Two Birds with One Stone: Series Saliency for Accurate and Interpretable Multivariate Time Series Forecasting

Qingyi Pan[1] , **Wenbo Hu**[2] and **Ning Chen**[1*]
[1]High Performance Computing Group, Dept. of Comp. Sci. and Tech., BNRist Center, Institute for AI, Tsinghua-Bosch Joint ML Center, THBI Lab, Tsinghua University, Beijing, 100084 China
[2]RealAI
panqingyi19@gmail.com, i@wbhu.net, ningchen@mail.tsinghua.edu.cn

Abstract

It is important yet challenging to perform accurate and interpretable time series forecasting. Though deep learning methods can boost the forecasting accuracy, they often sacrifice interpretability. In this paper, we present a new scheme of series saliency to boost both accuracy and interpretability. By extracting series images from sliding windows of the time series, we design series saliency as a mixup strategy with a learnable mask between the series images and their perturbed versions. Series saliency is model agnostic and performs as an adaptive data augmentation method for training deep models. Moreover, by slightly changing the objective, we optimize series saliency to find a mask for interpretable forecasting in both feature and time dimensions. Experimental results on several real datasets demonstrate that series saliency is effective to produce accurate time-series forecasting results as well as generate temporal interpretations.

1 Introduction

Time series forecasting is an important task with wide applications. Traditional parametric models often have a shallow architecture (e.g., [Box and Jenkins, 1976; Harvey, 1990]). By adopting some explicit assumptions, such methods are easy-to-interpret, but their predictive capabilities are often limited. Deep architectures have become increasingly popular for time-series [Gamboa, 2017], including recurrent neural networks (RNN), nonlinear autoregressive exogenous neural network (NARX) [Chen *et al.*, 1990], long-short term memory (LSTM) [Hochreiter and Schmidhuber, 1997], gated recurrent unit (GRU) [Chung *et al.*, 2014] and neural attention methods. Though effective in improving forecasting accuracy, deep models are hard to interpret the outputs [Castelvecchi, 2016], which may hinder their applications to high stakes applications (e.g., healthcare) where reliable interpretation is crucial. Though much progress has been made on interpreting deep visual or language models [Samek *et al.*, 2019], it is relatively unexplored to develop both accurate and interpretable methods for multivariate time series forecasting, where the 2D time-feature format imposes new challenges.

Existing work often considers either the time or feature domain, or treats them separately via a two-stage method. For example, some attempts have been made to apply the interpretation methods for general neural networks, such as LIME [Ribeiro *et al.*, 2016], DeepLift [Shrikumar *et al.*, 2017] and Shap [Lundberg and Lee, 2017]. They use gradient information to extract feature information for single-time forecasts after the back-propagation training, thereby ignoring the crucial temporal information and insufficient for forecasting interpretation [Mitrea *et al.*, 2009]. Another type of solutions transfers the attention methods from the fields of language or vision [Bahdanau *et al.*, 2014; Vaswani *et al.*, 2017; Assaf *et al.*, 2019; Shih *et al.*, 2019]. However, the attention values for explaining RNNs or CNNs are calculated via the relative importance of the different time steps and there are concerns that they are based on the intermediate feature importance instead of model interpretations [Serrano and Smith, 2019]. More recently, Ismail et al. [Ismail *et al.*, 2020] develop a two-stage saliency approach that decouples the time dimension and feature dimension, thereby may lead to sub-optimal solutions.

In this work, we present a new strategy of *series saliency* to boost both forecasting accuracy and interpretability of deep time series models, by considering the time and feature dimensions in a coherent manner. As shown in Fig. 1, we consider multivariate time series as a set of *window* $\times$ *feature series images*, and design series saliency as a masked mixup between series images and their perturbed versions, where the mask is a learnable matrix. Series saliency is model agnostic and can be used as an effective data augmentation method to boost the accuracy of deep forecasting models, where the augmentation strategy is learnable and adaptive, thereby different from the common augmentation methods (e.g., [Iwana and Uchida, 2020]) that typically apply some pre-fixed operations on a given training set. Furthermore, by simply changing the objective function, we can optimize the series saliency module to find a mask (i.e., heatmap) that identifies important regions for forecasting, thereby boosting interpretability. We present both quantitative and qualitative results on several typical time series datasets, which show that our method achieves better (or comparable) forecasting results and meanwhile provides temporal interpretations for the forecasts.

