MSGNet: Learning Multi-Scale Inter-Series Correlations for Multivariate Time Series Forecasting

Wanlin Cai[1], Yuxuan Liang[2], Xianggen Liu[1], Jianshuai Feng[3], Yuankai Wu[1] *

[1] Sichuan University
[2] The Hong Kong University of Science and Technology (Guangzhou)
[3] Beijing Institute of Technology

yozhiboo@gmail.com, yuxliang@outlook.com, liuxgcs@foxmail.com, 3120185238@bit.edu.cn, wuyk0@scu.edu.cn

Abstract

Multivariate time series forecasting poses an ongoing challenge across various disciplines. Time series data often exhibit diverse intra-series and inter-series correlations, contributing to intricate and interwoven dependencies that have been the focus of numerous studies. Nevertheless, a significant research gap remains in comprehending the varying inter-series correlations across different time scales among multiple time series, an area that has received limited attention in the literature. To bridge this gap, this paper introduces MSGNet, an advanced deep learning model designed to capture the varying inter-series correlations across multiple time scales using frequency domain analysis and adaptive graph convolution. By leveraging frequency domain analysis, MSGNet effectively extracts salient periodic patterns and decomposes the time series into distinct time scales. The model incorporates a self-attention mechanism to capture intra-series dependencies, while introducing an adaptive mixhop graph convolution layer to autonomously learn diverse inter-series correlations within each time scale. Extensive experiments are conducted on several real-world datasets to showcase the effectiveness of MSGNet. Furthermore, MSGNet possesses the ability to automatically learn explainable multi-scale inter-series correlations, exhibiting strong generalization capabilities even when applied to out-of-distribution samples. Code is available at https://github.com/YoZhibo/MSGNet.

Introduction

Throughout centuries, the art of forecasting has been an invaluable tool for scientists, policymakers, actuaries, and salespeople. Its foundation lies in recognizing that hidden outcomes, whether in the future or concealed, often reveal patterns from past observations. Forecasting involves skillfully analyzing available data, unveiling interdependencies and temporal trends to navigate uncharted territories with confidence and envision yet-to-be-encountered instances with clarity and foresight. In this context, time series forecasting emerges as a fundamental concept, enabling the analysis and prediction of data points collected over time, offering insights into variables like stock prices (Cao 2022), weather conditions (Bi et al. 2023), or customer behavior (Salinas et al. 2020).
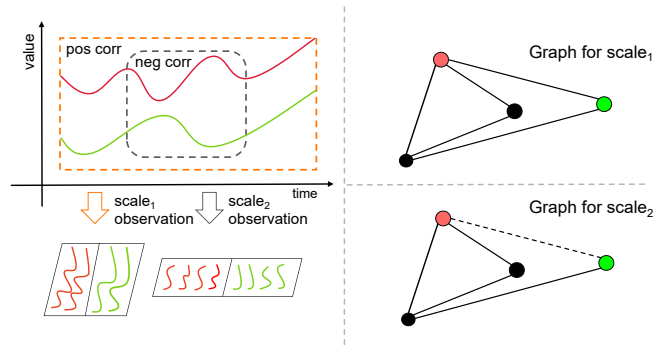

Figure 1: In the longer time $scale_1$, the green and red time series are positively correlated, whereas in the shorter time $scale_2$, they exhibit a negative correlation. Consequently, we observe two distinct graph structures in these two different time scales.

Two interconnected realms within time series forecasting come into play: *intra-series correlation* modeling, which predicts future values based on patterns within a specific time series, and *inter-series correlation* modeling, which explores relationships and dependencies between multiple time series. Recently, deep learning models have emerged as a catalyst for breakthroughs in time series forecasting. On one hand, Recurrent Neural Networks (RNNs) (Salinas et al. 2020), Temporal Convolution Networks (TCNs) (Yue et al. 2022), and Transformers (Zhou et al. 2021) have demonstrated exceptional potential in capturing temporal dynamics within individual series. Simultaneously, a novel perspective arises when considering multivariate time series as graph signals. In this view, the variables within a multivariate time series can be interpreted as nodes within a graph, interconnected through hidden dependency relationships. Consequently, Graph Neural Networks (GNNs) (Kipf and Welling 2017) offer a promising avenue for harnessing the intricate interdependencies among multiple time series.

Within the domain of time series analysis, there is a significant oversight concerning the varying inter-series correlations across different time scales among multiple time series, which the existing deep learning models fail to accurately describe. For instance, in the realm of finance, the correlations among various asset prices, encompassing

stocks, bonds, and commodities, during periods of market instability, asset correlations may increase due to a flight-to-safety phenomenon. Conversely, during economic growth, asset correlations might decrease as investors diversify their portfolios to exploit various opportunities (Baele et al. 2020). Similarly, in ecological systems, the dynamics governing species populations and environmental variables reveal intricate temporal correlations operating at multiple time scales (Whittaker, Willis, and Field 2001). In Figure 1, we provide an example where, in time $scale_1$, we can observe a positive correlation between two time series, whereas in the shorter $scale_2$, we might notice a negative correlation between them. By employing the graph-based approach, we obtain two distinct graph structures.

In the aforementioned examples, the limitation of existing deep learning models becomes apparent, as they often fail to capture the diverse interdependencies and time-varying correlations between the variables in consideration. For instance, when relying solely on one type of inter-series correlation, such as utilizing GNNs with one fixed graph structure (Yu, Yin, and Zhu 2018; Li et al. 2018), these models may suffer from diminished predictive accuracy and suboptimal forecasting performance in scenarios characterized by intricate and varying inter-series correlations. While some methods consider using dynamic and time-varying graph structures to model inter-series correlations (Zheng et al. 2020; Guo et al. 2021), they overlook the crucial fact that these correlations may be intimately tied to time scales of notable stability, exemplified by economic and environmental cycles.

