

# A Comprehensive Survey of Deep Learning for Multivariate Time Series Forecasting: A Channel Strategy Perspective

Xiangfei Qiu, Hanyin Cheng, Xingjian Wu, Jilin Hu, Chenjuan Guo, Bin Yang

School of Data Science and Engineering, East China Normal University, Shanghai, China  
 {xfqiu, hycheng, xjwu}@stu.ecnu.edu.cn, {jlhu, cjguo, byang}@dase.ecnu.edu.cn

## Abstract

Multivariate Time Series Forecasting (MTSF) plays a crucial role across diverse fields, ranging from economic, energy, to traffic. In recent years, deep learning has demonstrated outstanding performance in MTSF tasks. In MTSF, modeling the correlations among different channels is critical, as leveraging information from other related channels can significantly improve the prediction accuracy of a specific channel. This study systematically reviews the channel modeling strategies for time series and proposes a taxonomy organized into three hierarchical levels: the strategy perspective, the mechanism perspective, and the characteristic perspective. On this basis, we provide a structured analysis of these methods and conduct an in-depth examination of the advantages and limitations of different channel strategies. Finally, we summarize and discuss some future research directions to provide useful research guidance. Moreover, we maintain an up-to-date Github repository<sup>1</sup> which includes all the papers discussed in the survey.

## 1 Introduction

Multivariate time series forecasting (MTSF) is a fundamental yet challenging task in various domains, including economic, energy, and traffic [Qiu *et al.*, 2024; Wu *et al.*, 2024; Yu *et al.*, 2024]. The ability to accurately predict future values of multiple interdependent channels (a.k.a., variables) over time is crucial for making informed decisions, optimizing resource allocation, and improving operational efficiency.

In recent years, the rapid advancements in deep learning have significantly boosted the performance of MTSF. Researchers primarily model multivariate time series from the temporal and channel (a.k.a., variable) dimensions. In terms of the temporal dimension, researchers have utilized various modules such as CNN [Wu *et al.*, 2023a], MLP [Lin *et al.*, 2024], and Transformer [Nie *et al.*, 2023] to capture the non-linear and complex temporal dependencies within time series. In the channel dimension, researchers have designed different channel strategies to model the intricate correlations among

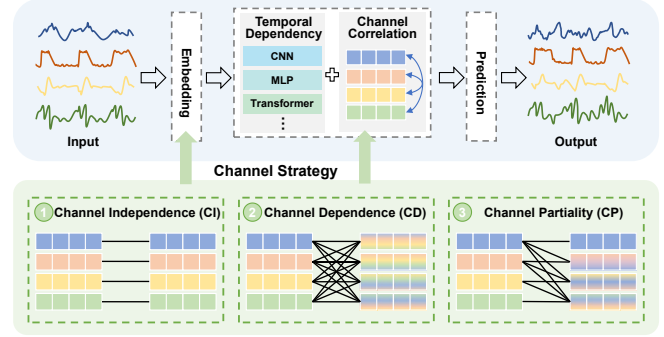


Figure 1: Channel strategy overview.

channels [Qiu *et al.*, 2025; Liu *et al.*, 2024b]. This is crucial in multivariate forecasting, as leveraging information from related channels can significantly improve the prediction accuracy of a specific channel. For example, in financial market forecasting, integrating data such as stock open price, trading volumes, and market indices can lead to more accurate predictions of a particular stock’s return ratio. The interconnections among these factors provide a more comprehensive perspective for capturing market dynamics.

Given the critical role of channel correlation in improving prediction accuracy, the selection of an appropriate channel strategy becomes a key design consideration. Overall, existing channel strategies—see Figure 1 can be categorized into three types: CI (Channel Independence) processing each channel independently without considering any potential interactions or correlations among them [Nie *et al.*, 2023; Lin *et al.*, 2024]; CD (Channel Dependence) treating all channels as a unified entity, assuming they are interrelated and dependent on each other [Zhang and Yan, 2022; Liu *et al.*, 2024b]; and CP (Channel Partiality) meaning that each channel maintains some degree of independence while simultaneously being influenced by some other related channels [Han *et al.*, 2024b; Qiu *et al.*, 2025]. Each strategy reflects a unique perspective on how to model the inter-channel dependencies, leading to diverse modeling architectures and applications.

While there are several surveys on MTSF [Wen *et al.*, 2023; Wang *et al.*, 2024d], they often lack a comprehensive discussion on the role of channel strategies in multivariate settings. This study aims to bridge this gap by summarizing the main developments of channel strategies of MTSF. We first

<sup>1</sup><https://github.com/decisionintelligence/CS4TS>

briefly introduce the MTSF task and propose a new taxonomy organized into three hierarchical levels. Starting with the strategy perspective, we systematically introduce the definitions of three channel strategies and their representative methods, providing researchers with a foundational framework to understand these strategies. Next, from the mechanism perspective, we further explore how each method implements the specific mechanisms of channel strategies, categorizing and summarizing them to help researchers gain a deeper understanding of the underlying implementation of different channel strategies. Finally, from the characteristic perspective, we focus on the different characteristics considered by channel strategies when modeling the correlations among channels. Following this, we provide a detailed comparison of the advantages and limitations of the three channel strategies, offering valuable insights for researchers. In conclusion, we discuss future research directions that will provide useful guidance for the further development of the field. To the best of our knowledge, this is the first work to comprehensively and systematically review the key developments of deep learning methods for MTSF through the lens of channel strategy. In summary, the main contributions of this survey include:

- **Comprehensive and up-to-date survey:** We provide an in-depth review of state-of-the-art deep learning models for MTSF, highlighting their use of channel strategies.
- **Novel channel-perspective taxonomy:** We introduce a structured taxonomy of channel strategies for deep learning-based MTSF, offering a comprehensive analysis of their strengths and limitations.
- **Future research opportunities:** We discuss and highlight future avenues for enhancing MTSF through diverse channel strategies, urging researchers to delve deeper into this area.

## 2 Preliminaries

**Time Series:** A time series  $X \in \mathbb{R}^{T \times N}$  is a time-oriented sequence of  $N$ -dimensional time points, where  $T$  is the number of timestamps, and  $N$  is the number of channels.

**Multivariate Time Series Forecasting:** Given a historical multivariate time series  $X \in \mathbb{R}^{T \times N}$  of  $T$  time points, multivariate time series forecasting aims to predict the next  $F$  future time points, i.e.,  $Y \in \mathbb{R}^{F \times N}$ , where  $F$  is called the forecasting horizon.