*Contact Author

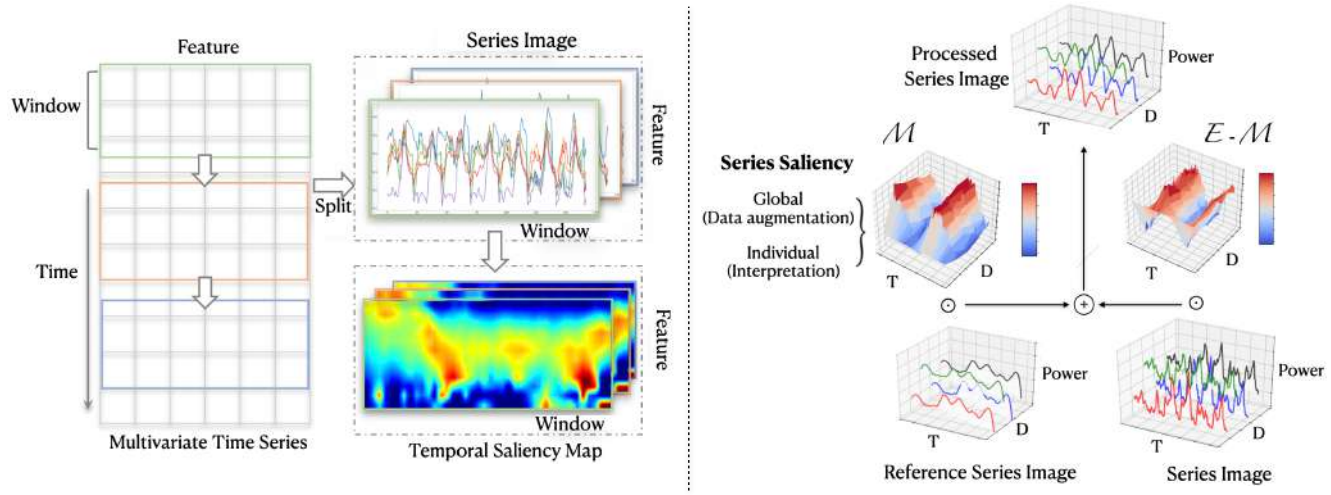

Figure 1: (Left): multivariate time series and the corresponding temporal saliency map, where we extract *series images* from the multivariate time series, and each series image is associated with a temporal saliency map to identify the informative features for forecasting; (Right): the proposed series saliency module, where for each series image we perturb the original series to obtain the reference series image, and the series image and its perturbed version are mixed up by the series saliency map.

2 Methods

We now present our method in detail, which consists of a series saliency module and its use in both training and interpretation phases.

2.1 Setup and Notations

As shown in Fig. 1, multivariate time series data are spatio-temporal with two dimensions – time and feature. Formally, we use S to denote a series of observed time series signal:

$$S = [s_1, s_2, \cdots, s_t, \cdots], \tag{1}$$

where $s_t \in \mathbb{R}^D$ denotes the feature vector with dimension D at time t. For every time step t, we aim to predict the future value $s_{t+\tau}$ after a given horizon τ. The horizon is chosen according to the forecasting task settings. For example, for the traffic usage, the horizon τ of interest ranges from an hour to a day; while for the stock market data, even a second or minute ahead forecasting can be meaningful for generating returns. Besides forecasting accuracy, we are also interested in interpretation: *which features contribute most and which contributes least for the final forecast predictions?*

As stated above, traditional statistical methods (e.g., [Box and Jenkins, 1976] and [Harvey, 1990]) are easy to interpret by making some explicit model assumptions that can extract interpretations directly from the learned model parameters, while they are often limited in forecasting accuracy. In contrast, recent progress on deep learning methods leads to superior prediction capabilities [Goodfellow *et al.*, 2016], however, they are hard to interpret since the deep model assumptions are stacked with multiple non-linear activation or blocks.