Addressing the identified gaps and aiming to overcome the limitations of prior models, we introduce MSGNet, which is comprised of three essential components: the scale learning and transforming layer, the multiple graph convolution module, and the temporal multi-head attention module. Recognizing the paramount importance of periodicity in time series data and to capture dominant time scales effectively, we leverage the widely recognized Fast Fourier transformation (FFT) method. By applying FFT to the original time series data, we project it into spaces linked to the most prominent time scales. This approach enables us to aptly capture and represent various inter-series correlations unfolding at distinct time scales. Moreover, we introduce a multiple adaptive graph convolution module enriched with a learnable adjacency matrix. For each time scale, a dedicated adjacency matrix is dynamically learned. Our framework further incorporates a multi-head self-attention mechanism adept at capturing intra-series temporal patterns within the data. Our contributions are summarized in three folds:

- We make a key observation that inter-series correlations are intricately associated with different time scales. To address this, we propose a novel structure named MSGNet that efficiently discovers and captures these multi-scale inter-series correlations.

- To tackle the challenge of capturing both intra-series and inter-series correlations simultaneously, we introduce a combination of multi-head attention and adaptive graph convolution modules.

- Through extensive experimentation on real-world datasets, we provide empirical evidence that MSGNet consistently outperforms existing deep learning models in time series forecasting tasks. Moreover, MSGNet exhibits better generalization capability.

Related Works

Time Series Forecasting

Time series forecasting has a long history, with classical methods like VAR (Kilian and Lütkepohl 2017) and Prophet (Taylor and Letham 2018) assuming that intra-series variations follow pre-defined patterns. However, real-world time series often exhibit complex variations that go beyond the scope of these pre-defined patterns, limiting the practical applicability of classical methods. In response, recent years have witnessed the emergence of various deep learning models, including MLPs (Oreshkin et al. 2020; Zeng et al. 2023), TCNs (Yue et al. 2022), RNNs (Rangapuram et al. 2018; Gasthaus et al. 2019; Salinas et al. 2020) and Transformer-based models (Zhou et al. 2021; Wu et al. 2021; Zhou et al. 2022; Wen et al. 2022; Wang et al. 2023), designed for time series analysis. Yet, an ongoing question persists regarding the most suitable candidate for modeling intra-series correlations, whether it be MLP or transformer-based architectures (Nie et al. 2023; Das et al. 2023). Some approaches have considered periodicities as crucial features in time series analysis. For instance, DEPTS (Fan et al. 2022) instantiates periodic functions as a series of cosine functions, while TimesNet (Wu et al. 2023a) performs periodic-dimensional transformations of sequences. Notably, none of these methods, though, give consideration to the diverse inter-series correlations present at different periodicity scales, which is a central focus of this paper.

GNNs for Inter-series Correlation Learning

Recently, there has been a notable rise in the use of GNNs (Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2017; Abu-El-Haija et al. 2019) for learning inter-series correlations. Initially introduced to address traffic prediction (Li et al. 2018; Yu, Yin, and Zhu 2018; Cini et al. 2023; Wu et al. 2023b) and skeleton-based action recognition (Shi et al. 2019), GNNs have demonstrated significant improvements over traditional methods in short-term time series prediction. However, it is important to note that most existing GNNs are designed for scenarios where a pre-defined graph structure is available. For instance, in traffic prediction, the distances between different sensors can be utilized to define the graph structure. Nonetheless, when dealing with general multivariate forecasting tasks, defining a general graph structure based on prior knowledge can be challenging. Although some methods have explored the use of learnable graph structures (Wu et al. 2019; Bai et al. 2020; Wu et al. 2020), they often consider a limited number of graph structures and do not connect the learned graph structures with different time scales. Consequently, these approaches may not fully capture the intricate and evolving inter-series correlations.
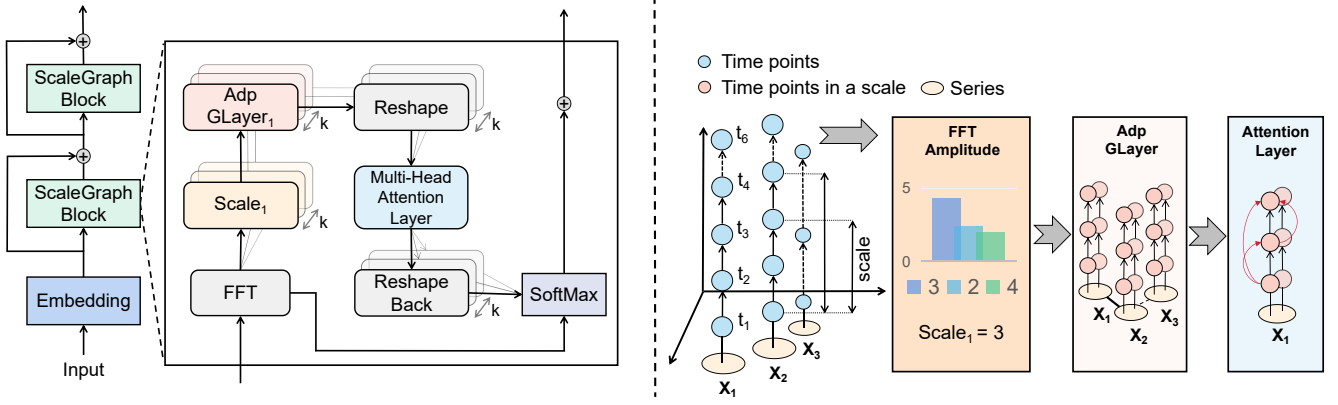
Figure 2: MSGNet employs several ScaleGraph blocks, each encompassing three pivotal modules: an FFT module for multi-scale data identification, an adaptive graph convolution module for inter-series correlation learning within a time scale, and a multi-head attention module for intra-series correlation learning.

