, the CI strategy simplifies the model, reducing the likelihood of overfitting and enhancing robustness. Therefore, the CI strategy has been able to demonstrate good performance on benchmark datasets where distribution shifts are present.

However, this does not imply that the CI strategy is superior to the CD strategy. It merely suggests that, in some datasets, the positive effects resulting from the robustness of the CI strategy outweigh the advantages of learning inter-variable relationships. Conversely, if distribution shifts can be effectively alleviated, the CD strategy can provide much more useful information than the CI strategy. To this end, even the application of simple regularization methods, such as Predict Residual with Regularization (PRReg), adapting low-rank layers, and using the MAE loss function, can reverse the performance gap. Furthermore, various distribution shift alleviation methodologies have been researched, and these can be applied in a model-agnostic manner. The details of these methodologies will be discussed in the next section, and the previous content has been illustrated in Fig. 15



Advantages
· The simpler model resulting in lower complexity
· Totally focusing on learning the temporal dependency of each channel
· Reduction of overfitting due to noise learning between variables
· Robustness to data changes
· Fast inference
CI
Channel Correlations Strategy
CD
· Interpreting the complex patterns formed by interwoven variables
· Providing important causality for understanding various local patterns in long sequences
· Emphasizing important features and complement missing information.
· Effectively extracts key information in a noisy system
High Robustness for Distribution Shift
Low Robustness for Distribution Shift
Existence of Distribution Shift in Dataset
Better Performance
Worse Performance
(+) Alleviation of Distribution Shift
Performance Improvement

Fig. 15: Comparison of CI and CD Strategies in Channel Correlations

Recent Approaches

As the effectiveness of learning channel correlation has been validated, recent multivariate time series forecasting problems have predominantly adopted the CD strategy. In contrast, the CI strategy is increasingly applied in a limited way, primarily in univariate models focused on temporal dependency.

Approaches to addressing recent MTS problems can be broadly categorized into three types. The first approach involves explicitly integrating modules for channel mixing into the backbone model.

Most models adopt this strategy, aiming to maximize effectiveness by explicitly incorporating channel correlation. However, as discussed earlier, distribution shifts in the dataset can potentially degrade the performance of the CD strategy, necessitating the introduction of appropriate alleviation methods. By employing this combination, the strengths of learning channel correlations can be applied more reliably, leading to enhanced model performance. The distribution shift alleviation methods will be discussed in detail in Section 5.2.

The second approach implicitly incorporates channel correlations into the input. Instead of explicitly integrating channel learning modules into the backbone model, this method takes a preprocessing approach by mixing channel information in advance to create a new input, which is then fed into the prediction model. This new input combines the mixed channel information with the original individual channel data before being processed by the prediction model. By inherently reflecting the relationships between channels, this method allows even a simple CI backbone to achieve the effectiveness of a more complex CD model. Models like **SOFTS** (Han et al, 2024) and **C-LoRA** (Nie et al, 2024) fall into this category.

The third approach adaptively selects between the CI and CD strategies. The model is designed to operate in both directions, allowing it to choose the most effective strategy based on the characteristics of the dataset being used. In some cases, like **TSMixer** (Ekambaram et al, 2023), the user can manually select the appropriate strategy, while in others, such as **Bi-Mamba+** (Liang et al, 2024a), the model automatically determines the optimal strategy. The second and third approaches both utilize the CI strategy alongside the CD strategy, aiming to integrate the advantages of both into the model. A summary of the criteria for selecting these strategies is presented in Fig. 16

In the past, the criteria for selecting channel strategies were ambiguous, but recent research has provided clear guidelines. Based on these advancements, users can now choose the appropriate channel strategy according to the characteristics of their dataset, effectively improving model performance through various approaches.

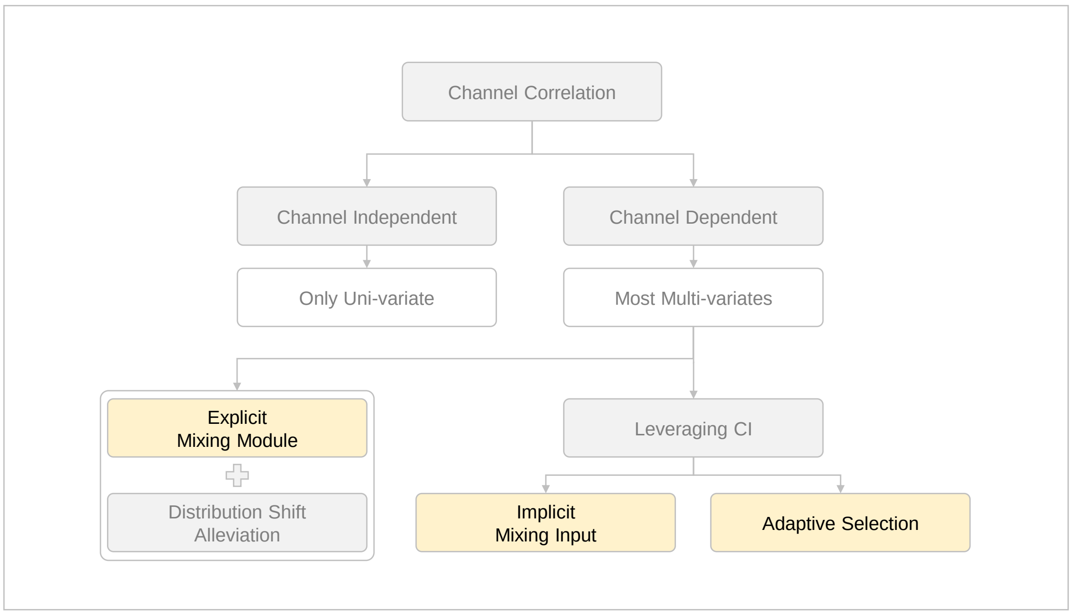


Fig. 16: Recent Approaches to Channel Strategies

5.2 Alleviation of Distribution Shift

Many real-world time series data exhibit non-stationarity, where their distribution gradually changes. For instance, trends such as increasing electricity consumption and shifts in consumer preferences can lead to changes in data distribution. This non-stationarity introduces challenges for the generalization of time series forecasting (TSF) models by creating distribution discrepancies within the training data and between the training and test data. To mitigate the issue of distribution shift caused by non-stationarity in time series forecasting, various research efforts have been proposed.

This survey focuses on one prominent approach that introduces normalization and denormalization modules within a normalization-TSF model-denormalization framework. Here, normalization is applied to the look-back window before feeding it into the TSF model, aiming to remove input statistics. As a result, the TSF model processes inputs from a less time-variant distribution. Denormalization is then applied to the forecasting window obtained from the TSF model to restore the original statistics. The denormalized forecasting window is ultimately used as the final forecast. However, calculating the appropriate statistics for normalization and denormalization in non-stationary scenarios is a non-trivial task, and numerous studies have been proposed within this framework to address this challenge.

DAIN (Passalis et al, 2019) introduces a Deep Adaptive Input Normalization layer that learns how much to shift and scale each observation end-to-end. Considering the changing distribution characteristics of time series data, this approach outperforms widely used normalization methods like batch normalization (Ioffe and Szegedy, 2015) and instance normalization (Ulyanov et al, 2016) across various domains, highlighting the importance of normalization techniques in TSF. However, DAIN omits a denormalization step, meaning it does not account for restoring statistics in the forecasting results. In contrast, **RevIN** (Kim et al, 2021), which extends instance normalization to be reversible, adopts a normalization-denormalization framework. RevIN performs instance normalization on each look-back window, followed by variable-wise multiplication of a learnable scale factor and addition of a bias factor. During de-normalization, the same parameters are applied in reverse. **NST** (Liu et al, 2022b) normalizes the look-back window in a non-parametric manner without a learnable affine transformation and then de-normalizes the prediction window using the mean and standard deviation of the look-back window. **Dish-TS** (Fan et al, 2023) learns a module that predicts the statistics of the prediction window, considering the distribution shift between the look-back window and subsequent prediction windows, and performs normalization and de-normalization accordingly. **SAN** (Liu et al, 2024d) considers the distribution shift within both the look-back window and prediction windows, performing statistical prediction on smaller slices.

