2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

S2LNet: Review the Non-Stationarity in Multivariate Time Series Forecasting

Zhuang Xing[1](✉)

[1] Xi'an Damai Network Technology Co., Ltd., Xi'an 710000, China
xingzhuang@damaikeji.com

Abstract. Transformer-based methods have achieved remarkable advances in multivariate time series forecasting for their long-range ability. However, the non-stationarity of real-world time series, make these models particularly prone to overfitting when data distribution changes over time. Recently, despite various attempts in existing studies, they either overlook cross-channel mutual information gains or struggle to effectively capture cross-time features. To overcome these limitations, we review the characteristics of time series and develop a novel **S**hort-term **to L**ong-term network called **S2LNet**, which combines short-term cross-time features into long-term distributions and then models cross-channel dependencies models cross-time and cross-channel dependencies. For cross-time features, S2LNet first decomposes the input sequence into seasonal and trend items, then employs Transformers for capturing seasonal features seasonal items and multilayer perceptrons (MLPs) for trend items modeling trend features. These modeled short-term features are then fused and downsampled into long-term relationships through the Long-term Fusion module, followed by a channel-wise Transformer for long-term cointegration across channels. Extensive experiments on various real-world benchmarks have verified the superiority of our model over other state-of-the-art baselines.

Keywords: Non-stationary Time Series, Cross-time Dependencies, Cross-channel Dependencies.

1 Introduction

1.1 A Subsection Sample

Multivariate time series forecasting (MTSF) promotes various real-world applications where future trends can be learned from historic multi-channel signals, such as weather prediction [1] and building energy consumption [2]. Recently, Transformer-based methods [3, 4] have been widely used in MTSF for their robust long-range ability [5, 6]. However, real-world time series data are often non-stationary, with shifting distributional statistics data distribution shifting over time (known as distribution drift). This problem leads to significant overfitting for Transformer-based methods [7-9].

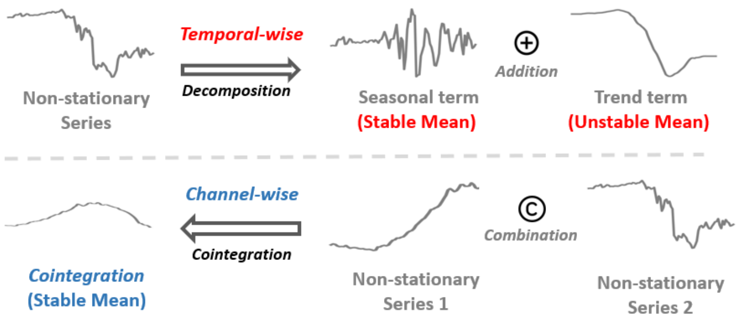

Fig. 1. Seasonal-trend decomposition and cointegration. A stable mean reflects consistent distribution over time, while non-stationary series show varying mean values. Cointegration describes the long-term relationships where two or more non-stationary time series are linearly combined to produce a stationary series.

To this issue, recent approaches have proposed to employ channel-independent (CI) strategies, ignoring correlations across channels and focus on temporal dependencies, named channel-independent (CI). It avoids the challenges of spurious regression that may arise from irrelevant channels. Surprisingly, by avoiding disruptions of spurious regression that may arise from irrelevant channels, CI models [10, 11] even exhibit superior performance that surpasses other contemporary methods [12, 13]. Afterward, a novel channel-dependent (CD) model, iTransformer [14], proposes a simple but novel operation that treats the entire input sequence as a token and captures channel correlations, outperforming previous elaborate CD methods [3, 15]. However, these CI and CD models still have certain limitations: the former typically overlooks the mutual information gain across channels, the latter lacks sufficient capability to model short-term temporal dependencies [16].

To illustrate our solution comprehensively, we review the non-stationary nature of the time series in Figure 1. For temporal modeling, decomposition effectively separates the seasonal item, with its stable mean, from the trend item. In channel-wise cointegration [17], non-stationary sequences are combined into a stationary one. Building on these concepts, we propose a **S**hort-term **to** **L**ong-term network **S2LNet**. S2LNet first decomposes each input series into seasonal and trend items, followed by a patching process which divides them into short-term subsequences. S2LNet leverages the fitting ability of the Transformer for seasonal features and the generalization of Linear for trend features. Finally, these short-term seasonal and trend features are fused into long-term dependencies via a Long-term Fusion module and used to model long-term cointegration with the Transformer. For these short-term components, S2LNet not only separately extracts their temporal-wise features, but also reconstruct a combined stationary distribution and further model their long-term cointegrated channel-wise dependencies.

