MTS-UNMixers: Multivariate Time Series Forecasting via Channel-Time Dual Unmixing

1st Xuanbing Zhu
School of Information and Communication Engineering
Dalian University of Technology
DaLian, China
zhuxuanbing@mail.dlut.edu.cn

2nd Dunbin Shen
School of Information and Communication Engineering
Dalian University of Technology
DaLian, China
sdb_2012@163.com

3rd Zhongwen Rao
Huawei Noah's Ark Lab
ShenZhen, China
raozhongwen@huawei.com

4th Huiyi Ma
School of Information and Communication Engineering
Dalian University of Technology
DaLian, China
mahuiyi@mail.dlut.edu.cn

5th Yingguang Hao
School of Information and Communication Engineering
Dalian University of Technology
DaLian, China
yghao@dlut.edu.cn

6th Hongyu Wang✉
School of Information and Communication Engineering
Dalian University of Technology
DaLian, China
whyu@dlut.edu.cn

***Abstract*—Multivariate time series data provide a robust framework for future predictions by leveraging information across multiple dimensions, ensuring broad applicability in practical scenarios. However, their high dimensionality and mixing patterns pose significant challenges in establishing an interpretable and explicit mapping between historical and future series, as well as extracting long-range feature dependencies. To address these challenges, we propose a channel-time dual unmixing network for multivariate time series forecasting (named MTS-UNMixer), which decomposes the entire series into critical bases and coefficients across both the time and channel dimensions. This approach establishes a robust sharing mechanism between historical and future series, enabling accurate representation and enhancing physical interpretability. Specifically, MTS-UNMixers represent sequences over time as a mixture of multiple trends and cycles, with the time-correlated representation coefficients shared across both historical and future time periods. In contrast, sequence over channels can be decomposed into multiple tick-wise bases, which characterize the channel correlations and are shared across the whole series. To estimate the shared time-dependent coefficients, a vanilla Mamba network is employed, leveraging its alignment with directional causality. Conversely, a bidirectional Mamba network is utilized to model the shared channel-correlated bases, accommodating noncausal relationships. Experimental results show that MTS-UNMixers significantly outperform existing methods on multiple benchmark datasets. The code is available at https://github.com/ZHU-0108/MTS-UNMixers.**

***Index Terms*—time series forecasting, unmixing, mamba network, shared.**

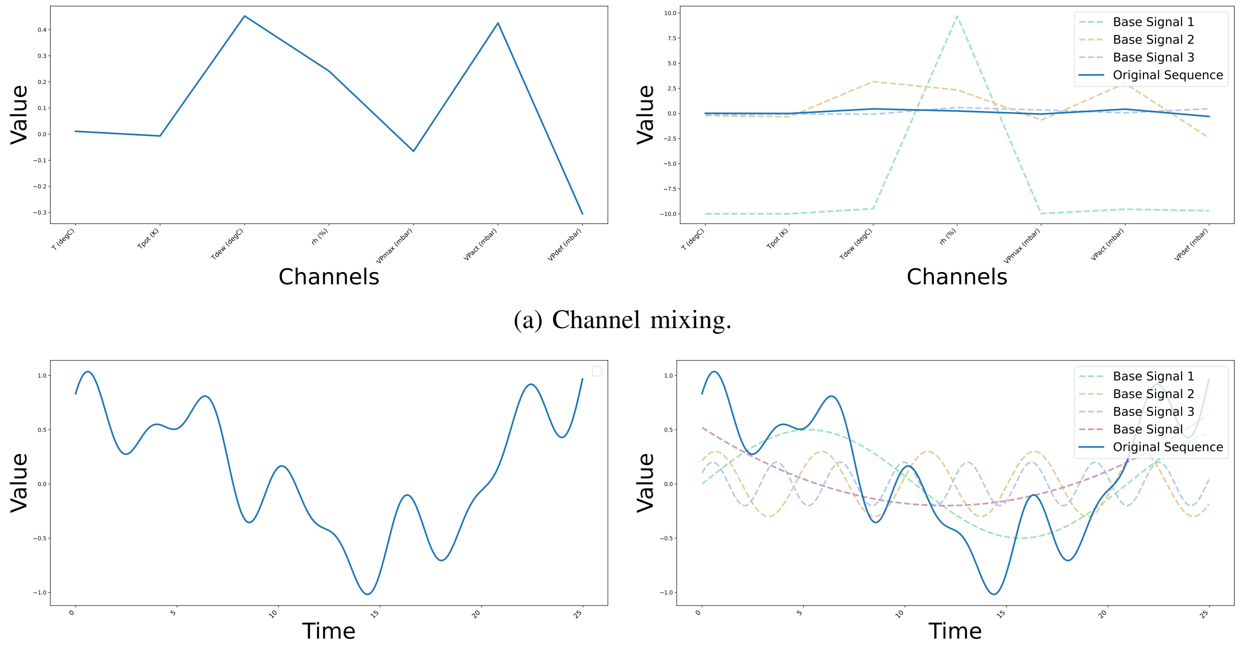
I. Introduction

The time series forecasting aims to provide accurate predictions of future series values by analyzing time-dependent patterns and trends in historical data. It is a core task in data analytics and is widely used in financial markets [1], weather forecasting [2], [3], electric power forecasting [4], [5], and traffic flow estimation [6]–[8], among other fields. With the exponential growth of data volumes and significant advancements in computational power, predictive models now face two key challenges. The first challenge is modeling complex nonlinear relationships over time to accurately capture essential features within long-term sequences. The second involves extracting interactions from multivariate data to better understand the dynamic variations in time series and identify latent patterns. To address these challenges, models have to effectively integrate data from multiple sources and also have robust prediction capabilities.

Recently, many deep learning models have been proposed to address the challenges of long series dependence and multivariate modeling in multivariate time series data. To capture long-series features more effectively, Chen et al. [9] introduced adaptive multiscale modeling with temporal dynamic inputs, which leverages both local and global information on the time axis. Wang et al. [10] integrated multiscale time series by incorporating micro-seasonal information and macro-trend data, utilizing complementary forecasting capabilities to improve overall performance. In order to extract the interaction information in multivariate time series, Liu et al. [11] redesigned the Transformer architecture to utilize the attention mechanism to capture the correlation between variables, and at the same time utilized the feed-forward neural network to extract the temporal features, which effectively enhances the cross-channel correlation extraction. Meanwhile, Li et al. [12] employed a decomposition module to capture inter-channel relationships. Its relatively simple structure offers significant efficiency advantages over more complex attention-based mechanisms.

Although recent advances have been made in dealing with

This work was supported by Dalian Science and Technology Innovation-Fund Project 2022JJ11CG002. The corresponding author is Hongyu Wang.



(a) Channel mixing.

(b) Time mixing.

Fig. 1: Mixing problem (taking weather data as an example). (a) shows the original plot of the seven variables and their principal component composition in the weather dataset. The high correlation between variables suggests that they may be influenced by common external environmental factors. (b) illustrates the time series of a single channel and shows how it can be decomposed into a mixture of different features.

long series features and multivariate interactions, these models still face limitations. The first challenge is the mixing problem in multivariate time series data, where different features intertwine with each other during the modeling process, making it difficult for the model to investigate individual contributions from each feature. Specifically:

- In the channel dimension, multivariables tend to exhibit high correlation, leading to feature redundancy. As shown in the weather dataset in Fig. 1 (a), variables such as temperature, humidity, and cloud cover are all affected by the dominant temperature trend, which results in similar variations between variables. This similarity reflects the natural correlation between weather variables and may lead to feature overlap and blending, which in turn increases computational complexity and amplifies noise during channel blending.
- In the time dimension, different periods often represent a mixture of patterns, such as cycles and trends, rather than a single feature. For example, in the time series data shown in Fig. 1 (b), we can observe multiple cyclical features (e.g., daily variations or seasonal fluctuations) coexisting with long-term trends. These patterns overlap each other on the time axis, which not only complicates component separation, but also masks localized patterns and interferes with long-term dependencies.

The second challenge lies in the limitations of traditional “black-box” mapping models in modeling the relationship between historical and future sequences. Black-box mappings typically rely on abstract feature representations and often lack the ability to precisely map between sequences. They also does not have an explicit physical meaning and lacks interpretability of the entire sequence characteristics.

To address the above limitations, we propose MTS-UNMixers, which establishes an explicit mapping of sequences through a ummixing mechanism. MTS-UNMixers decouples the historical sequences in time and channel, and extracts the significant components on the time and channel axes. Specifically, MTS-UNMixers uses a mixture model to represent a single-channel (multi-time) series as a combination of several trend and period components. For a single moment (multi-variate) it can be represented by a number of cardinalities which describe the correlation between the individual variables, called correlation components.The historical time horizon and the future time horizon are treated as a unified whole, allowing for unmixed and explicit mapping by sharing underlying signal and weight information. In temporal unmixing, we use the Mamba network. The network effectively extracts the nonlinear causal dependencies in the sequence through the dynamic causal mechanism in the state-space model, by utilising the nonlinear multiplication between the basis signal matrix and the component coefficient matrix. In channel unmixing, we employ a bidirectional Mamba network since there is no causal relationship between channels, but rather a bidirectional interaction between highly correlated variables. In this way, the model captures the bidirectional correlations and interactions between variables, realizes the decoupling of independent features in multiple channels, and improves the clarity and accuracy of feature extraction.

