Is Mamba Effective for Time Series Forecasting?

Zihan **Wang**, Fanheng **Kong**, Shi **Feng***, Ming **Wang**, Xiaocui **Yang**, Han **Zhao**, Daling **Wang** and Yifei **Zhang**

Department of Computer Science and Engineering, Northeastern University, Shenyang, China

ARTICLE INFO

ABSTRACT

In the realm of time series forecasting (TSF), it is imperative for models to adeptly discern and distill hidden patterns within historical time series data to forecast future states. Transformer-based models exhibit formidable efficacy in TSF, primarily attributed to their advantage in apprehending these patterns. However, the quadratic complexity of the Transformer leads to low computational efficiency and high costs, which somewhat hinders the deployment of the TSF model in real-world scenarios. Recently, Mamba, a selective state space model, has gained traction due to its ability to process dependencies in sequences while maintaining near-linear complexity. For TSF tasks, these characteristics enable Mamba to comprehend hidden patterns as the Transformer and reduce computational overhead compared to the Transformer. Therefore, we propose a Mamba-based model named Simple-Mamba (S-Mamba) for TSF. Specifically, we tokenize the time points of each variate autonomously via a linear layer. A bidirectional Mamba layer is utilized to extract inter-variate correlations and a Feed-Forward Network is set to learn temporal dependencies. Finally, the generation of forecast outcomes through a linear mapping layer. Experiments on thirteen public datasets prove that S-Mamba maintains low computational overhead and achieves leading performance. Furthermore, we conduct extensive experiments to explore Mamba's potential in TSF tasks. Our code is available at https://github.com/wzhwzhwzh0921/S-D-Mamba.

1. Introduction

Time series forecasting (TSF) involves leveraging historical information from lookback sequence to forecast states in the future [15] as Fig. 1. These data often have built-in patterns including the temporal dependency (TD), e.g. morning and evening peak patterns in traffic forecast tasks, and the inter-variate correlations (VC), e.g. temperature and humidity correlation patterns in weather forecast tasks. Discerning and distilling these patterns from time series data can bring better forecasting [5]. Transformer [52] exhibits
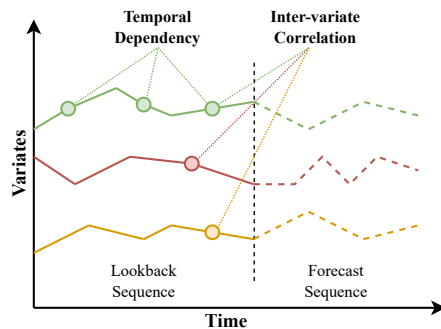

Figure 1: An example of Time Series Forecasting. Lines of different colors represent different variates, with solid lines indicating the historical changes of variates, and dotted lines indicating the future changes that need to be forecasted.

formidable efficacy in TSF, primarily attributed to their inherent advantages in apprehending TD and VC. Numerous

---

*Corresponding author

✉ 2310744@stu.neu.edu.cn (Z. Wang);
kongfanheng@stumail.neu.edu.cn (F. Kong); fengshi@cse.neu.edu.cn (S. Feng); sci.m.wang@gmail.com (M. Wang); yangxiaocui@stumail.neu.edu.cn (X. Yang); 2272065@stu.neu.edu.cn (H. Zhao); wangdaling@cse.neu.edu.cn (D. Wang); zhangyifei@cse.neu.edu.cn (Y. Zhang)

Transformer-based models with impressive capabilities have been introduced [59, 72], yet the Transformer architecture faces distinct challenges. Foremost is its quadratic computational complexity, which leads to a dramatic increase in calculation overhead as the number of variates or the lookback length increases. It hinders the deployment of Transformer-based models in real-world TSF scenarios that require the processing of large amounts of data simultaneously or have high real-time requirements. Many models attempt to reduce the computational complexity of the transformer in TSF by modifying its structure, such as focusing only on a portion of the sequence [28, 71, 31]. The loss of information in the above models may also lead to certain performance degradations. A more promising approach involves using linear models instead of transformer [32, 66], which possesses linear computational complexity. However, linear models relying solely on linear numerical calculations do not incorporate in-context information and are suboptimal compared to state-of-the-art Transformer models. And accurate forecasts can only be achieved when sufficient input information is available [66].

The State Space Models (SSM) [24, 51] demonstrate potential in simultaneously optimizing performance and computational complexity. SSMs employ convolutional calculation to capture sequence information and eliminate hidden states making it benefit from parallel computing and achieving near-linear complexity in processing speed. Rangapuram et al. [46] attempts to employ SSM for TSF, but the SSM architecture it used is unable to identify and filter content effectively, and the captured dependencies are solely based on distance, resulting in unsatisfactory performance. Mamba [22], introduces a selective mechanism into SSM, enabling it to discern valuable information like the attention mechanism. Numerous researchers develop models based on

Mamba [73, 61], demonstrating its considerable potential across both text and image domains. These Mamba-based models achieve a synergistic balance between enhanced performance and computational efficiency. Consequently, we are motivated to explore further the potential of Mamba in TSF.

We launch a Mamba-based model Simple-Mamba (S-Mamba) for TSF tasks. In the S-Mamba, the time points of each variate are tokenized by a linear layer. Subsequently, a Mamba VC (Inter-variate Correlation) Encoding layer encodes the VC by utilizing a bidirectional Mamba to leverage the global inter-variate mutual information. A Feed-Forward Network (FFN) TD (Temporal Dependency) Encoding Layer containing a simple FFN is followed to extract the TD. Ultimately, a mapping layer is utilized to output the forecast results. Experimental results on thirteen public datasets from traffic, electricity, weather, finance, and energy domains demonstrate that S-Mamba not only has low requirements in GPU memory usage and training time but also maintains superior performance compared to the state-of-the-art models in TSF. Concurrently, extensive experiments are conducted to assess the efficacy and potential of Mamba in TSF tasks. For instance, we evaluate whether Mamba demonstrates generalization capabilities comparable to those of the Transformer in handling TSF data. Our contributions are summarized as follows:

- We propose S-Mamba, a Mamba-based model for TSF, which delegates the extraction of inter-variate correlations and temporal dependencies to a bidirectional Mamba block and a Feed-Forward Network.

- We compare the performance of the S-Mamba against representative and state-of-the-art models in TSF. The results confirm that S-Mamba not only delivers superior forecast performance but also requires less computational resources.

- We conduct extensive experiments mainly focusing on exploring the characteristics of Mamba when facing TSF data to further discuss the potential of Mamba in TSF tasks.

2. Related Work

In conjunction with our work, two main areas of related work are investigated: (1) time series forecasting, and (2) applications of Mamba.

2.1. Time Series Forecasting

There have been two main architectures for TSF approaches, which are Transformer-based models [34, 43, 67] and linear models [5, 40, 48].

2.1.1. Transformer-based Models

Transformers are primarily designed for tasks that involve processing and generating sequences of tokens [52]. The excellent performance of Transformer-based models has also attracted numerous researchers to focus on time series forecasting tasks [3]. The transformer is utilized by Duong-Trung et al. [18] to solve the persistent challenge of long multi-horizon time series forecasting. Time Absolute Position Encoding (tAPE) and Efficient implementation of Relative Position Encoding (eRPE) are proposed in [20] to solve the position encoding problem encountered by Transformer in multivariate time series classification (MTSC). Wang et al. [53] replace the standard convolutional layer with an dilated convolutional layer and propose Graphformer to efficiently learn complex temporal patterns and dependencies between multiple variates. Some researchers have also considered the application of Transformer-based time series forecasting models in specific domains, such as piezometric level prediction [42], forecasting crude oil returns [1], predicting the power generation by solar panels [49], etc.

While they excel at capturing long-range dependencies in text, they may not be as effective in modeling sequential patterns. The use of content-based attention in Transformers is not effective in detecting essential temporal dependencies, especially for time-series data with weakening dependencies over time and strong seasonality patterns [56]. Particularly, the predictive capability and robustness of Transformer-based models may decrease rapidly when the input sequence is too long [55]. Moreover, the $O(N^2)$ time complexity makes Transformer-based models cost more computation and GPU memory resources. In addition, the previously mentioned issue of position encoding is also a challenge that deserves attention.

2.1.2. Linear Models

In addition to Transformer-based models, many researchers are keen to perform time series forecasting tasks using linear models [5]. Chen et al. [11] proposed TSMixer with all-MLP architecture to efficiently utilize cross-variate and auxiliary information to improve the performance of time series forecasting. LightTS [68] is dedicated to solving multivariate time series forecasting problems, and it can efficiently handle very long input series. Wang et al. [54] propose Time Series MLP to improve the efficiency and performance of multivariate time series forecasting. Yi et al. [64] explores MLP in the frequency domain for time series forecasting and proposes a novel architecture for FreTS that includes two phases: domain conversion and frequency learning.

Compared to Transformer-based models, MLP-based models are simpler in structure, less complex and more efficient. However, the MLP-based models also suffer from a number of shortcomings. In the case of high volatility and non-periodic, non-stationary patterns, MLP performance relying only on past observed temporal patterns is not satisfactory [11]. In addition, MLP is worse at capturing global dependencies compared to Transformers [64] and need longer input than Transformer-based models.

2.2. Applications of Mamba

As a new architecture, Mamba [22] swiftly attracted the attention of a large number of researchers in Natural

Language Processing (NLP), Computer Vision (CV), and other Artificial Intelligence communities.

2.2.1. Mamba in Natural Language Processing

Pióro et al. [45] and Anthony et al. [4] replaced the Transformer architecture in the Mixture of Experts (MoE) with the Mamba architecture, achieving a complete override of Mamba's and Transformer-MoE's performance. Mamba has demonstrated strong performance in clinical note generation [62]. Jiang et al. [27] replace Transformers with Mamba and demonstrate that Mamba can achieve match or outperform results on speech separation tasks with fewer parameters than Transformer. Empirical evidence is provided using simple NLP tasks (like translation) that Mamba can be an efficient alternative to Transformer for in-context learning tasks with long input sequences [21].

2.2.2. Mamba in Computer Vision

Mamba has been used to solve the long-range dependency problem in biomedical image segmentation tasks [39]. Cao et al. [8] propose a local-enhanced vision Mamba block named LEVM to improve local information perception, achieving state-of-the-art results on multispectral pansharpening and multispectral and hyperspectral image fusion tasks. Fusion-Mamba block [17] is designed to map features from images with different types (such as RGB and IR) into a hidden state space for interaction and enhance the representation consistency of features. Liu et al. [38] utilize the proposed HSIDMamba and Bidirectional Continuous Scanning Mechanism to improve the capture of long-range and local spatial-spectral information and improve denoise performance. In addition, Mamba has also been used in small target detection [12], medical image reconstruction[26] and classification [65], hyperspectral image classification [63], etc.

2.2.3. Mamba in Others

In addition to the two single modalities described, the application of Mamba to multimodal tasks has received a lot of attention. VideoMamba [30] achieves efficient long-term modeling using Mamba's linear complexity operator, showing advantages on long video understanding tasks. Zhao et al. [70] extend Mamba to a multi-modal large language model to improve the efficiency of inference, achieving comparable performance to LLaVA [35] with only about 43% of the number of parameters.