Our contributions are as follows: (1) We propose **S2LNet**, a **S**hort-term **to** **L**ong-term network that addresses the limitations of existing models in capturing both cross-time and cross-channel dependencies; (2) S2LNet introduces the Long-term Fusion module, which combines short-term non-stationary features into a long-term stationary one, boosting the ability combines short-term seasonal and trend components into long-term features, enhancing the effectiveness of channel-wise cointegration modeling; (3) S2LNet achieves state-of-the-art performance across a wide range of baselines on multiple real-world datasets. S2LNet presents stable advantages over other 8 state-of-the-art models, ranking top-1 in 20 out of the 32 settings with various datasets, prediction lengths and metrics.

2 Time Series Forecasting Via Transformer

In the realm of multivariate time series forecasting (MTSF), Recently, the Transformer has seen widespread adoption in MTSF due to the long-range ability of its multi-head self-attention (MHSA) mechanism. MHSA parallelly calculates the attention score for each head and concatenate weighed outputs from all heads to form the final output. For a set of N_P tokens $\boldsymbol{x} \in \mathbb{R}^{N_P \times P}$ with a feature dimension of P, the attention is calculated as follows:

$$\boldsymbol{Q} = \mathrm{Linear}(\boldsymbol{x}), \boldsymbol{K} = \mathrm{Linear}(\boldsymbol{x}), \boldsymbol{V} = \mathrm{Linear}(\boldsymbol{x}),$$
$$\mathrm{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\top}}{\sqrt{d_k}}\right)\boldsymbol{V},$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ represent the query, key, and value matrices projected by the respective linear layer, and d_k is the dimension of the key. Through this process, weights are dynamically reassigned to each token, allowing the model to capture how each element in a sequence relates to other parts.

To further illustrate, we reuse the previously defined notation $\boldsymbol{x} \in \mathbb{R}^{N_P \times P}$ as an example. In cross-time modeling, N_P represents the number of divided subsequences for each time series, with C different channels treated as independent samples. In cross-channel modeling, N_P often refers to the number of channels, with subsequences treated as independent period samples. It is worth noting that the number of period samples is greater than 1 in short-term channel-wise modeling [15], while set to 1 in long-term modeling [14].

3 S2LNet: Short-term to Long-term network

3.1 The overall framework

As shown in Figure 2, S2LNet has three main components: input preprocessing, forecasting model, and output postprocessing. In the preprocessing stage, S2LNet first applies z-score standardization to align the global distribution of different sequences and then decompose the normalized input into the stationary seasonal item $\boldsymbol{I}_s$ and the non-stationary trend item $\boldsymbol{I}_t$. In the forecasting model, $\boldsymbol{I}_s$ and $\boldsymbol{I}_t$ will be divided into subsequences for local semantic information aggregation through the Patching operation. These partitioned items are further fed into different modules for cross-time extraction.

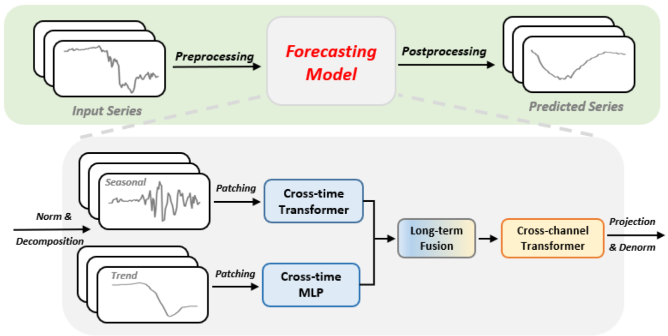

Fig. 2. The pipeline of S2LNet. (1)In Preprocessing, each input series will be normalized to align distribution and decomposed into seasonal and trend items; (2)The Forecasting Model captures cross-time and cross-channel features of two items the two decomposed items and then predicts future series; (3)In Postprocessing, denormalization restores the distribution, producing a final predicted series.

Afterward, these outputs are fused by a Long-term Fusion module to obtain long-term features. Following this, S2LNet captures long-term cointegrations by the Cross-channel Transformer, and its outputs will be projected to a future series by a linear layer. Finally, in the postprocessing stage, de-normalization is applied to restore the global distribution of the output series.