A key part of an effective model for multivariate time series forecasting would be the capability on handling the information from both time and feature dimensions in a coherent manner. Recent work develops an attention-based scheme [Assaf *et al.*, 2019; Shih *et al.*, 2019], which introduces an attention map to "selectively" combine time-feature information (see Fig. 2 (left)), with the primary focus on interpretation. However, as pointed out in [Ismail *et al.*, 2020], the attention-based methods can be insufficient for interpreting multivariate time-series data. We develop a new scheme of *series saliency*, which is model-agnostic and can boost both forecasting accuracy and interpretation, as detailed below.

2.2 Series Saliency

We develop series saliency by drawing inspirations from the saliency maps [Dabkowski and Gal, 2017] in computer vision. However, unlike previous work on saliency maps that mainly focuses on interpetability of deep models, series saliency is beneficial for improving both forecasting accurarcy and interpretation for time-series data.

Specifically, to consider the time-feature information jointly, we first represent the multivariate time series as a set of 2D *series images*. As shown in Fig. 1, each series image corresponds to a part of the multivariate time series within a given time window. Formally, let T be the window size. We simply set the value of T for various datasets with 2 periodic patterns (e.g., $p = 48$ for hourly electricity consumption). A series image is represented as a matrix $X \in \mathbb{R}^{D \times T}$, of which each row corresponds to one feature dimension in the multivariate time series. Then, we follow the perturbation strategy in the smallest destroying region (SDR) principle [Dabkowski and Gal, 2017] to design the series saliency scheme. We define a reference series image $\hat{X}$ by adding noise or Gaussian blur on each element of the original series image X:

$$\hat{x}_{t,i} = \begin{cases} x_{t,i} + \epsilon_{\sigma_1} & \text{noise} \\ g_{\sigma_2}(x_{t,i}) & \text{blur} \end{cases}, \tag{2}$$

where $\epsilon_{\sigma_1} \sim \mathcal{N}(\mu, \sigma_1^2)$ is a Gaussian noise and g_{σ_2} is a Gaussian blur kernel on element $x_{t,i}$ with the maximum isotropic standard deviation σ_2.