In summary, the main contributions of this paper are as follows:

- To address the problem of reduced model accuracy due to feature overlap and correlation in high-dimensional time series data, we propose a dual-mixing model to decompose the entire series into critical bases and coefficients along both the time and channel dimension. This decomposition can effectively investigate the individual contributions of intrinsic time-dependent and channel-correlated bases, reducing the noise and redundancy.
- For effective sharing, we adopt an explicit mapping model. Historical and future sequences are treated as a unified whole, sharing the component coefficient matrix in the channel dimension and the base signal matrix in the time dimension. This improves physical interpretability and prediction reliability while ensuring the conciseness and simplicity of the model.
- In the time dimension, we use Mamba to capture the characteristics of long-term temporal dependencies, while channel dimension where no causal relationship exists, we use bi-directional Mamba to effectively capture the bidirectional interactions between variables.

The remainder of this article is organized as follows. First, we provide a brief overview of the relevant research on time series forecasting in Section II. Second, we describe the problem in Section III. Next, the proposed MTS-UNMixers method is presented in detail in Section IV. Then, the experiments and corresponding analyses are presented in Section V. Finally, conclusions are drawn in Section VI.

II. Related Work

A. Deep learning Models for Time Series Forecasting

In recent years, deep learning has been widely applied in time series forecasting, and the models can be broadly categorized into attention-based and non-attention-based approaches. Attention-based models primarily include the Transformer series, which utilize self-attention mechanisms to capture long-range dependencies and relationships between different time points in time series data. These models are particularly suited for forecasting tasks involving long sequences and high-dimensional data. Transformer models, with self-attention mechanisms, excel in capturing long-range dependencies and inter-timepoint relationships, making them well-suited for long-sequence and high-dimensional forecasting tasks [13], [14], [15], [16] .Nie et al. proposed PatchTST [13], a Transformer-based model design that enhances efficiency and long-term forecasting accuracy for multivariate time series by leveraging time series segmentation and channel independence. Zhou et al. proposed FEDformer [14] a method that combines Transformer with seasonal-trend decomposition and frequency enhancement to effectively capture both global structure and detailed features of time series, improving long-term forecasting performance.

Non-attention-based models include recurrent neural network (RNN)-based, convolutional neural network (CNN)-based, mamba-based and multi-layer perceptron (MLP)-based models.These models capture local dependencies, time series characteristics, and multi-scale features in time series data through various modeling approaches. Technically, MLP-based methods apply multi-layer perceptrons along the temporal dimension, offering simplicity and competitive performance, especially with high-dimensional data [10], [17], [18]. Wang et al. [10] proposed TimeMixer, a fully MLP-based architecture that leverages Past-Decomposable-Mixing (PDM) and Future-Multipredictor-Mixing (FMM) modules to disentangle and capture multiscale temporal variations for enhanced time series forecasting. CNN-based models leverage convolutional kernels to effectively capture local patterns over time [19], [20], [21]. Liu et al. proposed SCINet [19], a novel neural network architecture that recursively downsamples, convolves, and interacts to effectively model complex temporal dynamics, achieving significant improvements in forecasting accuracy. RNN-based models utilize recurrent structures to manage temporal dependencies in sequential data [22], [23]. David Salinas et al. proposed the DeepAR method [22], which achieves high-precision probabilistic forecasting by training an auto-regressive recurrent network model on a large collection of related time series. Recently, Gu and Dao presented Mamba [24], integrating parameterized matrices and hardware-aware parallel computing, achieving superior performance in multiple tasks. Derived from Mamba, several models have further advanced time series forecasting. Ma et al. proposed FMamba [25], combining fast-attention with Mamba for efficient temporal and inter-variable dependency modeling. Patro and Agneeswaran introduced SiMBA [26], a simplified Mamba-based architecture using EinFFT for channel modeling, outperforming existing SSMs in both image and time series benchmarks. Tang et al. [27] introduced VMRNN, integrating Vision Mamba blocks with LSTM for enhanced spatiotemporal forecasting with a smaller model size. Ahamed and Cheng proposed TimeMachine [28], a quadruple-Mamba architecture achieving superior accuracy, scalability, and memory efficiency in long-term time series forecasting. However, despite the promising progress, the full potential of the Mamba model in more complex and challenging prediction tasks has not been fully utilised, indicating a need for further exploration and refinement.

B. State space models

State Space Models (SSMs) [24], [30]–[32] use intermediate state variables to achieve sequence-to-sequence mapping, enabling efficient handling of long sequences. The core concept of SSMs is to capture complex dynamic features through state variables, addressing the computational and memory bottlenecks encountered in traditional long-sequence modeling. However, early SSM models often required high-dimensional matrix operations, leading to substantial computational and memory demands. Rangapuram et al. proposed Deep State Space Models (DSSM) [31], integrating latent states with deep neural networks to better model complex dynamics. Lin et al. introduced SSDNet [32], which combines Transformer architecture with SSMs for efficient temporal pattern learning and direct parameter estimation, avoiding Kalman filters.

To overcome these limitations, structured state-space models like S4 [30] introduced innovative improvements. S4 employs low-rank corrections to regulate certain model parameters, ensuring stable diagonalization and reducing the SSM framework to a well-studied Cauchy kernel. This transformation not only significantly reduces computational costs but also maintains performance while drastically lowering memory requirements, making SSM more practical for long-sequence modeling.

Building on this foundation, Mamba [24] further optimized SSM by introducing a selection mechanism and hardware-friendly algorithm design. Mamba selection mechanism enables the model to dynamically adjust parameters based on the input sequence, effectively addressing the discrete modality problem and achieving more efficient processing for long sequences in domains like natural language and genomics. This hardware-optimized and parameter-dynamic strategy not only reduces computational complexity but also improves computational resource efficiency, making Mamba highly effective for long-sequence modeling tasks. Together, the advancements in S4 and Mamba have enhanced the applicability of SSMs in long-sequence data analysis, providing valuable insights for other complex sequence modeling tasks.

III. Channel-Time Mixing

In this section, we first give a brief introduction to the multivariate time series prediction problem and its bottlenecks. We then formulate the problem. Finally, we describe the process of optimally solving the problem.

A. Multivariate Time Series Forecasting

Multivariate Time Series Forecasting (MTSF) is a method that predicts future values by analyzing time series data consisting of multiple variables. Given a historical multivariate time series input $\mathbf{X} \in \mathbb{R}^{T \times N}$, where T represents the number of time steps and N represents the number of variables, the objective is to predict the future target values for the next H time steps. The prediction sequence is denoted as $\hat{\mathbf{X}} \in \mathbb{R}^{H \times N}$. In MTSF, the data typically contains complex mixed signals across both the temporal and channel dimensions, making it highly challenging to effectively capture meaningful patterns and relationships. The current challenges primarily include: 1) time series signals often contain various mixed components, making feature extraction and unmixing complex; 2) achieving stable mapping between historical and future sequences, as current features cannot be expressed in an explicit mapping across the entire sequence. To address these challenges, we propose a method called MTS-UNMixing, which employs unmixing techniques to mitigate these issues.

B. Problem Formulation

To solve the problem in MTSF, we formulate the sequence in an equation, that is, we build a mixed model to obtain the main components of each dimension. The components are shared by explicitly mapping them to the sequence, and finally an optimisation equation is proposed for solving.

a) Mixing Model: In multivariate time series data, each variable exhibits different feature patterns along the temporal axis. Consider a multivariate time series with T time steps and N variables, represented as $X = \{x_i\}_{i=1}^{T} \in \mathbb{R}^N$.

According to the temporal mixing model, the observation at a given time step consists of several primary temporal patterns (referred to as basis signals). The data at a single time step can be expressed as:

$$x_i = \sum_{k=1}^{k_1} s_{ik} a_k = A_t s_{i,t}, \tag{1}$$

where k_1 represents the number of basis signals, $\{a_k\}_{k=1}^{k_1}$ denotes k_1 basis signals, $\{s_{ik}\}_{k=1}^{k_1}$ represents the weight of the k-th basis signal at time step i, $A_t = [a_{t,1}, a_{t,2}, \ldots, a_{t,k_1}] \in \mathbb{R}^{N \times k_1}$ is the basis matrix containing k_1 distinct basis signals, and $s_{i,t} = [s_{i,t,1}, s_{i,t,2}, \ldots, s_{i,t,k_1}]^T \in \mathbb{R}^{k_1}$ is the weight vector for time step i.