Problem Formulation

In the context of multivariate time series forecasting, consider a scenario where the number of variables is denoted by N. We are given input data $\mathbf{X}_{t-L:t} \in \mathbb{R}^{N \times L}$, which represents a retrospective window of observations, comprising X_τ^i values at the τth time point for each variable i in the range from $t - L$ to $t - 1$. Here, L represents the size of the retrospective window, and t denotes the initial position of the forecast window. The objective of the time series forecasting task is to predict the future values of the N variables for a time span of T future time steps. The predicted values are represented by $\hat{\mathbf{X}}_{t:t+T} \in \mathbb{R}^{N \times T}$, which includes X_τ^i values at each time point τ from t to $t + T - 1$ for all the variables.

We assume the ability to discern varying inter-series correlations among N time series at different time scales, which can be represented by graphs. For instance, given a time scale $s_i < L$, we can identify a graph structure $\mathcal{G}_i = \{\mathcal{V}_i, \mathcal{E}_i\}$ from the time series $\mathbf{X}_{p-s_i:p}$. Here, $\mathcal{V}_i$ denotes a set of nodes with $|\mathcal{V}_i| = N$, $\mathcal{E}_i \subseteq \mathcal{V}_i \times \mathcal{V}_i$ represents the weighted edges, and p denotes an arbitrary time point. Considering a collection of k time scales, denoted as $\{s_1, \cdots, s_k\}$, we can identify k adjacency matrices, represented as $\{\mathbf{A}^1, \cdots, \mathbf{A}^k\}$, where each $\mathbf{A}^k \in \mathbb{R}^{N \times N}$. These adjacency matrices capture the varying inter-series correlations at different time scales.

Methodology

As previously mentioned, our work aims to bridge the gaps in existing time series forecasting models through the introduction of MSGNet, a novel framework designed to capture diverse inter-series correlations at different time scales. The overall model architecture is illustrated in Figure 2. Comprising multiple ScaleGraph blocks, MSGNet's essence lies in its ability to seamlessly intertwine various components. Each ScaleGraph block entails a four-step sequence: 1) Identifying the scales of input time series; 2) Unveiling scale-linked inter-series correlations using adaptive graph convo-

lution blocks; 3) Capturing intra-series correlations through multi-head attention; and 4) Adaptively aggregating representations from different scales using a SoftMax function.

Input Embedding and Residual Connection

We embed N variables at the same time step into a vector of size d_{model}: $\mathbf{X}_{t-L:t} \to \mathbf{X}_{\text{emb}}$, where $\mathbf{X}_{\text{emb}} \in \mathbb{R}^{d_{\text{model}} \times L}$. We employ the uniform input representation proposed in (Zhou et al. 2021) to generate the embedding. Specifically, $\mathbf{X}_{\text{emb}}$ is calculated using the following equation:

$$\mathbf{X}_{\text{emb}} = \alpha \text{Conv1D}(\hat{\mathbf{X}}_{t-L:t}) + \mathbf{PE} + \sum_{p=1}^{P} \mathbf{SE}_p. \quad (1)$$

Here, we first normalize the input $\mathbf{X}_{t-L:t}$ and obtain $\hat{\mathbf{X}}_{t-L:t}$, as the normalization strategy has been proven effective in improving stationarity (Liu et al. 2022). Then we project $\hat{\mathbf{X}}_{t-L:t}$ into a d_{model}-dimensional matrix using 1-D convolutional filters (kernel width=3, stride=1). The parameter α serves as a balancing factor, adjusting the magnitude between the scalar projection and the local/global embeddings. $\mathbf{PE} \in \mathbb{R}^{d_{\text{model}} \times L}$ represents the positional embedding of input $\mathbf{X}$, and $\mathbf{SE}_p \in \mathbb{R}^{d_{\text{model}} \times L}$ is a learnable global time stamp embedding with a limited vocabulary size (60 when minutes as the finest granularity).

We implement MSGNet in a residual manner (He et al. 2016). At the very outset, we set $\mathbf{X}^0 = \mathbf{X}_{\text{emb}}$, where $\mathbf{X}_{\text{emb}}$ represents the raw inputs projected into deep features by the embedding layer. In the l-th layer of MSGNet, the input is $\mathbf{X}^{l-1} \in \mathbb{R}^{d_{\text{model}} \times L}$, and the process can be formally expressed as follows:

$$\mathbf{X}^l = \text{ScaleGraphBlock}\left(\mathbf{X}^{l-1}\right) + \mathbf{X}^{l-1}. \quad (2)$$

Here, ScaleGraphBlock denotes the operations and computations that constitute the core functionality of the MSGNet layer.

Scale Identification

Our objective is to enhance forecasting accuracy by leveraging inter-series correlations at different time scales. The

choice of scale is a crucial aspect of our approach, and we place particular importance on selecting periodicity as the scale source. The rationale behind this choice lies in the inherent significance of periodicity in time series data. For instance, in the daytime when solar panels are exposed to sunlight, the time series of energy consumption and solar panel output tend to exhibit a stronger correlation. This correlation pattern would differ if we were to choose a different periodicity, such as considering the correlation over the course of a month or a day.

Inspired by TimesNet (Wu et al. 2023a), we employ the Fast Fourier Transform (FFT) to detect the prominent periodicity as the time scale:

$$\mathbf{F} = \text{Avg}\left(\text{Amp}\left(\text{FFT}(\mathbf{X}_{\text{emb}})\right)\right),$$

$$f_1, \cdots, f_k = \underset{f_* \in \{1, \cdots, \frac{L}{2}\}}{\text{argTopk}} (\mathbf{F}), s_i = \frac{L}{f_i}. \quad (3)$$

Here, $\text{FFT}(\cdot)$ and $\text{Amp}(\cdot)$ denote the FFT and the calculation of amplitude values, respectively. The vector $\mathbf{F} \in \mathbb{R}^L$ represents the calculated amplitude of each frequency, which is averaged across d_{model} dimensions by the function $\text{Avg}(\cdot)$.