Table 11: Normalization-Denormalization-based Approaches to Alleviate Distribution Shifts in Time Series Forecasting

Model Name	Main Improvement & Methodology	Publication
DAIN	· Introduce adaptive normalization · Adaptive scaling, shift, and gating layers to normalize look-back window	2019
RevIN	· Introduce normalization-denormalization framework · Denormalize prediction with reversible affine transformation parameters	2022
NST	· Non-parametric normalization-denormalization · Normalize and denormalize using mean and std of look-back window without learnable parameters	2022
Dish-TS	· Divides distribution shifts: 1) within look-back window and 2) between look-back window and forecasting window · Introduce Dual-CONET modules for statistics prediction	2023
SAN	· Predict statistics in slice-level · Slice forecasting windows and predict mean and std for each slice	2023

5.3 Enhancing Causality

Why Causal Analysis is Essential for Accurate Time Series Forecasting

Causal analysis is crucial in achieving accurate time series forecasting by better understanding the underlying factors driving data patterns. Time series data often exhibit correlations that can be misleading without a proper understanding of causality. For instance, two variables may show a strong correlation simply due to coincidence or because they are both influenced by a third, unobserved variable. Without distinguishing between correlation and causation, forecasting models risk attributing changes in the data to irrelevant or spurious factors, leading to inaccurate predictions. Causal analysis helps overcome this by identifying the true cause-and-effect relationships, ensuring the model is grounded in reality rather than coincidental patterns.

Moreover, causal analysis enhances the interpretability and practical application of forecasting models. By explaining how different variables influence the outcome, causal models provide valuable insights into the mechanisms driving the forecast, which is crucial for decision-making. Businesses and policymakers can use these insights to predict the impact of specific actions, such as policy changes or marketing strategies, and make informed decisions based on the likely effects. This ability to simulate interventions and conduct counterfactual senarios makes causal analysis an indispensable tool for accurate and actionable time series forecasting.

Research on TSF with Causality

As mentioned earlier, utilizing causality in time series forecasting not only improves prediction accuracy but also enhances model interpretability. For these reasons, research applying causal discovery information to time series forecasting models is actively ongoing. In particular, various methodologies have been proposed in fields such as healthcare, environmental science, and social sciences, where causal discovery is actively researched.

Qian et al (2023) introduce a model to predict Kuroshio Volume Transport (KVT). It employs multivariate causal analysis to discover causal relationships and selects only the variables with causal relationships to make predictions using an LSTM model. This approach captures meaningful information from causally related variables and prevents the model from being confused by unrelated variables. Mu et al (2023) propose a model for predicting the North Atlantic Oscillation. It uses information obtained through causal discovery not only for variable selection but also by directly applying it to a GCN (Graph Convolutional Network). The overall structure utilizes a symmetrical encoder and decoder of ConvLSTM, with the GCN acting as a coupler in between. Sharma et al (2022) focuses on a model for energy consumption. It applies the Granger causality test to determine the causal relationship between weather conditions and energy consumption and then integrates this causal information into a Bi-LSTM to improve energy consumption prediction accuracy. **Causal-GNN** (Wang et al, 2022) is a model aimed at epidemic forecasting. It extracts causal features using the SIRD model and incorporates them into an Attention-Based Dynamic GNN module to learn spatio-temporal patterns. **Caformer** (Zhang et al, 2024a) criticizes existing methods for failing to learn causal relationships effectively due to false correlations caused by environmental factors. To address this, it explicitly models environmental influences and removes them to obtain reliable causal relationships.

5.4 Time Series Feature Extraction

Time Series Feature Extraction is the process of extracting useful information from time series data to enhance model performance. In other words, it involves identifying and quantifying time series data into key patterns, trends, seasonality, outliers, periodicity, and statistical characteristics to transform it into a form that is suitable for model training. This process clarifies the unique patterns or characteristics of time series data, allowing models to learn them more effectively. This enhances the understanding of data, improves the predictive performance of models, and increases efficiency through data compression. The reasons why Time Series Feature Extraction is necessary are as follows:

- **Understanding the characteristics of data**
  Time series data, which is ordered sequentially over time, possesses unique characteristics such as various patterns, seasonality, and periodicity. However, these data are complex and high-dimensional, making these features not easily apparent. The continuous values in time series data tend to be similar to adjacent values, which makes it challenging to accurately distinguish the context and meaning of each data point, often resulting in a lack of semantic information. Therefore, the process of transforming time series data into an analyzable format and extracting key features is crucial for understanding the inherent behaviors and interactions within the data.
- **Explainability of data**
  Unlike traditional machine learning methods, deep learning models automatically learn useful features from data without requiring manual feature extraction. However, due to the nonlinear structure of deep learning models, it is often difficult for humans to interpret the learned features. The extracted features can better explain the structural characteristics of the data, helping to interpret and understand its meaning. This process provides critical insights for making data-driven decisions and contributes to improving explainability in complex model architectures.
- **Enhancing Model Performance**
  By summarizing complex time series data and removing noise, it emphasizes the essential signals of the data, reducing the computational burden on models and improving predictive performance.

Focusing on critical information allows the model to avoid overfitting to the training data and develop strong generalization capabilities.

Decomposing or transforming time series data into different forms before feeding it into the model is a widely used and researched approach for feature extraction. These approaches help the model focus on learning the critical information in the data, thereby reducing sensitivity to noise and overfitting. These methods are not limited to specific architectures and can be applied across various models. Fig. 17 presents the key techniques for time series feature extraction, followed by an explanation of related application models for these approaches.

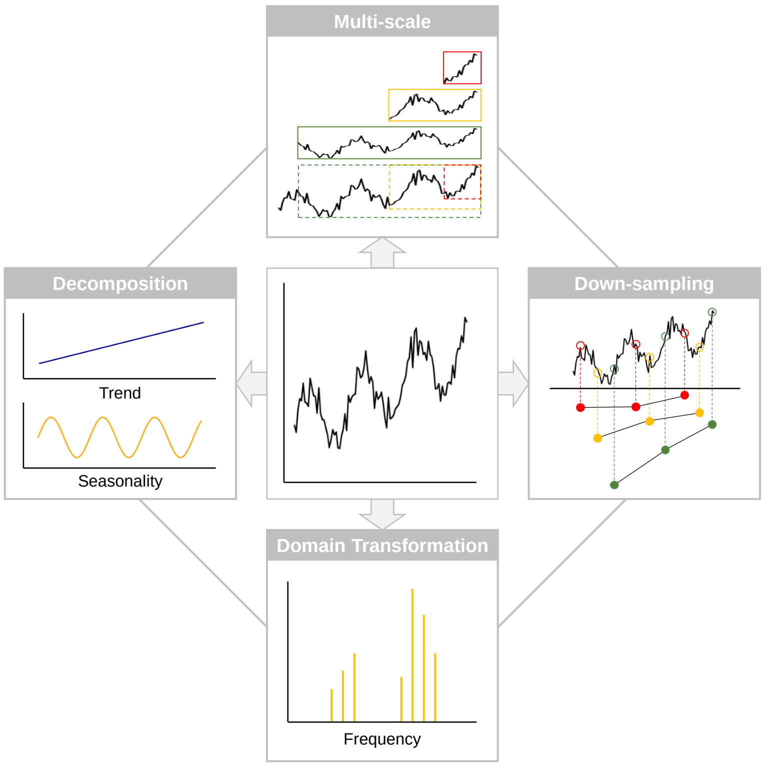


Fig. 17: Key Techniques in Time Series Feature Extraction

5.4.1 Decomposition

Time series decomposition has long been used as a fundamental time series feature extraction technique, which involves separating time series data into components such as trend, seasonality, periodicity, and residual. The advantage of decomposition is that it simplifies complex time series data by breaking it down into understandable, independent components, enabling more accurate model predictions and easier analysis and interpretation.