For T time steps, the matrix form can be defined as:

$$X = A_t S_t, \tag{2}$$

where $S_t = [s_{1,t}, s_{2,t}, \ldots, s_{T,t}] \in \mathbb{R}^{k_1 \times T}$ is the coefficient matrix. To ensure physical interpretability, the coefficient matrix must satisfy the following constraints:

- **Sum-to-one constraint**: The coefficients for each time step must sum to 1, i.e.,

$$E_{k_1}^T S_t = E_T^T, \tag{3}$$

  where $E_{k_1} = [1, 1, \ldots, 1]^T$ is a column vector with all elements equal to 1, and $[\cdot]^T$ denotes the transpose operation.
- **Non-negativity constraint**: All elements of the coefficient matrix must be non-negative, i.e.,

$$S_t \geq 0. \tag{4}$$

Thus, the temporal mixing model can be rewritten as:

$$X = A_t S_t \quad \text{s.t.} \quad E_{k_1}^T S_t = E_T^T, \quad S_t \geq 0. \tag{5}$$

Similar to the time-domain mixture model, the channel mixture model considers the relationship between different channels at each time step. The data of each channel is expressed as a linear combination of several primary channel patterns (base signals), which is expressed as:

$$X = A_c S_c \quad \text{s.t.} \quad E_{k_2}^T S_c = E_N^T, \quad S_c \geq 0, \tag{6}$$

where $A_c \in \mathbb{R}^{T \times k_2}$ is the basis matrix containing k_2 primary channel patterns, and $S_c \in \mathbb{R}^{k_2 \times N}$ is the coefficient matrix, describing the contributions of each basis signal to different channels. The sum-to-one constraint $E_{k_2}^T S_c = E_N^T$ ensures the relative contributions of the basis signals, while the non-negativity constraint $S_c \geq 0$ guarantees physical interpretability.

b) Explicit Mapping: To establish a clear and interpretable relationship between historical and future sequences, we propose an explicit mapping mechanism. This mechanism extracts global features in the temporal and channel dimensions to enable the sharing of key components. Explicit mapping assumes that historical and future sequences can be represented through a unified mixing model to capture the overall patterns. Specifically, the sequences can be expressed as:

$$X = \left[A'_c, \hat{A}_c\right] S_c, \quad X = A_t \left[S'_t, \hat{S}_t\right], \tag{7}$$

where A'_c and $\hat{A}_c$ are the channel basis matrices for the historical and future sequences, respectively. S'_t and $\hat{S}_t$ are the temporal coefficient matrices for the historical and future sequences. S_c is the shared channel coefficient matrix, ensuring consistent inter-variable relationships between the historical and future sequences. $\hat{A}_c \in \mathbb{R}^{H \times k_2}$ is the channel basis matrix for the future sequence. Similarly, A_t is the shared temporal basis matrix, capturing primary temporal features and ensuring consistency in trends and periodicity across the sequences. $\hat{S}_t \in \mathbb{R}^{k_1 \times H}$ is the temporal coefficient matrix for the future sequence, describing its temporal feature distribution. This unified representation enables consistent modeling of both historical and future sequences and achieves explicit feature decomposition in both dimensions.

Based on the equation 7, the historical sequence can be reconstructed as:

$$X_h = A_t S'_t, \quad X_h = A'_c S_c, \tag{8}$$

where A_t is the shared temporal basis matrix, capturing trends and periodic patterns in the historical sequence. S'_t is the temporal coefficient matrix, describing the contribution of each temporal feature in the historical data. Similarly, A'_c is the channel basis matrix, reflecting major patterns among channels, and S_c is the shared channel coefficient matrix, indicating the mixing weights for variables.

The future sequence is predicted as:

$$\hat{X} = A_t \hat{S}_t, \quad \hat{X} = \hat{A}_c S_c, \tag{9}$$

where shared components ensure continuity between historical and future features. The temporal basis A_t maintains consistency in trends and periodic patterns, while the channel coefficient S_c ensures stable inter-variable relationships.

The explicit mapping mechanism allows for a clear decoupling of features in the time and channel dimensions. Shared components ensure the continuity of features between historical and future sequences. In the time dimension, the shared basis matrix (A_t) ensures the consistency of trends and cyclical patterns, thereby enhancing the ability of the model to capture correlations. In the channel dimension, the shared coefficient matrix (S_c) maintains the stability of the relationship between variables and reduces redundancy during channel mixing. This explicit mapping design improves the interpretability of the model and enhances the accuracy and robustness of the predictions.

c) Optimization Process: To accurately model the feature relationships in both historical and future sequences, we need to estimate both the coefficient matrix and the basis matrix. Specifically, the goal is to minimize the reconstruction error to effectively capture the temporal and channel characteristics of the data. Our objective function is defined as:

$$(A, S) = \arg\min_{A,S} \|X - AS\|_1, \tag{10}$$

where $\|\cdot\|_1$ denotes the L1 norm, used here to measure reconstruction error. Since we effectively extract features from both the temporal and channel axes, the optimization process is integrated into the following form:

$$\begin{aligned} (A_c, S_c, A_t, S_t) = \arg\min_{A_c, S_c} &\|X - A_c S_c\|_1 \\ + \arg\min_{A_t, S_t} &\|X - A_t S_t\|_1 \qquad (11) \\ \text{s.t.} \quad E_{k_1}^T S_t = E_T^T, &\quad E_{k_2}^T S_c = E_N^T. \end{aligned}$$

With this formulation, we simultaneously extract features along the temporal and channel dimensions and address both reconstruction and prediction within a unified framework. Ultimately, our optimization task is divided into two parts: reconstructing the historical sequence and modeling the future sequence. To achieve accurate reconstruction and prediction, we only need to estimate the coefficient matrix and basis matrix for both historical and future sequences, ensuring feature consistency across different time points and variables.

IV. MTS-UNMixers Network

This section provides a detailed overview of the proposed method. We first present the overall architecture, followed by in-depth descriptions of three key components: the unmixing network, the prediction layer, and the loss function.

A. *Overall Architecture*

To address the issues outlined in the previous section (equation 11), we propose a network called MTS-UNMixers, which utilizes Mamba blocks for unmixing. The overall framework is shown in Fig. 2. This design includes two main dimensions: the horizontal dimension, which represents each time frame (channel-wise representation) and each channel (tick-wise representation), and the vertical dimension, which represents the stages of model unmixing and reconstruction-prediction. The first main path focuses on the temporal aspect. In the unmixing stage, Mamba blocks extract complex dynamic patterns and main features over long time steps using nonlinear computation, resulting in a sequence-based coefficient matrix. This matrix is then multiplied with the basis signals to obtain both the reconstructed and predicted sequences. The second main path addresses the channel aspect. Here, the unmixing stage employs bidirectional Mamba blocks, performing deep calculations in different directions to extract low-dimensional similar features between channels. This results in a shared

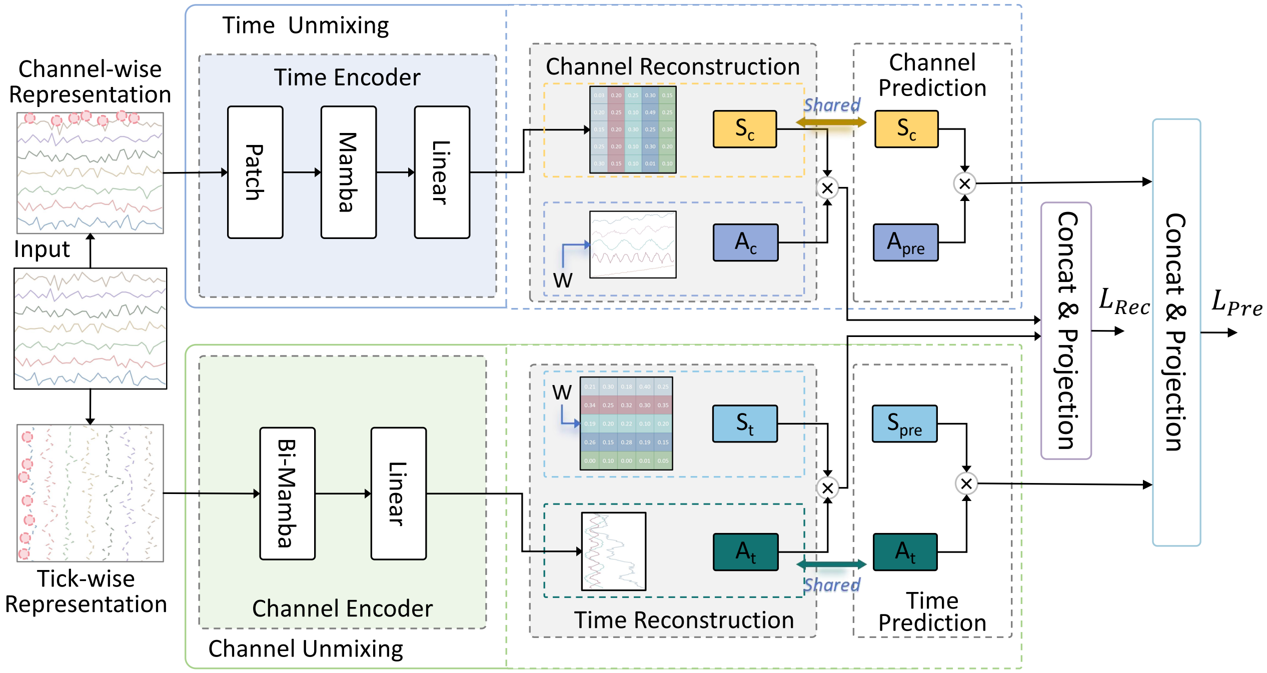


Fig. 2: The framework of MTS-UNMixers comprises two main components: temporal unmixing and channel unmixing.

basis signal for both historical and future sequences. To ensure consistent patterns between different channels in historical and future sequences, the basis signals are multiplied by reconstruction and prediction coefficient matrices, respectively, to produce the final reconstructed and predicted results. Finally, the outputs from the temporal and channel paths are fused to produce the final output.

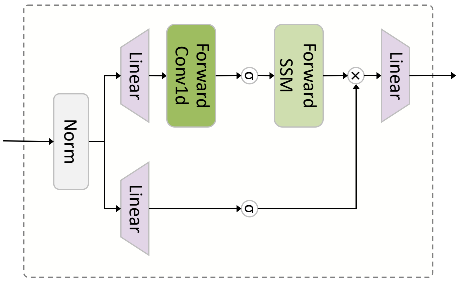


Fig. 3: Structure of the Mamba Block.