In this context, it is noteworthy that the temporally varying inputs may demonstrate distinct periodicities, thereby allowing our model to detect evolving scales. We posit that the correlations intrinsic to this time-evolving periodic scale remain stable. This viewpoint leads us to observe dynamic attributes in the inter-series and intra-series correlations learned by our model.

Based on the selected time scales $\{s_1, \ldots, s_k\}$, we can get several representations corresponding to different time scales by reshaping the inputs into 3D tensors using the following equations:

$$\mathcal{X}^i = \text{Reshape}_{s_i, f_i}(\text{Padding}(\mathbf{X}_{\text{in}})), \quad i \in \{1, \ldots, k\}, \quad (4)$$

where $\text{Padding}(\cdot)$ is used to extend the time series by zeros along the temporal dimension to make it compatible for $\text{Reshape}_{s_i, f_i}(\cdot)$. Note that $\mathcal{X}^i \in \mathbb{R}^{d_{\text{model}} \times s_i \times f_i}$ denotes the i-th reshaped time series based on time scale i. We use $\mathbf{X}_{\text{in}}$ to denote the input matrix of the ScaleGraph block.

Multi-scale Adaptive Graph Convolution

We propose a novel multi-scale graph convolution approach to capture specific and comprehensive inter-series dependencies. To achieve this, we initiate the process by projecting the tensor corresponding to the i-th scale back into a tensor with N variables, where N represents the number of time series. This projection is carried out through a linear transformation, defined as follows:

$$\mathcal{H}^i = \mathbf{W}^i \mathcal{X}^i. \quad (5)$$

Here, $\mathcal{H}^i \in \mathbb{R}^{N \times s_i \times f_i}$, and $\mathbf{W}^i \in \mathbb{R}^{N \times d_{\text{model}}}$ is a learnable weight matrix, tailored to the i-th scale tensor. One may raise concerns that inter-series correlation could be compromised following the application of linear mapping and subsequent linear mapping back. However, our comprehensive experiments demonstrate a noteworthy outcome: the proposed approach adeptly preserves the inter-series correlation by the graph convolution approach.

The graph learning process in our approach involves generating two trainable parameters, $\mathbf{E}_1^i$ and $\mathbf{E}_2^i \in \mathbb{R}^{N \times h}$. Subsequently, an adaptive adjacency matrix is obtained by multiplying these two parameter matrices, following the formula:

$$\mathbf{A}^i = \text{SoftMax}(\text{ReLu}(\mathbf{E}_1^i(\mathbf{E}_2^i)^T)). \quad (6)$$

In this formulation, we utilize the SoftMax function to normalize the weights between different nodes, ensuring a well-balanced and meaningful representation of inter-series relationships.

After obtaining the adjacency matrix $\mathbf{A}^i$ for the i-th scale, we utilize the Mixhop graph convolution method (Abu-El-Haija et al. 2019) to capture the inter-series correlation, as its proven capability to represent features that other models may fail to capture (See Appendix). The graph convolution is defined as follows:

$$\mathcal{H}_{\text{out}}^i = \sigma\left(\underset{j \in \mathcal{P}}{\|} (\mathbf{A}^i)^j \mathcal{H}^i\right), \quad (7)$$

where $\mathcal{H}_{\text{out}}^i$ represents the output after fusion at scale i, $\sigma()$ is the activation function, the hyper-parameter P is a set of integer adjacency powers, $(\mathbf{A}^i)^j$ denotes the learned adjacency matrix $\mathbf{A}^i$ multiplied by itself j times, and $\|$ denotes a column-level connection, linking intermediate variables generated during each iteration. Then, we proceed to utilize a multi-layer perceptron (MLP) to project $\mathcal{H}_{\text{out}}^i$ back into a 3D tensor $\hat{\mathcal{X}}^i \in \mathbb{R}^{d_{\text{model}} \times s_i \times f_i}$.

Multi-head Attention and Scale Aggregation

In each time scale, we employ the Multi-head Attention (MHA) to capture the intra-series correlations. Specifically, for each time scale tensor $\hat{\mathcal{X}}^i$, we apply self MHA on the time scale dimension of the tensor:

$$\hat{\mathcal{X}}_{\text{out}}^i = \text{MHA}_s(\hat{\mathcal{X}}^i). \quad (8)$$

Here, $\text{MHA}_s(\cdot)$ refers to the multi-head attention function proposed in (Vaswani et al. 2017) in the scale dimension. For implementation, it involves reshape the input tensor of size $B \times d_{\text{model}} \times s_i \times f_i$ into a $Bf_i \times d_{\text{model}} \times s_i$ tensor, B is the batch size. Although some studies have raised concerns about the effectiveness of MHA in capturing long-term temporal correlations in time series (Zeng et al. 2023), we have successfully addressed this limitation by employing scale transformation to convert long time spans into periodic lengths. Our results, as presented in the Appendix, show that MSGNet maintains its performance consistently even as the input time increases.

Finally, to proceed to the next layer, we need to integrate k different scale tensors $\hat{\mathcal{X}}_{\text{out}}^1, \cdots, \hat{\mathcal{X}}_{\text{out}}^k$. We first reshape the tensor of each scale back to a 2-way matrix $\hat{\mathbf{X}}_{\text{out}}^i \in \mathbb{R}^{d_{\text{model}} \times L}$. Then, we aggregate the different scales based on their amplitudes:

$$\hat{a}_1, \cdots, \hat{a}_k = \text{SoftMax}(\mathbf{F}_{f_1}, \cdots, \mathbf{F}_{f_k}),$$

$$\hat{\mathbf{X}}_{\text{out}} = \sum_{i=1}^{k} \hat{a}_i \hat{\mathbf{X}}_{\text{out}}^i. \quad (9)$$

In this process, $\mathbf{F}_{f_1}, \cdots, \mathbf{F}_{f_k}$ are amplitudes corresponding to each scale, calculated using the FFT. The SoftMax function is then applied to compute the amplitudes $\hat{a}_1, \cdots, \hat{a}_k$. This Mixture of Expert (MoE) (Jacobs et al. 1991) strategy enables the model to emphasize information from different scales based on their respective amplitudes, facilitating the effective incorporation of multi-scale features into the next layer (Appendix).