## 3 Taxonomy of Channel Strategies in MTSF

The taxonomy presented in Table 1 provides a structured classification to enhance the understanding of channel strategies in MTSF. It is organized into three hierarchical levels, starting with the strategy perspective, followed by the mechanism perspective, and finally the characteristic perspective.

### 3.1 Strategy Perspective

A channel strategy refers to the approach employed to process, integrate, or utilize information from multiple input channels. As illustrated in Figure 1, the explored strategies can be broadly categorized as follows. We will discuss the pros and cons of these strategies in Section 4.

**Channel Independence (CI):** The CI strategy treats each channel independently, without considering any potential interactions or correlations among channels. Each channel is processed as a separate input, and no shared information or dependencies are utilized. The representative method PatchTST [Nie *et al.*, 2023] employs CI and demonstrates outstanding performance in MTSF. This design significantly reduces model complexity, enabling faster inference while mitigating the risk of overfitting caused by noise or spurious correlations among channels. Furthermore, the CI strategy offers flexibility, as the addition of new channels does not require changes to the model architecture, allowing it to seamlessly adapt to evolving datasets. These advantages have made the CI strategy increasingly popular in recent research, contributing to improved forecasting performance [Lin *et al.*, 2024; Zeng *et al.*, 2023].

**Channel Dependence (CD):** The CD strategy assumes that all channels in a multivariate time series are inherently correlated and interdependent, treating them as a unified entity during the forecasting process. Based on the phases when inter-channel interactions are learned, the existing CD methods can be divided into two categories: I) **Embedding fusion:** These models fuse data from different channels when obtaining their time series embedding representations. For example, Informer [Zhou *et al.*, 2021], Autoformer [Wu *et al.*, 2021], and TimesNet [Wu *et al.*, 2023a] use 1D or 2D convolutions to extract temporal representations. In the convolutional operation, each convolutional kernel first performs a sliding convolution within each input channel to obtain the corresponding feature maps. These feature maps for all channels are then weighted and combined, capturing the dependencies among the channels. II) **Explicit correlation:** These models often design specialized modules to explicitly model channel correlations, facilitating more structured channel modeling based on the acquired time series embedding representations. Representative algorithms include iTransformer [Liu *et al.*, 2024b] and TSMixer [Ekambaram *et al.*, 2023]. iTransformer adopts a self-attention module among channels, treating independent time series as tokens and capturing multivariate correlations using the self-attention mechanism. In contrast, TSMixer uses an MLP module among channels to capture the intricate correlations among channels, with these correlations represented by multi-level features extracted through fully connected layers.

**Channel Partiality (CP):** The CP strategy strikes a balance between CI and CD, allowing each channel to retain a degree of independence while simultaneously interacting with other related channels. This approach emphasizes a hybrid state where channels selectively interact and exhibit partial correlations. Based on whether the number of related channels for each channel is fixed or dynamic, the existing CP methods can be divided into two categories: I) **Fixed partial channels:** These models fix the number of related channels for each channel, which means the set of associated channels remains constant overtime. For example, in MTGNN [Wu *et al.*, 2020], the channel relationships are modeled as a  $K$ -regular graph, where each channel interacts with  $K$  other channels using the CD strategy to model interdependencies, while the remaining channels interact through the CI strategy.

Table 1: A taxonomy of channel strategy in multivariate time series forecasting.