Furthermore, Mamba's sequence modeling capabilities have also received attention from researchers. Schiff et al. [47] extend long-range Mamba to a BiMamba component that supports bi-directionality, and to a MambaDNA block as the basis of long-range DNA language models. Mamba has also been shown to be effective on the tasks of predicting sequences of sensor data [6] and stock prediction [50]. Sequence Reordering Mamba [60] are proposed to exploit the inherent valuable information embedded within the long sequences. Ahamed and Cheng [2] propose Mamba-based TimeMachine to capture long-term dependencies in multivariate time series data.

As can be seen from the application of Mamba in these areas, Mamba can effectively reduce the parameter size and improve the efficiency of model inference while achieving similar or outperforming performance. It captures global dependencies better in a lightweight structure and has a better sense of position relationships. In addition, the Mamba architecture is more robust. Furthermore, the performance of Mamba in sequence modelling tasks further inspired us to explore whether Mamba can effectively mitigate the issues faced by Transformer-based models and linear models on TSF tasks.

3. Preliminaries

3.1. Problem Statement

In time series forecasting tasks, the model receives input as a history sequence $U_{in} = [u_1, u_2, \ldots, u_L] \in \mathbb{R}^{L \times V}$ and $u_n = [p_1, p_2, \ldots, p_V]$. and then uses the information to predict a future sequence $U_{out} = [u_{L+1}, u_{L+2}, \ldots, u_{L+T}] \in \mathbb{R}^{T \times V}$. The preceding L and T are referred to as the review window and prediction horizon respectively, representing the lengths of the past and future time windows, while p is a variate and V represents the total number of variates.

3.2. State Space Models

State Space Models can represent any cyclical process with latent states. By using first-order differential equations to represent the evolution of the system's internal state and another set to describe the relationship between latent states and output sequences, input sequences $x(t) \in \mathbb{R}^D$ can be mapped to output sequences $y(t) \in \mathbb{R}^N$ through latent states $h(t) \in \mathbb{R}^N$ in (1):

$$\begin{aligned} h(t)' &= Ah(t) + Bx(t), \\ y(t) &= Ch(t), \end{aligned} \tag{1}$$

where $A \in \mathbb{R}^{N \times N}$ and $B, C \in \mathbb{R}^{N \times D}$ are learnable matrices. Then, the continuous sequence is discretized by a step size Δ, and the discretized SSM model is represented as (2).

$$\begin{aligned} h_t &= \overline{A}h_{t-1} + \overline{B}x_t, \\ y_t &= Ch_t, \end{aligned} \tag{2}$$

where $\overline{A} = \exp(\Delta A)$ and $\overline{B} = (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B$. Since transitioning from continuous form (Δ, A, B, C) to discrete form $(\overline{A}, \overline{B}, C)$, the model can be efficiently calculated using a linear recursive approach [25]. The structured state space model (S4) [24], originating from the vanilla SSM, utilizes HiPPO [23] for initialization to add structure to the state matrix A, thereby improving long-range dependency modeling.

3.3. Mamba Block

Mamba [22] introduces a data-dependent selection mechanism into the S4 and incorporates hardware-aware parallel algorithms in its looping mode. The mechanism enables Mamba to capture contextual information in long sequences

| **Algorithm 1:** The process of Mamba Block |
|---|
| **Input**: $X : (B, V, D)$ |
| **Output**: $Y : (B, V, D)$ |
| 1: $x, z : (B, V, ED) \leftarrow$ Linear(U) {Linear projection} |
| 2: $x' : (B, V, ED) \leftarrow$ SiLU(Conv1D(x)) |
| 3: $A : (D, N) \leftarrow Parameter$ {Structured state matrix} |
| 4: $B, C : (B, V, N) \leftarrow$ Linear(x'), Linear(x') |
| 5: $\Delta : (B, V, D) \leftarrow$ Softplus($Parameter +$ Broadcast(Linear(x'))) |
| 6: $\overline{A}, \overline{B} : (B.V.D.N) \leftarrow discretize(\Delta, A, B)$ {Input-dependent parameters and discretization} |
| 7: $y : (B, V, ED) \leftarrow$ SelectiveSSM($\overline{A}, \overline{B}, C$)($x'$) |
| 8: $y' : (B, V, ED) \leftarrow y \otimes$ SiLU(z) |
| 9: $Y : (B, V, D) \leftarrow$ Linear(y') {Linear Projection} |
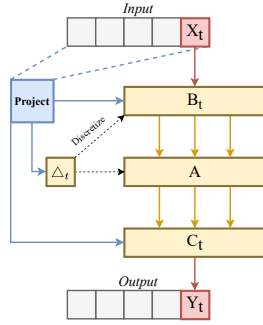
Figure 2: The structure of selective SSM (Mamba).

while maintaining computational efficiency. As an approximately linear perplexity series model, Mamba demonstrates potential in long sequence tasks, compared to transformers, in both efficiency enhancement and performance improvement. The details are presented in the algorithm related to the mamba layer in Alg.1 and the description in Fig. 2, where the former illustrates the complete data processing procedure, while the latter depicts the formation process of the output at sequence position t. The Mamba layer takes a sequence $X \in \mathbb{R}^{B \times V \times D}$ as input, where B denotes the batch size, V denotes the number of variates, and D denotes hidden dimension.

The block first expands the hidden dimension to ED through linear projection, obtaining x and z. Then, it processes the projection obtained earlier using convolutional functions and a SiLU [19] activation function to get x'. Based on the discretized SSM selected by the input parameters, denoted as the core of the Mamba Block, together with x', it generates the state representation y. Finally, y is combined with a residual connection from z after activation, and the final output y_t at time step t is obtained through a linear transformation. In summary, the Mamba Block effectively handles sequential information by leveraging selective state space models and input-dependent adaptations. The parameters involved in the Mamba Block include an SSM state expansion factor N, a size of convolutional kernel k, and a block expansion factor E for input-output linear projection. The larger the values of N and E, the higher the

Algorithm 2: The Forecasting Procedure of S-Mamba
Input: $Batch(U_{in}) = [u_1, u_2, \dots u_L] : (B, L, V)$
Output: $Batch(U_{out}) = [u_{L+1}, u_{L+2}, \dots u_{L+T}] : (B, T, V)$
1: **Linear Tokenization Layer:**
2: $Batch(U_{in}^{\top}) : (B, V, L) \leftarrow$ Transpose($Batch(U_{in})$)
3: $U^{tok} : (B, V, D) \leftarrow$ LinearTokenize($Batch(U_{in}^{\top})$) {Tokenization}
4: **for** l in $Mamba\ Layers$ **do**
5:    **Mamba VC Encoding Layer:**
6:    $\overrightarrow{Y} : (B, V, D) \leftarrow \overrightarrow{\text{Mamba Block}}(U)$
7:    $\overleftarrow{Y} : (B, V, D) \leftarrow \overleftarrow{\text{Mamba Block}}(U)$
8:    $Y : (B, V, D) \leftarrow \overrightarrow{Y} + \overleftarrow{Y}$ {Fusion Bidirectional Information}
9:    $U' : (B, V, D) \leftarrow Y + U$ {Residual Connection}
10:   **FFN TD Encoding Layer:**
11:   $U' : (B, V, D) \leftarrow$ LayerNorm(U')
12:   $U' : (B, V, D) \leftarrow$ Feed $-$ Forward(U')
13:   $U' : (B, V, D) \leftarrow$ LayerNorm(U')
14: **end for**
15: **Projection:**
16: $U' : (B, V, T) \leftarrow$ Projection(U')
17: $Batch(U_{out}) : (B, T, V) \leftarrow$ Transpose(U')

computational cost. The final output of the Mamba block is $Y \in \mathbb{R}^{B \times V \times D}$.

4. Methodology

In this section, we provide a detailed introduction of S-Mamba. Fig. 3 illustrates the overall structure of S-Mamba, which is primarily composed of four layers. The first layer, the Linear Tokenization Layer, tokenizes the time series with a linear layer. The second layer, the Mamba inter-variate correlation (VC) Encoding layer, employs a bidirectional Mamba block to capture mutual information among variates. The third layer, the FFN Temporal Dependencies (TD) Encoding Layer, further learns the temporal sequence information and finally generates future series representations by a Feed-Forward Network. Then the final layer, the Projection Layer, is only responsible for mapping the processed information of the above layers as the model's forecast. Alg. 2 demonstrates the operation process of S-Mamba.

4.1. Linear Tokenization Layer

The input for the Linear Tokenization Layer is U_{in}. Similar to iTransformer [37], we commence by tokenizing the time series, a method analogous to the tokenization of sequential text in natural language processing, to standardize the temporal series format. This pivotal task is executed by a single linear layer in Eq. (3).

$$U = \text{Linear}(Batch(U_{in})), \tag{3}$$

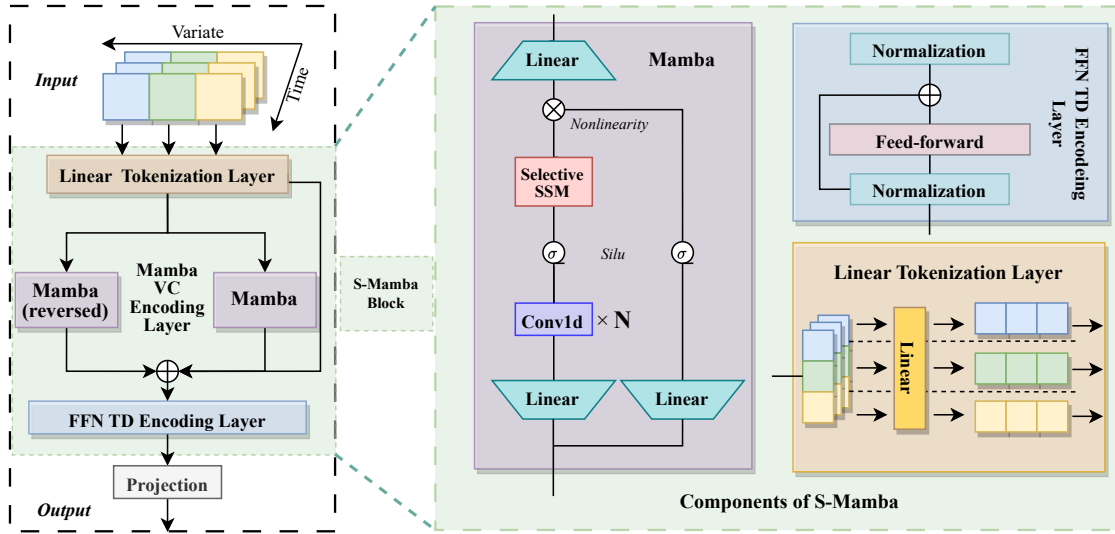where U is the output of this layer.