Moving Average Kernel

Many models apply a Moving Average kernel to the input sequence to separate trend and seasonality components. In this process, high-frequency noise or short-term fluctuations are removed, allowing long-term trends to be more clearly captured and overall patterns of increase and decrease to be emphasized. The method of using a Moving Average Kernel is widely adopted due to its computational simplicity and efficiency. However, it has limitations when dealing with complex nonlinear patterns.

Autoformer (Wu et al, 2021) goes beyond using decomposition techniques merely as preprocessing for forecasting tasks by progressively decomposing time series data throughout the prediction process within the model itself. Similarly, **CrossWaveNet** (Huang et al, 2024b)employs a dual-channel network to perform gradual deep cross-decomposition, enabling it to capture complex temporal patterns effectively. Likewise, models such as **FEDformer** (Zhou et al, 2022), **LTSF-Linear** (Zeng et al, 2023a), and **PDMLP** (Tang and Zhang, 2024) utilize a moving average kernel to individually model each component of the time series data, which are then recombined to make effective predictions. However, the Moving Average method lacks robustness because it is not learnable by the model. Additionally, since it assigns equal weights to each data point within the sliding window, it has limited ability to identify specific patterns. To address these limitations, **Leddam** (Yu et al, 2024) use learnable 1D convolutional kernels, which can better capture nonlinear structures and dynamic variations. Meanwhile, diffusion-based models face challenges as components like trend and seasonality can easily collapse during the diffusion process. Models like **Diffusion-TS** (Yuan and Qiao, 2024) overcome this by applying the diffusion process after decomposition, thereby preserving the characteristics of each component more effectively. Through this approach, the traditional limitations of time series decomposition are overcome, allowing for a more effective separation of trend and seasonal components.

Downsampling

The technique of using downsampling for decomposition is also frequently employed. Originating from signal processing, downsampling involves reducing the number of samples from the original signal, effectively decreasing the data by a specific ratio. Time series forecasting is typically achieved through pooling methods, where representative values are extracted from specific segments of the time series data, thereby suppressing high-frequency components and emphasizing low-frequency components. By adjusting the pooling size, various downsampling levels can be achieved, allowing for the exploration of multiple patterns within the data.

SparseTSF (Lin et al, 2024b) and **SutraNets** (Bergsma et al, 2023) use downsampling to separate data into trend and periodic components, predicting each subseries independently. This allows the model to learn each component separately and better understand the influence of each on the complex time series data, enhancing prediction accuracy. By dividing the complex time series data into various sub-series, it can be decomposed into a simpler form, thereby reducing the overall complexity of the model. Therefore, this approach enhances the model's generalization ability and helps prevent overfitting.

5.4.2 Multi-scale

The multi-scale approach analyzes data at various time scales, capturing patterns and trends at multiple levels simultaneously. This method is beneficial not only for time series data but also in fields like computer vision and natural language processing (Fan et al, 2021; Nawrot et al, 2021). By integrating information from longer time frames, the multi-scale approach plays a significant role in improving performance in time series forecasting.

MTST (Zhang et al, 2024d), **PDMLP** (Tang and Zhang, 2024), and **FTMixer** (Li et al, 2024c) extend the patching application to a multi-scale framework, where shorter patches effectively learn high-frequency patterns and longer patches capture long-term trends. **TimeMixer** (Wang et al, 2024a) and **AMD** (Hu et al, 2024b) leverage multi-scale decomposition to capitalize on the strengths of each scale, enabling the model to make more accurate predictions. **HD-TTS** (Marisca et al, 2024) uses spatiotemporal downsampling to decompose data into various scales across both time and space. **Scaleformer** (Shabani et al, 2022) and **Pathformer** (Chen et al, 2024b) apply the multi-scale concept to the overall model architecture, while HD-TTS hierarchically implements temporal and spatial downsampling. **MG-TSD** (Fan et al, 2024b) and **mr-Diff** (Shen et al, 2024) successfully integrate multi-scale by incorporating inductive biases that prioritize generating coarse data in the early stages of the diffusion reverse process.

5.4.3 Domain transformation

A common method for enhancing the expression of latent periodic features in time series data is to transform the data into the frequency domain. Techniques such as Fourier transformation, Wavelet transformation, and Cosine transformation are commonly used for this purpose. Generally, time series data exhibits periodicity, which refers to patterns that repeat at regular intervals. Frequency transformations effectively explore this periodicity. In time series analysis, frequency transformations are used with two main approaches: extracting periodic components and direct learning in the frequency domain.

Periodicity Extraction

Periodicity in time series data serves as critical information for predictive models, where features that may be hidden in the time domain can be more easily uncovered in the frequency domain. Frequency transformations are advantageous because they can remove high-frequency noise while retaining important low-frequency components, thereby improving data quality. By selectively extracting key frequency components and feeding them into the model, the process helps in learning essential patterns, enhances computational efficiency, and reduces model complexity. Some models leverage these characteristics to identify and extract various periodic patterns in the data. These adaptively extracted patterns are used either as inputs for the model or integrated into the learning process, thereby enhancing the model's predictive performance.

Autoformer (Wu et al, 2021) utilizes periodicity extracted through auto-correlation within the Attention mechanism. **TimesNet** (Wu et al, 2023) transforms periodic components into a 2D format to train the time series with CNNs. **MSGNet** (Cai et al, 2024a) uses periodic components to allow the model to determine appropriate scale levels for multi-scale analysis autonomously. However, since the periodicity extracted in this manner is based on selective sampling, it does not encompass all the latent information, and if critical information is not selected, performance degradation becomes inevitable.

Training in the Frequency Domain

The approach of directly training models in the frequency domain is widely studied as it overcomes these limitations by evenly learning all latent frequency components, ensuring that critical information is not missed. While time series data can exhibit complex and varied patterns in the time domain, these patterns can often be succinctly represented by a few dominant frequency components when transformed into the frequency domain (Zhou et al, 2022). This simplification makes the learning process more straightforward, as most of the information can be captured with a minimal set of frequency components. This method allows for an easy transition to the frequency domain while preserving the original information, thanks to various transformation and inverse transformation techniques. When time series data exhibit nonlinear characteristics, learning in the frequency domain can better capture these complex patterns than in the time domain. The frequency domain approach doesn't require consideration of time-axis variations, enabling stable performance even with non-stationary data. Furthermore, this method allows for efficient learning by compressing the data without losing important characteristics.

In the frequency domain, each frequency component is represented as a complex number, consisting of a real part and an imaginary part, each conveying different information. The real part is related to the magnitude (amplitude) of the time series data, indicating how prominent the periodic components are. The imaginary part, on the other hand, relates to the phase information of the data, determining the temporal positioning of the frequency components. By simultaneously understanding the amplitude and phase information in the frequency domain, models can accurately capture and predict complex periodic patterns in the data. The research initially focused on simply changing the domain, but more recently, it has been moving towards expanding the expressiveness of the frequency domain.