Strategy	Mechanism	Characteristic						Method	Paradigm	Venue	Year	Code
		Asym.	Lag.	Pol.	Gw.	Dyn.	Ms.					
CI	-	-	-	-	-	-	-	PatchTST [Nie <i>et al.</i> , 2023]	Specific	ICLR	2023	PatchTST
	-	-	-	-	-	-	-	CycleNet [Lin <i>et al.</i> , 2024]	Specific	NIPS	2024	CycleNet
	-	-	-	-	-	-	-	DLinear [Zeng <i>et al.</i> , 2023]	Specific	AAAI	2023	DLinear
	-	-	-	-	-	-	-	Timer [Liu <i>et al.</i> , 2024c]	Foundation	ICML	2024	Timer
	-	-	-	-	-	-	-	Chronos [Ansari <i>et al.</i> , 2024]	Foundation	ICML	2024	Chronos
	-	-	-	-	-	-	-	LLM4TS [Chang <i>et al.</i> , 2023]	Foundation	NIPS	2023	LLM4TS
	-	-	-	-	-	-	-	Time-LLM [Jin <i>et al.</i> , 2024]	Foundation	ICLR	2024	Time-LLM
	-	-	-	-	-	-	-	RevIN [Kim <i>et al.</i> , 2021]	Plugin	ICLR	2021	RevIN
CD	CNN-based	✓	-	-	-	-	-	Informer [Zhou <i>et al.</i> , 2021]	Specific	AAAI	2021	Informer
	CNN-based	✓	-	-	-	-	-	Autoformer [Wu <i>et al.</i> , 2021]	Specific	NIPS	2021	Autoformer
	CNN-based	✓	-	-	-	-	-	FEDformer [Zhou <i>et al.</i> , 2022]	Specific	ICML	2022	FEDformer
	CNN-based	✓	-	-	-	-	-	TimesNet [Wu <i>et al.</i> , 2023a]	Specific	ICLR	2023	TimesNet
	MLP-based	✓	-	-	-	-	-	TSMixer [Ekambaram <i>et al.</i> , 2023]	Specific	KDD	2023	TSMixer
	MLP-based	✓	-	-	-	-	-	TTM [Ekambaram <i>et al.</i> , 2024]	Foundation	NIPS	2024	TTM
	Transformer-based	✓	-	-	-	✓	-	iTransformer [Liu <i>et al.</i> , 2024b]	Specific	ICLR	2024	iTransformer
	Transformer-based	✓	-	-	-	✓	-	Crossformer [Zhang and Yan, 2022]	Specific	ICLR	2023	Crossformer
	Transformer-based	✓	✓	-	-	-	-	VCformer [Yang <i>et al.</i> , 2024]	Specific	IJCAI	2024	VCformer
	Transformer-based	✓	✓	-	-	✓	-	MOIRAI [Woo <i>et al.</i> , 2024]	Foundation	ICML	2024	MOIRAI
	Transformer-based	✓	-	-	-	✓	-	UniTS [Gao <i>et al.</i> , 2024]	Foundation	NIPS	2024	UniTS
	GNN-based	✓	-	-	-	✓	-	GTS [Shang <i>et al.</i> , 2021]	Specific	ICLR	2021	GTS
	GNN-based	✓	-	-	-	-	✓	MSGNet [Cai <i>et al.</i> , 2024]	Specific	AAAI	2024	MSGNet
	GNN-based	-	✓	-	-	-	-	FourierGNN [Yi <i>et al.</i> , 2023]	Specific	NIPS	2023	FourierGNN
	GNN-based	-	✓	-	-	-	-	FC-STGNN [Wang <i>et al.</i> , 2024c]	Specific	AAAI	2024	FC-STGNN
	GNN-based	✓	-	-	-	✓	-	TPGNN [Liu <i>et al.</i> , 2022]	Specific	NIPS	2022	TPGNN
	GNN-based	✓	-	-	-	✓	✓	ESG [Ye <i>et al.</i> , 2022]	Specific	KDD	2022	ESG
	GNN-based	✓	-	-	-	✓	✓	EnhanceNet [Cirstea <i>et al.</i> , 2021]	Plugin	ICDE	2021	EnhanceNet
	Others	✓	-	-	-	-	-	SOFTS [Han <i>et al.</i> , 2024a]	Specific	NIPS	2024	SOFTS
	Others	✓	-	-	-	✓	-	C-LoRA [Nie <i>et al.</i> , 2024]	Plugin	CIKM	2024	C-LoRA
CP	CNN-based	✓	-	-	-	-	-	ModernTCN [Luo and Wang, 2024]	Specific	ICLR	2024	ModernTCN
	Transformer-based	✓	-	-	✓	-	-	DUET [Qiu <i>et al.</i> , 2025]	Specific	KDD	2025	DUET
	Transformer-based	✓	-	-	-	✓	-	MCformer [Han <i>et al.</i> , 2024b]	Specific	IITJ*	2024	-
	Transformer-based	✓	-	-	✓	✓	-	DGCformer [Liu <i>et al.</i> , 2024a]	Specific	arXiv	2024	-
	Transformer-based	✓	-	-	-	✓	-	CM [Lee <i>et al.</i> , 2024]	Plugin	NIPS	2024	-
	GNN-based	✓	-	-	-	-	-	MTGNN [Wu <i>et al.</i> , 2020]	Specific	KDD	2020	MTGNN
	GNN-based	✓	-	✓	-	-	-	CrossGNN [Huang <i>et al.</i> , 2023a]	Specific	NIPS	2023	CrossGNN
	GNN-based	✓	-	-	-	✓	-	WaveForM [Yang <i>et al.</i> , 2023]	Specific	AAAI	2023	WaveForM
	GNN-based	-	-	-	-	✓	-	MTSF-DG [Zhao <i>et al.</i> , 2023]	Specific	VLDB	2023	MTSF-DG
	GNN-based	✓	-	-	✓	-	-	ReMo [Wu <i>et al.</i> , 2023b]	Specific	IJCAI	2023	-
	GNN-based	✓	-	-	✓	✓	✓	Ada-MSHyper [Shang <i>et al.</i> , 2024]	Specific	NIPS	2024	Ada-MSHyper
	Others	✓	✓	-	-	-	-	LIFT [Zhao and Shen, 2024]	Plugin	ICLR	2024	LIFT
	Others	✓	-	-	✓	-	-	CCM [Chen <i>et al.</i> , 2024]	Plugin	NIPS	2024	CCM

Asym.: Asymmetry, Lag.: Lagginess, Pol.: Polarity, Gw.: group-wise, Dyn.: Dynamism, and Ms.: Multi-scale. We will discuss them in Section 3.3. Since the CI models do not consider the correlations among channels, the corresponding positions for mechanism and characteristic are marked as "-". For the plugin model, its mechanism is complex, so we exclude it, marking the corresponding position as "-". IITJ\*: IEEE Internet Things J.

Similarly, in MCformer [Han *et al.*, 2024b], each channel interacts with only  $K$  other channels, maintaining the CI strategy with the rest to ensure computational efficiency and prevent overfitting. II) **Dynamic partial channels:** These models allow the number of related channels for each channel to be dynamic, changing over time and providing greater flexibility to adapt to varying scenarios. For instance, DUET [Qiu *et al.*, 2025] calculates channel similarity using metric learning in the frequency domain and then sparsifies the result. This creates a mask matrix, which is integrated into the attention mechanism of the fusion module, ensuring that each channel interacts only with relevant channels, reducing interference from noisy ones. Another example, CCM [Chen *et al.*, 2024], dynamically clusters channels based on their intrinsic similarities. To effectively capture the underlying time series patterns within these clusters, CCM utilizes a cluster-aware feed forward mechanism, enabling tailored management and processing for each individual cluster.

### 3.2 Mechanism Perspective

This section presents mainly the various mechanisms designed to model the relationships among channels.

**Transformer-based:** In recent years, Transformer has

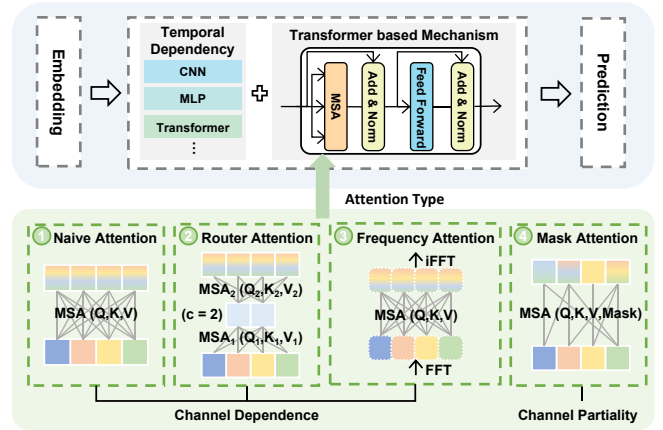


Figure 2: Transformer-based mechanism for channel strategy.

been widely applied to MTSF tasks, leveraging its powerful global modeling capability to effectively capture complex temporal dependencies and channel interactions. As shown in Figure 2, existing channel strategies based on the attention mechanism can be categorized into the following types. I)