Figure 3: Overall framework of S-Mamba, the left side of the figure presents the overall architecture of our model. The right side of the figure details the components of the S-Mamba Block.

4.2. Mamba VC Encoding Layer

Within this layer, the primary objective is to extract the VC by linking variates that exhibit analogous trends aiming to learn the mutual information therein. The Transformer architecture confers the capacity for global attention [52], enabling the computation of the impact of all other variates upon a given variate, which facilitates the learning of precise information. However, the computational load of global attention escalates exponentially with an increase in the number of variates, potentially rendering it impractical. This limitation could restrict the application of Transformer-based algorithms in real-world scenarios. In contrast, Mamba's selective mechanism can discern the significance of different variates akin to an attention mechanism, and it exhibits a computational overhead that escalates in a near-linear fashion with an increasing count of variates. Yet, the unilateral nature of Mamba precludes it from attending to global variates in the manner of the Transformer; its selection mechanism is unidirectional, capable only of incorporating antecedent variates. To surmount this limitation, we employ two Mamba blocks to be combined as a bidirectional Mamba layer as Eq. (4), which facilitates the acquisition of correlations among all variates.

$$
\begin{aligned}
\overrightarrow{Y} &= \overrightarrow{\text{Mamba Block}}(U), \\
\overleftarrow{Y} &= \overleftarrow{\text{Mamba Block}}(U).
\end{aligned}
\tag{4}
$$

The VC encoded by the bidirectional Mamba is aggregated $Y = \overrightarrow{Y} + \overleftarrow{Y}$ and connected with another residual network to form the output of this layer $U' = Y + U$.

4.3. FFN TD Encoding Layer

At this layer, we further process the output of the Mamba VC Encoding Layer. Firstly, we employ a normalization layer [37] to enhance convergence and training stability in deep networks by standardizing all variates to a Gaussian

distribution, thereby minimizing disparities resulting from inconsistent measurements. Then, the feed-forward network (FFN) is used on the series representation of each variate. The FFN layer encodes observed time series and decodes future series representations using dense non-linear connections. During this procedure, FFN implicitly encodes TD by keeping the sequential relationships. Finally, another normalization layer is set to adjust the future series representations.

4.4. Projection Layer

Based on the output of the FFN TD Encoding layer, the tokenized temporal information is reconstructed into the time series requiring prediction via a mapping layer, subsequently undergoing transposition to yield the final predictive outcome.

5. Experiments

5.1. Datasets and Baselines

We conduct experiments on thirteen real-world datasets. For convenience of comparison, we divide them into three types. (1) Traffic-related datasets: Traffic [59] and PEMS [10]. Traffic is a collection of hourly road occupancy rates from the California Department of Transportation, capturing data from 862 sensors across San Francisco Bay area freeways from January 2015 to December 2016. And PEMS is a complicated spatial-temporal time series for public traffic networks in California including four public subsets (PEMS03, PEMS04, PEMS07, PEMS08), which are the same as SCINet [36]. Traffic-related datasets are characterized by a large number of variates, most of which are periodic. (2) ETT datasets: ETT [71] (Electricity Transformer Temperature) comprises data on load and oil temperature, collected from electricity transformers over the period from July 2016 to July 2018. It contains four subsets, ETTm1,

Table 1
The statistics of the thirteen public datasets.

Datasets	Variates	Timesteps	Granularity
Traffic	862	17,544	1hour
PEMS03	358	26,209	5min
PEMS04	307	16,992	5min
PEMS07	883	28,224	5min
PEMS08	170	17,856	5min
ETTm1 & ETTm2	7	17,420	15min
ETTh1 & ETTh2	7	69,680	1hour
Electricity	321	26,304	1hour
Exchange	8	7,588	1day
Weather	21	52,696	10min
Solar-Energy	137	52,560	10min

ETTm2, ETTh1 and ETTh2. ETT datasets have few varieties and weak regularity. (3) Other datasets: Electricity [59], Exchange [59], Weather [59], and Solar-Energy [29]. Electricity records the hourly electricity consumption of 321 customers from 2012 to 2014. Exchange collects daily exchange rates of eight countries from 1990 to 2016. Weather contains 21 meteorological indicators collected every 10 minutes from the Weather Station of the Max Planck Biogeochemistry Institute in 2020. Solar-Energy contains solar power records in 2006 from 137 PV plants in Alabama State which are sampled every 10 minutes. Among them, the Electricity and Solar-Energy datasets contain many variates, and most of them are periodic, while the Exchange and Weather datasets contain fewer variates, and most of them are aperiodic. Tab. 1 shows the statistics of these datasets.

Our models are fairly compared with 9 representative and state-of-the-art (SOTA) forecasting models, including (1) Transformer-based methods: iTransformer [37], PatchTST [44], Crossformer [69], FEDformer [72], Autoformer [59]; (2) Linear-based methods: RLinear [33], TiDE [14], DLinear [66]; and (3) Temporal Convolutional Network-based methods: TimesNet [57]. The brief introductions of these models are as follows:

- iTransformer reverses the order of information processing, which first analyzes the time series information of each individual variate and then fuses the information of all variates. This unique approach has positioned iTransformer as the current SOTA model in TSF.

- PatchTST segments time series into subseries patches as input tokens and uses channel-independent shared embeddings and weights for efficient representation learning.

- Crossformer introduces a cross-attention mechanism that allows the model to interact with information between different time steps to help the model capture long-term dependencies in time series.

- FEDformer is a frequency-enhanced Transformer that takes advantage of the fact that most time series tend to have a sparse representation in well-known basis such as Fourier transform to improve performance.

- Autoformer takes a decomposition architecture that incorporates an auto-correlation mechanism and updates traditional sequence decomposition into the basic inner blocks of the depth model.

- RLinear is the SOTA linear model, which employs reversible normalization and channel independence into pure linear structure.

- TiDE is a Multi-layer Perceptron (MLP) based encoder-decoder model.

- DLinear is the first linear model in TSF and a simple one-layer linear model with decomposition architecture.

- TimesNet uses TimesBlock as a task-general backbone, transforms 1D time series into 2D tensors, and captures intraperiod and interperiod variations using 2D kernels.

5.2. Overall Performance

Tab. 2, Tab. 3, and Tab. 4 present a comparative analysis of the overall performance of our models and other baseline models across all datasets. The best results are highlighted in bold red font, while the second best results are presented in underlined purple font. From the data presented in these tables, we summarize three observations and attach the analysis: (1) S-Mamba attain commendable outcomes on the traffic-related, Electricity, and Solar-Energy datasets. These datasets are distinguished by their numerous variates, most of which are periodic. It is worth noting that period variates are more likely to contain learnable VC. Mamba VC Fusion Layer benefits from this characteristic and improves S-Mamba performance. (2) In the context of the ETT, and Exchange datasets, S-Mamba does not demonstrate a pronounced superiority in performance; indeed, it exhibits a suboptimal outcome. This can be attributed to the fact that these datasets are characterized by a few number of variates, predominantly of an aperiodic nature. Consequently, there are weak VCs between these variates, and the employment of Mamba VC Encoding layer by S-Mamba can't bring useful information and even may inadvertently introduce noise into the predictive model, thus impeding its predictive accuracy. (3) The Weather dataset is special in that it has fewer variates and most variates are aperiodic, but S-Mamba still achieves the best performance on it. We think that this phenomenon arises from the tendency of variates in the Weather dataset to exhibit simultaneous trends of either falling or rising despite the absence of periodic patterns So the Mamba VC Encoding Layer of S-Mamba can still benefit from these data. Moreover, the Weather dataset exhibits large sections of rising or falling trends. The Feed-Forward Network (FFN) layer accurately records these relationships, which is also beneficial for S-Mamba's comprehension.

Furthermore, to provide a more intuitive assessment of S-Mamba's forecast capabilities, we visually compare the

Table 2
Full results of S-Mamba and baselines on traffic-related datasets. The lookback length L is set to 96 and the forecast length T is set to 12, 24, 48, 96 for PEMS and 96, 192, 336, 720 for Traffic.

Models		S-Mamba		iTransformer		RLinear		PatchTST		Crossformer		TiDE		TimesNet		DLinear		FEDformer		Autoformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Traffic	96	0.382	0.261	0.395	0.268	0.649	0.389	0.462	0.295	0.522	0.290	0.805	0.493	0.593	0.321	0.650	0.396	0.587	0.366	0.613	0.388
	192	0.396	0.267	0.417	0.276	0.601	0.366	0.466	0.296	0.530	0.293	0.756	0.474	0.617	0.336	0.598	0.370	0.604	0.373	0.616	0.382
	336	0.417	0.276	0.433	0.283	0.609	0.369	0.482	0.304	0.558	0.305	0.762	0.477	0.629	0.336	0.605	0.373	0.621	0.383	0.622	0.337
	720	0.460	0.300	0.467	0.302	0.647	0.387	0.514	0.322	0.589	0.328	0.719	0.449	0.640	0.350	0.645	0.394	0.626	0.382	0.660	0.408
	Avg	0.414	0.276	0.428	0.282	0.626	0.378	0.481	0.304	0.550	0.304	0.760	0.473	0.620	0.336	0.625	0.383	0.610	0.376	0.628	0.379
PEMS03	12	0.065	0.169	0.071	0.174	0.126	0.236	0.099	0.216	0.090	0.203	0.178	0.305	0.085	0.192	0.122	0.243	0.126	0.251	0.272	0.385
	24	0.087	0.196	0.093	0.201	0.246	0.334	0.142	0.259	0.121	0.240	0.257	0.371	0.118	0.223	0.201	0.317	0.149	0.275	0.334	0.440
	48	0.133	0.243	0.125	0.236	0.551	0.529	0.211	0.319	0.202	0.317	0.379	0.463	0.155	0.260	0.333	0.425	0.227	0.348	1.032	0.782
	96	0.201	0.305	0.164	0.275	1.057	0.787	0.269	0.370	0.262	0.367	0.490	0.539	0.228	0.317	0.457	0.515	0.348	0.434	1.031	0.796
	Avg	0.122	0.228	0.113	0.221	0.495	0.472	0.180	0.291	0.169	0.281	0.326	0.419	0.147	0.248	0.278	0.375	0.213	0.327	0.667	0.601
PEMS04	12	0.076	0.180	0.078	0.183	0.138	0.252	0.105	0.224	0.098	0.218	0.219	0.340	0.087	0.195	0.148	0.272	0.138	0.262	0.424	0.491
	24	0.084	0.193	0.095	0.205	0.258	0.348	0.153	0.275	0.131	0.256	0.292	0.398	0.103	0.215	0.224	0.340	0.177	0.293	0.459	0.509
	48	0.115	0.224	0.120	0.233	0.572	0.544	0.229	0.339	0.205	0.326	0.409	0.478	0.136	0.250	0.355	0.437	0.270	0.368	0.646	0.610
	96	0.137	0.248	0.150	0.262	1.137	0.820	0.291	0.389	0.402	0.457	0.492	0.532	0.190	0.303	0.452	0.504	0.341	0.427	0.912	0.748
	Avg	0.103	0.211	0.111	0.221	0.526	0.491	0.195	0.307	0.209	0.314	0.353	0.437	0.129	0.241	0.295	0.388	0.231	0.337	0.610	0.590
PEMS07	12	0.063	0.159	0.067	0.165	0.118	0.235	0.095	0.207	0.094	0.200	0.173	0.304	0.082	0.181	0.115	0.242	0.109	0.225	0.199	0.336
	24	0.081	0.183	0.088	0.190	0.242	0.341	0.150	0.262	0.139	0.247	0.271	0.383	0.101	0.204	0.210	0.329	0.125	0.244	0.323	0.420
	48	0.093	0.192	0.110	0.215	0.562	0.541	0.253	0.340	0.311	0.369	0.446	0.495	0.134	0.238	0.398	0.458	0.165	0.288	0.390	0.470
	96	0.117	0.217	0.139	0.245	1.096	0.795	0.346	0.404	0.396	0.442	0.628	0.577	0.181	0.279	0.594	0.553	0.262	0.376	0.554	0.578
	Avg	0.089	0.188	0.101	0.204	0.504	0.478	0.211	0.303	0.235	0.315	0.380	0.440	0.124	0.225	0.329	0.395	0.165	0.283	0.367	0.451
PEMS08	12	0.076	0.178	0.079	0.182	0.133	0.247	0.168	0.232	0.165	0.214	0.227	0.343	0.112	0.212	0.154	0.276	0.173	0.273	0.436	0.485
	24	0.104	0.209	0.115	0.219	0.249	0.343	0.224	0.281	0.215	0.260	0.318	0.409	0.141	0.238	0.248	0.353	0.210	0.301	0.467	0.502
	48	0.167	0.228	0.186	0.235	0.569	0.544	0.321	0.354	0.315	0.355	0.497	0.510	0.198	0.283	0.440	0.470	0.320	0.394	0.966	0.733
	96	0.245	0.280	0.221	0.267	1.166	0.814	0.408	0.417	0.377	0.397	0.721	0.592	0.320	0.351	0.674	0.565	0.442	0.465	1.385	0.915
	Avg	0.148	0.224	0.150	0.226	0.529	0.487	0.280	0.321	0.268	0.307	0.441	0.464	0.193	0.271	0.379	0.416	0.286	0.358	0.814	0.659