B. Unmixing Encoder

a) Temporal (Channel-wise Representation) Unmixing Encoder: The Time Encoder first processes the historical sequence through a blocking operation, resulting in $X_p \in \mathbb{R}^{N \times T_p \times P}$, where the sequence is divided into small segments (tokens) for feature extraction. Subsequently, these tokens are processed by the Mamba block, which leverages a SSM to capture complex temporal dependencies. As shown in Fig. 3. The Mamba block is designed to extract long-range features across time steps, effectively handling non-linear dynamics within the sequence. Additionally, Mamba enhances computational efficiency through parallelized processing, enabling efficient modeling while preserving crucial information. The core equations are as follows:

$$h'(t) = Ah(t) + Bx(t), \quad y(t) = Ch(t), \tag{12}$$

where A describes the state transition, B integrates the input signal, and C maps the state to the output. To handle discrete data, the Zero-Order Hold (ZOH) method is applied for discretization:

$$h_t = \tilde{A}h_{t-1} + \tilde{B}x_t, \quad y_t = Ch_t. \tag{13}$$

Mamba introduces the Selective State Space Model (S6), which parameterizes the matrices A, B, and C dynamically, enabling the model to adapt to complex time series tasks. The final Mamba model is formalized as:

$$X_m = SSM(Conv(Linear(X_p))) * \sigma(Linear(X_p)), \tag{14}$$

where σ is a nonlinear activation function, and X_m represents the processed output. The encoded output is then passed through a linear layer followed by a softmax operation to generate the weight matrix S_c:

$$S_c = Softmax(Linear(X_m)). \tag{15}$$

b) Channel (Tick-wise Representation) Unmixing Encoder: The channel encoder processes the historical sequence across channels at each time step using a bidirectional Mamba block, as shown in Fig. 4. The bidirectional Mamba block leverages a bidirectional state space model (SSM) to capture complex dependencies between channels, allowing the model to obtain complete contextual information at each time step. Since there is typically no causal relationship between channels, but rather bidirectional interactions, the bidirectional Mamba is better equipped to comprehensively capture the complex interdependencies among variables. Compared to linear models, the Mamba block enhances precision in capturing complex inter-channel interactions through nonlinear feature extraction. The main operations are:

$$\begin{aligned} X_f &= SSM(Conv(X_{\text{forward}})) * \sigma(X), \\ X_b &= SSM(Conv(X_{\text{backward}})) * \sigma(X), \\ X_e &= Linear(X_f + X_b), \end{aligned} \tag{16}$$

where X represents the linearly transformed historical sequence, and X_{forward} and X_{backward} denote the forward and reverse inputs of X, respectively. X_f and X_b are the forward and backward temporal features processed through the bidirectional Mamba block, while X_e is the combined encoding output. The function σ is a nonlinear activation function. Here, we use $ReLU$ to introduce nonlinearity and enhance the expressiveness of the model. Specifically, X represents the linearly transformed historical sequence. This encoded result, X_e, is used to generate the temporal basis matrix A_t, which captures the primary temporal patterns of the sequence.

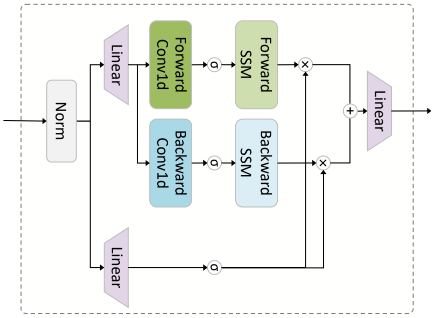


Fig. 4: Structure of the Bi-Mamba Block.

C. Unmixing Decoder

The unmixing decoder reconstructs the historical sequence and predicts the future sequence using the extracted features, followed by a projection layer to generate the final model output.

a) Channel Reconstruction and Prediction: In this phase, the Time Encoder generates the component coefficient matrix S_c, where $S_c = \text{softmax}(S_c)$, representing the mixing ratio of variables across feature patterns. Based on Equations 7, 8, 9, the historical and future sequences are treated as a unified sequence, resulting in a shared S_c.

For reconstruction and prediction, the shared S_c is used as follows:

$$X'_c = A_c S_c, \quad \hat{X}_c = A_p S_c, \tag{17}$$

where A_c and A_p are the learnable basis signal matrices for reconstruction and prediction, respectively. This ensures consistency between historical and future features while enabling accurate representation and forecasting within the channel-wise representation.

b) Time Reconstruction and Prediction: Based on Equations 7, 8 ,9, the shared basis signal matrix A_t is generated through the Channel Encoder to represent the core features of each channel in the temporal dimension. During the unmixing process, the historical and future sequences are treated as a unified sequence, resulting in a consistent basis signal matrix A_t across both historical and future sequences, thereby forming a shared matrix with a physically meaningful feature pattern representation.

For reconstruction and prediction, the shared A_t is used as follows:

$$X'_t = A_t S_t, \quad \hat{X}_t = A_t S_p, \tag{18}$$

where S_t and S_p are the learnable coefficient matrices for reconstruction and prediction, respectively. Both S_t and S_p undergo a softmax operation to ensure that their elements represent the relative contributions across different time steps. This approach enables reconstruction and prediction in the temporal dimension while ensuring consistency of temporal features between historical and future sequences, thereby better capturing the primary temporal patterns of the series.

c) Projection Layer: In obtaining the prediction results from the temporal and channel axes, the two are concatenated, and a linear layer is applied to generate the final output:

$$\hat{X} = Linear(Concat(\hat{X}_c, \hat{X}_t)). \tag{19}$$

Through concatenation and linear transformation, the model integrates features from both dimensions to enhance prediction accuracy. Similarly, the reconstructed data undergoes fusion to obtain the reconstructed historical sequence:

$$X^{'} = Linear(Concat(X^{'}_c, X^{'}_t)). \tag{20}$$

The linear layer processes the fused data to generate the final reconstructed sequence.

D. Loss Function

Based on the optimization formulas 11 presented earlier, we employ the L1 loss function to solve the sequence reconstruction and prediction tasks by measuring the absolute differences between predicted and actual values. Compared to L2 loss, L1 loss is more robust to outliers, making it better suited for

TABLE I: Multivariate time series forecasting results. The input length T = 96 . The best results are highlighted in bold, and the second-best results are underlined.