**Naive Attention:** These approaches all adopt CD strategy,

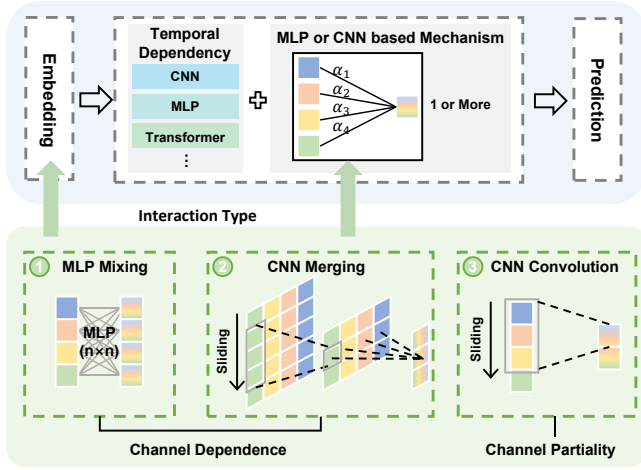


Figure 3: MLP, CNN-based mechanism for channel strategy.

treating time series segments (patches) or the entire sequence of each channel as individual tokens, and directly applying attention mechanisms to model channel correlations. For instance, CARD [Wang *et al.*, 2024b] and iTransformer [Liu *et al.*, 2024b] represent the patches and series of each channel as independent tokens, respectively, and explicitly capture channel correlation using attention mechanisms. II) **Router Attention**: When the number of channels ( $N$ ) is large, the computational complexity of channel attention reaches  $O(N^2)$ , resulting in high computational costs. To address this, some methods propose optimization strategies to mitigate the computational complexity caused by CD strategy. For example, Crossformer [Zhang and Yan, 2022] introduces a Router Mechanism for Naive Attention, which uses a small fixed number of  $c$  “routers” ( $c \ll N$ ) to gather information from all channels and redistribute it. This reduces the complexity to  $O(2cN) = O(N)$ . This mechanism effectively balances the modeling of channel correlation and computational efficiency. III) **Frequency Attention**: Some CD methods suggest that frequency-domain information is more effective for capturing inter-channel dependencies than time-domain information. For example, FECAM [Jiang *et al.*, 2023] transforms the time series data into the frequency domain and then employs Naive Attention to model inter-channel relationships in this domain. IV) **Mask Attention**: In naive attention, each channel calculates attention scores with all channels, which can be negatively affected by irrelevant channels. To mitigate this, Mask Attention provides an approach to avoid irrelevant noise by constructing CP strategy. For example, DUET [Qiu *et al.*, 2025] generates mask matrices for Naive Attention, allowing each channel to focus on those beneficial for downstream prediction tasks, while mitigating the impact of noisy or irrelevant channels. This approach explicitly constrains the attention computation, improving the accuracy of channel correlation modeling.

**MLP-based**: The Multilayer Perceptron (MLP), as a backbone network, possesses powerful feature learning capabilities according to the universal approximation theorem. Existing MLP-based models use **MLP Mixing** in a CD manner, to capture the intricate correlations among channels,

with these correlations represented by multi-level features extracted through fully connected layers—see Figure 3. From the perspective of channel strategies, models in the MLP Mixing category, such as TSMixer [Ekambaram *et al.*, 2023] and Tiny-TTM [Ekambaram *et al.*, 2024], employ this approach to efficiently capture correlations among all channels, achieving strong performance with low computational cost, and all fall under the CD strategy.

**CNN-based**: Convolutional Neural Network (CNN) is deep learning model that utilize convolutional layers to extract local features from data. As illustrated in Figure 3, the explored CNN-based approaches can be broadly categorized as follows: I) **Merging**: Many models, such as Informer [Zhou *et al.*, 2021], Autoformer [Wu *et al.*, 2021], and FEDformer [Zhou *et al.*, 2022], use 1D convolution with a sliding operation along the temporal dimension in the initial feature extraction layers. These models treat different channels as distinct inputs to the convolution, whose features are then weighted and merged during the convolution process, enabling inter-channel interactions. Although TimesNet [Wu *et al.*, 2023a] employs 2D convolutions, it folds the temporal dimension into a 2D format, with variable channels still serving as independent input for weighted merging via convolution. Such models are all under the CD strategy. II) **Convolution**: Given the slight spatial dependence among channels, ModerTCN [Luo and Wang, 2024] directly applies convolution operations to facilitate information interaction among channels within local scopes. Within the same convolution window, channels interact with each other in a CD manner through the convolution kernel, while channels that cannot be assigned to the same window remain independent of each other. This results in an efficient method for CP modeling.

**GNN-based**: By dividing the time series into different windows along time, where each channel within a window is treated as a node and the correlations between channels are considered as edges, multivariate time series can be transformed into graph-based data. The GNN-based methods can be classified into dense and sparse graphs. In *dense graphs*, each node is typically connected to almost all other nodes, with the edges often representing the strength of correlation or the probability of correlated influence. Methods based on dense graphs, such as GTS [Shang *et al.*, 2021] and FourierGNN [Yi *et al.*, 2023], generally follow a CD strategy. In contrast, *sparse graphs* only retain necessary edges, with most nodes remaining independent. For instance, MTGNN [Wu *et al.*, 2020] preserves  $K$  edges per node, constructing a sparse  $K$ -regular graph. Different from this, MTSF-DG [Zhao *et al.*, 2023] sparsifies the adjacency matrix by filtering out low-probability edges based on a pre-set threshold. Methods based on sparse graphs belong to the CP strategy.

The sparsity of the constructed graph determines whether the method follows a CD or CP strategy. Additionally, GNN-based models often rely on the type of graph they construct when implementing the CD or CP strategy. As shown in Figure 4, we classify graph types as follows: I) **Simple Graph**: A simple graph is the most basic graph model, where there is at most one edge between each pair of nodes. A well-defined graph structure is required for effective message passing. Researchers have used channel similarity metrics (MTGNN,



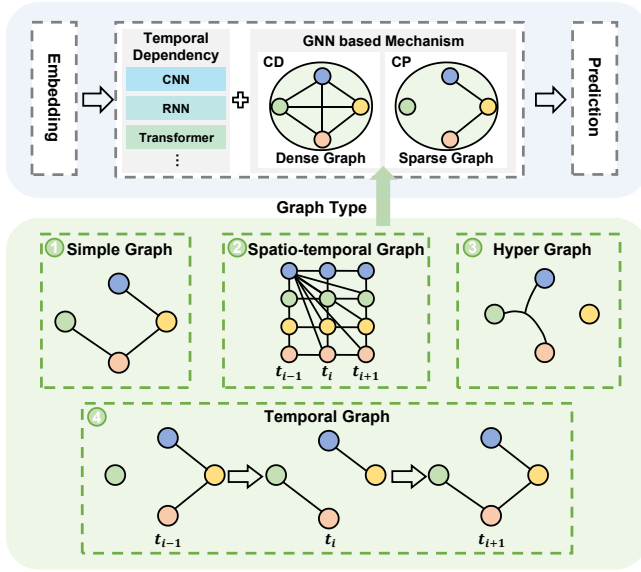


Figure 4: GNN-based mechanism for channel strategy.