Table 3
Full results of S-Mamba and baselines on ETT datasets. The lookback length L is set to 96 and the forecast length T is set to 96, 192, 336, 720.

Models		S-Mamba		iTransformer		RLinear		PatchTST		Crossformer		TiDE		TimesNet		DLinear		FEDformer		Autoformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	0.333	0.368	0.334	0.368	0.355	0.376	0.329	0.367	0.404	0.426	0.364	0.387	0.338	0.375	0.345	0.372	0.379	0.419	0.505	0.475
	192	0.376	0.390	0.377	0.391	0.391	0.392	0.367	0.385	0.450	0.451	0.398	0.404	0.374	0.387	0.380	0.389	0.426	0.441	0.553	0.496
	336	0.408	0.413	0.426	0.420	0.424	0.415	0.399	0.410	0.532	0.515	0.428	0.425	0.410	0.411	0.413	0.413	0.445	0.459	0.621	0.537
	720	0.475	0.448	0.491	0.459	0.487	0.450	0.454	0.439	0.666	0.589	0.487	0.461	0.478	0.450	0.474	0.453	0.543	0.490	0.671	0.561
	Avg	0.398	0.405	0.407	0.410	0.414	0.407	0.387	0.400	0.513	0.496	0.419	0.419	0.400	0.406	0.403	0.407	0.448	0.452	0.588	0.517
ETTm2	96	0.179	0.263	0.180	0.264	0.182	0.265	0.175	0.259	0.287	0.366	0.207	0.305	0.187	0.267	0.193	0.292	0.203	0.287	0.255	0.339
	192	0.250	0.309	0.250	0.309	0.246	0.304	0.241	0.302	0.414	0.492	0.290	0.364	0.249	0.309	0.284	0.362	0.269	0.328	0.281	0.340
	336	0.312	0.349	0.311	0.348	0.307	0.342	0.305	0.343	0.597	0.542	0.377	0.422	0.321	0.351	0.369	0.427	0.325	0.366	0.339	0.372
	720	0.411	0.406	0.412	0.407	0.407	0.398	0.402	0.400	1.730	1.042	0.558	0.524	0.408	0.403	0.554	0.522	0.421	0.415	0.433	0.432
	Avg	0.288	0.332	0.288	0.332	0.286	0.327	0.281	0.326	0.757	0.610	0.358	0.404	0.291	0.333	0.350	0.401	0.305	0.349	0.327	0.371
ETTh1	96	0.386	0.405	0.386	0.405	0.386	0.395	0.414	0.419	0.423	0.448	0.479	0.464	0.384	0.402	0.386	0.400	0.376	0.419	0.449	0.459
	192	0.443	0.437	0.441	0.436	0.437	0.424	0.460	0.445	0.471	0.474	0.525	0.492	0.436	0.429	0.437	0.432	0.420	0.448	0.500	0.482
	336	0.489	0.468	0.487	0.458	0.479	0.446	0.501	0.466	0.570	0.546	0.565	0.515	0.491	0.469	0.481	0.459	0.459	0.465	0.521	0.496
	720	0.502	0.489	0.503	0.491	0.481	0.470	0.500	0.488	0.653	0.621	0.594	0.558	0.521	0.500	0.519	0.516	0.506	0.507	0.514	0.512
	Avg	0.455	0.450	0.454	0.447	0.446	0.434	0.469	0.454	0.529	0.522	0.541	0.507	0.458	0.450	0.456	0.452	0.440	0.460	0.496	0.487
ETTh2	96	0.296	0.348	0.297	0.349	0.288	0.338	0.302	0.348	0.745	0.584	0.400	0.440	0.340	0.374	0.333	0.387	0.358	0.397	0.346	0.388
	192	0.376	0.396	0.380	0.400	0.374	0.390	0.388	0.400	0.877	0.656	0.528	0.509	0.402	0.414	0.477	0.476	0.429	0.439	0.456	0.452
	336	0.424	0.431	0.428	0.432	0.415	0.426	0.426	0.433	1.043	0.731	0.643	0.571	0.452	0.452	0.594	0.541	0.496	0.487	0.482	0.486
	720	0.426	0.444	0.427	0.445	0.420	0.440	0.431	0.446	1.104	0.763	0.874	0.679	0.462	0.468	0.831	0.657	0.463	0.474	0.515	0.511
	Avg	0.381	0.405	0.383	0.407	0.374	0.398	0.387	0.407	0.942	0.684	0.611	0.550	0.414	0.427	0.559	0.515	0.437	0.449	0.450	0.459

predictions of S-Mamba and the leading baseline, iTransformer, on four datasets: Electricity, Weather, Traffic, Exchange, and ETTh1, through graphical representation. Specifically, we randomly select a variate and then input its lookback sequence, where the true subsequent sequence is depicted as a blue line and the model's forecast is represented by a red line in Fig. 4. It is evident that on the Electricity, Weather, and Traffic datasets, S-Mamba's predictions closely approximate the actual values, with nearly perfect alignment observed on the Electricity and Traffic

Table 4

Full results of S-Mamba and baselines on Electricity, Exchange, Weather and Solar-Energy datasets. The lookback length L is set to 96 and the forecast length T is set to 96, 192, 336, 720.

Models		S-Mamba		iTransformer		RLinear		PatchTST		Crossformer		TiDE		TimesNet		DLinear		FEDformer		Autoformer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	96	**0.139**	**0.235**	0.148	0.240	0.201	0.281	0.181	0.270	0.219	0.314	0.237	0.329	0.168	0.272	0.197	0.282	0.193	0.308	0.201	0.317
	192	**0.159**	**0.255**	0.162	0.253	0.201	0.283	0.188	0.274	0.231	0.322	0.236	0.330	0.184	0.289	0.196	0.285	0.201	0.315	0.222	0.334
	336	**0.176**	**0.272**	0.178	0.269	0.215	0.298	0.204	0.293	0.246	0.337	0.249	0.344	0.198	0.300	0.209	0.301	0.214	0.329	0.231	0.338
	720	**0.204**	**0.298**	0.225	0.317	0.257	0.331	0.246	0.324	0.280	0.363	0.284	0.373	0.220	0.320	0.245	0.333	0.246	0.355	0.254	0.361
	Avg	**0.170**	**0.265**	0.178	0.270	0.219	0.298	0.205	0.290	0.244	0.334	0.251	0.344	0.192	0.295	0.212	0.300	0.214	0.327	0.227	0.338
Exchange	96	**0.086**	0.207	**0.086**	0.206	0.093	0.217	0.088	**0.205**	0.256	0.367	0.094	0.218	0.107	0.234	0.088	0.218	0.148	0.278	0.197	0.323
	192	0.182	0.304	0.177	0.299	0.184	0.307	0.176	0.299	0.470	0.509	0.184	0.307	0.226	0.344	0.176	0.315	0.271	0.315	0.300	0.369
	336	0.332	0.418	0.331	0.417	0.351	0.432	0.301	0.397	1.268	0.883	0.349	0.431	0.367	0.448	0.313	0.427	0.460	0.427	0.509	0.524
	720	0.867	0.703	0.847	0.691	0.886	0.714	0.901	0.714	1.767	1.068	0.852	0.698	0.964	0.746	0.839	0.695	1.195	0.695	1.447	0.941
	Avg	0.367	0.408	0.360	0.403	0.378	0.417	0.367	0.404	0.940	0.707	0.370	0.413	0.416	0.443	0.354	0.414	0.519	0.429	0.613	0.539
Weather	96	0.165	0.210	0.174	0.214	0.192	0.232	0.177	0.218	0.158	0.230	0.202	0.261	0.172	0.220	0.196	0.255	0.217	0.296	0.266	0.336
	192	0.214	0.252	0.221	0.254	0.240	0.271	0.225	0.259	0.206	0.277	0.242	0.298	0.219	0.261	0.237	0.296	0.276	0.336	0.307	0.367
	336	0.274	0.297	0.278	0.296	0.292	0.307	0.278	0.297	0.272	0.335	0.287	0.335	0.280	0.306	0.283	0.335	0.339	0.380	0.359	0.395
	720	0.350	0.345	0.358	0.347	0.364	0.353	0.354	0.348	0.398	0.418	0.351	0.386	0.365	0.359	0.345	0.381	0.403	0.428	0.419	0.428
	Avg	0.251	0.276	0.258	0.278	0.272	0.291	0.259	0.281	0.259	0.315	0.271	0.320	0.259	0.287	0.265	0.317	0.309	0.360	0.338	0.382
Solar-Energy	96	0.205	0.244	0.203	0.237	0.322	0.339	0.234	0.286	0.310	0.331	0.312	0.399	0.250	0.292	0.290	0.378	0.242	0.342	0.884	0.711
	192	0.237	0.270	0.233	0.261	0.359	0.356	0.267	0.310	0.734	0.725	0.339	0.416	0.296	0.318	0.320	0.398	0.285	0.380	0.834	0.692
	336	0.258	0.288	0.248	0.273	0.397	0.369	0.290	0.315	0.750	0.735	0.368	0.430	0.319	0.330	0.353	0.415	0.282	0.376	0.941	0.723
	720	0.260	0.288	0.249	0.275	0.397	0.356	0.289	0.317	0.769	0.765	0.370	0.425	0.338	0.337	0.356	0.413	0.357	0.427	0.882	0.717
	Avg	0.240	0.273	0.233	0.262	0.369	0.356	0.270	0.307	0.641	0.639	0.347	0.417	0.301	0.319	0.330	0.401	0.291	0.381	0.885	0.711