3.2 Input Preprocessing

In the preprocessing stage, we first apply z-score normalization to normalize the input sequence. Specifically, given a channel of, the normalized output $\hat{\boldsymbol{x}}$ is defined as follows:

$$\mu = \frac{1}{I}\sum_{k=1}^{I} x_k, \sigma^2 = \frac{1}{I}\sum_{k=1}^{I} (x_k - \mu)^2, \hat{\boldsymbol{x}} = \frac{x-\mu}{\sigma},$$

where μ and σ are the mean and standard deviation of $\boldsymbol{x}$, respectively. The counted global distribution statistics of different channels and sample sequences are used for aligning their distribution.

Later, as demonstrated in existing works [3, 4], we leverage the seasonal-trend decomposition to facilitate cross-time modeling. Similarly, a decomposition is employed to extract the mean variations within the input sequence as a trend item (I_t). Then, $\hat{\boldsymbol{x}}$ is decomposed into a relative stationary seasonal item (I_s) via subtracting I_t. The process is as follows:

$$\boldsymbol{I}_t = \mathrm{AvgPool}(\mathrm{Padding}((\hat{\boldsymbol{x}})), \boldsymbol{I}_s = \hat{\boldsymbol{x}} - \boldsymbol{I}_t,$$

where AvgPool is used to obtain the trend item $\boldsymbol{I}_t \in \mathbb{R}^I$ with an average kernel, with a kernel size of 25 and a stride of 1. Padding is applied using terminal values to ensure a consistent length of all the input sequences.

3.3 Forecasting Model

Patching aggregates adjacent time steps into cohesive patch-based tokens, which can effectively aggregate local semantic information and promote subsequent dependencies modeling. Notably, to further enhance the stationary of the trend item to improve predictive ability [9] and facilitate subsequent modeling [16], the following operations are conducted:

Notably, before patching, re-normalization and sliding aggregation [18] are additionally applied to I_t to further enhance the stationarity of the trend item and to smooth away outlier values:

$$\bar{I}_t = \frac{1}{I}\sum_{k=1}^{I} I_{t_k}, \hat{\boldsymbol{I}}_t = \boldsymbol{I}_t - \bar{I}_t \quad \text{(re-normalization)}$$

$$\hat{\boldsymbol{I}}_t = \mathrm{DWC}(\hat{\boldsymbol{I}}_t) + \hat{\boldsymbol{I}}_t \quad \text{(sliding aggregation)}$$

Here, $\bar{I}_t$ represents the mean of $\boldsymbol{I}_t$, and DWC refers to a depth-wise 1D convolution. Afterward, the patching step divides $\boldsymbol{I}_s$ and $\hat{\boldsymbol{I}}_t$ into two sets of N_P subsequences with length P. This process is equivalent to transforming the shape of $\boldsymbol{I}_s$ and $\hat{\boldsymbol{I}}_t$ and from $\mathbb{R}^I$ into $\mathbb{R}^{N_P \times P}$. Finally, the patched matrices of $\boldsymbol{I}_s$ and $\hat{\boldsymbol{I}}_t$ are mapped to latent space $\mathbb{R}^{N_P \times d_1}$ and $\mathbb{R}^{N_P \times d_2}$ respectively, where d_1 and d_2 are the last dimensions of the output matrices.

Cross-time Transformer aims at capturing the cross-time seasonal features $\boldsymbol{z}_s \in \mathbb{R}^{N_p \times d_1}$. Due to the stable distribution within the seasonal item, a stack of Transformer layers is ideally suited for fitting robust seasonal features. The modeling process of each layer is defined in the following equations:

$$\boldsymbol{z}_s = \mathrm{LayerNorm}\big(\boldsymbol{z}_s + \mathrm{MHSA}(\boldsymbol{z}_s, \boldsymbol{z}_s, \boldsymbol{z}_s)\big),$$
$$\boldsymbol{z}_s = \mathrm{LayerNorm}\big(\boldsymbol{z}_s + \mathrm{FFN}(\boldsymbol{z}_s)\big),$$

where $\boldsymbol{z}_s$ is initially set to $\boldsymbol{I}_s$, LayerNorm refers to layer normalization, FFN stands for feed-forward network, and MHSA represents temporal-wise multi-head self-attention. These components are generally adopted to form the Transformer layer. We use two Transformer layers in this module.