FreTS (Yi et al, 2024) and Crabbé et al (2024) transform data into the frequency domain before feeding it into the model and then convert the predicted values back into the time domain. Simply converting data to the frequency domain has limitations in achieving good model performance, leading to the development of models that overcome these challenges. Research continues to expand the representation in the frequency domain to fully utilize its rich features. **FEDformer** (Zhou et al, 2022) applies season-trend decomposition and then performs attention in the frequency domain, facilitating independent modeling of key information components. **Fredformer** (Piao et al, 2024) not only transforms data into the frequency domain but also investigates frequency bias through experiments, addressing issues in the frequency domain using frequency normalization techniques. **FITS** (Xu et al, 2024c) utilizes distinct complex-valued linear layers in the frequency domain to learn amplitude scaling and phase changes, thereby enhancing the frequency representation of input time series through interpolation learning. **DERITS** (Fan et al, 2024a) effectively handles the non-stationarity of time series data by differentiating frequency components and representing them in a static form that is easier to predict. **SiMBA** (Patro and Agneeswaran, 2024b) transforms time series data into the frequency domain using Fourier transforms, learning real and imaginary components separately, thus allowing more precise analysis of data in the frequency domain and improving model prediction performance. While frequency bands capture global dependencies well, they often struggle with local dependency capture. To overcome this, **WaveForM** (Yang et al, 2023) uses discrete wavelet transforms (DWT) to decompose time series data into various frequency bands while preserving the time information of each band. This approach captures frequency changes within specific time intervals, simultaneously capturing features from both time and frequency domains. **FTMixer** (Li et al, 2024c) proposes a method that directly utilizes data from both domains to leverage the strengths of global dependency extraction in the frequency domain and local dependency extraction in the time domain.

5.4.4 Additional approach

In addition to the methods previously described, various other approaches have been explored. A notable example is the use of High-Dimensional Embedding, which generates high-dimensional representations to better capture the essential information of the data. This approach extracts complex, multidimensional information from the original time series data and integrates it to enhance the model's predictive capabilities. For instance, **CATS** (Lu et al, 2024) creates a new Auxiliary Time Series (ATS) by combining variables from the input data and then utilizes these for prediction. **SOFTS** (Han et al, 2024) focuses on extracting common features (Core) from the variables and incorporating them into the learning process.

Efforts to effectively represent the complex features of time series data for better model learning are being explored in various forms. These efforts are essential research topics for improving performance in the challenging field of time series forecasting. Additionally, integrating these feature extraction techniques appropriately within the basic structure of existing deep learning models can create synergy, enhancing the model's predictive accuracy.

6 Conclusion

This survey has been prepared to broaden the expertise of existing researchers and to assist beginners in gaining a fundamental understanding amid the rapid growth of time series forecasting (TSF) research. By integrating key concepts of time series and the latest techniques, we aim to provide researchers with clear direction and insights, fostering the continued advancement of this field. This survey paper comprehensively reviews recent advancements in TSF, focusing on deep learning models. This survey carefully selects and includes key papers from major AI and ML conferences. Moreover, given the rapid development in this field and the significant importance of the latest research, we have also included papers published on arXiv, selected based on their experimental results and innovative contributions. TSF is a highly regarded topic, gradually expanding from the prominence of Transformer models to various architecture-based models. Traditional deep learning models like MLPs, CNNs, RNNs, and GNNs are being reassessed, and Transformers are also advancing by overcoming their previous limitations. Additionally, not only are innovative models like Mamba emerging, but models such as Diffusion are also entering the field and experiencing rapid growth. Furthermore, as the demand for foundation models in the field of TSF intensifies, pre-trained models based on large language models (LLMs) are emerging. To create a powerful predictive model, researchers need to be aware of these trends and understand both the characteristics of time series data and the strengths and weaknesses of different architectures. Taking it a step further, we aim to address common challenges in TSF and provide deep insights into future development directions. In particular, we reviewed various studies that focus on approaches to channel correlation and address the challenges of distribution shift, which frequently occur in real-world scenarios. Additionally, we emphasized the need for research that leverages causality to eliminate spurious correlations, thereby enabling a deeper understanding of the underlying essence. Lastly, we discussed the importance of studies focused on extracting features from complex time series data. Through this, we aim to present readers with potential directions for future research.

Limitations and Future Work

This survey acknowledges several limitations that could be addressed in future research.

- We skipped the detailed theoretical backgrounds of the models. This survey aims to provide a comprehensive overview, facilitating comparison and analysis, thereby allowing researchers to explore areas of interest more effectively. Although comprehensive information is primarily provided, readers can access additional details through reference links if needed.

- We left the specific differences in characteristics across various time series datasets for future work. Time series data is often domain-specific, requiring expert knowledge, which can be integrated into models to enhance performance. This underscores the importance of interdisciplinary collaboration, and future research should continue to develop in this direction.
- The aspect of interpretability could be further covered. Interpretability is crucial since understanding and trusting the model's results are critical from an application perspective. Therefore, this could also be a potential topic for subsequent studies.

Acknowledgements. We would like to express our heartfelt gratitude to Jisoo Mok, Hyeongrok Han, Bonggyun Kang, Junyong Ahn, Jiin Kim, Youngwoo Kimh, Hyungyu Lee, Juhyeon Shin, Jaihyun Lew, and Jieun Byeon from our research lab for their invaluable assistance in reviewing and providing feedback on this paper. Their insights and suggestions have significantly enhanced the quality of our work.

References

Abu-Mostafa YS, Atiya AF (1996) Introduction to financial forecasting. Applied intelligence 6:205–213

Achiam J, Adler S, Agarwal S, et al (2023) Gpt-4 technical report. arXiv preprint arXiv:230308774

Ahamed MA, Cheng Q (2024) Timemachine: A time series is worth 4 mambas for long-term forecasting. arXiv preprint arXiv:240309898

Ahmed DM, Hassan MM, Mstafa RJ (2022) A review on deep sequential models for forecasting time series data. Applied Computational Intelligence and Soft Computing 2022(1):6596397

Alcaraz JML, Strodthoff N (2022) Diffusion-based time series imputation and forecasting with structured state space models. arXiv preprint arXiv:220809399

Alghamdi T, Elgazzar K, Bayoumi M, et al (2019) Forecasting traffic congestion using arima modeling. In: 2019 15th international wireless communications & mobile computing conference (IWCMC), IEEE, pp 1227–1232

Ansari AF, Stella L, Turkmen C, et al (2024) Chronos: Learning the language of time series. arXiv preprint arXiv:240307815

Athanasopoulos G, Hyndman RJ, Song H, et al (2011) The tourism forecasting competition. International Journal of Forecasting 27(3):822–844

Bai S, Kolter JZ, Koltun V (2018) An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:180301271

Barlin JN, Zhou Q, Clair CMS, et al (2013) Classification and regression tree (cart) analysis of endometrial carcinoma: seeing the forest for the trees. Gynecologic oncology 130(3):452–456

Bartholomew DJ (1971) Time series analysis forecasting and control.