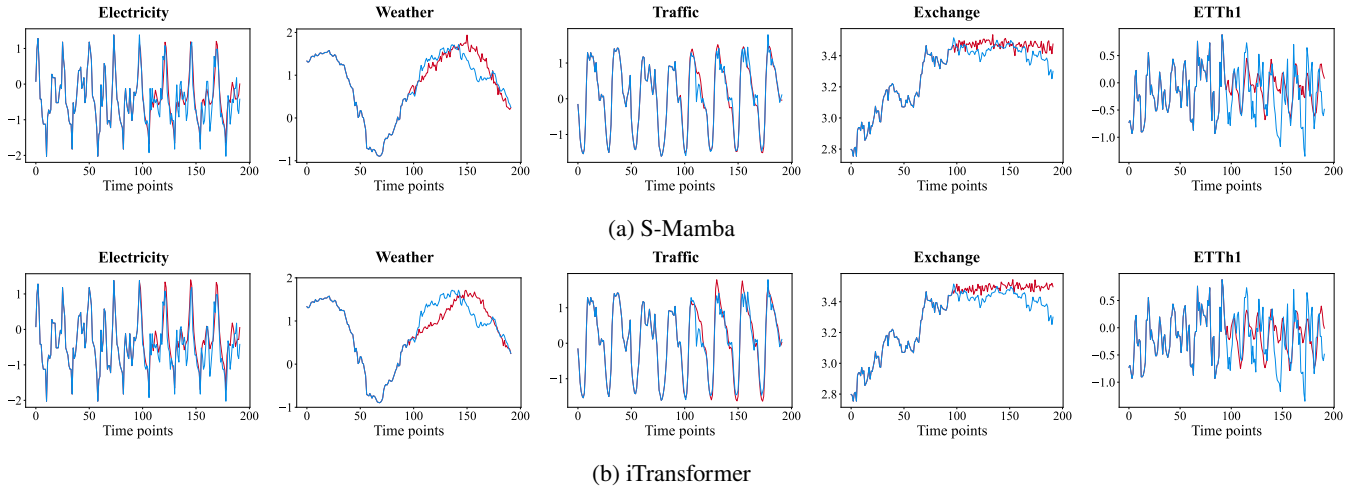(a) S-Mamba

(b) iTransformer

Figure 4: Comparison of forecasts between S-Mamba and iTransformer on five datasets when the input length is 96 and the forecast length is 96. The blue line represents the ground truth and the red line represents the forecast.

datasets and are better than iTransformer. On the Exchange and ETTh1, the two models exhibit similar performance because the two datasets contain few variates, so there is no evident gap between using bidirectional Mamba or using Transformer for information fusion between variates.

5.3. Model Efficiency

To evaluate the computational efficiency of the models, we compare the memory usage and computing time of S-Mamba with several baselines on PEMS07, Electricity, Traffic, and ETTm1. Independent runs are conducted on a single NVIDIA RTX3090 GPU with the batch size set to 16 and meticulously document the results in Fig. 5. In our analysis, bubble charts are used to depict the measurement outcomes, wherein the vertical axis denotes the

Mean Squared Error (MSE), the horizontal axis quantifies the training duration, and the bubble magnitude correlates with the allocated GPU memory. The visualization reveals that the S-Mamba algorithm attains the most favorable MSE metric across the PEMS07, Electricity, and Traffic datasets. When benchmarked against Transformer-based models, S-Mamba typically necessitates short training time and low allocated GPU memory. The RLinear model does utilize minimal GPU memory and curtails training time, it does not confer a competitive edge in terms of forecast precision. Overall, S-Mamba manifests exemplary predictive accuracy with a low computational resource footprint.
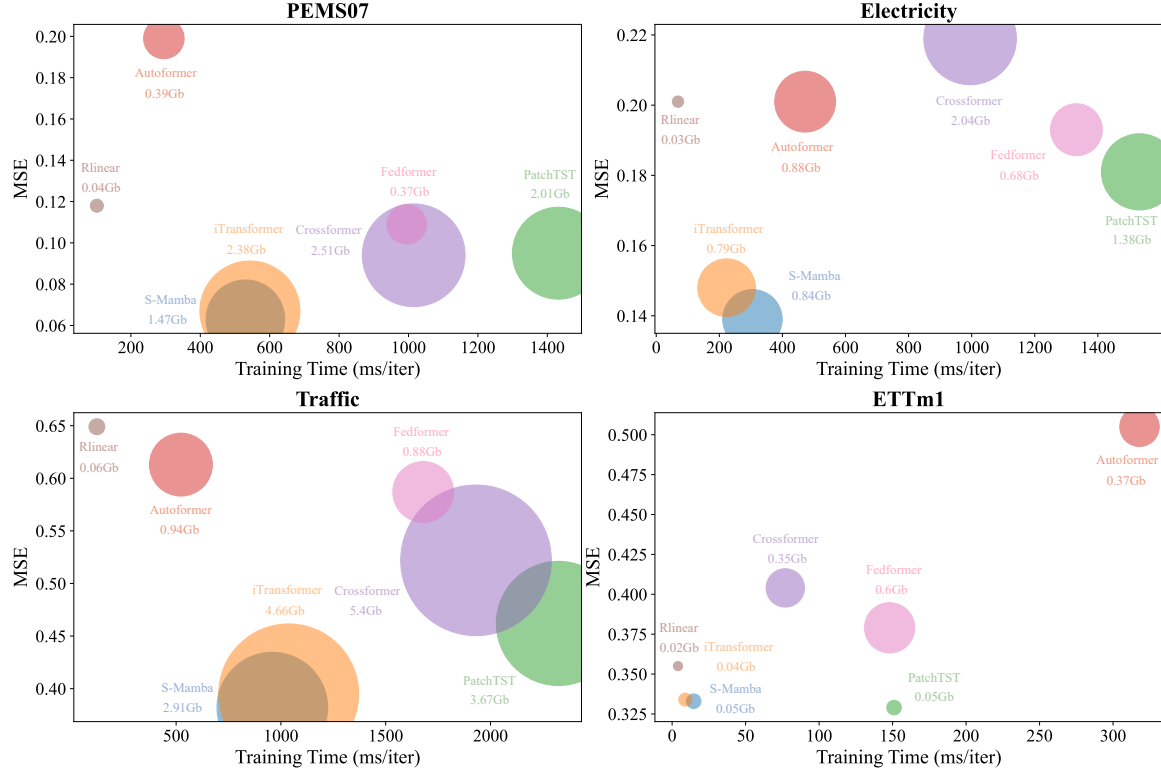
Figure 5: Comparison of S-Mamba and six baselines on MSE, Training Time, and GPU Memory. The lookback length $L = 96$, and the forecast length $T = 12$ for PEMS07 and $T = 96$ for other datasets

Table 5
Ablation study on Electricity, Traffic, Weather, Solar-Energy, and ETTh2. The lookback length $L = 96$, while the forecast length $T \in \{96, 192, 336, 720\}$.

Design	VC Encoding	TD Encoding	Forecast Lengths	Electricity MSE	Electricity MAE	Traffic MSE	Traffic MAE	Weather MSE	Weather MAE	Solar-Energy MSE	Solar-Energy MAE	ETTh2 MSE	ETTh2 MAE
S-Mamba	**bi-Mamba**	**FFN**	96	**0.139**	**0.235**	<u>0.382</u>	<u>0.261</u>	0.165	0.210	<u>0.205</u>	<u>0.244</u>	**0.296**	**0.348**
			192	**0.159**	**0.255**	**0.396**	**0.267**	0.214	0.252	<u>0.237</u>	<u>0.270</u>	**0.376**	**0.396**
			336	**0.176**	<u>0.272</u>	**0.417**	**0.276**	0.274	0.297	<u>0.258</u>	<u>0.288</u>	**0.424**	**0.431**
			720	**0.204**	**0.298**	**0.460**	<u>0.300</u>	0.350	0.345	<u>0.260</u>	<u>0.288</u>	**0.426**	**0.444**
Replace	bi-Mamba	uni-Mamba	96	0.155	0.260	0.488	0.329	**0.161**	**0.204**	0.213	0.255	<u>0.297</u>	<u>0.349</u>
			192	0.173	0.271	0.511	0.341	**0.208**	**0.249**	0.247	0.280	0.378	0.399
			336	0.188	0.281	0.531	0.347	**0.265**	**0.280**	0.267	0.298	0.428	0.437
			720	0.210	0.308	0.621	0.352	**0.343**	**0.339**	0.272	0.295	0.436	0.451
	bi-Mamba	bi-Mamba	96	0.154	0.259	0.512	0.348	<u>0.162</u>	<u>0.205</u>	0.221	0.261	<u>0.297</u>	<u>0.349</u>
			192	0.175	0.273	0.505	0.344	<u>0.210</u>	<u>0.250</u>	0.271	0.291	<u>0.377</u>	<u>0.398</u>
			336	0.184	0.276	0.527	0.369	<u>0.266</u>	<u>0.288</u>	0.271	0.291	0.428	0.437
			720	0.216	0.315	0.661	0.423	<u>0.344</u>	**0.339**	0.278	0.296	0.436	0.451
	bi-Mamba	Attention	96	0.153	0.259	0.514	0.351	0.163	0.207	0.230	0.268	0.299	0.350
			192	0.167	0.266	0.512	0.348	0.211	0.252	0.255	0.287	0.382	0.401
			336	0.183	0.277	0.534	0.377	<u>0.266</u>	<u>0.288</u>	0.275	0.295	0.430	0.438
			720	<u>0.213</u>	0.311	0.685	0.441	0.346	<u>0.340</u>	0.284	0.301	0.433	0.449
	Attention	FFN	96	0.148	0.240	0.395	0.268	0.174	0.214	**0.203**	**0.237**	<u>0.297</u>	<u>0.349</u>
			192	0.162	<u>0.253</u>	0.417	0.276	0.221	0.254	**0.233**	**0.261**	0.380	0.400
			336	<u>0.178</u>	**0.269**	0.433	<u>0.283</u>	0.278	0.296	**0.248**	**0.273**	0.428	<u>0.432</u>
			720	0.225	0.317	0.467	0.302	0.358	0.349	**0.249**	**0.275**	<u>0.427</u>	<u>0.445</u>
w/o	bi-Mamba	w/o	96	<u>0.141</u>	<u>0.238</u>	**0.380**	**0.259**	0.167	0.214	0.210	0.250	0.298	<u>0.349</u>
			192	<u>0.160</u>	0.256	**0.400**	**0.270**	0.217	0.255	0.245	0.276	0.381	0.400
			336	0.181	0.279	<u>0.426</u>	<u>0.283</u>	0.276	0.300	0.263	0.291	0.430	0.437
			720	0.214	<u>0.304</u>	<u>0.466</u>	**0.299**	0.353	0.348	0.268	0.296	0.433	0.446
	w/o	FFN	96	0.169	0.253	0.437	0.283	0.183	0.220	0.228	0.263	0.299	0.350
			192	0.177	0.261	0.449	0.287	0.231	0.262	0.261	0.283	0.380	0.399
			336	0.194	0.278	0.464	0.294	0.285	0.300	0.279	0.294	<u>0.427</u>	0.435
			720	0.233	0.311	0.496	0.313	0.362	0.350	0.276	0.291	0.431	0.449