Cross-time MLP is used to model the cross-time dependencies of the trend item $\hat{\boldsymbol{I}}_t \in \mathbb{R}^{N_p \times d_2}$. For a less stationary trend item $\hat{\boldsymbol{I}}_t$, a simple linear module can provide strong generalization capability. The process is as follows:

$$\boldsymbol{z}_t = \hat{\boldsymbol{I}}_t + \mathrm{FFN}(\hat{\boldsymbol{I}}_t),$$

where FFN stands for a two-layer feed-forward network. Similar to the Cross-time Transformer, the shape of the trend feature $\boldsymbol{z}_t$ is kept as the same of the original trend item $\hat{\boldsymbol{I}}_t$.

$$\boldsymbol{z}_t = \hat{\boldsymbol{I}}_t + \mathrm{FFN}(\hat{\boldsymbol{I}}_t)_{\times l_2},$$

where l_2 (default l_2=1) layers of FFN are stacked for cross-time modeling, resulting in trend features $\boldsymbol{z}_t \in \mathbb{R}^{N_p \times d_1}$.

Long-term Fusion is used to restore the decomposed seasonal and trend items, resulting in long-term features with original mean variations within a sequence. For each univariate sequence, the modeled seasonal and trend features are fused via the following operations:

$$\boldsymbol{z} = \mathrm{Linear}(\boldsymbol{z}_s) + \mathrm{Linear}(\boldsymbol{z}_t) + \bar{I}_t,$$
$$\boldsymbol{z}^l = \mathrm{Linear}(\boldsymbol{z}),$$

where $\boldsymbol{z}_s$ and $\boldsymbol{z}_t$ are aligned to same shape, then integrated to $\boldsymbol{z} \in \mathbb{R}^{N_p \times D}$ by addition. Later, $\boldsymbol{z}$ is aggregated along the temporal dimension to produce long-term relationships $\boldsymbol{z}^l \in \mathbb{R}^{\times D}$, which compresses information of the entire sequence.

Cross-channel Transformer models the Long-term Fusion outputs of non-stationary period features for cross-channel correlations. In contrast, existing works either overly focus on short-term channel dependencies or directly disregard channel-wise mutual information gains. Given a multivariate series $\boldsymbol{X} \in \mathbb{R}^{C \times I}$, after all the above cross-time modules, the long-term features$\boldsymbol{z}^l$ of C channels will be mixed to form the multi-channel long term features $\boldsymbol{Z} \in \mathbb{R}^{C \times D}$. The cross-channel correlation will be obtained and modeled as follows:

$$\boldsymbol{Z}_c = \mathrm{LayerNorm}\big(\boldsymbol{Z} + \mathrm{MHSA}(\boldsymbol{Z}, \boldsymbol{Z}, \boldsymbol{Z})\big),$$
$$\boldsymbol{Z}_c = \mathrm{LayerNorm}\big(\boldsymbol{Z}_c + \mathrm{FFN}(\boldsymbol{Z}_c)\big),$$

where $\boldsymbol{Z}_c \in \mathbb{R}^{C \times D}$ represents long-term cointegrated features of C channels, LayerNorm and FFN are similar to those mentioned in the Cross-time Transformer. Specifically, MHSA here is channel-wise multi-head self-attention.

Projection maps the intermediate features to a future sequence. Regarding the superior performance of encoder-only architectures [19], a linear layer is applied to project $\boldsymbol{Z}_c$ from $\mathbb{R}^{C \times D}$ to $\mathbb{R}^{C \times O}$, resulting in a output sequence $\overline{\boldsymbol{Y}}$.

3.4 Output Postprocessing

In the postprocessing stage, reversed z-score normalization is applied to each channel sequence based on its distribution statistics. Given a channel of sequence $\boldsymbol{x}$ and its predicted output $\overline{\boldsymbol{y}} \in \overline{\boldsymbol{Y}}$, the final predicted series $\widehat{\boldsymbol{y}}$ is generated by restoring the original distribution:

$$\widehat{\boldsymbol{y}} = \overline{\boldsymbol{y}} \cdot \sigma + \mu,$$

where μ and σ are the mean and standard deviation of $\boldsymbol{x}$ respectively, counted in the input preprocessing.