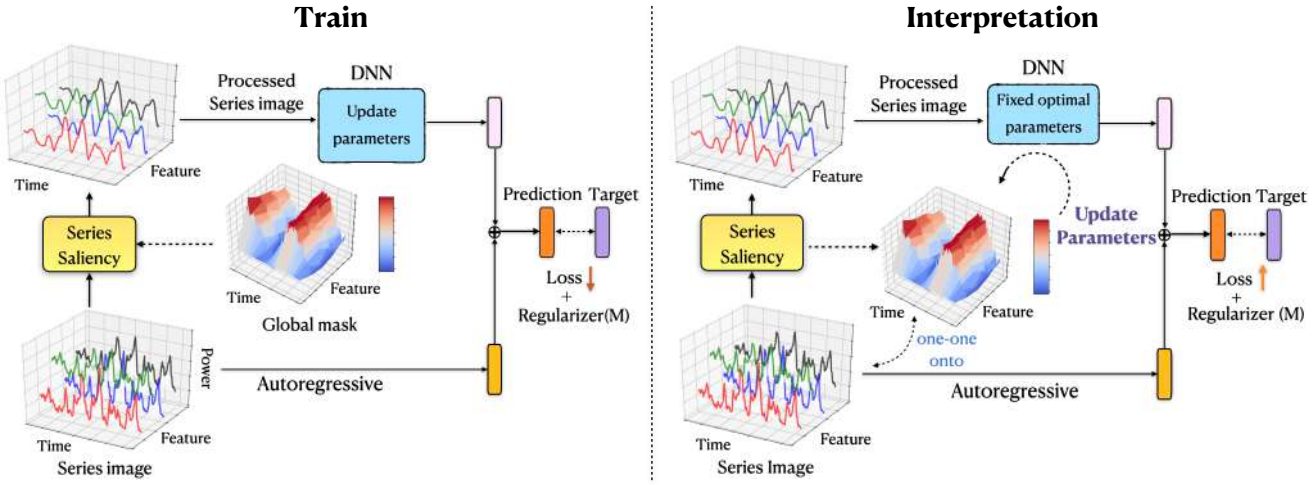

Figure 2: The training and interpretation phases with series saliency. Best viewed in color.

In time series data, each series can be composed of three parts — trend, seasonality and residual [Hyndman and Athanasopoulos, 2018]. As shown in [Hyndman and Athanasopoulos, 2018], blurring is helpful to extract the trend information of a time series, and adding some noise can enhance the local information. When the amount of injected noise or blurring is small, the reference series images can be treated as data augmentation in the time domain [Iwana and Uchida, 2020] to learn deep models. However, if the noise (or blurring) is not set properly (e.g., too large), the noise (or blurring) will introduce irregular roughness to cover the original series, making it difficult for DNNs to learn temporal patterns in reference data.

To relieve the sensitivity over noise (or blurring), the series saliency module further introduces a learnable mask $M \in [0,1]^{D\times T}$ and selectively combines the reference series image and the original one:

$$\widetilde{X} = M \odot \hat{X} + (E - M) \odot X, \tag{3}$$

where E is the matrix with all elements be the unit one. As illustrated in Fig. 1, in such a design, the series saliency module can generate data (i.e., $\widetilde{X}$) that cover the unexplored input space while maintaining the important characteristics of the original series image (i.e., X). As we shall see, series saliency is an effective data augmentation strategy for training deep models and it is learnable.

Besides augmenting the data for improving training, series saliency can also be used for interpretation by simply changing the objective to be optimized. We defer the details to Section 2.4 after presenting the training procedure.

2.3 Training with Series Saliency

With the series saliency module, we now present the entire model architecture and its training details. One essential problem in deep learning models is that the scale of the outputs is not sensitive to the scale of inputs. In specific real time series datasets, the scale of input data often changes in a non-periodic manner, which can significantly lower the forecasting accuracy of deep learning models. We propose to decompose the final prediction into a linear part, which primarily focuses on the local scaling issue, plus a non-linear part containing complex temporal patterns. Fig. 2 (left) shows the dual-path architecture — a nonlinear deep learning (DL) module and a linear auto-regressive (AR) module.

The original series images X are converted to the processed versions $\widetilde{X}$ by our proposed module, and then $\widetilde{X}$ are fed into the DL module to obtain $y^{(r)}$. Here, the DL module can be any time series network, e.g., convolution neural networks, recurrent neural networks, Long short-term memory (LSTM) [Hochreiter and Schmidhuber, 1997], long- and short-term temporal pattern neural network (LSTNet) [Lai *et al.*, 2018] and Self-Attention encoder [Vaswani *et al.*, 2017] (see experiments for details). For the linear part, we choose an auto-regressive model and denote its result as $y^{(o)}$:

$$y^{(o)} = \sum_{k=0}^{p} W_k^\top y_{t-k} + b, \tag{4}$$

where p is the order of the AR model, $W_k \in R^D$ and $b \in R$ are its coefficients. p can be determined for various datasets with periodic patterns (e.g., $p = 24$ for the hourly electricity consumption). In our experiments, we empirically searched a good p through validation. The final forecasting is obtained by combining the outputs of both DL and AR:

$$\hat{y} = y^{(o)} + y^{(r)}. \tag{5}$$

Let N be the number of series images in the training set and $\phi = \{\theta, W, b\}$ denote all the unknown parameters. For notation simplicity, we will use y_i to denote the ground-truth forecasting result of X_i at a particular horizon. We adopt stochastic gradient descent (SGD) to jointly optimize over ϕ and M. Specifically, at each iteration, we draw a mini-batch $B = \{(X_i, y_i)\}_{i=1}^n$ with size n. Our loss function $\ell(\phi, M)$ consists of two parts. First, for each $(X_i, y_i) \in B$, we follow the diagram in Fig. 2 to get the prediction $\hat{y}_i$ (a function of both ϕ and M) via Eq. (5), and accumulate the squared error