5.4. Ablation Study

To evaluate the efficacy of the components within S-Mamba, we conduct ablation studies by substituting or eliminating the VC and TD encoding layers. Specifically, the TD encoding layer is replaced with Attention, bidirectional Mamba, unidirectional Mamba, or omitted altogether (w/o). The choice of bidirectional Mamba (bi-Mamba), which is set to benchmark Attention, is made to facilitate global temporal information extraction. The rationale behind employing unidirectional Mamba is its resemblance to RNN models, so inherently possesses the capacity to preserve sequential relationships, thereby making it a suitable candidate for evaluating the impact of sequential encoding on TD. The VC encoding layer was replaced with an Attention mechanism or entirely removed. This modification is predicated on the empirical evidence from iTransformer experiments [37], which demonstrate that Attention was the optimal encoder for VC. We do not use a unidirectional Mamba as the VC Encoding Layer, because Mamba, like RNNs, can only observe information from one direction. A unidirectional Mamba setting would result in the loss of half of the information, making it less effective than bidirectional Mamba or Attention in capturing global information.

Our experimental investigations are conducted on five datasets: Electricity, Traffic, Weather, Solar Energy, and ETTh2. The findings from these experiments in Tab.5 indicate that Mamba exhibits superior performance in VC encoding, whereas the Feed-Forward Network (FFN) maintained its dominance in TD encoding. These findings demonstrate that S-Mamba's current framework is the most efficient.

	MSE	MAE
S-Mamba	0.139	0.235
iTransformer	0.148	0.240

	MSE	MAE
S-Mamba	0.140	0.237
iTransformer	0.147	0.239

	MSE	MAE
S-Mamba	0.139	0.235
iTransformer	0.148	0.239
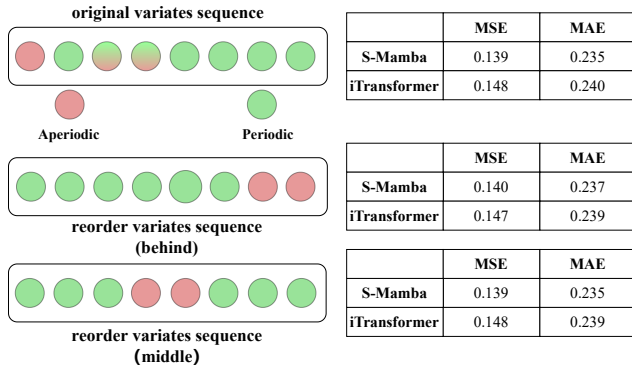
Figure 6: In the Electricity dataset, adjust the distribution of periodic and aperiodic variates. The left side represents the distribution, while the right side indicates the two models' performance when lookback length $L = 96$ and forecast length $T = 96$.

5.5. Can Variate Order Affect the Performance of S-Mamba?

In S-Mamba, each variate is treated as an independent channel, so variates themselves are not inherently ordered. But in the Mamba VC Encoding Layer, the Mamba Block interprets the sequence like RNN, implying that it apprehends the variates as a sequence with implicit order. Mamba's Selective mechanism is closely linked to the Hippo matrix

[23], which causes Mamba to prioritize closer variates in sequences at initialization, leading to a bias against more distant variates. The initial bias towards neighboring variates may potentially impede the acquisition of a global inter-variate correlation. inspiring us to investigate the impact of the variate order on the performance of S-Mamba.

We first use the Fourier transform [7] to categorize the variates into periodic and aperiodic groups and then consider periodic variates as containing reliable information and aperiodic variates as potential noise. This distinction is based on the assumption that periodic variates are more likely to exhibit consistent patterns that can be learned, while aperiodic variates may contain unreliable information due to irregular fluctuations. Next, we decide to alter the variate order by changing the positions of these noise variates for these noise variates have the greatest impact on performance by affecting VC Encoding. Instead of randomly shuffling the overall variate order, it is more effective to adjust the distribution of these noisy variates. Subsequent trials involve repositioning the aperiodic variates towards the middle or end of the variates sequence, followed by evaluating the predictive capabilities of the models trained on these modified datasets. For comparative analysis, we also included experiments with the iTransformer.

The variate distribution and corresponding model performance are illustrated in the Fig. 6. We conduct this experiment only on the Electricity dataset because it requires a dataset with a large number of both periodic and aperiodic variates and Electricity is the only one that satisfies the condition. Our findings suggest that the S-Mamba model's performance remains largely unaffected by the perturbation of variate order. It implies that through adequate training, the S-Mamba can effectively mitigate the initial bias of the Hippo matrix to get accurate inter-variate correlations.

5.6. Can Mamba Outperform Advanced Transformers?

Beyond the foundational Transformer architecture, some advanced Transformers have been introduced, predominantly focusing on the augmentation of the self-attention mechanism. We aim to determine whether Mamba can still maintain an advantage over these advanced Transformers. To the end, we conduct a comparative experiment in which we directly replace the Encoder layer of three advanced self-attention mechanism in three Transformer: Autoformer [59], Flashformer [13] and Flowformer [58] with a unidirectional Mamba (uni-Mamba) for TSF tasks to get Auto-M, Flash-M and Flow-M to compare their performance. The reason behind using a uni-Mamba is that the Encoder layer of these three Transformers handles Temporal Dependency (TD), which is inherently ordered. Therefore, a uni-Mamba is more suitable than a bidirectional Mamba, for apprehending the sequential nature of TD.

We compare the GPU Memory, training time, and MSE of three advanced Transformer models and their Mamba Encoder counterparts on Electricity, Traffic, PEMS07 and ETTm1 as Fig. 7. The findings indicate that employing
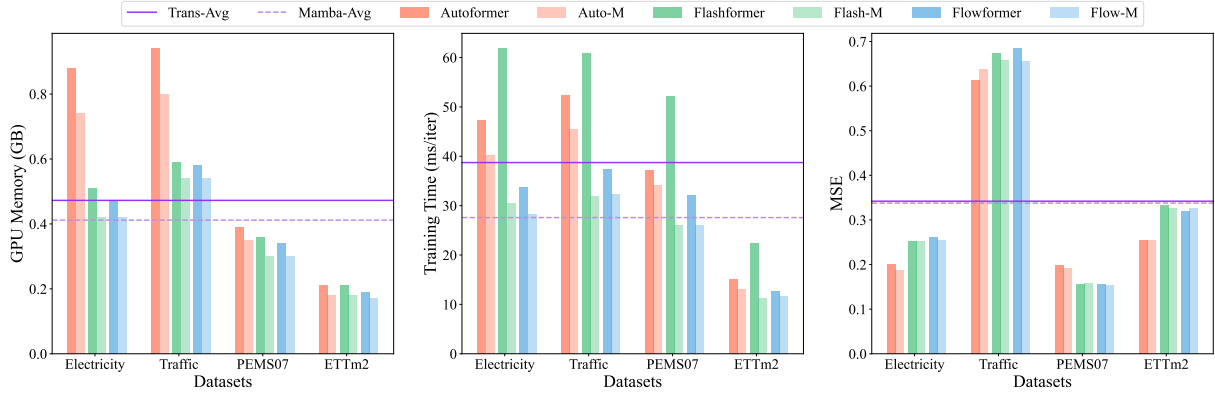
Figure 7: Allocated GPU Memory, Training Time, and MSE of Autoformer, Flashformer, Flowformer and Auto-M, Flash-M, and Flow-M on four datasets. The lookback length $L = 96$ and the forecast length $T = 12$ for PEMS07, $T = 96$ for other datasets. The purple horizontal line represents the average performance of "Transformer models", and the purple dotted line represents the average performance of "Mamba models".
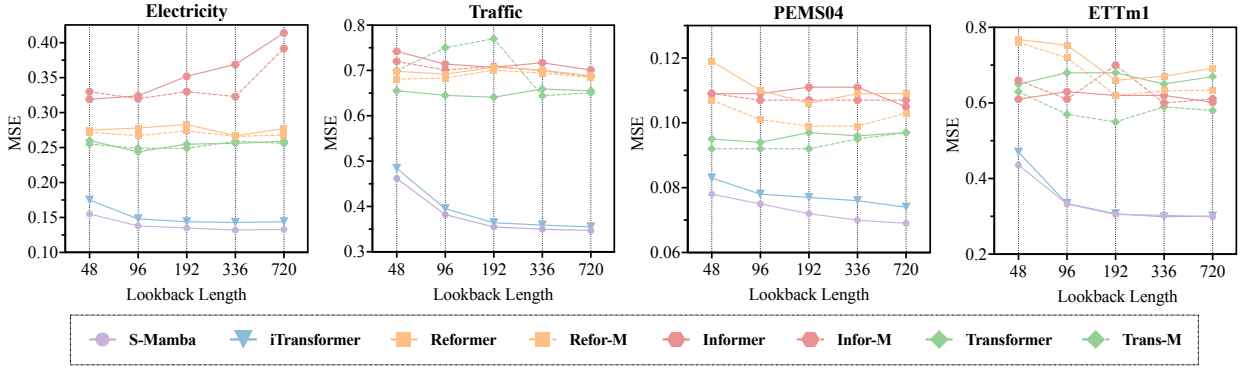

Figure 8: Forecasting performance on four datasets with the lookback length $L \in \{96, 192, 336, 720\}$ while the forecast length $T = 12$ for PEMS04 and $T = 96$ for other datasets.

Mamba as the Encoder directly resulted in reduced GPU usage and training time consumption while achieving slightly improved overall performance. It means that Mamba can still maintain its advantage compared to these advanced self-attention mechanisms or, in other words, these advanced Transformers.