MSGNet [Cai *et al.*, 2024], CrossGNN [Huang *et al.*, 2023a]) and data similarity metrics (GTS, WaveForM [Yang *et al.*, 2023]) to learn the correlation graph structure among multivariate channels. They utilize time-domain (MTGNN, MSGNet, CrossGNN, GTS) or frequency-domain (WaveForM) information as node learning features. Graph convolution-based message passing is applied within the simple graph to facilitate the transmission of dependency information among channels. II) **Spatio-temporal Graph**: Unlike a simple graph, a Spatio-temporal Graph incorporates multiple channels at different time steps into a single graph, further considering the relationships between channels across different time steps. This approach allows GNNs to simultaneously model both temporal and channel dependencies, effectively addressing potential compatibility issues between the temporal module and the GNNs. The main challenge of Spatio-temporal Graph-based methods is to address the efficiency issues in the graph construction and message passing stages. For example, FourierGNN uses fully connected graph construction and employs Fourier domain convolution operators to achieve a time complexity of  $O(N \log(N))$ . Similarly, FC-STGNN [Wang *et al.*, 2024c] adopts the same graph construction method and employs moving-pooling convolution to achieve the same time complexity. III) **Hyper Graph**: Hypergraphs are an extension of graphs that allow hyperedges to connect multiple vertices, enabling the modeling of higher-order group interactions. Models based on hypergraphs assume that the interactions among channels are not pairwise but involve group-based interactions among multiple channels. Therefore, hypergraph-based models are inherently suitable for constructing CP strategies. ReMo [Wu *et al.*, 2023b] and Ada-MSHyper construct multi-view and multi-scale hypergraphs, respectively, and design message passing mechanisms on these hypergraphs to enable group-wise message propagation. It is noteworthy that they use different MLPs or clustering constraints to promote the expression of heterogeneity among groups. IV) **Temporal Graph**: In the real

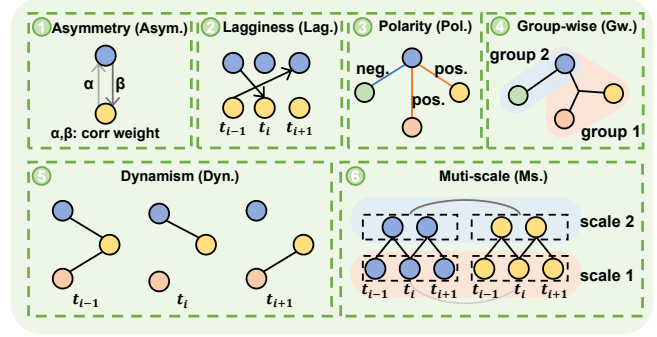


Figure 5: Characteristics perspective overview.

world, the correlation of time series data often changes over time, forming dynamic relational graphs. MTSF-DG and TPGNN [Liu *et al.*, 2022] use dynamic graphs and polynomial graphs, respectively, to model the variation patterns of these correlations. The CP model MTSF-DG combines historical and future relational graphs, leveraging memory networks and logical symbol learning to capture the impact of historical correlations on future correlations. The CD model TPGNN represents the correlation matrix as a matrix polynomial with time-varying coefficients to learn the evolving patterns of correlations.

**Others:** In addition to the mechanisms mentioned above, some models have proposed alternative approaches. For example: I) The CD model SOFTS [Han *et al.*, 2024a] introduces the STAR module, which utilizes a centralized structure to first aggregate information from all channels using MLPs, and then distribute the aggregated information to each channel. This interaction not only reduces the complexity of inter-channel interactions but also minimizes reliance on individual channel quality. II) The CP model LIFT [Zhao and Shen, 2024] proposes a novel plugin, adaptable to all MTSF specific models, that efficiently estimates leading indicators and their lead steps at each time step. This approach enables lagging channels to utilize advanced information from a pre-defined set of leading indicators. III) C-LoRA [Nie *et al.*, 2024] introduces a channel-aware low-rank adaptation (C-LoRA) plugin, which is adaptable to all MTSF specific models. It first parameterizes each channel with a low-rank factorized adapter to enable individualized treatment. The specialized channel adaptation is then conditioned on the series information to form an identity-aware embedding. Additionally, cross-channel relational dependencies are captured by integrating a globally shared CD model.

### 3.3 Characteristics Perspective

To better explore the channel correlation in MTSF, it is often necessary to delve into the different characteristics of correlations among time-series channels. This section will explain the six key characteristics (Figure 5) commonly considered in current methods.

**Asymmetry:** Asymmetry refers to the unequal relationships among channels in multivariate time series, where the degree of mutual influence is not identical across channels. Methods based on transformers and MLP, due to the specific

nature of their computational processes, inherently possess asymmetry, enabling them to effectively capture the asymmetric correlations. On the other hand, methods based on GNN establish directed, weighted graphs through asymmetric distance metrics, allowing interaction edges to have varying weights in different transmission directions, as seen in models such as MTGNN [Wu *et al.*, 2020], MSGnet [Cai *et al.*, 2024].

**Lagginess:** Lagginess refers to the fact that the current state of a certain channel not only depends on the current states of the other channels but may also be influenced by the past states of the other channels. Based on Lagginess characteristic, VCformer [Yang *et al.*, 2024] incorporates the joint effects of multi-step delays among channels when calculating the attention matrix. In contrast, FourierGNN [Yi *et al.*, 2023] and FC-STGNN [Wang *et al.*, 2024c] directly perform message passing between representations across different channels and time steps using spatiotemporal fully connected graphs. LIFT [Zhao and Shen, 2024], on the other hand, combines prior knowledge with neural network predictions to estimate the lag step.