Output Layer

To perform forecasting, our model utilizes linear projections in both the time dimension and the variable dimension to transform $\hat{\mathbf{X}}_{\text{out}} \in \mathbb{R}^{d_{\text{model}} \times L}$ into $\hat{\mathbf{X}}_{t:t+T} \in \mathbb{R}^{N \times T}$. This transformation can be expressed as:

$$\hat{\mathbf{X}}_{t:t+T} = \mathbf{W_s} \hat{\mathbf{X}}_{\text{out}} \mathbf{W_t} + \mathbf{b}. \quad (10)$$

Here, $\mathbf{W_s} \in \mathbb{R}^{N \times d_{\text{model}}}$, $\mathbf{W_t} \in \mathbb{R}^{L \times T}$, and $\mathbf{b} \in \mathbb{R}^T$ are learnable parameters. The $\mathbf{W_s}$ matrix performs the linear projection along the variable dimension, and $\mathbf{W_t}$ does the same along the time dimension. The resulting $\hat{\mathbf{X}}_{t:t+T}$ is the forecasted data, where N represents the number of variables, L denotes the input sequence length, and T signifies the forecast horizon.

Experiments

Datasets

To evaluate the advanced capabilities of MSGNet in time series forecasting, we conducted experiments on 8 datasets, namely Flight, Weather, ETT (h1, h2, m1, m2) (Zhou et al. 2021), Exchange-Rate (Lai et al. 2018) and Electricity. With the exception of the Flight dataset, all these datasets are commonly used in existing literature. The Flight dataset's raw data is sourced from the OpenSky official website[1], and it includes flight data related to the COVID-19 pandemic. In Figure 1 and 2 of Appendix, we visualize the changes in flight data during this period. Notably, the flights were significantly affected by the pandemic, resulting in out-of-distribution (OOD) samples for all deep learning models. This provides us with an opportunity to assess the robustness of the proposed model against OOD samples.

Baselines

We have chosen six time series forecasting methods for comparison, encompassing models such as Informer (Zhou et al. 2021), and Autoformer (Wu et al. 2021), which are based on transformer architectures. Furthermore, we included MTGnn (Wu et al. 2020), which relies on graph convolution, as well as DLinear and NLinear (Zeng et al. 2023), which are linear models. Lastly, we considered TimesNet (Wu et al. 2023a), which is based on periodic decomposition and currently holds the state-of-the-art performance.

Experimental Setups

The experiment was conducted using an NVIDIA GeForce RTX 3090 24GB GPU, with the Mean Squared Error (MSE) used as the training loss function. The review window size

of all models was set to $L = 96$ (for fair comparison), and the prediction lengths were $T = \{96, 192, 336, 720\}$. It should be noted that our model can achieve better performance with longer review windows (see Appendix). These settings were applied to all models. The initial learning rate was $LR = 0.0001$, batch size was $Batch = 32$, and the number of epochs was $Epochs = 10$, and early termination was used where applicable. For more details on hyperparameter settings of our model, please refer to Appendix. (0.7, 0.1, 0.2) or (0.6, 0.2, 0.2) of the data are used as training, validation, and test data, respectively. As for baselines, relevant data from the papers (Wu et al. 2023a) or official code (Wu et al. 2020) was utilized.
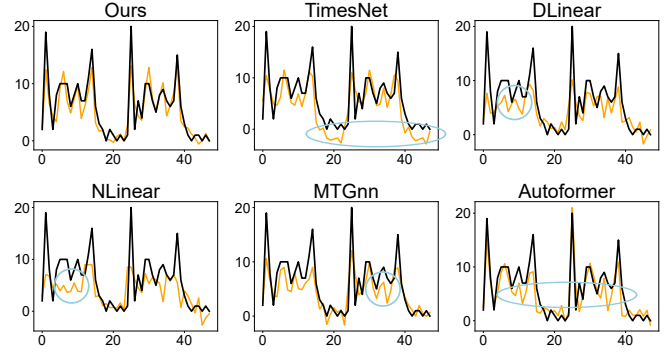

Figure 3: Visualization of Flight prediction results: black lines for true values, orange lines for predicted values, and blue markings indicating significant deviations.

Results and Analysis

Table 1 summarizes the predictive performance of all methods on 8 datasets, showcasing MSGNet's excellent results. Specifically, regarding the average Mean Squared Error (MSE) with different prediction lengths, it achieved the best performance on 5 datasets and the second-best performance on 2 datasets. In the case of the Flight dataset, MSGNet outperformed TimesNet (current SOTA), reducing MSE and MAE by 21.5% (from 0.265 to 0.208) and 13.7% (from 0.372 to 0.321) in average, respectively. Although TimesNet uses multi-scale information, it adopts a pure computer vision model to capture inter and intra-series correlations, which is not very effective for time series data. Autoformer demonstrated outstanding performance on the Flight dataset, likely attributed to its established autocorrelation mechanism. Nevertheless, even with GNN-based inter-series correlation modeling, MTGnn remained significantly weaker than our model due to a lack of attention to different scales. Furthermore, we assessed the model's generalization ability by calculating its average rank across all datasets. Remarkably, MSGNet outperforms other models on average ranking.