Behrouz A, Santacatterina M, Zabih R (2024) Chimera: Effectively modeling multivariate time series with 2-dimensional state space models. arXiv preprint arXiv:240604320

Benidis K, Rangapuram SS, Flunkert V, et al (2022) Deep learning for time series forecasting: Tutorial and literature survey. ACM Computing Surveys 55(6):1–36

Bergsma S, Zeyl T, Guo L (2023) Sutranets: sub-series autoregressive networks for long-sequence, probabilistic forecasting. Advances in Neural Information Processing Systems 36:30518–30533

Brown RG (1959) Statistical forecasting for inventory control. (No Title)

Cai W, Liang Y, Liu X, et al (2024a) Msgnet: Learning multi-scale inter-series correlations for multivariate time series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 11141–11149

Cai W, Wang K, Wu H, et al (2024b) Forecastgrapher: Redefining multivariate time series forecasting with graph neural networks. arXiv preprint arXiv:240518036

Cai X, Zhu Y, Wang X, et al (2024c) Mambats: Improved selective state space models for long-term time series forecasting. arXiv preprint arXiv:240516440

Cao H, Tan C, Gao Z, et al (2024) A survey on generative diffusion models. IEEE Transactions on Knowledge and Data Engineering

Center JMK (2020) Dominick's dataset

Chang S, Zhang Y, Han W, et al (2017) Dilated recurrent neural networks. Advances in neural information processing systems 30

Chen J, Lenssen JE, Feng A, et al (2024a) From similarity to superiority: Channel clustering for time series forecasting. arXiv preprint arXiv:240401340

Chen P, Zhang Y, Cheng Y, et al (2024b) Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. arXiv preprint arXiv:240205956

Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp 785–794

Chen X, Li X, Liu B, et al (2024c) Biased temporal convolution graph network for time series forecasting with missing values. In: The Twelft International Conference on Learning Representations

Chen Z, Ma M, Li T, et al (2023) Long sequence time-series forecasting with deep learning: A survey. Information Fusion 97:101819

Cheng H, Wen Q, Liu Y, et al (2024a) Robusttsf: Towards theory and design of robust time series forecasting with anomalies. ICRL

Cheng M, Yang J, Pan T, et al (2024b) Convtimenet: A deep hierarchical fully convolutional model for multivariate time series analysis. arXiv preprint arXiv:240301493

Cheng X, Chen X, Li S, et al (2024c) Leveraging 2d information for long-term time series forecasting with vanilla transformers. arXiv preprint arXiv:240513810

Cho K (2014) Learning phrase representations using rnn encoder–decoder for statistical machine translation. arXiv preprint arXiv:14061078

Cortes C (1995) Support-vector networks. Machine Learning

Coskunuzer B, Segovia-Dominguez I, Chen Y, et al (2024) Time-aware knowledge representations of dynamic objects with multidimensional persistence. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 11678–11686

Crabbé J, Huynh N, Stanczuk J, et al (2024) Time series diffusion in the frequency domain. arXiv preprint arXiv:240205933

Cryer JD (1986) Time series analysis, vol 286. Duxbury Press Boston

Danese P, Kalchschmidt M (2011) The role of the forecasting process in improving forecast accuracy and operational performance. International Journal of Production Economics 131(1):204–214. https://doi.org/https://doi.org/10.1016/j.ijpe.2010.09.006, URL https://www.sciencedirect.com/science/article/pii/S0925527310003282, innsbruck 2008

Das A, Kong W, Sen R, et al (2024) A decoder-only foundation model for time-series forecasting. In: Forty-first International Conference on Machine Learning, URL https://openreview.net/forum?id=jn2iTJas6h

Dimri T, Ahmad S, Sharif M (2020) Time series analysis of climate variables using seasonal arima approach. Journal of Earth System Science 129:1–16

Dosovitskiy A, Beyer L, Kolesnikov A, et al (2020) An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:201011929

Dubey A, Jauhri A, Pandey A, et al (2024) The llama 3 herd of models. arXiv preprint arXiv:240721783

Ekambaram V, Jati A, Nguyen N, et al (2023) Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge

Discovery and Data Mining, pp 459–469

Ekambaram V, Jati A, Dayama P, et al (2024) Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. CoRR

Eldele E, Ragab M, Chen Z, et al (2024) Tslanet: Rethinking transformers for time series representation learning. International Conference on Machine Learning

Fan H, Xiong B, Mangalam K, et al (2021) Multiscale vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 6824–6835

Fan W, Wang P, Wang D, et al (2023) Dish-ts: a general paradigm for alleviating distribution shift in time series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 7522–7529

Fan W, Yi K, Ye H, et al (2024a) Deep frequency derivative learning for non-stationary time series forecasting. IJCAI

Fan X, Wu Y, Xu C, et al (2024b) Mg-tsd: Multi-granularity time series diffusion models with guided learning process. arXiv preprint arXiv:240305751

Feng S, Miao C, Zhang Z, et al (2024) Latent diffusion transformer for probabilistic time series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 11979–11987

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Annals of statistics pp 1189–1232

Fu DY, Dao T, Saab KK, et al (2023) Hungry hungry hippos: Towards language modeling with state space models. In The International Conference on Learning Representations (ICLR)

Fukushima K (1980) Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological cybernetics 36(4):193–202

Godahewa R, Bergmeir C, Webb G, et al (2021a) Solar power dataset (4 seconds observations). https://doi.org/10.5281/zenodo.4656027, URL https://doi.org/10.5281/zenodo.4656027

Godahewa R, Bergmeir C, Webb G, et al (2021b) Wind farms dataset (with missing values). https://doi.org/10.5281/zenodo.4654909, URL https://doi.org/10.5281/zenodo.4654909

Godahewa R, Bergmeir C, Webb G, et al (2021c) Wind farms dataset (without missing values). https://doi.org/10.5281/zenodo.4654858, URL https://doi.org/10.5281/zenodo.4654858

Godahewa R, Bergmeir C, Webb G, et al (2021d) Wind power dataset (4 seconds observations). https://doi.org/10.5281/zenodo.4656032, URL https://doi.org/10.5281/zenodo.4656032

Godahewa R, Bergmeir C, Webb G, et al (2021e) Australian electricity demand dataset. https://doi.org/10.5281/zenodo.4659727, URL https://doi.org/10.5281/zenodo.4659727

Godahewa R, Bergmeir C, Webb G, et al (2021f) Bitcoin dataset with missing values. https://doi.org/10.5281/zenodo.5121965, URL https://doi.org/10.5281/zenodo.5121965

Godahewa R, Bergmeir C, Webb G, et al (2021g) Bitcoin dataset without missing values. https://doi.org/10.5281/zenodo.5122101, URL https://doi.org/10.5281/zenodo.5122101

Godahewa R, Bergmeir C, Webb G, et al (2021h) Rideshare dataset with missing values. https://doi.org/10.5281/zenodo.5122114, URL https://doi.org/10.5281/zenodo.5122114

Godahewa R, Bergmeir C, Webb G, et al (2021i) Rideshare dataset without missing values. https://doi.org/10.5281/zenodo.5122232, URL https://doi.org/10.5281/zenodo.5122232

Godahewa R, Bergmeir C, Webb G, et al (2021j) Temperature rain dataset with missing values. https://doi.org/10.5281/zenodo.5129073, URL https://doi.org/10.5281/zenodo.5129073

Godahewa R, Bergmeir C, Webb G, et al (2021k) Temperature rain dataset without missing values. https://doi.org/10.5281/zenodo.5129091, URL https://doi.org/10.5281/zenodo.5129091

Godahewa R, Bergmeir C, Webb GI, et al (2021l) Monash time series forecasting archive. arXiv preprint arXiv:210506643

Gong Z, Tang Y, Liang J (2023) Patchmixer: A patch-mixing architecture for long-term time series forecasting. arXiv preprint arXiv:231000655

Gruver N, Finzi MA, Qiu S, et al (2023) Large language models are zero-shot time series forecasters. In: Thirty-seventh Conference on Neural Information Processing Systems, URL https://openreview.net/forum?id=md68e8iZK1

Gu A, Dao T (2023) Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:231200752

Gu A, Goel K, Ré C (2021a) Efficiently modeling long sequences with structured state spaces. arXiv preprint arXiv:211100396