**Polarity:** Polarity refers to the distinction between positive and negative correlations in the interactions among channels. During modeling, it is important to distinguish between these two types of interactions to avoid confusion. Cross-GNN [Huang *et al.*, 2023b] utilizes a sign graph approach, categorizing correlations into positive, negative, and neutral relationships. During message passing, it integrates both positive and negative information exchanges, thereby capturing the heterogeneity of correlations more effectively.

**Group-wise:** Group-wise refers to a phenomenon in which correlations among channels exhibit a grouping structure, characterized by strong correlations within the same group, weak correlations among different groups, and varying correlation dependencies across different groups. CCM [Chen *et al.*, 2024] and DUET [Qiu *et al.*, 2025] use clustering techniques to group channels for interaction, while ReMo [Wu *et al.*, 2023b] and Ada-MSHyper [Shang *et al.*, 2024] establish intra-group message passing through hyperedges. Furthermore, CCM and ReMo apply different MLPs for feature extraction within different groups, and Ada-MSHyper constrains hyperedges based on the loss function. These varying approaches facilitate the expression of differences among different groups.

**Dynamism:** The correlation among channels in multivariate time series exhibits different behaviors at different time steps, showing an overall dynamic change. First, methods based on MLP (such as TimeMixer [Wang *et al.*, 2024a], TTM [Ekambaram *et al.*, 2024]), where the weights remain constant across time steps, fail to capture dynamism. Methods that use Transformer to consider channel correlation typically employ series tokens or patch tokens. Methods based on series tokens, such as iTransformer [Liu *et al.*, 2024b] and DUET [Qiu *et al.*, 2025], cannot capture dynamism. However, methods based on patch tokens, such as Crossformer [Zhang and Yan, 2022], assign different attention scores at different time patches, enabling the modeling of dynamism. In GNNs, only approaches where the graph structure changes over time can capture dynamism, such as MS-

Table 2: Comparison among different channel strategies.

Dimension	CI	CD	CP
Efficiency	High	Low	Moderate
Robustness	High	Low	Moderate
Generalizability	Low	Moderate	High
Capacity	Low	High	Moderate
Ease of Implementation	High	Moderate	Low

GNet [Cai *et al.*, 2024], which computes the graph structure at each timestep. However, the aforementioned methods for modeling dynamism only consider different channel relationships at different time steps. In contrast, MSTF-DG [Zhao *et al.*, 2023], TPGNN [Liu *et al.*, 2022], and ESG [Ye *et al.*, 2022] propose that there is a direct connection between channel relationships across different time steps. For example, MSTF-DG uses previous channel relationships to directly infer the current channel relationships.

**Muti-scale:** Multi-scale refers to the phenomenon where the correlations among channels exhibit different behaviors at various time scales (such as hours, minutes, or seconds). MSGNet [Cai *et al.*, 2024] and Ada-MSHyper [Shang *et al.*, 2024] establish different graph structures across scales to describe the variations in correlation at different levels, and they achieve the fusion of correlation information at different scales through varying degrees of interaction. Considering the multi-scale heterogeneity of correlations helps the model better understand the multi-scale features of time series data, thereby generating more accurate predictions.

## 4 Comparison within the Taxonomy

In this section, we compare the strengths and limitations of CI, CD, and CP across multiple dimensions—see Table 2.

**Efficiency:** Efficiency measures the amount of resources consumed by a model during its operation, such as time and memory. CI is the most efficient strategy as it processes each channel independently without modeling inter-channel relationships. This results in the lowest computational complexity and excellent scalability for large datasets. CD is the least efficient strategy as it requires modeling all inter-channel dependencies. The computational complexity increases sharply with the number of channels, making it less scalable. The CP strategy strikes a balance by dynamically capturing interactions among channels while allowing each channel to focus only on the ones most relevant to itself. Thanks to its dynamical mechanism, which limits the modeling scope, CP remains more efficient than CD.

**Robustness:** Robustness refers to the ability of a model to maintain stability and effectiveness against noise, data variations, or interference. CI has a certain degree of robustness to noise as it processes each channel independently, avoiding interference among channels. CD has higher robustness for strongly correlated channels but is highly sensitive to noise. Spurious correlations can significantly degrade its performance, making it the least robust strategy. CP exhibits the high robustness by capturing flexible and dynamic relationships, making it highly effective in handling noise and variations in data distribution.

**Generalizability:** Generalizability refers to the ability of a model to perform well on unseen data or different datasets by leveraging patterns and relationships beyond the training data. CI has the weakest generalizability as it cannot leverage channel correlations, which is a critical drawback for multivariate time series tasks. CD demonstrates strong generalizability when channel correlations are consistent and significant. However, its performance deteriorates when relationships are weak or vary significantly across datasets. CP exhibits the strongest generalizability by handling overlapping and dynamic channel correlations. It adapts well to various datasets, particularly in complex real-world scenarios.

**Capacity:** Capacity refers to the ability of a model to capture and represent complex relationships and dependencies within the data. CI has the lowest capacity as it completely ignores inter-channel relationships and can only model the dynamics of individual channels. CD has the highest capacity as it simultaneously models all inter-channel relationships, allowing it to capture complex global dependencies. The CP strategy offers a balanced approach with moderate capacity. It models interactions selectively, allowing each channel to focus only on the channels most relevant to it.

**Ease of Implementation:** Ease of Implementation refers to how straightforward or complex it is to put a model into practice, considering the required components and design. CI is the easiest strategy to implement due to its simple structure, as it does not require modeling inter-channel relationships. CD is more complex to implement because it requires designing modules to capture inter-channel dependencies. CP is the most challenging strategy to implement as it requires a dynamic and flexible mechanism to model inter-channel dependencies. Typically, this involves the use of attention mechanisms or graph-based methods, which add to the difficulty of implementation.

## 5 Future Research Opportunities

### 5.1 Channel Correlation in Future Horizon

Currently, few models address the correlation relationships within the prediction horizon. The correlation within the prediction horizon directly impacts the quality of the prediction results. Although some methods, such as TPGNN [Liu *et al.*, 2022] and MTSF-DG [Zhao *et al.*, 2023], predict the channel correlation of the future horizon using temporal graph-based approaches and apply them accordingly, they focus on short-term forecasting and, due to performance limitations, are difficult to scale to long-term forecasting.