MSGNet's excellence is evident in Figure 3, as it closely mirrors the ground truth, while other models suffer pronounced performance dips during specific time periods. The depicted peaks and troughs in the figure align with crucial flight data events, trends, or periodic dynamics. The inability of other models to accurately follow these variations likely

Models		Ours		TimesNet		DLinear		NLinear		MTGnn		Autoformer		Informer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Flight	96	**0.183**	**0.301**	0.237	0.350	0.221	0.337	0.270	0.379	0.196	0.316	0.204	0.319	0.333	0.405
	192	**0.189**	**0.306**	0.224	0.337	0.220	0.336	0.272	0.380	0.272	0.379	0.200	0.314	0.358	0.421
	336	0.206	0.320	0.289	0.394	0.229	0.342	0.280	0.385	0.260	0.369	**0.201**	**0.318**	0.398	0.446
	720	**0.253**	**0.358**	0.310	0.408	0.263	0.366	0.316	0.409	0.390	0.449	0.345	0.426	0.476	0.484
Weather	96	**0.163**	**0.212**	0.172	0.220	0.196	0.255	0.196	0.235	0.171	0.231	0.266	0.336	0.300	0.384
	192	**0.212**	**0.254**	0.219	0.261	0.237	0.296	0.241	0.271	0.215	0.274	0.307	0.367	0.598	0.544
	336	0.272	**0.299**	0.280	0.306	0.283	0.335	0.293	0.308	**0.266**	0.313	0.359	0.395	0.578	0.523
	720	0.350	**0.348**	0.365	0.359	0.345	0.381	0.366	0.356	**0.344**	0.375	0.419	0.428	1.059	0.741
ETTm1	96	**0.319**	**0.366**	0.338	0.375	0.345	0.372	0.350	0.371	0.381	0.415	0.505	0.475	0.672	0.571
	192	0.376	0.397	**0.374**	**0.387**	0.380	0.389	0.389	0.390	0.442	0.451	0.553	0.496	0.795	0.669
	336	0.417	0.422	**0.410**	**0.411**	0.413	0.413	0.422	0.412	0.475	0.475	0.621	0.537	1.212	0.871
	720	0.481	0.458	0.478	0.450	**0.474**	0.453	0.482	**0.446**	0.531	0.507	0.671	0.561	1.166	0.823
ETTm2	96	**0.177**	**0.262**	0.187	0.267	0.193	0.292	0.188	0.272	0.240	0.343	0.255	0.339	0.365	0.453
	192	**0.247**	**0.307**	0.249	0.309	0.284	0.362	0.253	0.312	0.398	0.454	0.281	0.340	0.533	0.563
	336	**0.312**	**0.346**	0.321	0.351	0.369	0.427	0.314	0.350	0.568	0.555	0.339	0.372	1.363	0.887
	720	0.414	**0.403**	**0.408**	0.403	0.554	0.522	0.414	0.405	1.072	0.767	0.433	0.432	3.379	1.338
ETTh1	96	0.390	0.411	**0.384**	0.402	0.386	**0.400**	0.393	0.400	0.440	0.450	0.449	0.459	0.865	0.713
	192	0.442	0.442	**0.436**	**0.429**	0.437	0.432	0.449	0.433	0.449	0.433	0.500	0.482	1.008	0.792
	336	**0.480**	0.468	0.491	0.469	0.481	0.459	0.485	**0.448**	0.598	0.554	0.521	0.496	1.107	0.809
	720	0.494	0.488	0.521	0.500	0.519	0.516	**0.469**	**0.461**	0.685	0.620	0.514	0.512	1.181	0.865
ETTh2	96	0.328	0.371	0.340	0.374	0.333	0.387	**0.322**	**0.369**	0.496	0.509	0.346	0.388	3.755	1.525
	192	**0.402**	**0.414**	0.402	0.414	0.477	0.476	0.410	0.419	0.716	0.616	0.456	0.452	5.602	1.931
	336	**0.435**	**0.443**	0.452	0.452	0.594	0.541	0.444	0.449	0.718	0.614	0.482	0.486	4.721	1.835
	720	**0.417**	**0.441**	0.462	0.468	0.831	0.657	0.450	0.462	1.161	0.791	0.515	0.511	3.647	1.625
Electricity	96	**0.165**	0.274	0.168	**0.272**	0.197	0.282	0.198	0.274	0.211	0.305	0.201	0.317	0.274	0.368
	192	**0.184**	0.292	0.184	0.289	0.196	0.285	0.197	**0.277**	0.225	0.319	0.222	0.334	0.296	0.386
	336	**0.195**	0.302	0.198	0.300	0.209	0.301	0.211	**0.292**	0.247	0.340	0.231	0.338	0.300	0.394
	720	0.231	0.332	**0.220**	**0.320**	0.245	0.333	0.253	0.325	0.287	0.373	0.254	0.361	0.373	0.439
Exchange	96	0.102	0.230	0.107	0.234	0.088	0.218	**0.088**	**0.205**	0.267	0.378	0.197	0.323	0.847	0.752
	192	0.195	0.317	0.226	0.344	**0.176**	0.315	0.177	**0.297**	0.590	0.578	0.300	0.369	1.204	0.895
	336	0.359	0.436	0.367	0.448	**0.313**	0.427	0.323	**0.409**	0.939	0.749	0.509	0.524	1.672	1.036
	720	0.940	0.738	0.964	0.746	**0.839**	**0.695**	0.923	0.725	1.107	0.834	1.447	0.941	2.478	1.310
Avg Rank		**1.813**		2.750		3.563		2.813		5.313		4.750		7.000	

Table 1: Forecast results with 96 review window and prediction length $\{96, 192, 336, 720\}$. The best result is represented in bold, followed by underline.