5.7. Can Mamba Help Benefit from Increasing Lookback Length?

Prior research shows that Transformer-based models' performance does not consistently improve with increasing lookback sequence length L, which is somewhat unexpected. A plausible explanation is that the temporal sequence relationship is overlooked under the self-attention mechanism, as it disregards the sequential order, and in some instances, even inverts it. Mamba, resembling a Recurrent Neural Network [41], concentrates on the preceding window during information extraction, thereby preserving certain sequential attributes. It prompts an exploration of Mamba's potential effectiveness in temporal sequence information fusion, aiming to address the issue of diminishing or stagnant performance with increasing lookback length. Consequently, we add an additional Mamba block between the encoder Layer and decoder layer of Transformers-based models. The role of the Mamba Block is to add a layer of time

sequence dependence from the information output by the encoder layer, to add some information similar to position embedding before the decoder layer processes it. We experiment with Reformer [59], Informer [71], and Transformer [52] to get Refor-M, Infor-M, and Trans-M, and evaluate their performance with varying lookback lengths. We also test the performance of S-Mamba and iTransformer as the lookback length increases. The experiment is conducted on four datasets: Electricity, Traffic, PEMS04 and ETTm1. The results are in Fig. 8, from which we can observe four results. (1) S-Mamba and iTransformer can enhance their performance as the input lengthens, but we believe it is not solely due to the Mamba Block or Transformer Block, but rather to the FFN TD Encoding Layer they both possess. (2) S-Mamba consistently outperforms iTransformer, primarily due to the superior performance of S-Mamba's Mamba VC Encoding layer compared to iTransformer's Transformer VC Encoding layer. (3) After incorporating the Mamba Block between the Encoder and Decoder layer, performance enhancements are typically observed in the original model across the four datasets. (4) Despite these variants' performance gains sometimes, they do not achieve optimization with longer lookback lengths. It is consistent with the findings of Zeng et al. [66], which also suggests
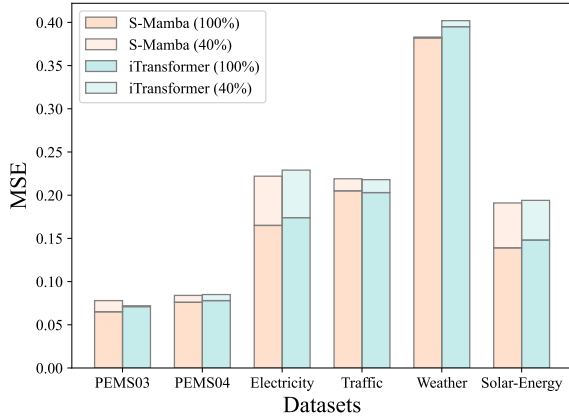
Figure 9: Forecasting performance comparison between S-Mamba and iTransformer trained on 100% variates with on 40% variates. The lookback length $L = 96$ for all datasets. For the PEMS dataset, the output length $T = 12$, while for the other datasets $T = 96$.

that encoding temporal sequence information into the model beforehand does not resolve the issue.

5.8. Is Mamba Generalizable in TSF?

The emergence of pretrained models [16] and large language models [9] based on the Transformer architecture has underscored the Transformer's ability to discern similar patterns across diverse data, highlighting its generalization capabilities. In the context of TSF, it is observed that some variates exhibit a similar pattern of differences, so the generalization potential of the Transformer for sequence data may also take effect on TSF tasks. In this vein, iTransformer [37] conducts a pivotal experiment. The study involves masking a majority of the variates in a dataset and training the model on a limited subset of variates. Subsequently, the model was tasked with forecasting all variates, including those previously unseen, based on the learned information from the few varieties. The results show that Transformer can use its generalization ability to make accurate predictions for unseen variates in TSF tasks. Building on it, we seek to evaluate the generalization capabilities of Mamba in TSF tasks. An experiment is proposed wherein the S-Mamba are trained on merely 40% of the variates in the PEMS03, PEMS04, Electricity, Weather, Traffic, and Solar datasets. We selected these datasets for testing because they contain a large number of variates, which makes it fair to evaluate the models' generalization ability. Then they are employed to predict 100% variates, and the results are subjected to statistical analysis. The outcomes of this investigation in 9 reveal that S-Mamba exhibits generalization potential on the six datasets, which proves their generalizability in TSF tasks.

6. Conclusion

Transformer-based models have consistently exhibited outstanding performance in the field of time series forecasting (TSF), while Mamba has recently gained popularity, and has been shown to surpass the Transformer in

various domains by delivering superior performance while reducing memory and computational overhead. Motivated by these advancements, we seek to investigate the potential of Mamba-based models in the TSF domain, to uncover new research avenues for this field. To this end, we introduce a Mamba-based model for TSF, Simple-Mamba (S-Mamba). It transfers the task of inter-variate correlation (VC) encoding from the Transformer architecture to a bidirectional Mamba block and uses a Feed-Forward Network to extract Temporal Dependencies (TD). We compare S-Mamba with nine representative and state-of-the-art models on thirteen public datasets including Traffic, Weather, Electricity, and Energy forecasting tasks. The results indicate that S-Mamba requires low computational overhead and achieves leading performance. The advantage is primarily attributed to the bidirectional Mamba (bi-Mamba) block within the Mamba VC Encoding Layer, which offers an enhanced understanding of VC at a lower overhead compared to the Transformer. Furthermore, we conduct extensive experiments to prove Mamba possesses robust capabilities in TSF tasks. We demonstrate that the Mamba maintains the same stability as the Transformer in extracting VC and still can offer advantages over advanced Transformer architectures. Transformer architectures can see performance gains by simply integrating or substituting with Mamba blocks. Additionally, Mamba exhibits comparable generalization capabilities to the Transformer. In a word, Mamba exhibits remarkable potential to outperform the Transformer in the TSF tasks.

7. Future Work

As the number of variates grows, global inter-variate correlations (VC) become increasingly valuable and the extraction of them becomes more difficult and consumes more computational resources. Mamba excels at detecting long-range dependencies and controlling the escalation of computational demands, thus equipping it to meet the challenges outlined. In real-life scenarios where resources are limited, compared with Transformer, Mamba is capable of processing more variates information simultaneously and delivering more accurate predictions. For example, in traffic forecasting, Mamba can rapidly assess traffic flows at more intersections, and in hydrological forecasting, it can provide insights into conditions across more tributaries. Looking forward, Mamba-based models are expected to be applicable to a broader spectrum of time series prediction tasks that involve processing extensive variate data.

Pretrained models utilizing the Transformer architecture capitalize on its robust generalization capabilities, achieving notable success in TSF. These models demonstrate effectiveness across various tasks through fine-tuning. Our experimental results indicate that Mamba matches the Transformer in both generalization and stability, suggesting that the development of a Mamba-based pre-training model for TSF tasks could be a fruitful direction to explore.

References

[1] Abdollah Pour, M.M., Hajizadeh, E., Farineya, P., 2022. A New Transformer-Based Hybrid Model for Forecasting Crude Oil Returns. AUT Journal of Modeling and Simulation 54, 19–30. URL: https://miscj.aut.ac.ir/article_4853.html, doi:10.22060/miscj.2022.20734.5263. publisher: Amirkabir University of Technology.

[2] Ahamed, M.A., Cheng, Q., 2024. Timemachine: A time series is worth 4 mambas for long-term forecasting. arXiv:2403.09898.

[3] Ahmed, S., Nielsen, I.E., Tripathi, A., Siddiqui, S., Rasool, G., Ramachandran, R.P., 2023. Transformers in Time-series Analysis: A Tutorial. Circuits, Systems, and Signal Processing 42, 7433–7466. URL: http://arxiv.org/abs/2205.01138, doi:10.1007/s00034-023-02454-8. arXiv:2205.01138 [cs].

[4] Anthony, Q., Tokpanov, Y., Glorioso, P., Millidge, B., 2024. Blackmamba: Mixture of experts for state-space models. ArXiv abs/2402.01771. URL: https://api.semanticscholar.org/CorpusID:267413070.

[5] Benidis, K., Rangapuram, S.S., Flunkert, V., Wang, Y., Maddix, D., Turkmen, C., Gasthaus, J., Bohlke-Schneider, M., Salinas, D., Stella, L., Aubet, F.X., Callot, L., Januschowski, T., 2023. Deep Learning for Time Series Forecasting: Tutorial and Literature Survey. ACM Computing Surveys 55, 1–36. URL: http://arxiv.org/abs/2004.10240, doi:10.1145/3533382. arXiv:2004.10240 [cs, stat].

[6] Bhirangi, R., Wang, C., Pattabiraman, V., Majidi, C., Gupta, A., Hellebrekers, T., Pinto, L., 2024. Hierarchical state space models for continuous sequence-to-sequence modeling. arXiv:2402.10211.

[7] Bracewell, R.N., 1989. The fourier transform. Scientific American 260, 86–95.

[8] Cao, Z., Wu, X., Deng, L.J., Zhong, Y., 2024. A novel state space model with local enhancement and state sharing for image fusion. arXiv:2404.09293.

[9] Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., et al., 2023. A survey on evaluation of large language models. ACM Transactions on Intelligent Systems and Technology .

[10] Chen, C., Petty, K., Skabardonis, A., Varaiya, P., Jia, Z., 2001. Freeway performance measurement system: mining loop detector data. Transportation research record 1748, 96–102.

[11] Chen, S.A., Li, C.L., Yoder, N., Arik, S.O., Pfister, T., 2023. TSMixer: An All-MLP Architecture for Time Series Forecasting. URL: http://arxiv.org/abs/2303.06053, doi:10.48550/arXiv.2303.06053. arXiv:2303.06053 [cs].

[12] Chen, T., Tan, Z., Gong, T., Chu, Q., Wu, Y., Liu, B., Ye, J., Yu, N., 2024. MiM-ISTD: Mamba-in-Mamba for Efficient Infrared Small Target Detection. URL: http://arxiv.org/abs/2403.02148, doi:10.48550/arXiv.2403.02148. arXiv:2403.02148 [cs].

[13] Dao, T., Fu, D., Ermon, S., Rudra, A., Ré, C., 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. Advances in Neural Information Processing Systems 35, 16344–16359.

[14] Das, A., Kong, W., Leach, A., Mathur, S.K., Sen, R., Yu, R., 2023. Long-term forecasting with tide: Time-series dense encoder. Transactions on Machine Learning Research .

[15] De Gooijer, J.G., Hyndman, R.J., 2006. 25 years of time series forecasting. International journal of forecasting 22, 443–473.

[16] Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 .

[17] Dong, W., Zhu, H., Lin, S., Luo, X., Shen, Y., Liu, X., Zhang, J., Guo, G., Zhang, B., 2024. Fusion-mamba for cross-modality object detection. arXiv:2404.09146.

[18] Duong-Trung, N., Nguyen, D.M., Le-Phuoc, D., 2023. Temporal Saliency Detection Towards Explainable Transformer-based Time-series Forecasting. URL: http://arxiv.org/abs/2212.07771, doi:10.48550/arXiv.2212.07771. arXiv:2212.07771 [cs] version: 3.

[19] Elfwing, S., Uchibe, E., Doya, K., 2017. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural networks : the official journal of the International Neural Network Society 107, 3–11.

[20] Foumani, N.M., Tan, C.W., Webb, G.I., Salehi, M., 2024. Improving position encoding of transformers for multivariate time series classification. Data Mining and Knowledge Discovery 38, 22–48. URL: https://doi.org/10.1007/s10618-023-00948-2, doi:10.1007/s10618-023-00948-2.