Gu A, Johnson I, Goel K, et al (2021b) Combining recurrent, convolutional, and continuous-time models with linear state space layers. Advances in neural information processing systems 34:572–585

Hahn Y, Langer T, Meyes R, et al (2023) Time series dataset survey for forecasting with deep learning. Forecasting 5(1):315–335

Han L, Chen XY, Ye HJ, et al (2024) Softs: Efficient multivariate time series forecasting with series-core fusion. arXiv preprint arXiv:240414197

Ho J, Jain A, Abbeel P (2020) Denoising diffusion probabilistic models. Advances in neural information processing systems 33:6840–6851

Hochreiter S, Schmidhuber J (1997a) Long short-term memory. Neural computation 9(8):1735–1780

Hochreiter S, Schmidhuber J (1997b) Long short-term memory. Neural computation 9(8):1735–1780

Holt CC (1957) Forecasting trends and seasonals by exponentially weighted averages. carnegie institute of technology. Pittsburgh ONR memorandum

Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. Proceedings of the national academy of sciences 79(8):2554–2558

Hou H, Yu FR (2024) Rwkv-ts: Beyond traditional recurrent neural network for time series tasks. arXiv preprint arXiv:240109093

Hounie I, Porras-Valenzuela J, Ribeiro A (2024) Transformers with loss shaping constraints for long-term time series forecasting. In: Forty-first International Conference on Machine Learning

Hu J, Lan D, Zhou Z, et al (2024a) Time-ssm: Simplifying and unifying state space models for time series forecasting. arXiv preprint arXiv:240516312

Hu Y, Liu P, Zhu P, et al (2024b) Adaptive multi-scale decomposition framework for time series forecasting. arXiv preprint arXiv:240603751

Huang Q, Shen L, Zhang R, et al (2024a) Hdmixer: Hierarchical dependency with extendable patch for multivariate time series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 12608–12616

Huang S, Liu Y, Zhang F, et al (2024b) Crosswavenet: A dual-channel network with deep cross-decomposition for long-term time series forecasting. Expert Systems with Applications 238:121642

Ilbert R, Odonnat A, Feofanov V, et al (2024) Unlocking the potential of transformers in time series forecasting with sharpness-aware minimization and channel-wise attention. arXiv preprint arXiv:240210198

Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning, pmlr, pp 448–456

Jhin SY, Kim S, Park N (2024) Addressing prediction delays in time series forecasting: A continuous gru approach with derivative regularization. arXiv preprint arXiv:240701622

Jia Y, Lin Y, Hao X, et al (2024) Witran: Water-wave information transmission and recurrent acceleration network for long-range time series forecasting. Advances in Neural Information Processing Systems 36

Jin M, Wang S, Ma L, et al (2024) Time-LLM: Time series forecasting by reprogramming large language models. In: International Conference on Learning Representations (ICLR)

Kalman RE (1960) A new approach to linear filtering and prediction problems. Transactions of the ASME–Journal of Basic Engineering 82(1):35–45

Kim D, Park J, Lee J, et al (2024) Are self-attentions effective for time series forecasting? arXiv preprint arXiv:240516877

Kim T, Kim J, Tae Y, et al (2021) Reversible instance normalization for accurate time-series forecasting against distribution shift. In: International Conference on Learning Representations

Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:160902907

Kitaev N, Kaiser Ł, Levskaya A (2020) Reformer: The efficient transformer. arXiv preprint arXiv:200104451

Kollovieh M, Ansari AF, Bohlke-Schneider M, et al (2024) Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. Advances in Neural Information Processing Systems 36

Kong Z, Ping W, Huang J, et al (2020) Diffwave: A versatile diffusion model for audio synthesis. arXiv preprint arXiv:200909761

Kontopoulou VI, Panagopoulos AD, Kakkos I, et al (2023) A review of arima vs. machine learning approaches for time series forecasting in data driven networks. Future Internet 15(8):255

LeCun Y, Bottou L, Bengio Y, et al (1998) Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11):2278–2324

Li J, Li D, Savarese S, et al (2023) Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: International conference on machine learning, PMLR, pp 19730–19742

Li S, Jin X, Xuan Y, et al (2019) Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. Advances in neural information processing systems 32

Li Y, Yu R, Shahabi C, et al (2017) Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:170701926

Li Y, Chen W, Hu X, et al (2024a) Transformer-modulated diffusion models for probabilistic multivariate time series forecasting. In: The Twelfth International Conference on Learning Representations, URL https://openreview.net/forum?id=qae04YACHs

Li Y, Xu J, Anastasiu D (2024b) Learning from polar representation: An extreme-adaptive model for long-term time series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 171–179

Li Z, Qin Y, Cheng X, et al (2024c) Ftmixer: Frequency and time domain representations fusion for time series modeling. arXiv preprint arXiv:240515256

Liang A, Jiang X, Sun Y, et al (2024a) Bi-mamba4ts: Bidirectional mamba for time series forecasting. arXiv preprint arXiv:240415772

Liang D, Zhang H, Yuan D, et al (2024b) Minusformer: Improving time series forecasting by progressively learning residuals. arXiv preprint arXiv:240202332

Liang Y, Wen H, Nie Y, et al (2024c) Foundation models for time series analysis: A tutorial and survey. arXiv preprint arXiv:240314735

Lim B, Zohren S (2021) Time-series forecasting with deep learning: a survey. Philosophical Transactions of the Royal Society A 379(2194):20200209

Lin L, Li Z, Li R, et al (2024a) Diffusion models for time-series applications: a survey. Frontiers of Information Technology & Electronic Engineering 25(1):19–41

Lin S, Lin W, Wu W, et al (2023) Petformer: Long-term time series forecasting via placeholder-enhanced transformer. arXiv preprint arXiv:230804791

Lin S, Lin W, Wu W, et al (2024b) Sparsetsf: Modeling long-term time series forecasting with 1k parameters. International Conference on Machine Learning

Liu H, Li C, Wu Q, et al (2024a) Visual instruction tuning. Advances in neural information processing systems 36

Liu J, Liu C, Woo G, et al (2024b) Unitst: Effectively modeling inter-series and intra-series dependencies for multivariate time series forecasting. arXiv preprint arXiv:240604975

Liu S, Yu H, Liao C, et al (2021a) Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In: International conference on learning representations

Liu Y, Wu H, Wang J, et al (2022a) Non-stationary transformers: Exploring the stationarity in time series forecasting. Advances in Neural Information Processing Systems 35:9881–9893

Liu Y, Wu H, Wang J, et al (2022b) Non-stationary transformers: Rethinking the stationarity in time series forecasting. NeurIPS

Liu Y, Hu T, Zhang H, et al (2023) itransformer: Inverted transformers are effective for time series forecasting. arXiv preprint arXiv:231006625

Liu Y, Li C, Wang J, et al (2024c) Koopa: Learning non-stationary time series dynamics with koopman predictors. Advances in Neural Information Processing Systems 36

Liu Z, Zhu Z, Gao J, et al (2021b) Forecast methods for time series data: a survey. Ieee Access 9:91896–91912

Liu Z, Cheng M, Li Z, et al (2024d) Adaptive normalization for non-stationary time series forecasting: A temporal slice perspective. Advances in Neural Information Processing Systems 36

Lu J, Han X, Sun Y, et al (2024) Cats: Enhancing multivariate time series forecasting by constructing auxiliary time series as exogenous variables. International Conference on Machine Learning

Lu Han DCZHan-Jia Ye (2024) The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting. IEEE Transactions on Knowledge and Data Engineering pp 1–14

Luo D, Wang X (2024) Moderntcn: A modern pure convolution structure for general time series analysis. In: The Twelfth International Conference on Learning Representations