4 Performance Evaluation

4.1 Experimental Setup

Setups. Our experiments are mainly conducted on extensively used real-world datasets open-sourced in previous works [3, 12]: ETT-small (ETTh1), Weather, Solar-Energy, Electricity, with the. For a comprehensive evaluation, we select a range of representative baselines in time series forecasting, including the following categories: (1) Transformer-based models: iTransformer [14], Crossformer [15], Autoformer [3], and PatchTST [10]; (2) CNN-based model: TimesNet [12] and SCINet [13]; and (3) Linear models: DLinear [20] and TiDE [21].

Setting. In our training configuration, we primarily adhere to the settings from [14]. For our architecture, we set $P = 24, N_p = [\frac{I}{24}], d_1 = 96$, $d_2 = D = 512$. We utilize Mean Square Error (MSE) and Mean Absolute Error (MAE) as performance metrics, aligning with previous methods [3, 22].

4.2 Main Results

As shown in Table 1, S2LNet consistently demonstrates the best or second-best performance across various datasets and prediction ranges. Specifically, S2LNet outperforms channel-independent approaches such as PatchTST and DLinear, as well as traditional channel-dependent transformers such as iTransformer, Crossformer, and Autoformer. Moreover, compared to the most state-of-the-art method, iTransformer, our approach also shows considerable advantages, with the Count of best performance being four times those of iTransformer.

4.3 Channel-wise Attentions

To validate the feasibility of modeling long-term channel dependencies, we visualize the channel-wise attention maps on the Weather dataset in Figure 3, where the Long-term map is obtained using our full model with the Long-term Fusion module and the Short-term map is obtained without it. The two maps share significant commonalities, while the attention maps for long-term channel dependencies are typically less blurred and clearer, highlighting the effectiveness of our **S**hort-term **to L**ong-term mechanism.

Moreover, the quantitative comparison is shown in Table 2. Cross-channel correlations between long-term sequences outperform the short-term variant. This result is reasonable because short-term cross-channel correlations are highly susceptible to distribution shifts, noise, and other short-term fluctuations, leading to severe spurious regression issues [23]. In contrast, long-term cross-channel correlations can better avoid these fluctuations and effectively capture cointegration, which typically exists between a long-term temporal span of channels [17].

Table 1. Results for four prediction lengths $O \in \{96,192,336,720\}$, with lower MSE and MAE indicating better performance. All baselines use an input length of $I = 96$. We highlight the best results in **bold** and the second-best with <u>underline</u>. The last row (Count) shows the times each method achieves the best results.