$\|\hat{y}_i - y_i\|^2$ into the loss $\ell(\phi, M)$. Second, to fully leverage the training set, we also send the original series image X_i into both the DNN and AR model (i.e., bypassing the series saliency module in Fig. 2 to get another prediction $\hat{y}'_i$ (a function of ϕ only) via Eq. (5), and accumulate the squared error $\|\hat{y}'_i - y_i\|^2$ into the loss $\ell(\phi, M)$. Our overall training objective is to minimize a regularized version of loss $\ell(\phi, M)$:

$$\mathcal{L}_1(\phi, M) = \ell(\phi, M) + \lambda_1 \ell_m(M) + \lambda_2 \ell_r(M), \quad (6)$$

where $\ell_m(M)$ and $\ell_r(M)$ are regularization terms on the mask M, λ_1 and λ_2 are the corresponding coefficients.

The effect of the regularizer $\ell_m(M)$ is to control the amount of perturbation. As stated before, if M is close to E, the reference series image will dominate and may degenerate the performance. To discourage such degenerating case, we choose the common 2-norm, i.e., $\ell_m(M) = \|M\|_2$.

The design of $\ell_r(M)$ is a bit more subtle. Note that a key difference exists between our series images and the natural images in computer vision — the rows (i.e., features) of series images are permutable without affecting the content, while switching the rows of a nature image would destroy its semantic meaning. In our case, we would expect the feature importance matrix M to be row-wisely orthogonal. Therefore, we define the regularizer term $\ell_r(M)$:

$$\ell_r(M) = \|MM^\top - I\|_F, \quad (7)$$

where I is the identity matrix and $||\cdot||_F$ is the Frobenius norm. Such orthogonality is beneficial for interpretation, and can boost the training as well (see ablation study in Section 3.2).

With the overall objective $\mathcal{L}_1$, we use back-propagation to calculate its gradient and apply SGD to update (ϕ, M). We would like to highlight a key difference from the traditional data augmentation [Iwana and Uchida, 2020]. Unlike the traditional methods, which typically augment the data for once with some pre-fixed strategies (e.g., perturbation with a fixed noise), our augmentation strategy is adaptive by iteratively updating the mask M of the series saliency module. Such a scheme makes the model to better focus on informative parts that are good for forecasting, as we shall see in experiments.

2.4 Interpretation with Series Saliency

Now, we show that with the series saliency module, we can easily derive an interpretation strategy for forecasting.

Let X^* be the a series image in the testing set with ground-truth forecasting y^* at a particular horizon τ. Let ϕ^* denote the optimal parameters of the DNN and AR models after training. We follow the diagram in Fig. 2 to get the prediction $\hat{y}^*$ (a function of M) via Eq. (5), and calculate the squared error $\|\hat{y}^* - y^*\|^2$. The goal of interpretation is to find the most salient evidence that influences the model performance measured by square error. In series saliency, the most salient feature regions are found by identifying the highly representative mask, which summarizes compactly the effect of deleting feature regions, either setting values to zeros or Gaussian noise, to explain the behaviour of the DL part. For the AR part, the weights are easy-to-interpret because of its linearity [Hyndman and Athanasopoulos, 2018]. On the other hand, the AR part mainly assists DL models to address the local scaling issue, so it is not essential to focus on the explanation of the AR part. Formally, we formulate interpretation as minimizing the following objective:

$$L_2(M; X^*) = -\|\hat{y}^* - y^*\|^2 + \lambda_1 \ell_m(M) + \lambda_2 \ell_r(M), \quad (8)$$

where $\ell_m(M)$ and $\ell_r(M)$ are the same regularization terms as in training. Then, we use back-propagation to calculate the gradient of L_2 and apply SGD to update M **only**. After convergence, the optimal M provides an interpretation of important features for forecasting on instance X^*.