Visualization of Learned Inter-series Correlation

Figure 4 illustrates three learned adjacency matrices for distinct time scales. In this instance, our model identifies three significant scales, corresponding to 24, 6, and 4 hours, respectively. As depicted in this showcase, our model learns different adaptive adjacency matrices for various scales, effectively capturing the interactions between airports in the flight data set. For instance, in the case of Airport 6, which is positioned at a greater distance from Airports 0, 1, and 3, it exerts a substantial influence on these three airports primarily over an extended time scale (24 hours). However, the impact diminishes notably as the adjacency matrix values decrease during subsequent shorter periods (6 and 4 hours). On the other hand, airports 0, 3, and 5, which are closer in distance, exhibit stronger mutual influence at shorter time scales. These observations mirror real-life scenarios, indicating that there might be stronger spatial correlations between flights at certain time scales, linked to their physical proximity.

stems from architecture constraints, hindering their capacity to grasp multi-scale patterns, sudden shifts, or intricate inter-series and intra-series correlations.

Dataset	Flight		Weather		ETTm2	
Metric	MSE	MAE	MSE	MAE	MSE	MAE
MSGNet	**0.195**	**0.311**	**0.218**	**0.255**	**0.245**	**0.304**
w/o-AdapG	0.302	0.401	0.232	0.270	0.253	0.313
w/o-MG	0.213	0.331	0.226	0.261	0.250	0.307
w/o-A	0.198	0.314	0.224	0.259	0.247	0.306
w/o-Mix	0.202	0.318	0.224	0.260	0.247	0.304
TimesNet	0.263	0.372	0.226	0.263	0.254	0.309

Table 2: Ablation analysis of Flight, Weather and ETTm2 datasets. Results represent the average error of prediction length $\{96, 336\}$, with the best performance highlighted in bold black.

Models	Ours		TimesNet		DLinear		NLinear		MTGnn		Autoformer	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Flight(7:1:2)	**0.208**	**0.321**	0.265	0.372	0.233	0.345	0.285	0.388	0.280	0.378	0.238	0.344
Flight(4:4:2)	**0.252**	**0.366**	0.335	0.426	0.332	0.448	0.365	0.447	0.407	0.501	0.307	0.424
Decrease(%)	**21.29**	**13.80**	26.47	14.32	42.29	29.87	28.19	15.17	45.74	32.52	29.17	23.09

Table 3: Generalization test under COVID-19 influence: mean error for all prediction lengths, black bold indicates best performance. **Decrease** shows the percentage of performance reduction after partition modification.
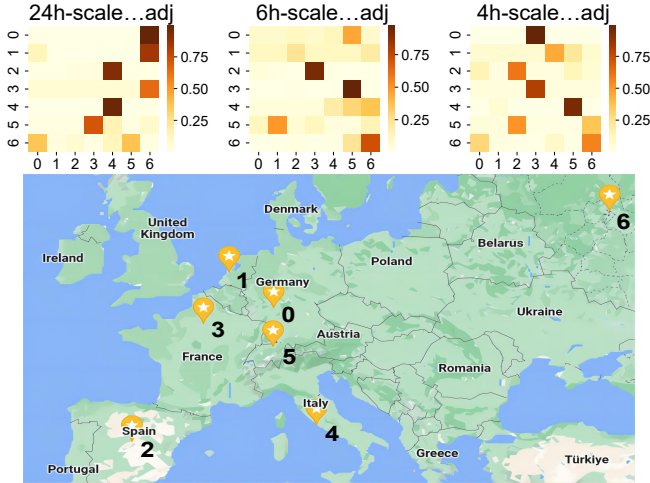

Figure 4: Learned adjacency matrices (24h, 6h, and 4h of the first layer) and airport map for Flight dataset.

Ablation Analysis

We conducted ablation testing to verify the effectiveness of the MSGNet design. We considered 5 ablation methods and evaluated them on 3 datasets. The following will explain the variants of its implementation:

1. **w/o-AdapG:** We removed the adaptive graph convolutional layer (graph learning) from the model.

2. **w/o-MG:** We removed multi-scale graph convolution and only used a shared graph convolution layer to learn the overall inter-series dependencies.

3. **w/o-A:** We removed multi-head self-attention and eliminated intra-series correlation learning.

4. **w/o-Mix:** We replaced the mixed hop convolution method with the traditional convolution method (Kipf and Welling 2017).

Table 2 shows the results of the ablation study. Specifically, we have summarized the following four improvements:

1. **Improvement of graph learning layer:** After removing the graph structure, the performance of the model showed a significant decrease. This indicates that learning the inter-series correlation between variables is crucial in predicting multivariate time series.

2. **Improvement of multi-scale graph learning:** Based on the results of the variant w/o-MG, it can be concluded

that the multi-scale graph learning method significantly contributes to improving model performance. This finding suggests that there exist varying inter-series correlations among different time series at different scales.

3. **Improvement of MHA layer:** Examining the results from w/o-A and TimesNet, it becomes apparent that employing multi-head self-attention yields marginal enhancements in performance.

4. **Improvement of mix-hop convolution:** The results of variant w/o-Mix indicate that the mix-hop convolution method is effective in improving the model's performance as w/o-Mix is slightly worse than MSGNet.

Generalization Capabilities

To verify the impact of the epidemic on flight predictions and the performance of MSGNet in resisting external influences, we designed a new ablation test by modifying the partitioning of the Flight dataset to 4:4:2. This design preserved the same test set while limiting the training set to data before the outbreak of the epidemic, and using subsequent data as validation and testing sets. The specific results are shown in Table 3. By capturing multi-scale inter-series correlations, MSGNet not only achieved the best performance under two different data partitions but also exhibited the least performance degradation and strongest resistance to external influences. The results demonstrate that MSGNet possesses a robust generalization capability to out-of-distribution (OOD) samples. We hypothesize that this strength is attributed to MSGNet's ability to capture multiple inter-series correlations, some of which continue to be effective even under OOD samples of multivariate time series. This hypothesis is further supported by the performance of TimesNet, which exhibits a relatively small performance drop, ranking second after our method. It is worth noting that TimesNet also utilizes multi-scale information, similar to our approach.