Models		S2Lnet (Ours)		iTransformer [14]		Crossformer [15]		Autoformer [3]		PatchTST [10]		TimesNet [12]		SCINet [13]		DLinear [20]		TiDE [21]	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.383**	<u>0.401</u>	0.386	0.405	0.423	0.448	0.449	0.459	0.414	0.419	<u>0.384</u>	0.402	0.654	0.599	0.386	**0.4**	0.479	0.464
	192	**0.434**	<u>0.43</u>	0.441	0.436	0.471	0.474	0.5	0.482	0.46	0.445	<u>0.436</u>	**0.429**	0.719	0.631	0.437	0.432	0.525	0.492
	336	**0.475**	**0.453**	0.487	0.458	0.57	0.546	0.521	0.496	0.501	0.466	0.491	0.469	0.778	0.659	0.565	0.515	<u>0.481</u>	<u>0.459</u>
	720	**0.468**	**0.471**	0.503	0.491	0.653	0.621	0.514	0.512	<u>0.5</u>	<u>0.488</u>	0.521	0.5	0.836	0.699	0.519	0.516	0.594	0.558
Weather	96	<u>0.17</u>	**0.211**	0.174	<u>0.214</u>	**0.158**	0.23	0.266	0.336	0.177	0.218	0.172	0.22	0.221	0.306	0.196	0.255	0.202	0.261
	192	<u>0.216</u>	**0.254**	0.221	**0.254**	**0.206**	0.277	0.307	0.367	0.225	0.259	0.219	0.261	0.261	0.34	0.237	0.296	0.242	0.298
	336	<u>0.273</u>	**0.295**	0.278	<u>0.296</u>	**0.272**	0.335	0.359	0.395	0.278	0.297	0.28	0.306	0.309	0.378	0.283	0.335	0.287	0.335
	720	0.353	<u>0.348</u>	0.358	**0.347**	0.398	0.418	0.419	0.428	0.354	<u>0.348</u>	0.365	0.359	0.377	0.427	**0.345**	0.381	<u>0.351</u>	0.386
Solar-Energy	96	**0.198**	<u>0.238</u>	<u>0.203</u>	**0.237**	0.31	0.331	0.884	0.711	0.234	0.286	0.25	0.292	0.237	0.344	0.29	0.378	0.312	0.399
	192	**0.232**	<u>0.266</u>	<u>0.233</u>	**0.261**	0.734	0.725	0.834	0.692	0.267	0.31	0.296	0.318	0.28	0.38	0.32	0.398	0.339	0.416
	336	**0.244**	<u>0.277</u>	<u>0.248</u>	**0.273**	0.75	0.735	0.941	0.723	0.29	0.315	0.319	0.33	0.304	0.389	0.353	0.415	0.368	0.43
	720	**0.247**	<u>0.281</u>	<u>0.249</u>	**0.275**	0.769	0.765	0.882	0.717	0.289	0.317	0.338	0.337	0.308	0.388	0.356	0.413	0.37	0.425
Electricity	96	**0.139**	**0.233**	<u>0.148</u>	<u>0.24</u>	0.219	0.314	0.201	0.317	0.181	0.27	0.168	0.272	0.247	0.345	0.197	0.282	0.237	0.329
	192	**0.155**	**0.248**	<u>0.162</u>	<u>0.253</u>	0.231	0.322	0.222	0.334	0.188	0.274	0.184	0.289	0.257	0.355	0.196	0.285	0.236	0.33
	336	**0.17**	**0.265**	<u>0.178</u>	<u>0.269</u>	0.246	0.337	0.231	0.338	0.204	0.293	0.198	0.3	0.269	0.369	0.209	0.301	0.249	0.344
	720	0.226	**0.312**	<u>0.225</u>	<u>0.317</u>	0.28	0.363	0.254	0.361	0.246	0.363	0.22	0.32	0.299	0.39	0.245	0.333	0.284	0.373
Count		**20**		<u>5</u>		3		0		0		2		0		2		0	

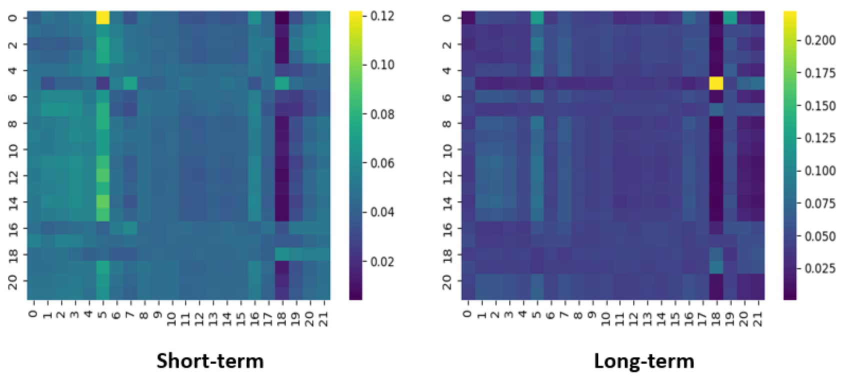

Fig. 3. Visualization of attention maps from the channel-wise Transformer on the Weather dataset, showcasing the channel correlations between Short-term period and Long-term series. The input length I and the prediction length O are fixed to 96.

Table 2. Ablation of Short-term period and Long-term series on Weather, Solar-Energy, and Electricity datasets, **bold** notes better results. The input length I is fixed to 96.

Datasets		Weather				Solar-Enegry				Electricity			
Type	Metric	96	192	336	720	96	192	336	720	96	192	336	720
Long-term	MSE	**0.170**	**0.216**	**0.273**	**0.353**	**0.198**	0.232	**0.244**	**0.247**	**0.139**	**0.155**	**0.170**	0.226
	MAE	**0.211**	**0.254**	**0.295**	**0.348**	**0.238**	**0.266**	**0.277**	**0.281**	**0.233**	**0.248**	**0.265**	**0.312**
Short-term	MSE	0.173	0.220	0.275	0.359	0.201	**0.230**	0.246	0.248	0.144	0.164	0.177	**0.219**
	MAE	0.216	0.258	0.298	0.350	0.239	**0.266**	0.280	0.284	0.237	0.252	0.268	0.316