Model		MTS-UNMixers		TimeXer		TimeMixer		PatchTST		TimesNet		FITS		DLinear		FEDformer		TiDE		Stationary		Autoformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.368**	**0.388**	0.377	0.397	0.375	0.400	0.393	0.408	0.384	0.402	0.701	0.558	0.386	0.432	0.376	0.419	0.427	0.450	0.513	0.491	0.449	0.459
	192	0.427	**0.419**	0.425	0.426	0.429	0.421	0.445	0.434	0.436	0.429	0.718	0.570	0.437	0.459	**0.420**	0.448	0.472	0.486	0.534	0.504	0.500	0.482
	336	**0.443**	**0.433**	0.457	0.441	0.484	0.458	0.484	0.451	0.491	0.469	0.723	0.581	0.481	0.516	0.459	0.465	0.527	0.527	0.588	0.535	0.521	0.496
	720	**0.454**	**0.429**	0.464	0.463	0.498	0.482	0.480	0.471	0.521	0.500	0.712	0.595	0.519	0.452	0.506	0.507	0.644	0.605	0.643	0.616	0.514	0.512
	Avg.	**0.423**	**0.417**	0.431	0.432	0.447	0.440	0.451	0.441	0.458	0.450	0.714	0.576	0.456	0.465	0.440	0.460	0.518	0.517	0.570	0.537	0.496	0.487
ETTh2	96	**0.273**	**0.327**	0.289	0.340	0.289	0.341	0.294	0.343	0.340	0.374	0.353	0.387	0.333	0.476	0.346	0.388	0.304	0.359	0.476	0.458	0.346	0.388
	192	**0.344**	**0.372**	0.370	0.391	0.372	0.392	0.377	0.393	0.402	0.414	0.428	0.429	0.477	0.541	0.429	0.439	0.394	0.422	0.512	0.493	0.456	0.452
	336	**0.356**	**0.390**	0.422	0.434	0.386	0.414	0.381	0.409	0.452	0.452	0.454	0.455	0.594	0.657	0.496	0.487	0.385	0.421	0.552	0.551	0.482	0.486
	720	**0.405**	**0.426**	0.429	0.445	0.412	0.434	0.412	0.433	0.462	0.468	0.451	0.460	0.831	0.515	0.463	0.474	0.463	0.475	0.562	0.560	0.515	0.511
	AVG.	**0.345**	**0.379**	0.378	0.403	0.365	0.395	0.366	0.395	0.414	0.427	0.422	0.433	0.559	0.547	0.434	0.447	0.387	0.419	0.526	0.516	0.450	0.459
ETTm1	96	0.317	**0.350**	**0.309**	0.352	0.320	0.357	0.321	0.360	0.338	0.375	0.693	0.548	0.345	0.372	0.378	0.418	0.356	0.381	0.386	0.398	0.505	0.475
	192	0.360	**0.378**	**0.355**	0.378	0.361	0.381	0.362	0.384	0.374	0.387	0.710	0.557	0.380	0.389	0.426	0.441	0.391	0.399	0.459	0.444	0.553	0.496
	336	**0.388**	**0.400**	0.387	0.399	0.390	0.404	0.392	0.402	0.410	0.411	0.722	0.566	0.413	0.413	0.445	0.459	0.424	0.423	0.495	0.464	0.621	0.537
	720	**0.454**	**0.429**	0.448	0.435	0.454	0.441	0.461	0.439	0.478	0.450	0.746	0.581	0.474	0.453	0.543	0.490	0.480	0.456	0.585	0.516	0.671	0.561
	AVG.	**0.380**	**0.389**	**0.375**	0.391	0.381	0.396	0.384	0.396	0.400	0.406	0.718	0.563	0.403	0.407	0.448	0.452	0.413	0.415	0.481	0.456	0.588	0.517
ETTm2	96	**0.170**	**0.250**	0.171	0.255	0.175	0.258	0.178	0.260	0.187	0.267	0.229	0.307	0.193	0.292	0.203	0.287	0.182	0.264	0.192	0.274	0.255	0.339
	192	**0.236**	**0.293**	0.238	0.300	0.237	0.299	0.249	0.307	0.249	0.309	0.284	0.337	0.284	0.362	0.269	0.328	0.256	0.323	0.280	0.339	0.281	0.340
	336	**0.297**	**0.333**	0.301	0.340	0.298	0.340	0.313	0.346	0.321	0.351	0.338	0.369	0.369	0.427	0.325	0.366	0.313	0.354	0.334	0.361	0.339	0.372
	720	0.392	**0.390**	0.401	0.397	**0.391**	0.392	0.400	0.398	0.408	0.403	0.433	0.419	0.554	0.522	0.421	0.415	0.419	0.410	0.417	0.413	0.433	0.432
	AVG.	**0.274**	**0.316**	0.278	0.323	0.275	0.322	0.285	0.328	0.291	0.333	0.321	0.358	0.350	0.401	0.305	0.349	0.293	0.338	0.306	0.347	0.327	0.371
Weather	96	**0.158**	**0.195**	0.168	0.209	0.163	0.209	0.178	0.219	0.172	0.220	0.215	0.271	0.196	0.255	0.217	0.296	0.202	0.261	0.173	0.223	0.266	0.223
	192	**0.206**	**0.242**	0.220	0.254	0.208	0.250	0.224	0.259	0.219	0.261	0.264	0.305	0.237	0.296	0.276	0.336	0.242	0.298	0.245	0.285	0.307	0.285
	336	0.254	**0.286**	0.276	0.294	**0.251**	0.287	0.278	0.298	0.280	0.306	0.312	0.336	0.283	0.335	0.339	0.380	0.287	0.335	0.321	0.338	0.359	0.338
	720	**0.339**	**0.336**	0.353	0.347	**0.339**	0.341	0.353	0.346	0.365	0.359	0.381	0.377	0.345	0.381	0.403	0.428	0.351	0.386	0.414	0.410	0.419	0.410
	AVG.	**0.239**	**0.265**	0.254	0.276	0.240	0.272	0.258	0.281	0.259	0.287	0.293	0.322	0.265	0.317	0.309	0.360	0.271	0.320	0.288	0.314	0.338	0.314
Traffic	96	0.437	**0.274**	**0.416**	0.280	0.462	0.285	0.500	0.315	0.593	0.321	1.410	0.805	0.650	0.396	0.562	0.349	0.568	0.352	0.612	0.338	0.613	0.338
	192	0.454	**0.288**	**0.435**	**0.288**	0.473	0.296	0.498	0.299	0.617	0.336	1.413	0.806	0.598	0.370	0.562	0.346	0.612	0.371	0.613	0.340	0.616	0.340
	336	0.471	**0.295**	**0.451**	0.296	0.498	0.296	0.504	0.319	0.629	0.336	1.429	0.809	0.605	0.373	0.570	0.323	0.605	0.374	0.618	0.328	0.622	0.328
	720	0.504	**0.312**	**0.484**	0.314	0.506	0.313	0.542	0.335	0.640	0.350	1.502	0.820	0.645	0.394	0.596	0.368	0.647	0.410	0.653	0.355	0.660	0.355
	AVG.	0.466	**0.292**	**0.447**	0.295	0.485	0.298	0.511	0.317	0.620	0.336	1.439	0.810	0.625	0.383	0.573	0.347	0.608	0.377	0.624	0.340	0.628	0.340
Electricity	96	**0.147**	**0.246**	0.151	0.247	0.153	0.247	0.180	0.259	0.168	0.272	0.846	0.762	0.197	0.282	0.183	0.297	0.194	0.277	0.169	0.273	0.201	0.317
	192	**0.165**	0.264	**0.165**	**0.261**	0.166	0.256	0.188	0.268	0.184	0.289	0.849	0.761	0.196	0.285	0.195	0.308	0.193	0.280	0.182	0.286	0.222	0.334
	336	**0.181**	**0.277**	0.183	0.280	0.185	0.277	0.203	0.288	0.198	0.300	0.861	0.765	0.209	0.301	0.212	0.313	0.206	0.296	0.200	0.304	0.231	0.338
	720	**0.210**	**0.304**	0.220	0.309	0.225	0.310	0.239	0.321	0.220	0.320	0.892	0.775	0.245	0.333	0.231	0.343	0.242	0.328	0.222	0.321	0.254	0.361
	AVG.	**0.176**	**0.273**	0.180	0.274	0.182	0.273	0.203	0.284	0.193	0.295	0.862	0.766	0.212	0.300	0.205	0.315	0.209	0.295	0.193	0.296	0.227	0.338
1st Count		23	33	9	3	3	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

capturing the main patterns and trends in time series data. The overall L1 loss function, incorporating both reconstruction and prediction objectives, is defined as follows:

$$\mathcal{L} = \lambda_1 \cdot \|X^{'} - X_{\text{history}}\|_1 + \lambda_2 \cdot \|\hat{X} - X_{\text{future}}\|_1, \tag{21}$$

where λ_1 and λ_2 are weighting factors that balance the importance of the reconstruction and prediction tasks.

V. Experiment Resluts

A. Datasets Descriptions

We conducted experiments on seven real-world datasets to evaluate the performance of the proposed MTS-UNMixers model. These datasets include: (1) ETT dataset (ETTh1, ETTh2, ETTm1, ETTm2), which contains data from two power transformers (from two distinct sites) recorded from July 2016 to July 2018, including variables such as load and oil temperature, totaling seven features. Specifically, ETTm1 and ETTm2 are datasets sampled at a minute-level frequency, while ETTh1 and ETTh2 are sampled at an hourly frequency. (2) Weather dataset, collected by the Max Planck Institute for Biogeochemistry in 2020, which contains 21 meteorological factors, including temperature and humidity, sampled every 10 minutes. This dataset is suitable for evaluating the ability of a model to process complex, multivariate meteorological data. (3) Traffic dataset, which describes the road occupancy rates recorded by 862 sensors on highways in the San Francisco Bay Area from January 2015 to December 2016, with data sampled at an hourly frequency. (4) Electricity dataset, which includes the hourly electricity consumption records of 321 customers from 2012 to 2014. It captures the power usage patterns of different customers over time, serving as a common benchmark dataset for evaluating time series models with high-dimensional multivariate inputs. We followed the same data processing protocol and prediction horizon settings as used in TimesNet [20], varying the forecasting length among $\{96, 192, 336, 720\}$ to assess model performance across different forecast windows.

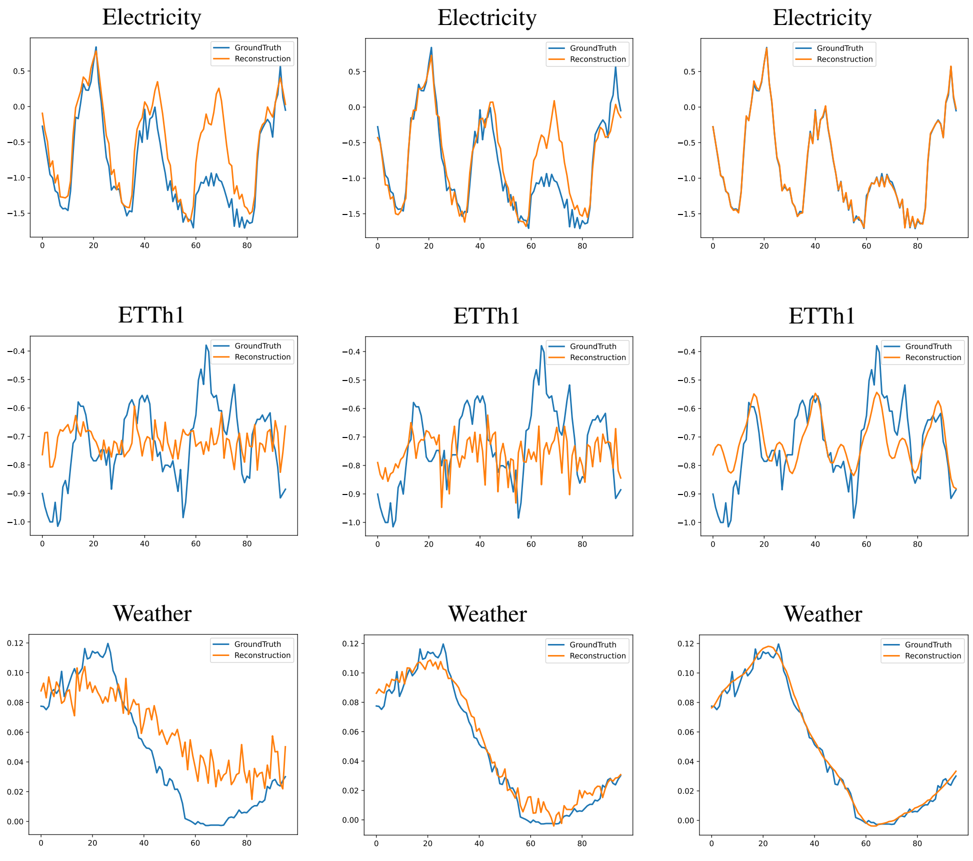
B. Forecasting Performance

a) Comparison Methods: We compared the proposed MTS-UNMixers with nine well-established and advanced models for time series forecasting, including Transformer-based approaches: PatchTST [13], FEDformer [14], Autoformer [15], Stationary Transformer [16], and TimeXer [29]; MLP-based models: DLinear [18], FITS [33], and TiDE [34]; as well as CNN-based models: TimesNet [20] and TimeMixer [10]. These models were selected to provide a comprehensive benchmark for evaluating the performance of MTS-UNMixers.