### 5.2 Other Correlation Characteristics

Existing research methods have explored and analyzed six characteristics of channel correlations—see section 3.3. However, in real-world scenarios, correlations also contain additional characteristics, such as: I) **Multi-component:** DLinear [Zeng *et al.*, 2023] and AutoFormer [Wu *et al.*, 2021] have demonstrated that decomposing time series into multiple components, such as trend and seasonality, significantly contributes to MTSF. Future research could explore how to model the channel correlations within each component

separately, as well as how to integrate the channel correlations across multiple components. II) **Multi-frequency:** Correlations may manifest differently across various frequency components of time series data, and so on. Further exploration of these characteristics can help models better understand and utilize the correlations between channels, ultimately enhancing their predictive and inferential capabilities.

### 5.3 Multi-modality for Channel Correlations

Multiple modalities can be introduced to more comprehensively model the correlation among channels. Compared to a single time series modality, multimodal data can provide richer information sources, such as text, images, or other sensor data, which can compensate for potential gaps in time series data. By extracting features from multimodal data, the unique characteristics of channels across different modalities can be captured. Subsequently, cross-modal relational modeling mechanisms, such as cross-modal attention mechanisms or GNN, can be employed to uncover the dynamic dependencies among channels. Additionally, to further enhance the modeling of channel correlation, an adaptive fusion mechanism can be designed to dynamically adjust interaction weights based on the correlations among different modalities.

### 5.4 Channel Strategy of Foundation Models

Multivariate time series foundation models follow two main approaches: LLM-based models and time series pre-trained models. LLM-based models, lacking a channel dimension in language modality, typically adopt a CI strategy [Jin *et al.*, 2024; Chang *et al.*, 2023]. Due to the high heterogeneity in the number of channels in time series data, most time series pre-trained models, such as Timer [Liu *et al.*, 2024c] and Chronos [Ansari *et al.*, 2024], use a CI strategy to ensure robust predictions while avoiding complex channel correlation. In contrast, models like MOIRIA [Woo *et al.*, 2024] and UniTS [Gao *et al.*, 2024] incorporate channel correlation during pretraining. MOIRIA flattens all channels, using positional embeddings to distinguish them, capturing both temporal and channel relationships with self-attention, while UniTS directly captures channel correlations via self-attention in the channel dimension. However, time series pre-trained models only consider CD strategy to capture the channel correlations, without fully considering the intricate and diverse channel correlations in different pretrain datasets, where CP strategy may achieve better performance. There also lacks works in LLM-based models to consider the channel correlations combined with the multimodal data. Existing approaches remain relatively basic, leaving significant room for improving channel strategies in foundation models.

## 6 Conclusion

In this survey, we provide a comprehensive review of deep learning methods for MTSF from a channel strategy perspective. We categorize and summarize existing approaches using a proposed methodological taxonomy, providing a structured understanding of the field. Additionally, we offer insights into the strengths and limitations of various channel strategies and outline future research directions to further advance MTSF.

## References

- [Ansari *et al.*, 2024] Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Türkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda-Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *CoRR*, abs/2403.07815, 2024.
- [Cai *et al.*, 2024] Wanlin Cai, Yuxuan Liang, Xianggen Liu, Jianshuai Feng, and Yuankai Wu. Msgnet: Learning multi-scale inter-series correlations for multivariate time series forecasting. In *AAAI*, pages 11141–11149, 2024.
- [Chang *et al.*, 2023] Ching Chang, Wen-Chih Peng, and Tien-Fu Chen. LLM4TS: two-stage fine-tuning for time-series forecasting with pre-trained llms. *CoRR*, abs/2308.08469, 2023.
- [Chen *et al.*, 2024] Jialin Chen, Jan Eric Lenssen, Aosong Feng, Weihua Hu, Matthias Fey, Leandros Tassioulas, Jure Leskovec, and Rex Ying. From similarity to superiority: Channel clustering for time series forecasting. In *NeurIPS*, 2024.
- [Cirstea *et al.*, 2021] Razvan-Gabriel Cirstea, Tung Kieu, Chenjuan Guo, Bin Yang, and Sinno Jialin Pan. Enhancenet: Plugin neural networks for enhancing correlated time series forecasting. In *ICDE*, pages 1739–1750, 2021.
- [Ekambaram *et al.*, 2023] Vijay Ekambaram, Arindam Jati, Nam Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting. In *SIGKDD*, 2023.
- [Ekambaram *et al.*, 2024] Vijay Ekambaram, Arindam Jati, Nam H. Nguyen, Pankaj Dayama, Chandra Reddy, Wesley M. Gifford, and Jayant Kalagnanam. Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. *CoRR*, abs/2401.03955, 2024.
- [Gao *et al.*, 2024] Shanghua Gao, Teddy Koker, Owen Queen, Thomas Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. Units: Building a unified time series model. *arXiv preprint arXiv:2403.00131*, 2024.
- [Han *et al.*, 2024a] Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. Softs: Efficient multivariate time series forecasting with series-core fusion. In *NeurIPS*, 2024.
- [Han *et al.*, 2024b] Wenyong Han, Tao Zhu, Liming Chen, Huansheng Ning, Yang Luo, and Yaping Wan. Mcformer: Multivariate time series forecasting with mixed-channels transformer. *IEEE Internet of Things Journal*, 2024.
- [Huang *et al.*, 2023a] Qihe Huang, Lei Shen, Ruixin Zhang, Shouhong Ding, Binwu Wang, Zhengyang Zhou, and Yang Wang. Crossgnn: Confronting noisy multivariate time series via cross interaction refinement. In *NeurIPS*, 2023.
- [Huang *et al.*, 2023b] Qihe Huang, Lei Shen, Ruixin Zhang, Shouhong Ding, Binwu Wang, Zhengyang Zhou, and Yang Wang. Crossgnn: Confronting noisy multivariate time series via cross interaction refinement. In *NeurIPS*, volume 36, pages 46885–46902, 2023.
- [Jiang *et al.*, 2023] Maowei Jiang, Pengyu Zeng, Kai Wang, Huan Liu, Wenbo Chen, and Haoran Liu. Fecan: Frequency enhanced channel attention mechanism for time series forecasting. *Advanced Engineering Informatics*, 58:102158, 2023.
- [Jin *et al.*, 2024] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-llm: Time series forecasting by reprogramming large language models. In *ICLR*, 2024.
- [Kim *et al.*, 2021] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *ICLR*, 2021.
- [Lee *et al.*, 2024] Seunghan Lee, Taeyoung Park, and Kibok Lee. Partial channel dependence with channel masks for time series foundation model. In *NeurIPS Workshop on Time Series in the Age of Large Models*, 2024.
- [Lin *et al.*, 2024] Shengsheng Lin, Weiwei Lin, HU Xinyi, Wentai Wu, Ruichao Mo, and Haocheng Zhong. Cyclenet: Enhancing time series forecasting through modeling periodic patterns. In *NeurIPS*, 2024.
- [Liu *et al.*, 2022] Yijing Liu, Qinxian Liu, Jian-Wei Zhang, Haozhe Feng, Zhongwei Wang, Zihan Zhou, and Wei Chen. Multivariate time-series forecasting with temporal polynomial graph neural networks. In *NeurIPS*, 2022.
- [Liu *et al.*, 2024a] Qinshuo Liu, Yanwen Fang, Pengtao Jiang, and Guodong Li. Dgcformer: Deep graph clustering transformer for multivariate time series forecasting. *arXiv preprint arXiv:2405.08440*, 2024.
- [Liu *et al.*, 2024b] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *ICLR*, 2024.
- [Liu *et al.*, 2024c] Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *ICML*, 2024.
- [Luo and Wang, 2024] Donghao Luo and Xue Wang. ModernTCN: A modern pure convolution structure for general time series analysis. In *ICLR*, 2024.
- [Nie *et al.*, 2023] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *ICLR*, 2023.
- [Nie *et al.*, 2024] Tong Nie, Yuewen Mei, Guoyang Qin, Jian Sun, and Wei Ma. Channel-aware low-rank adaptation in time series forecasting. In *CIKM*, pages 3959–3963, 2024.
- [Qiu *et al.*, 2024] Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, Zhenli Sheng, and Bin