5 Conclusion

In this paper, we propose S2LNet, a novel approach for long-term time series forecasting that effectively models both cross-time and cross-channel dependencies. S2LNet tackles the challenges of distribution shift posed by the non-stationary nature of real-world time series. Concretely, S2LNet decomposes input sequences into seasonal and trend components, utilizing the Transformer for seasonal features and the MLP for trend features. These features are then fed to a Long-term Fusion module and a cross-channel Transformer for capturing long-term integrated relationships. Experimental results on four real-world datasets demonstrate the ability of S2LNet to outperform state-of-the-art baselines.

References

1. Angryk, R.A., Martens, P.C., Aydin, B., Kempton, D., Mahajan, S.S., Basodi, S., Ahmadzadeh, A., Cai, X., Filali Boubrahimi, S., Hamdi, S.M., Others: Multivariate time series dataset for space weather data analytics. Scientific data 7, 227 (2020)
2. Deb, C., Zhang, F., Yang, J., Lee, S.E., Shah, K.W.: A review on time series forecasting techniques for building energy consumption. Renewable and Sustainable Energy Reviews 74, 902-924 (2017)
3. Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. Advances in Neural Information Processing Systems, (2021)
4. Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R.: FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. International Conference on Machine Learning, pp. 27268-27286 (2022)
5. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Others: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010. 11929 (2020)
6. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems 30, (2017)
7. Liu, Z., Cheng, M., Li, Z., Huang, Z., Liu, Q., Xie, Y., Chen, E.: Adaptive Normalization for Non-stationary Time Series Forecasting: A Temporal Slice Perspective. Thirty-seventh Conference on Neural Information Processing Systems, (2023)
8. Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.-H., Choo, J.: Reversible Instance Normalization for Accurate Time-Series Forecasting against Distribution Shift. International Conference on Learning Representations, (2021)
9. Liu, Y., Wu, H., Wang, J., Long, M.: Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting. (2022)

10.Nie, Y., H. Nguyen, N., Sinthong, P., Kalagnanam, J.: A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. International Conference on Learning Representations, (2023)
11.Dai, T., Wu, B., Liu, P., Li, N., Bao, J., Jiang, Y., Xia, S.-T.: Periodicity decoupling framework for long-term series forecasting. The Twelfth International Conference on Learning Representations, (2024)
12.Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M.: TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. International Conference on Learning Representations, (2023)
13.Liu, M., Zeng, A., Chen, M., Xu, Z., Lai, Q., Ma, L., Xu, Q.: Scinet: Time series modeling and forecasting with sample convolution and interaction. Advances in Neural Information Processing Systems 35, 5816-5828 (2022)
14.Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., Long, M.: iTransformer: Inverted transformers are effective for time series forecasting. arXiv preprint arXiv:2310. 06625 (2023)
15.Zhang, Y., Yan, J.: Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. International Conference on Learning Representations, (2023)
16.Wang, Y., Wu, H., Dong, J., Liu, Y., Qiu, Y., Zhang, H., Wang, J., Long, M.: Timexer: Empowering transformers for time series forecasting with exogenous variables. arXiv preprint arXiv:2402. 19072 (2024)
17.Perman, R.: Cointegration: an introduction to the literature. Journal of Economic Studies 18, (1991)
18.Lin, S., Lin, W., Wu, W., Chen, H., Yang, J.: SparseTSF: Modeling Long-term Time Series Forecasting with 1k Parameters. arXiv preprint arXiv:2405. 00946 (2024)
19.Lin, S., Lin, W., Wu, W., Wang, S., Wang, Y.: Petformer: Long-term time series forecasting via placeholder-enhanced transformer. arXiv preprint arXiv:2308. 04791 (2023)
20.Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? Proceedings of the AAAI conference on artificial intelligence, vol. 37, pp. 11121-11128 (2023)
21.Das, A., Kong, W., Leach, A., Sen, R., Yu, R.: Long-term Forecasting with TiDE: Time-series Dense Encoder. arXiv preprint arXiv:2304. 08424 (2023)
22.Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. Proceedings of the AAAI conference on artificial intelligence, vol. 35, pp. 11106-11115 (2021)
23.Granger, C.W.J., Newbold, P.: Spurious regressions in econometrics. Journal of econometrics 2, 111-120 (1974)