b) Main Results: Table I summarizes the performance comparison between MTS-UNMixers and the benchmark models across seven datasets. The results show that MTS-UNMixers consistently ranks within the top two in all scenarios, achieving or approaching state-of-the-art performance. Notably, MTS-UNMixers ranked first in 56 cases. Compared to the second-ranking model, TimeMixer, MTS-UNMixers achieved a 5.23% relative reduction in average MSE and a 5.49% reduction in MAE on the ETTh1 dataset, as well as a 5.21% reduction in average MSE on the ETTm2 dataset. These results demonstrate that MTS-UNMixers performs exceptionally well on most datasets, such as ETTh1, ETTm2, and Weather, effectively removing redundant information and reducing sequence complexity. Its strong feature separation capability enables outstanding performance in tasks involving complex temporal dependencies and multivariate interactions. However, on the Traffic dataset, MTS-UNMixers slightly underperforms compared to TimeXer. This may be due to the fact that the traffic dataset has a large number of channels and significant spatial features, while TimeXer excels with its cross-variable attention mechanism and effective use of exogenous variables, which allows it to better capture spatial dependencies and complex variable interactions.

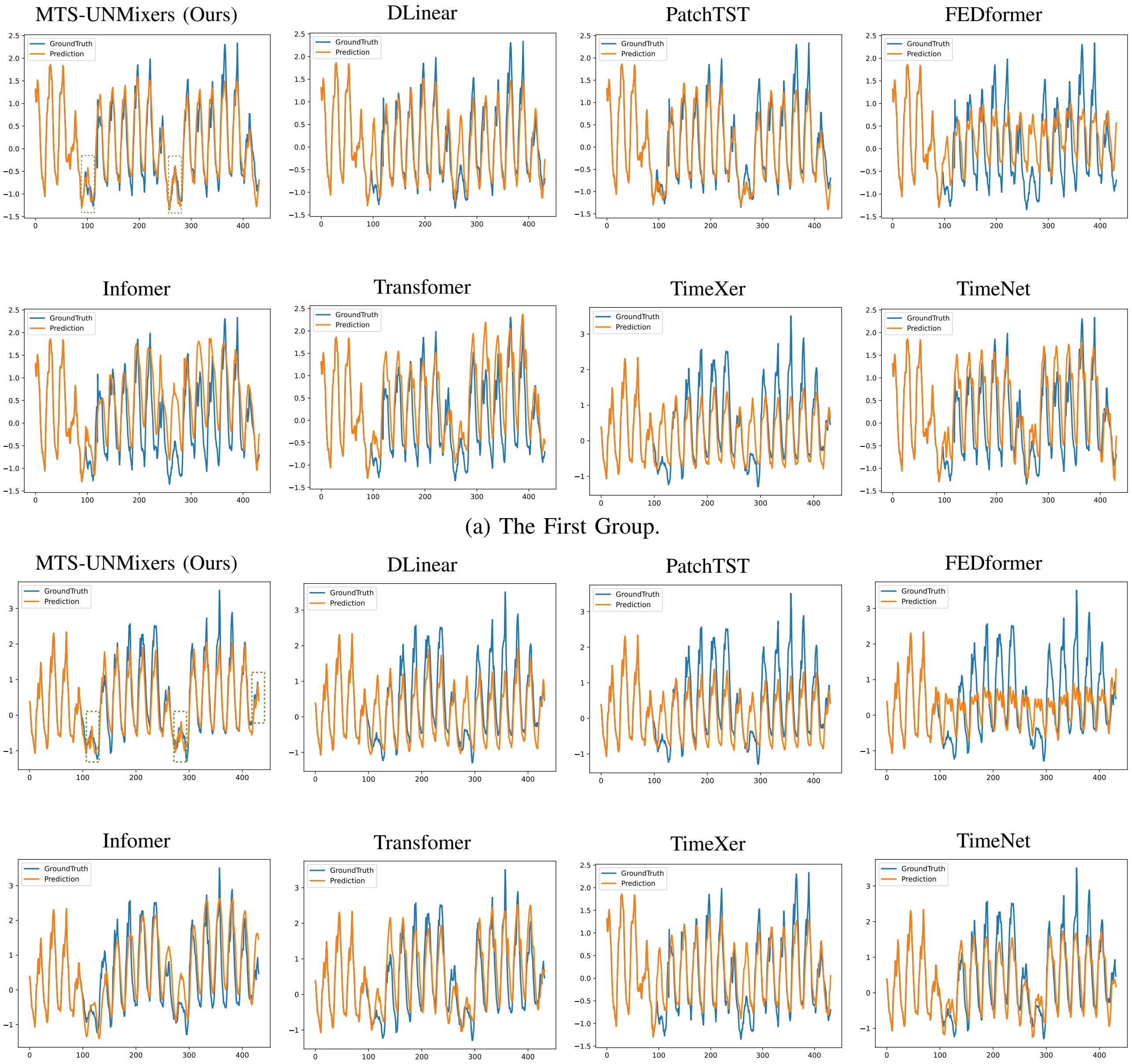
c) Reconstruction visualization: To better understand the performance of MTS-UNMixers, we designed reconstruction experiments with different module configurations and conducted prediction visualization on the electricity dataset. These experiments utilized a history sequence length of 96 for the ETTh1, Weather, and Electricity datasets, with the following configurations:

- Channel Unmixing Only: In this setup, we retained only the channel unmixing module to reconstruct the input data. As shown in Fig. 5 (a), channel unmixing effectively captures inter-channel relationships, allowing feature extraction along the channel dimension. However, without temporal unmixing, the model struggles with time dependencies, leading to visible discrepancies in trend continuity, especially in datasets with long-term dependencies.
- Temporal Unmixing Only: In this setup, only the temporal unmixing module was used. Fig. 5 (b) shows that temporal unmixing better captures time dependencies and dynamic trends, resulting in more accurate temporal reconstructions. However, it lacks inter-channel feature extraction, leading to weaker reconstruction of inter-variable interactions.



(a) Channel Unmixing (b) Temporal Unmixing (c) Dual Unmixing

Fig. 5: Reconstruction visualization of MTS-UNMixers under different configurations on ETTh1, Weather, and Electricity datasets using a history sequence length of 96. (a) Channel Unmixing Only, (b) Temporal Unmixing Only, (c) Dual Unmixing. Dual-path unmixing achieves the best reconstruction accuracy by effectively capturing both temporal dependencies and inter-channel relationships.



(a) The First Group.

(b) The Second Group.

Fig. 6: Visualization of MTS-UNMixers' prediction results on Electricity. The input length $T = 96$. The results of two experimental groups are provided.

- Dual Unmixing: In this complete configuration, both temporal and channel unmixing modules were applied. As illustrated in Fig. 5 (c), dual unmixing enables the model to deeply extract features across both dimensions, capturing both temporal dependencies and inter-channel details with high accuracy. This dual-path setup demonstrates superior reconstruction compared to single-path configurations.

Through a comparative analysis of these visualizations, we observe that dual-path unmixing demonstrates a clear advantage in information extraction and reconstruction. In contrast, the channel-only unmixing configuration performs well in capturing channel-specific features but lacks temporal dependency representation. Meanwhile, the temporal-only configuration fails to capture intricate inter-channel details. By integrating both temporal and channel information, dual-path unmixing achieves optimal accuracy and information integrity in reconstruction, further validating the effectiveness of dual-path unmixing in complex time-series forecasting tasks and providing strong support for the design of MTS-UNMixers.

d) Prejection visualization: To provide an intuitive understanding of the forecasting process, we present the prediction results from the electricity dataset in Fig. 6. The results of five models are recorded for a 96-input, 336-prediction setting.

From the figure, we can observe that our model performs relatively well during cyclical fluctuations. As shown in the Fig. 6, MTS-UNMixers can respond accurately to fluctuations in signal generation cycles and trends, with precise predictions, especially in the areas highlighted by the green boxes.

TABLE II: Performance comparison at various prediction lengths of MTS-UNMixers and its ablated versions on the ETTh1, ETTm1, and Weather datasets.