3 Experiments

We now present experimental results by comparing the performance of four widely-used deep learning models with and without the proposed series saliency module on various datasets. We also give an ablation study on the mask and AR model components. Then we show extensive qualitative results on how saliency heatmaps interpret the forecasts.

3.1 Experimental Setting

(1) Datasets: We use three time series datasets *electricity*, *Air-quality*, *Industry data* The three datasets are representative (from difficult to easy). **(2) Metrics:** We use two evaluation metrics, namely relative squared error (RSE) and empirical correlation coefficient (CORR). For RSE, lower values are better, while for CORR higher values are better. The datasets and the metrics have been widely used in many papers about time series forecasting (e.g., LSTNet, TPA-LSTM) [Lai *et al.*, 2018; Shih *et al.*, 2019]. **(3) Deep Learning Module:** We use four state-of-the-art architectures for comparison (i.e., CNN, GRU+Attention, LSTNet and Self-Attention encoder). These four deep learning models achieve superior forecasting performance and in the following results they are used as the deep learning module to get interpreted by the proposed series saliency method.

3.2 Results on Forecasting

Table 1 presents the CORR values of various methods on the three datasets with different forecasting horizons. We can see that using the series saliency module can boost the forecasting accuracy of various deep models on the *Air quality* dataset, especially when the forecasting horizon is large. This is because the model complexity increases with the horizon, and data augmentation may help explore more useful temporal information. We also observe that among all the deep architectures in comparison, the Self-Attention encoder with the series saliency gives the best performance in most settings, mainly because of the powerful representation capability of the transformer encoder model. Therefore, in the sequel, we will use the Self-Attention encoder as the main architecture to do further analysis.

Ablation Study

We now present an ablation study to demonstrate the effectiveness of our design components, including regularization terms and and the auto-regressive path in our architecture.

First, we examine how the auto-regressive module helps on the forecasting when series saliency (i.e., data augmentation) is used. We compare the following variants:

Methods	Air Quality			Industry			Electricity		
	$\tau = 3$	$\tau = 6$	$\tau = 12$	$\tau = 3$	$\tau = 6$	$\tau = 12$	$\tau = 3$	$\tau = 6$	$\tau = 12$
CNN	0.775 ± 0.003	0.701 ± 0.001	0.636 ± 0.001	0.919 ± 0.022	0.909 ± 0.019	0.841 ± 0.008	0.883 ± 0.004	0.871 ± 0.002	0.866 ± 0.004
GRU	0.804 ± 0.003	0.712 ± 0.002	0.639 ± 0.003	0.953 ± 0.003	0.936 ± 0.013	0.904 ± 0.011	0.878 ± 0.001	0.877 ± 0.003	0.867 ± 0.002
LSTNet	0.777 ± 0.001	0.708 ± 0.004	0.624 ± 0.004	0.949 ± 0.004	0.934 ± 0.003	0.876 ± 0.011	0.922 ± 0.004	0.913 ± 0.002	0.906 ± 0.002
SA	0.813 ± 0.002	0.722 ± 0.003	0.643 ± 0.003	0.961 ± 0.002	0.942 ± 0.005	0.905 ± 0.009	0.919 ± 0.007	0.907 ± 0.001	0.902 ± 0.003
CNN w/ SS	0.779 ± 0.005	0.723 ± 0.009	0.641 ± 0.007	0.941 ± 0.006	0.927 ± 0.004	0.881 ± 0.001	0.898 ± 0.004	0.893 ± 0.002	0.892 ± 0.007
GRU w/ SS	0.809 ± 0.003	0.716 ± 0.012	0.649 ± 0.003	0.955 ± 0.001	0.935 ± 0.002	0.912 ± 0.003	0.905 ± 0.004	0.889 ± 0.008	0.878 ± 0.003
LSTNet w/ SS	0.794 ± 0.008	0.724 ± 0.002	0.641 ± 0.003	0.959 ± 0.004	0.938 ± 0.001	0.901 ± 0.002	$\mathbf{0.928} \pm 0.003$	$\mathbf{0.918} \pm 0.003$	0.907 ± 0.001
SA w/ SS	$\mathbf{0.819} \pm 0.003$	$\mathbf{0.732} \pm 0.009$	$\mathbf{0.658} \pm 0.001$	$\mathbf{0.965} \pm 0.003$	$\mathbf{0.955} \pm 0.016$	$\mathbf{0.916} \pm 0.004$	0.923 ± 0.003	0.915 ± 0.001	$\mathbf{0.911} \pm 0.002$