[21] Grazzi, R., Siems, J., Schrodi, S., Brox, T., Hutter, F., 2024. Is mamba capable of in-context learning? arXiv:2402.03170.

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

[23] Gu, A., Dao, T., Ermon, S., Rudra, A., Ré, C., 2020. Hippo: Recurrent memory with optimal polynomial projections. ArXiv abs/2008.07669.

[24] Gu, A., Goel, K., Ré, C., 2021a. Efficiently modeling long sequences with structured state spaces. arXiv preprint arXiv:2111.00396 .

[25] Gu, A., Johnson, I., Goel, K., Saab, K.K., Dao, T., Rudra, A., R'e, C., 2021b. Combining recurrent, convolutional, and continuous-time models with linear state-space layers, in: Neural Information Processing Systems.

[26] Huang, J., Yang, L., Wang, F., Wu, Y., Nan, Y., Aviles-Rivero, A.I., Schönlieb, C.B., Zhang, D., Yang, G., 2024. Mambamir: An arbitrary-masked mamba for joint medical image reconstruction and uncertainty estimation. arXiv:2402.18451.

[27] Jiang, X., Han, C., Mesgarani, N., 2024. Dual-path mamba: Short and long-term bidirectional selective structured state space models for speech separation. arXiv:2403.18257.

[28] Kitaev, N., Kaiser, Ł., Levskaya, A., 2020. Reformer: The efficient transformer. arXiv preprint arXiv:2001.04451 .

[29] Lai, G., Chang, W.C., Yang, Y., Liu, H., 2018. Modeling long-and short-term temporal patterns with deep neural networks, in: The 41st international ACM SIGIR conference on research & development in information retrieval, pp. 95–104.

[30] Li, K., Li, X., Wang, Y., He, Y., Wang, Y., Wang, L., Qiao, Y., 2024. VideoMamba: State Space Model for Efficient Video Understanding. URL: http://arxiv.org/abs/2403.06977, doi:10.48550/arXiv.2403.06977. arXiv:2403.06977 [cs].

[31] Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.X., Yan, X., 2019. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. Advances in neural information processing systems 32.

[32] Li, Z., Qi, S., Li, Y., Xu, Z., 2023a. Revisiting long-term time series forecasting: An investigation on linear mapping. arXiv preprint arXiv:2305.10721 .

[33] Li, Z., Qi, S., Li, Y., Xu, Z., 2023b. Revisiting long-term time series forecasting: An investigation on linear mapping. arXiv preprint arXiv:2305.10721 .

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

[35] Liu, H., Li, C., Wu, Q., Lee, Y.J., 2023a. Visual instruction tuning.

[36] 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.

[37] Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., Long, M., 2023b. itransformer: Inverted transformers are effective for time series forecasting. arXiv preprint arXiv:2310.06625 .

[38] Liu, Y., Xiao, J., Guo, Y., Jiang, P., Yang, H., Wang, F., 2024. Hsidmamba: Exploring bidirectional state-space models for hyperspectral denoising. arXiv:2404.09697.

[39] Ma, J., Li, F., Wang, B., 2024. U-Mamba: Enhancing Long-range Dependency for Biomedical Image Segmentation. URL: http://arxiv.org/abs/2401.04722, doi:10.48550/arXiv.2401.04722. arXiv:2401.04722 [cs, eess].

[40] Mahmoud, A., Mohammed, A., 2021. A Survey on Deep Learning for Time-Series Forecasting. Springer International Publishing, Cham. pp. 365–392. URL: https://doi.org/10.1007/978-3-030-59338-4_19, doi:10.1007/978-3-030-59338-4_19.

[41] Medsker, L.R., Jain, L., et al., 2001. Recurrent neural networks. Design and Applications 5, 2.

[42] Mellouli, N., Rabah, M.L., Farah, I.R., 2022. Transformers-based time series forecasting for piezometric level prediction, in: 2022 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS), pp. 1–6. URL: https://ieeexplore.ieee.org/abstract/document/9787530, doi:10.1109/EAIS51927.2022.9787530. iSSN: 2473-4691.

[43] Midilli, Y.E., Parshutin, S., 2023. A review for pre-trained transformer-based time series forecasting models. 2023 IEEE 64th International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS) , 1–8URL: https://api.semanticscholar.org/CorpusID:265256497.

[44] Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J., 2022. A time series is worth 64 words: Long-term forecasting with transformers, in: The Eleventh International Conference on Learning Representations.

[45] Pióro, M., Ciebiera, K., Król, K., Ludziejewski, J., Krutul, M., Krajewski, J., Antoniak, S., Miłoś, P., Cygan, M., Jaszczur, S., 2024. MoE-Mamba: Efficient Selective State Space Models with Mixture of Experts. URL: http://arxiv.org/abs/2401.04081, doi:10.48550/arXiv.2401.04081. arXiv:2401.04081 [cs].

[46] Rangapuram, S.S., Seeger, M.W., Gasthaus, J., Stella, L., Wang, Y., Januschowski, T., 2018. Deep state space models for time series forecasting. Advances in neural information processing systems 31.

[47] Schiff, Y., Kao, C.H., Gokaslan, A., Dao, T., Gu, A., Kuleshov, V., 2024. Caduceus: Bi-directional equivariant long-range dna sequence modeling. ArXiv abs/2403.03234. URL: https://api.semanticscholar.org/CorpusID:268253280.

[48] Sezer, O.B., Gudelek, M.U., Ozbayoglu, A.M., 2020. Financial time series forecasting with deep learning : A systematic literature review: 2005–2019. Applied Soft Computing 90, 106181. URL: https://www.sciencedirect.com/science/article/pii/S1568494620301216, doi:https://doi.org/10.1016/j.asoc.2020.106181.

[49] Sherozbek, J., Park, J., Akhtar, M.S., Yang, O.B., 2023. Transformers-Based Encoder Model for Forecasting Hourly Power Output of Transparent Photovoltaic Module Systems. Energies 16, 1353. URL: https://www.mdpi.com/1996-1073/16/3/1353, doi:10.3390/en16031353. number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

[50] Shi, Z., 2024. Mambastock: Selective state space model for stock prediction. arXiv:2402.18959.

[51] Smith, J.T., Warrington, A., Linderman, S.W., 2022. Simplified state space layers for sequence modeling. arXiv preprint arXiv:2208.04933 .

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

[53] Wang, Y., Long, H., Zheng, L., Shang, J., 2024a. Graphformer: Adaptive graph correlation transformer for multivariate long sequence time series forecasting. Knowledge-Based Systems 285, 111321. URL: https://www.sciencedirect.com/science/article/pii/S0950705123010699, doi:https://doi.org/10.1016/j.knosys.2023.111321.

[54] Wang, Z., Ruan, S., Huang, T., Zhou, H., Zhang, S., Wang, Y., Wang, L., Huang, Z., Liu, Y., 2024b. A lightweight multi-layer perceptron for efficient multivariate time series forecasting. Knowledge-Based Systems 288, 111463. URL: https://www.sciencedirect.com/science/article/pii/S0950705124000984, doi:https://doi.org/10.1016/j.knosys.2024.111463.

[55] Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., Sun, L., 2023. Transformers in Time Series: A Survey. URL: http://arxiv.org/abs/2202.07125, doi:10.48550/arXiv.2202.07125. arXiv:2202.07125 [cs, eess, stat].

[56] Woo, G., Liu, C., Sahoo, D., Kumar, A., Hoi, S., 2022. ETS-former: Exponential Smoothing Transformers for Time-series Forecasting. URL: http://arxiv.org/abs/2202.01381, doi:10.48550/arXiv.2202.01381. arXiv:2202.01381 [cs].

[57] Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M., 2022a. Timesnet: Temporal 2d-variation modeling for general time series analysis,

[58] Wu, H., Wu, J., Xu, J., Wang, J., Long, M., 2022b. Flowformer: Linearizing transformers with conservation flows. arXiv preprint arXiv:2202.06258 .

[59] Wu, H., Xu, J., Wang, J., Long, M., 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. Advances in neural information processing systems 34, 22419–22430.

[60] Yang, S., Wang, Y., Chen, H., 2024a. Mambamil: Enhancing long sequence modeling with sequence reordering in computational pathology. arXiv:2403.06800.

[61] Yang, Y., Xing, Z., Zhu, L., 2024b. Vivim: a video vision mamba for medical video object segmentation. arXiv preprint arXiv:2401.14168 .

[62] Yang, Z., Mitra, A., Kwon, S., Yu, H., 2024c. ClinicalMamba: A Generative Clinical Language Model on Longitudinal Clinical Notes. URL: http://arxiv.org/abs/2403.05795. arXiv:2403.05795 [cs].

[63] Yao, J., Hong, D., Li, C., Chanussot, J., 2024. Spectralmamba: Efficient mamba for hyperspectral image classification. arXiv:2404.08489.

[64] Yi, K., Zhang, Q., Fan, W., Wang, S., Wang, P., He, H., An, N., Lian, D., Cao, L., Niu, Z., 2023. Frequency-domain MLPs are More Effective Learners in Time Series Forecasting. Advances in Neural Information Processing Systems 36, 76656–76679. URL: https://proceedings.neurips.cc/paper_files/paper/2023/hash/f1d16af76939f476b5f040fd1398c0a3-Abstract-Conference.html.

[65] Yue, Y., Li, Z., 2024. Medmamba: Vision mamba for medical image classification. arXiv:2403.03849.

[66] Zeng, A., Chen, M., Zhang, L., Xu, Q., 2023. Are transformers effective for time series forecasting?, in: Proceedings of the AAAI conference on artificial intelligence, pp. 11121–11128.

[67] Zeng, A., Chen, M.H., Zhang, L., Xu, Q., 2022. Are transformers effective for time series forecasting?, in: AAAI Conference on Artificial Intelligence. URL: https://api.semanticscholar.org/CorpusID:249097444.

[68] Zhang, T., Zhang, Y., Cao, W., Bian, J., Yi, X., Zheng, S., Li, J., 2022. Less Is More: Fast Multivariate Time Series Forecasting with Light Sampling-oriented MLP Structures. URL: http://arxiv.org/abs/2207.01186, doi:10.48550/arXiv.2207.01186. arXiv:2207.01186 [cs] version: 1.

[69] Zhang, Y., Yan, J., 2022. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting, in: The eleventh international conference on learning representations.

[70] Zhao, H., Zhang, M., Zhao, W., Ding, P., Huang, S., Wang, D., 2024. Cobra: Extending mamba to multi-modal large language model for efficient inference. arXiv:2403.14520.

[71] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W., 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the AAAI conference on artificial intelligence, pp. 11106–11115.

[72] Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R., 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting, in: International conference on machine learning, PMLR. pp. 27268–27286.

[73] Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., Wang, X., 2024. Vision mamba: Efficient visual representation learning with bidirectional state space model. arXiv preprint arXiv:2401.09417 .