Model	Length	ETTh1		ETTm1		Weather	
		MSE	MAE	MSE	MAE	MSE	MAE
MTS-UNMixers	96	**0.368**	**0.388**	**0.317**	**0.350**	**0.158**	**0.195**
	192	**0.427**	**0.419**	**0.360**	**0.378**	**0.206**	**0.242**
	336	**0.443**	**0.433**	**0.388**	**0.400**	**0.254**	**0.286**
	720	**0.454**	**0.429**	**0.454**	**0.429**	**0.339**	**0.336**
w/o Channel Unmixing	96	0.375	0.393	0.321	0.355	0.159	0.196
	192	0.433	0.425	0.366	0.384	0.220	0.252
	336	0.454	0.442	0.396	0.406	0.264	0.295
	720	0.466	0.436	0.461	0.432	0.346	0.350
w/o Time Unmixing	96	0.477	0.458	0.383	0.364	0.204	0.270
	192	0.518	0.479	0.478	0.418	0.261	0.284
	336	0.537	0.498	0.476	0.441	0.379	0.372
	720	0.547	0.516	0.560	0.530	0.471	0.372
w/o Mamba	96	0.369	0.393	0.320	0.351	0.162	0.226
	192	0.439	0.430	0.375	0.389	0.218	0.260
	336	0.454	0.443	0.411	0.418	0.275	0.294
	720	0.461	0.448	0.465	0.449	0.352	0.356
w/o Bi-Mamba	96	0.374	0.391	0.319	0.353	0.163	0.200
	192	0.433	0.428	0.366	0.381	0.213	0.248
	336	0.449	0.438	0.396	0.407	0.260	0.287
	720	0.460	0.437	0.460	0.435	0.342	0.345

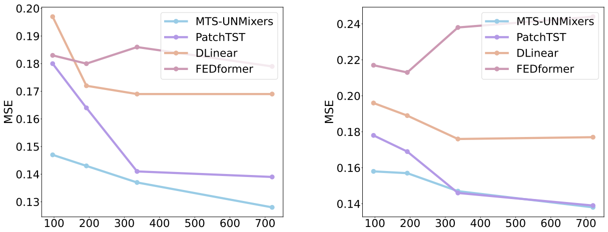
C. Ablation Study

In the ablation study, we systematically evaluated the importance of the main modules within MTS-UNMixers in time-series forecasting, as shown in Table II. We successively removed the Channel Unmixing, Time Unmixing, Mamba block and Bi-Mamba modules, while keeping other components unchanged. Experiments were conducted on the ETTh1, ETTm1, and Weather datasets, with MSE and MAE recorded at different prediction horizons (96, 192, 336, 720) to assess each module's contribution.

The results indicated that removing these modules led to significant performance declines, especially for long-term predictions. For example, on the ETTh1 dataset, the MSE for the 96-step forecast dropped by approximately 1.9% after removing the Channel Unmixing module. When the Time Unmixing module was removed, the MSE for the same horizon dropped significantly by about 29.6%, highlighting the critical role of Time Unmixing in reducing aliasing errors and capturing temporal dependencies. Overall, the full MTS-UNMixers model consistently achieved the best performance across datasets and prediction horizons, validating the synergistic role of each module in mitigating aliasing issues and enhancing prediction accuracy. Removing Mamba results in a significant performance drop for long prediction lengths, highlighting its importance in capturing nonlinear causal relationships and long-term trends in the temporal dimension. Removing Bi-Mamba leads to noticeable performance degradation, especially on the Weather dataset with complex inter-channel interactions, showing its role in modeling variable relationships. Comparatively, Mamba has a greater impact on long-sequence forecasting tasks, while Bi-Mamba is better at handling multivariable interactions. Together, they complement each other in unmixing across temporal and channel dimensions, improving the model's accuracy and robustness.

TABLE III: Prediction performance of MTS-UNMixers under different history lengths (96, 192, 336, 720). The best performance is highlighted in bold.

History Length		96		192		336		720	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.368	0.388	0.361	0.379	0.359	0.377	**0.355**	**0.371**
	192	0.427	0.419	0.424	0.413	0.419	0.409	**0.416**	**0.412**
	336	0.443	0.433	0.441	0.431	0.438	0.431	**0.432**	**0.428**
	720	0.454	0.429	0.448	0.427	0.444	0.414	**0.439**	**0.409**
ETTm1	96	0.317	0.350	0.308	0.340	0.291	0.339	**0.291**	**0.326**
	192	0.360	0.378	0.351	0.374	**0.343**	0.369	0.346	**0.355**
	336	0.388	0.400	0.376	0.396	**0.365**	0.386	0.371	**0.369**
	720	0.454	0.429	0.449	0.422	0.425	0.419	**0.414**	**0.417**
Weather	96	0.158	0.195	0.158	0.191	0.153	0.189	**0.149**	**0.183**
	192	0.206	0.242	0.200	0.240	0.194	0.239	**0.193**	**0.235**
	336	0.254	0.286	0.254	0.279	0.248	0.274	**0.243**	**0.271**
	720	0.339	0.336	0.335	0.329	0.328	0.328	**0.316**	**0.324**



Electricity H = 96

Weather H = 96

Fig. 7: The predictive performance of different models for the future sequence $H = 96$ under varying history lengths $\{96, 192, 336, 720\}$.

D. Varying Lookback Window

The results in Table III illustrate the impact of varying history lengths on the prediction performance of MTS-

TABLE IV: Model Efficiency Analysis.

(a) Running Time Efficiency Analysis (s/iter)

Model		Informer	Autoformer	FEDformer	PatchTST	TimesNet	TimeXer	MTS-UNMixers
Future Sequence Length	96	0.0078	0.0109	0.0859	0.0033	0.0428	0.0059	0.0054
	192	0.0095	0.0111	0.0860	0.0033	0.0438	0.0050	0.0054
	336	0.0098	0.0111	0.0864	0.0033	0.0528	0.0056	0.0054
	720	0.0102	0.0111	0.0867	0.0033	0.0754	0.0056	0.0054
History Sequence Length	96	0.0078	0.0109	0.0859	0.0036	0.0428	0.0059	0.0054
	192	0.0079	0.0103	0.0777	0.0036	0.0408	0.0061	0.0054
	336	0.0080	0.0104	0.0669	0.0036	0.0588	0.0061	0.0054
	720	0.0082	0.0104	0.0669	0.0036	0.0791	0.0049	0.0054

(b) Model Parameter Efficiency Analysis (M)

Model		Informer	Autoformer	FEDformer	PatchTST	TimesNet	TimeXer	MTS-UNMixers
Future Sequence Length	96	11.33	10.54	16.12	7.49	37.53	0.15	3.32
	192	11.33	10.54	16.12	7.49	37.53	0.16	3.38
	336	11.33	10.54	16.12	7.49	37.53	0.18	3.99
	720	11.33	10.54	16.12	7.49	37.53	0.22	4.34
History Sequence Length	96	11.33	10.54	16.12	7.49	37.53	0.15	3.32
	192	11.33	10.54	16.12	7.49	37.53	0.16	4.27
	336	11.33	10.54	16.12	7.49	37.53	0.18	5.76
	720	11.33	10.54	16.12	7.49	37.53	0.22	10.18

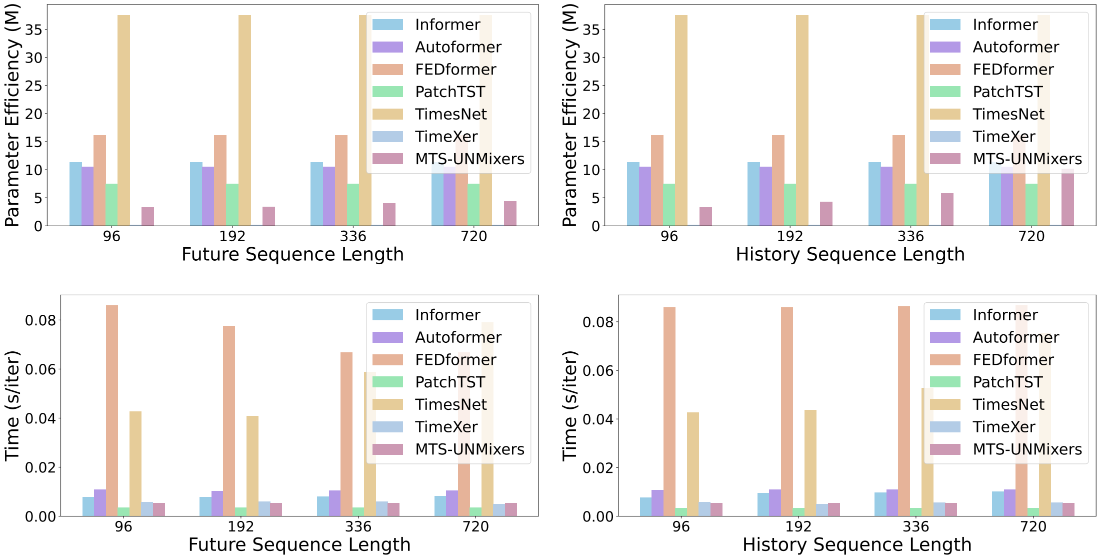


Fig. 8: These four graphs show the speed and parameter efficiency of the model at different historical and future sequence lengths.

UNMixers across different prediction steps (96, 192, 336, and 720) on the ETTh1, ETTm1, and Weather datasets. Overall, longer history lengths contribute to improved prediction accuracy, particularly for longer prediction steps, highlighting the advantages of utilizing extended historical information. Theoretically, prediction performance should increase as the input history sequence length grows.