- Yang. TFB: towards comprehensive and fair benchmarking of time series forecasting methods. *Proc. VLDB Endow.*, 17(9):2363–2377, 2024.
- [Qiu *et al.*, 2025] Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. Duet: Dual clustering enhanced multivariate time series forecasting. In *SIGKDD*, 2025.
- [Shang *et al.*, 2021] Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. *arXiv preprint arXiv:2101.06861*, 2021.
- [Shang *et al.*, 2024] Zongjiang Shang, Ling Chen, Binqing Wu, and Dongliang Cui. Ada-mshyper: Adaptive multi-scale hypergraph transformer for time series forecasting. *CoRR*, abs/2410.23992, 2024.
- [Wang *et al.*, 2024a] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *ICLR*, 2024.
- [Wang *et al.*, 2024b] Xue Wang, Tian Zhou, Qingsong Wen, Jinyang Gao, Bolin Ding, and Rong Jin. Card: Channel aligned robust blend transformer for time series forecasting. In *ICLR*, 2024.
- [Wang *et al.*, 2024c] Yucheng Wang, Yuecong Xu, Jianfei Yang, Min Wu, Xiaoli Li, Lihua Xie, and Zhenghua Chen. Fully-connected spatial-temporal graph for multivariate time-series data. In *AAAI*, pages 15715–15724, 2024.
- [Wang *et al.*, 2024d] Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. *arXiv preprint arXiv:2407.13278*, 2024.
- [Wen *et al.*, 2023] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. In *IJCAI*, pages 6778–6786, 2023.
- [Woo *et al.*, 2024] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *ICML*, 2024.
- [Wu *et al.*, 2020] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *SIGKDD*, pages 753–763, 2020.
- [Wu *et al.*, 2021] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *NeurIPS*, pages 22419–22430, 2021.
- [Wu *et al.*, 2023a] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *ICLR*, 2023.
- [Wu *et al.*, 2023b] Jinming Wu, Qi Qi, Jingyu Wang, Haifeng Sun, Zhikang Wu, Zirui Zhuang, and Jianxin Liao. Not only pairwise relationships: Fine-grained relational modeling for multivariate time series forecasting. In *IJCAI*, pages 4416–4423, 2023.
- [Wu *et al.*, 2024] Xinle Wu, Xingjian Wu, Bin Yang, Lekui Zhou, Chenjuan Guo, Xiangfei Qiu, Jilin Hu, Zhenli Sheng, and Christian S Jensen. Autocts++: zero-shot joint neural architecture and hyperparameter search for correlated time series forecasting. *The VLDB Journal*, 33(5):1743–1770, 2024.
- [Yang *et al.*, 2023] Fuhao Yang, Xin Li, Min Wang, Hongyu Zang, Wei Pang, and Mingzhong Wang. Waveform: Graph enhanced wavelet learning for long sequence forecasting of multivariate time series. In *AAAI*, volume 37, pages 10754–10761, 2023.
- [Yang *et al.*, 2024] Yingnan Yang, Qingling Zhu, and Jianyong Chen. Vcformer: Variable correlation transformer with inherent lagged correlation for multivariate time series forecasting. *arXiv preprint arXiv:2405.11470*, 2024.
- [Ye *et al.*, 2022] Junchen Ye, Zihan Liu, Bowen Du, Leilei Sun, Weimiao Li, Yanjie Fu, and Hui Xiong. Learning the evolutionary and multi-scale graph structure for multivariate time series forecasting. In *SIGKDD*, page 2296–2306, 2022.
- [Yi *et al.*, 2023] Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhen-dong Niu. Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective. In *NeurIPS*, 2023.
- [Yu *et al.*, 2024] Chengqing Yu, Fei Wang, Zezhi Shao, Tangwen Qian, Zhao Zhang, Wei Wei, and Yongjun Xu. Ginar: An end-to-end multivariate time series forecasting model suitable for variable missing. In *SIGKDD*, pages 3989–4000, 2024.
- [Zeng *et al.*, 2023] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *AAAI*, volume 37, pages 11121–11128, 2023.
- [Zhang and Yan, 2022] Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *ICLR*, 2022.
- [Zhao and Shen, 2024] Lifan Zhao and Yanyan Shen. Rethinking channel dependence for multivariate time series forecasting: Learning from leading indicators. In *ICLR*, 2024.
- [Zhao *et al.*, 2023] Kai Zhao, Chenjuan Guo, Yunyao Cheng, Peng Han, Miao Zhang, and Bin Yang. Multiple time series forecasting with dynamic graph modeling. *Proc. VLDB Endow.*, 17(4):753–765, 2023.
- [Zhou *et al.*, 2021] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, volume 35, pages 11106–11115, 2021.
- [Zhou *et al.*, 2022] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *ICML*, pages 27268–27286, 2022.