Conclusion

In this paper, we introduced MSGNet, a novel framework designed to address the limitations of existing deep learning models in time series analysis. Our approach leverages periodicity as the time scale source to capture diverse inter-series correlations across different time scales. Through extensive experiments on various real-world datasets, we demonstrated that MSGNet outperforms existing models in forecasting accuracy and captures intricate interdependencies among multiple time series. Our findings underscore the importance of discerning the varying inter-series correlation of different time scales in the analysis of time series data.

Acknowledgements

References

Abu-El-Haija, S.; Perozzi, B.; Kapoor, A.; Alipourfard, N.; Lerman, K.; Harutyunyan, H.; Ver Steeg, G.; and Galstyan, A. 2019. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *ICML*, 21–29. PMLR.

Baele, L.; Bekaert, G.; Inghelbrecht, K.; and Wei, M. 2020. Flights to safety. *The Review of Financial Studies*, 33(2): 689–746.

Bai, L.; Yao, L.; Li, C.; Wang, X.; and Wang, C. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. *In Neurips*, 33: 17804–17815.

Bi, K.; Xie, L.; Zhang, H.; Chen, X.; Gu, X.; and Tian, Q. 2023. Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, 1–6.

Cao, L. 2022. Ai in finance: challenges, techniques, and opportunities. *ACM Computing Surveys (CSUR)*, 55(3): 1–38.

Cini, A.; Marisca, I.; Bianchi, F. M.; and Alippi, C. 2023. Scalable spatiotemporal graph neural networks. In *AAAI*, volume 37, 7218–7226.

Das, A.; Kong, W.; Leach, A.; Sen, R.; and Yu, R. 2023. Long-term Forecasting with TiDE: Time-series Dense Encoder. *arXiv preprint arXiv:2304.08424*.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *In Neurips*, 29.

Fan, W.; Zheng, S.; Yi, X.; Cao, W.; Fu, Y.; Bian, J.; and Liu, T.-Y. 2022. DEPTS: Deep Expansion Learning for Periodic Time Series Forecasting. In *ICLR*.

Gasthaus, J.; Benidis, K.; Wang, Y.; Rangapuram, S. S.; Salinas, D.; Flunkert, V.; and Januschowski, T. 2019. Probabilistic forecasting with spline quantile function RNNs. In *ICML*, 1901–1910. PMLR.

Guo, S.; Lin, Y.; Wan, H.; Li, X.; and Cong, G. 2021. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 34(11): 5415–5428.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Jacobs, R. A.; Jordan, M. I.; Nowlan, S. J.; and Hinton, G. E. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1): 79–87.

Kilian, L.; and Lütkepohl, H. 2017. *Structural vector autoregressive analysis*. Cambridge University Press.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. *In ICLR*.

Lai, G.; Chang, W.-C.; Yang, Y.; and 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*, 95–104.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *ICLR*.

Liu, Y.; Wu, H.; Wang, J.; and Long, M. 2022. Non-stationary transformers: Exploring the stationarity in time series forecasting. *In Neurips*, 35: 9881–9893.

Nie, Y.; H. Nguyen, N.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *ICLR*.

Oreshkin, B. N.; Carpov, D.; Chapados, N.; and Bengio, Y. 2020. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *ICLR*.

Rangapuram, S. S.; Seeger, M. W.; Gasthaus, J.; Stella, L.; Wang, Y.; and Januschowski, T. 2018. Deep state space models for time series forecasting. *In Neurips*, 31.

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

Shi, L.; Zhang, Y.; Cheng, J.; and Lu, H. 2019. Skeleton-based action recognition with directed graph neural networks. In *CVPR*, 7912–7921.

Taylor, S. J.; and Letham, B. 2018. Forecasting at scale. *The American Statistician*, 72(1): 37–45.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *In Neurips*, 30.

Wang, H.; Peng, J.; Huang, F.; Wang, J.; Chen, J.; and Xiao, Y. 2023. MICN: Multi-scale Local and Global Context Modeling for Long-term Series Forecasting. In *ICLR*.

Wen, Q.; Zhou, T.; Zhang, C.; Chen, W.; Ma, Z.; Yan, J.; and Sun, L. 2022. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*.

Whittaker, R. J.; Willis, K. J.; and Field, R. 2001. Scale and species richness: towards a general, hierarchical theory of species diversity. *Journal of biogeography*, 28(4): 453–470.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023a. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *ICLR*.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *In Neurips*, 34: 22419–22430.

Wu, Y.; Yang, H.; Lin, Y.; and Liu, H. 2023b. Spatiotemporal Propagation Learning for Network-Wide Flight Delay Prediction. *IEEE Transactions on Knowledge and Data Engineering*.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; and Zhang, C. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *KDD*, 753–763.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; and Zhang, C. 2019. Graph wavenet for deep spatial-temporal graph modeling. In *IJCAI*, 1907–1913.

Yu, B.; Yin, H.; and Zhu, Z. 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *IJCAI*, 3634–3640.

Yue, Z.; Wang, Y.; Duan, J.; Yang, T.; Huang, C.; Tong, Y.; and Xu, B. 2022. Ts2vec: Towards universal representation of time series. In *AAAI*, volume 36, 8980–8987.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *AAAI*, volume 37, 11121–11128.

Zheng, C.; Fan, X.; Wang, C.; and Qi, J. 2020. Gman: A graph multi-attention network for traffic prediction. In *AAAI*, volume 34, 1234–1241.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, volume 35, 11106–11115.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *ICML*, 27268–27286. PMLR.