Ma X, Li X, Fang L, et al (2024) U-mixer: An unet-mixer architecture with stationarity correction for time series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 14255–14262

Ma Y, Guo Z, Ren Z, et al (2020) Streaming graph neural networks. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, pp 719–728

Makridakis S, Hibon M (2000) The m3-competition: results, conclusions and implications. International journal of forecasting 16(4):451–476

Makridakis S, Andersen A, Carbone R, et al (1982) The accuracy of extrapolation (time series) methods: Results of a forecasting competition. Journal of forecasting 1(2):111–153

Makridakis S, Spiliotis E, Assimakopoulos V (2020) The m4 competition: 100,000 time series and 61 forecasting methods. International Journal of Forecasting 36(1):54–74

Marisca I, Alippi C, Bianchi FM (2024) Graph-based forecasting with missing data through spatiotemporal downsampling. International Conference on Machine Learning

Masini RP, Medeiros MC, Mendes EF (2023) Machine learning advances for time series forecasting. Journal of economic surveys 37(1):76–111

McCracken MW, Ng S (2016) Fred-md: A monthly database for macroeconomic research. Journal of Business & Economic Statistics 34(4):574–589

McLeod A, Gweon H (2013) Optimal deseasonalization for monthly and daily geophysical time series. Journal of Environmental statistics 4(11):1–11

Meijer C, Chen LY (2024) The rise of diffusion models in time-series forecasting. arXiv preprint arXiv:240103006

Mu B, Jiang X, Yuan S, et al (2023) Nao seasonal forecast using a multivariate air–sea coupled deep learning model combined with causal discovery. Atmosphere 14(5):792

Nawrot P, Tworkowski S, Tyrolski M, et al (2021) Hierarchical transformers are more efficient language models. arXiv preprint arXiv:211013711

Ni Z, Yu H, Liu S, et al (2024) Basisformer: Attention-based time series forecasting with learnable and interpretable basis. Advances in Neural Information Processing Systems 36

Nie T, Mei Y, Qin G, et al (2024) Channel-aware low-rank adaptation in time series forecasting. Conference on Information and Knowledge Management

Nie Y, Nguyen NH, Sinthong P, et al (2022) A time series is worth 64 words: Long-term forecasting with transformers. arXiv preprint arXiv:221114730

Nti IK, Teimeh M, Nyarko-Boateng O, et al (2020) Electricity load forecasting: a systematic review. Journal of Electrical Systems and Information Technology 7:1–19

Passalis N, Tefas A, Kanniainen J, et al (2019) Deep adaptive input normalization for time series forecasting. IEEE transactions on neural networks and learning systems 31(9):3760–3765

Patro BN, Agneeswaran VS (2024a) Mamba-360: Survey of state space models as transformer alternative for long sequence modelling: Methods, applications, and challenges. arXiv preprint arXiv:240416112

Patro BN, Agneeswaran VS (2024b) Simba: Simplified mamba-based architecture for vision and multivariate time series. arXiv preprint arXiv:240315360

Patwardhan N, Marrone S, Sansone C (2023) Transformers in the real world: A survey on nlp applications. Information 14(4):242

Piao X, Chen Z, Murayama T, et al (2024) Fredformer: Frequency debiased transformer for time series forecasting. arXiv preprint arXiv:240609009

Qi S, Wen L, Li Y, et al (2024) Enhancing multivariate time series forecasting with mutual information-driven cross-variable and temporal modeling. arXiv preprint arXiv:240300869

Qian J, Wang Q, Wu Y, et al (2023) Causality-based deep learning forecast of the kuroshio volume transport in the east china sea. Earth and Space Science 10(2):e2022EA002722

Qin Y, Song D, Chen H, et al (2017) A dual-stage attention-based recurrent neural network for time series prediction. arXiv preprint arXiv:170402971

Quinlan JR (1986) Induction of decision trees. Machine learning 1:81–106

Radford A, Kim JW, Hallacy C, et al (2021) Learning transferable visual models from natural language supervision. In: International conference on machine learning, PMLR, pp 8748–8763

Ramesh A, Dhariwal P, Nichol A, et al (2022) Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:220406125 1(2):3

Rasul K, Seward C, Schuster I, et al (2021) Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In: International Conference on Machine Learning, PMLR, pp 8857–8868

Rasul K, Ashok A, Williams AR, et al (2024) Lag-llama: Towards foundation models for probabilistic time series forecasting. 2310.08278

Rombach R, Blattmann A, Lorenz D, et al (2022a) High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10684–10695

Rombach R, Blattmann A, Lorenz D, et al (2022b) High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10684–10695

Rossi E, Chamberlain B, Frasca F, et al (2020) Temporal graph networks for deep learning on dynamic graphs. arXiv preprint arXiv:200610637

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. nature 323(6088):533–536

Saharia C, Chan W, Saxena S, et al (2022) Photorealistic text-to-image diffusion models with deep language understanding. Advances in neural information processing systems 35:36479–36494

Salinas D, Flunkert V, Gasthaus J, et al (2020) Deepar: Probabilistic forecasting with autoregressive recurrent networks. International journal of forecasting 36(3):1181–1191

Scarselli F, Gori M, Tsoi AC, et al (2008) The graph neural network model. IEEE transactions on neural networks 20(1):61–80

Shabani A, Abdi A, Meng L, et al (2022) Scaleformer: Iterative multi-scale refining transformers for time series forecasting. arXiv preprint arXiv:220604038

Shafer G (1976) A Mathematical Theory of Evidence. Princeton University Press

Sharma K, Dwivedi YK, Metri B (2022) Incorporating causality in energy consumption forecasting using deep neural networks. Annals of Operations Research pp 1–36

Shen L, Kwok J (2023) Non-autoregressive conditional diffusion models for time series prediction. In: International Conference on Machine Learning, PMLR, pp 31016–31029

Shen L, Chen W, Kwok J (2024) Multi-resolution diffusion models for time series forecasting. In: The Twelfth International Conference on Learning Representations

Shih SY, Sun FK, Lee Hy (2019) Temporal pattern attention for multivariate time series forecasting. Machine Learning 108:1421–1441

Shu Y, Lampos V (2024) Deformtime: Capturing variable dependencies with deformable attention for time series forecasting. arXiv preprint arXiv:240607438

Song Y, Sohl-Dickstein J, Kingma DP, et al (2020) Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:201113456

Soyiri IN, Reidpath DD (2013) An overview of health forecasting. Environmental health and preventive medicine 18:1–9

Sparks AH, Carroll J, Goldie J, et al (2020) Bomrang: Australian government bureau of meteorology (bom) data client. R package version 07 0 URL https://CRAN R-project org/package= bomrang

Štěpnička M, Burda M (2017) On the results and observations of the time series forecasting competition cif 2016. In: 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, pp 1–6

Sun Y, Xie Z, Chen D, et al (2024) Hierarchical classification auxiliary network for time series forecasting. arXiv preprint arXiv:240518975

Taieb SB, Bontempi G, Atiya AF, et al (2012) A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition. Expert systems with applications 39(8):7067–7083

Tang P, Zhang W (2024) Pdmlp: Patch-based decomposed mlp for long-term time series forecastin. arXiv preprint arXiv:240513575

Tashiro Y, Song J, Song Y, et al (2021) Csdi: Conditional score-based diffusion models for probabilistic time series imputation. Advances in Neural Information Processing Systems 34:24804–24816

Tay Y, Dehghani M, Bahri D, et al (2020) Efficient transformers: A survey. ACM Computing Surveys 55:1 – 28. URL https://api.semanticscholar.org/CorpusID:221702858