For the ETTh1, ETTm1, and Weather datasets, extending the history length significantly enhances the prediction accuracy of MTS-UNMixers, with especially strong improvements for long prediction steps. For instance, increasing the history length from 96 to 720 across different datasets yields substantial improvements in both MSE and MAE, indicating that a longer history window helps the model better capture long-term dependencies and periodic features in time series data. In summary, the prediction performance of MTS-UNMixers consistently improves with extended history lengths, particularly in tasks involving longer prediction steps. These experimental results demonstrate that extending the historical sequence length effectively enhances the model's forecasting capability, underscoring the importance of capturing long-term dependencies in time series forecasting.

We also conducted a visual comparison with other models, as shown in Fig. 7. It can be seen that our results remain ahead, with a steady decrease in MSE and a gradual improvement in performance.

E. Model Efficiency Analysis

To summarize the model performance and efficiency, we calculate relative performance rankings to compare the baselines. The ranking is based on the common models used across all five tasks: Informer, Autoformer, FEDformer, PatchTST, TimesNet, TimeXer, and our proposed MTS-UNMixers, totaling six models. We compare the models using three efficiency metrics under different input and output sequence lengths: the number of parameters (Params) and runtime (s/iter). All experiments were conducted at a unified level to ensure fairness. The results are presented in Table IV and visualized in Figure 3. As shown in the Fig. 8, significant runtime differences exist between models with the same input and output sequence lengths. Although MTS-UNMixers does not outperform PatchTST, it still demonstrates competitive performance.

VI. Conclusion

In conclusion, this paper presents MTS-UNMixers, a novel approach to time-series forecasting that leverages unmixing and sharing mechanisms within a Mamba-based network. By using Mamba blocks to separate channel coefficients and temporal basis signals, MTS-UNMixers captures complex interchannel relationships and temporal dependencies with high precision. The integration of these unmixing mechanisms with a sharing phase enables efficient mapping of historical patterns to future predictions, effectively addressing challenges related to signal aliasing and redundancy. Extensive experiments on seven public datasets demonstrate that the model achieves superior performance compared to nine state-of-the-art baselines in various long-term forecasting tasks. The robust performance of MTS-UNMixers highlights its ability to effectively model intricate temporal dynamics and dependencies, offering significant advancements for multivariate time-series forecasting applications.

References

[1] Chen, Cathy WS, Gerlach, Richard, Lin, Edward MH, and Lee, WCW. ”Bayesian forecasting for financial risk management, pre and post the global financial crisis.” *Journal of Forecasting*, vol. 31, no. 8, pp. 661–687, 2012.

[2] Angryk, Rafal A, Martens, Petrus C, Aydin, Berkay, Kempton, Dustin, Mahajan, Sushant S, Basodi, Sunitha, Ahmadzadeh, Azim, Cai, Xumin, Filali Boubrahimi, Soukaina, Hamdi, Shah Muhammad, et al. ”Multivariate time series dataset for space weather data analytics.” *Scientific Data*, vol. 7, no. 1, pp. 227, 2020.

[3] Schultz, Martin G, Betancourt, Clara, Gong, Bing, Kleinert, Felix, Langguth, Michael, Leufen, Lukas Hubert, Mozaffari, Amirpasha, and Stadtler, Scarlet. ”Can deep learning beat numerical weather prediction?” *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, pp. 20200097, 2021.

[4] Khan, Zulfiqar Ahmad, Hussain, Tanveer, Ullah, Amin, Rho, Seungmin, Lee, Miyoung, and Baik, Sung Wook. ”Towards efficient electricity forecasting in residential and commercial buildings: A novel hybrid CNN with a LSTM-AE based framework.” *Sensors*, vol. 20, no. 5, pp. 1399, 2020.

[5] Zhu, Zhaoyang, Chen, Weiqi, Xia, Rui, Zhou, Tian, Niu, Peisong, Peng, Bingqing, Wang, Wenwei, Liu, Hengbo, Ma, Ziqing, Gu, Xinyue, et al. ”Energy forecasting with robust, flexible, and explainable machine learning algorithms.” *AI Magazine*, vol. 44, no. 4, pp. 377–393, 2023.

[6] Chen, Chao, Petty, Karl, Skabardonis, Alexander, Varaiya, Pravin, and Jia, Zhanfeng. ”Freeway performance measurement system: mining loop detector data.” *Transportation Research Record*, vol. 1748, no. 1, pp. 96–102, 2001.

[7] Cirstea, Razvan-Gabriel, Yang, Bin, Guo, Chenjuan, Kieu, Tung, and Pan, Shirui. ”Towards spatio-temporal aware traffic time series forecasting.” In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 2900–2913, IEEE, 2022.

[8] Ma, Yu, Yang, Bin, and Jensen, Christian S. ”Enabling time-dependent uncertain eco-weights for road networks.” In *Proceedings of Workshop on Managing and Mining Enriched Geo-Spatial Data*, pp. 1–6, 2014.

[9] P. Chen, Y. Zhang, Y. Cheng, Y. Shu, Y. Wang, Q. Wen, B. Yang, and C. Guo, ”Pathformer: Multi-scale Transformers with Adaptive Pathways for Time Series Forecasting,” *International Conference on Learning Representations (ICLR)*, 2024.

[10] Wang, Shiyu, Wu, Haixu, Shi, Xiaoming, Hu, Tengge, Luo, Huakun, Ma, Lintao, Zhang, James Y., and Zhou, Jun. TimeMixer: Decomposable Multiscale Mixing for Time Series Forecasting. In *International Conference on Learning Representations (ICLR)*, 2024.

[11] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, ”iTransformer: Inverted Transformers Are Effective for Time Series Forecasting,” *arXiv preprint arXiv:2310.06625*, 2023.

[12] Z. Li, Z. Rao, L. Pan, and Z. Xu, “MTS-Mixers: Multivariate Time Series Forecasting via Factorized Temporal and Channel Mixing,” arXiv preprint arXiv:2302.04501, 2023. [Online]. Available: https://arxiv.org/abs/2302.04501

[13] Nie, Yuqi, Nguyen, Nam H., Sinthong, Phanwadee, and Kalagnanam, Jayant. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *The Eleventh International Conference on Learning Representations*.

[14] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, “FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting,” in *Proceedings of the 39th International Conference on Machine Learning (ICML)*, Baltimore, Maryland, 2022.

[15] Wu, Haixu, Xu, Jiehui, Wang, Jianmin, and Long, Mingsheng. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

[16] Liu, Yong, Wu, Haixu, Wang, Jianmin, and Long, Mingsheng. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems*, 35:9881–9893, 2022.

[17] V. Ekambaram, A. Jati, N. Nguyen, P. Sinthong, and J. Kalagnanam, “TSMixer: Lightweight MLP-Mixer Model for Multivariate Time Series Forecasting,” in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 459-469.

[18] Zeng, Ailing, Chen, Muxi, Zhang, Lei, and Xu, Qiang. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, number 9, pages 11121–11128, 2023.

[19] Liu, M., Zeng, A., Chen, M., Xu, Z., Lai, Q., Ma, L., & Xu, Q. (2022). Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35, 5816–5828.

[20] Wu, Haixu, Hu, Tengge, Liu, Yong, Zhou, Hang, Wang, Jianmin, and Long, Mingsheng. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *The Eleventh International Conference on Learning Representations*.

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

[22] Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3), 1181–1191. Elsevier.

[23] Zhao, Z., Chen, W., Wu, X., Chen, P. C. Y., & Liu, J. (2017). LSTM network: A deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2), 68–75. Wiley Online Library.

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

[25] S. Ma, Y. Kang, P. Bai, and Y.-B. Zhao, “FMamba: Mamba based on Fast-attention for Multivariate Time-series Forecasting,” arXiv preprint arXiv:2407.14814, 2024. [Online]. Available: https://arxiv.org/abs/2407.14814

[26] B. N. Patro and V. S. Agneeswaran, “SiMBA: Simplified Mamba-Based Architecture for Vision and Multivariate Time series,” arXiv preprint arXiv:2403.15360, 2024. [Online]. Available: https://arxiv.org/abs/2403.15360

[27] Y. Tang, P. Dong, Z. Tang, X. Chu, and J. Liang, “VMRNN: Integrating Vision Mamba and LSTM for Efficient and Accurate Spatiotemporal Forecasting,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2024, pp. 5663-5673.

[28] M. A. Ahamed and Q. Cheng, ”TimeMachine: A Time Series is Worth 4 Mambas for Long-term Forecasting,” *arXiv preprint arXiv:2403.09898*, 2024.

[29] Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, Mingsheng Long, *Timexer: Empowering transformers for time series forecasting with exogenous variables*, arXiv preprint arXiv:2402.19072, 2024.

[30] A. Gu, K. Goel, and C. Ré, “Efficiently Modeling Long Sequences with Structured State Spaces,” in *The International Conference on Learning Representations (ICLR)*, 2022.

[31] S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski, “Deep state space models for time series forecasting,” Advances in Neural Information Processing Systems, vol. 31, 2018.

[32] Y. Lin, I. Koprinska, and M. Rana, “SSDNet: State Space Decomposition Neural Network for Time Series Forecasting,” in *2021 IEEE International Conference on Data Mining (ICDM)*, 2021, pp. 370-378, doi: 10.1109/ICDM51629.2021.00048.

[33] Xu, Zhijian, Zeng, Ailing, and Xu, Qiang. FITS: Modeling Time Series with 10k Parameters. In *The Twelfth International Conference on Learning Representations*.

[34] Das, Abhimanyu, Kong, Weihao, Leach, Andrew, Mathur, Shaan K., Sen, Rajat, and Yu, Rose. Long-term Forecasting with TiDE: Time-series Dense Encoder. *Transactions on Machine Learning Research*.