Touvron H, Martin L, Stone K, et al (2023) Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:230709288

Ulyanov D, Vedaldi A, Lempitsky V (2016) Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:160708022

Van Den Oord A, Dieleman S, Zen H, et al (2016a) Wavenet: A generative model for raw audio. arXiv preprint arXiv:160903499 12

Van Den Oord A, Dieleman S, Zen H, et al (2016b) Wavenet: A generative model for raw audio. arXiv preprint arXiv:160903499 12

Vapnik V, Golowich S, Smola A (1996) Support vector method for function approximation, regression estimation and signal processing. Advances in neural information processing systems 9

Vaswani A (2017) Attention is all you need. arXiv preprint arXiv:170603762

Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. Advances in neural information processing systems 30

Veličković P, Cucurull G, Casanova A, et al (2017) Graph attention networks. arXiv preprint arXiv:171010903

Wang L, Adiga A, Chen J, et al (2022) Causalgnn: Causal-based graph neural networks for spatio-temporal epidemic forecasting. In: Proceedings of the AAAI conference on artificial intelligence, pp 12191–12199

Wang S, Wu H, Shi X, et al (2024a) Timemixer: Decomposable multiscale mixing for time series forecasting. International Conference on Learning Representations

Wang X, Zhou T, Wen Q, et al (2024b) Card: Channel aligned robust blend transformer for time series forecasting. In: The Twelfth International Conference on Learning Representations

Wang Y, Wu H, Dong J, et al (2024c) Timexer: Empowering transformers for time series forecasting with exogenous variables. arXiv preprint arXiv:240219072

Wang Z, Kong F, Feng S, et al (2024d) Is mamba effective for time series forecasting? arXiv preprint arXiv:240311144

Wang Z, Ruan S, Huang T, et al (2024e) A lightweight multi-layer perceptron for efficient multivariate time series forecasting. Knowledge-Based Systems 288:111463

Wen Q, Zhou T, Zhang C, et al (2022a) Transformers in time series: A survey. arXiv preprint arXiv:220207125

Wen Q, Zhou T, Zhang C, et al (2022b) Transformers in time series: A survey. arXiv preprint arXiv:220207125

Wen R, Torkkola K, Narayanaswamy B, et al (2017) A multi-horizon quantile recurrent forecaster. arXiv preprint arXiv:171111053

Winters PR (1960) Forecasting sales by exponentially weighted moving averages. Management science 6(3):324–342

Woo G, Liu C, Kumar A, et al (2024) Unified training of universal time series forecasting transformers. arXiv preprint arXiv:240202592

Wu H, Xu J, Wang J, et al (2021) Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. Advances in neural information processing systems 34:22419–22430

Wu H, Hu T, Liu Y, et al (2023) Timesnet: Temporal 2d-variation modeling for general time series analysis. International Conference on Learning Representations

Wu Z, Gong Y, Zhang A (2024) Dtmamba: Dual twin mamba for time series forecasting. arXiv preprint arXiv:240507022

Xiong Q, Tang K, Ma M, et al (2024) Tdt loss takes it all: Integrating temporal dependencies among targets into non-autoregressive time series forecasting. arXiv preprint arXiv:240604777

Xu X, Liang Y, Huang B, et al (2024a) Integrating mamba and transformer for long-short range time series forecasting. arXiv preprint arXiv:240414757

Xu Z, Bian Y, Zhong J, et al (2024b) Beyond trend and periodicity: Guiding time series forecasting with textual cues. arXiv preprint arXiv:240513522

Xu Z, Zeng A, Xu Q (2024c) Fits: Modeling time series with $10k$ parameters. International Conference on Learning Representations

Xue H, Salim FD (2023) Promptcast: A new prompt-based learning paradigm for time series forecasting. IEEE Transactions on Knowledge and Data Engineering

Yan S, Xiong Y, Lin D (2018) Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Proceedings of the AAAI conference on artificial intelligence

Yan T, Zhang H, Zhou T, et al (2021) Scoregrad: Multivariate probabilistic time series forecasting with continuous energy-based generative models. arXiv preprint arXiv:210610121

Yan T, Gong H, Yongping H, et al (2024) Probabilistic time series modeling with decomposable denoising diffusion model. In: Salakhutdinov R, Kolter Z, Heller K, et al (eds) Proceedings of the 41st International Conference on Machine Learning, Proceedings of Machine Learning Research, vol 235. PMLR, pp 55759–55777, URL https://proceedings.mlr.press/v235/yan24b.html

Yang F, Li X, Wang M, et al (2023) Waveform: Graph enhanced wavelet learning for long sequence forecasting of multivariate time series. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 10754–10761

Yang Y, Zhu Q, Chen J (2024) Vcformer: Variable correlation transformer with inherent lagged correlation for multivariate time series forecasting. arXiv preprint arXiv:240511470

Ye J, Zhang W, Yi K, et al (2024) A survey of time series foundation models: Generalizing time series representation with large language mode. arXiv preprint arXiv:240502358

Yi K, Zhang Q, Fan W, et al (2024) Frequency-domain mlps are more effective learners in time series forecasting. Advances in Neural Information Processing Systems 36

Yu C, Wang F, Shao Z, et al (2023) Dsformer: A double sampling transformer for multivariate time series long-term prediction. In: Proceedings of the 32nd ACM international conference on information and knowledge management, pp 3062–3072

Yu G, Zou J, Hu X, et al (2024) Revitalizing multivariate time series forecasting: Learnable decomposition with inter-series dependencies and intra-series variations modeling. International Conference on Machine Learning

Yuan X, Qiao Y (2024) Diffusion-ts: Interpretable diffusion for general time series generation. arXiv preprint arXiv:240301742

Zeng A, Chen M, Zhang L, et al (2023a) Are transformers effective for time series forecasting? In: Proceedings of the AAAI conference on artificial intelligence, pp 11121–11128

Zeng A, Chen M, Zhang L, et al (2023b) Are transformers effective for time series forecasting? In: Proceedings of the AAAI conference on artificial intelligence, pp 11121–11128

Zeng C, Liu Z, Zheng G, et al (2024) C-mamba: Channel correlation enhanced state space models for multivariate time series forecasting. arXiv preprint arXiv:240605316

Zhan T, He Y, Li Z, et al (2024) Time evidence fusion network: Multi-source view in long-term time series forecasting. arXiv preprint arXiv:240506419

Zhang D, Wang Y (2024) Adaptive extraction network for multivariate long sequence time-series forecasting. arXiv preprint arXiv:240512038

Zhang K, Zou X, Tang Y (2024a) Caformer: Rethinking time series analysis from causal perspective. arXiv preprint arXiv:240308572

Zhang M, Sun Y, Liang F (2024b) Sparse deep learning for time series data: theory and applications. Advances in Neural Information Processing Systems 36

Zhang X, Chowdhury RR, Gupta RK, et al (2024c) Large language models for time series: A survey. arXiv preprint arXiv:240201801

Zhang Y, Yan J (2023) Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In: The eleventh international conference on learning representations

Zhang Y, Ma L, Pal S, et al (2024d) Multi-resolution time-series transformer for long-term forecasting. In: International Conference on Artificial Intelligence and Statistics, PMLR, pp 4222–4230

Zhou H, Zhang S, Peng J, et al (2021) Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI conference on artificial intelligence, pp 11106–11115

Zhou T, Ma Z, Wen Q, et al (2022) Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In: International conference on machine learning, PMLR, pp 27268–27286

Zhou T, Niu P, Wang X, et al (2023) One fits all: Power general time series analysis by pre-trained LM. In: Thirty-seventh Conference on Neural Information Processing Systems, URL https://openreview.net/forum?id=gMS6FVZvmF