Table 1: The CORR values of various methods on air quality, industry and electricity datasets when horizon $\tau = \{3, 6, 12\}$. Best performance in boldface. We report the mean and standard deviations in ten runs.

- **w/ SS**: The model in Fig. 2 with series saliency (SS).
- **w/o SS**: The model in Fig. 2, but without series saliency;
- **w/o AR**: The model in Fig. 2 with series saliency, but without the auto-regressive component.

Fig. 3 (left) shows the results. We do not use any perturbation in **w/o SS**. We can see that both the series saliency (i.e., adaptive data augmentation) and the auto-regressive modules are helpful to boost the performance — dropping either one would lead to degenerated performance.

Second, we do a finer investigation on the effects of the regularization terms in learning the series saliency module during training. We compare the following variants:

- **w/o SS**: The model in Fig. 2, but without the series saliency module (i.e., no data augmentation);
- **Fixed augmentation**: Augment the series images for once with some prefixed strategies (perturbation with a fixed noise) [Iwana and Uchida, 2020].
- **Type1**: The entire model in Fig. 2 trained by minimizing the loss only (i.e., without ℓ_r and ℓ_m).
- **Type2**: The entire model in Fig. 2 trained by minimizing the loss with ℓ_m regularizer (i.e., without ℓ_r).
- **w/ SS**: The entire model in Fig. 2 trained with both ℓ_r and ℓ_m regularizers as in Eq. (6).

Fig. 3 (right) shows the results. Again, we can see that series saliency can boost the forecasting results (i.e., higher CORR). Also, being an adaptive strategy, series saliency is more effective for training deep models than the pre-fixed data augmentation method [Iwana and Uchida, 2020]. The regularizer term ℓ_m has little effect on improving the model performance, while ℓ_r is much more influential. By considering the decouple of feature importance in time series help the neural networks understand time series data accurately.

3.3 Results on Interpretation

We present both qualitative and quantitative results.

Qualitative Analysis

We apply the series saliency methods with the Self-Attention encoder model on the air quality dataset. Fig. 5 visualizes the learned mask component when forecasting the selected future value with horizon $\tau = 6$. As the color bar shows, from blue to red means that the features become more and more important. We can see that the features from time interval $[20, 30]$ have the largest saliency, which corresponds to the extreme

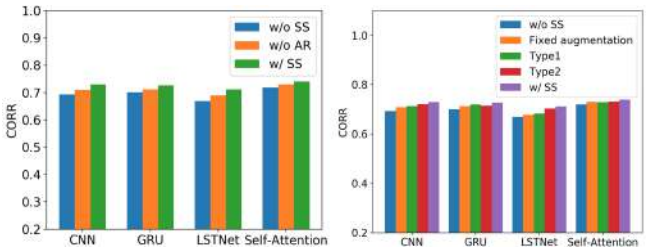

Figure 3: The CORR values of various methods on the air quality dataset with horizon $\tau = 6$.

low value in feature 1 representing the concentration of CO and high value in feature 5 representing the concentration of benzene.

We also visualize the correspondence between saliency mask, the original time series data and the data frequencies on the electricity dataset. Fig. 4 shows the results. We can see that the data of channel 250 has the high value of the saliency mask and reflects a periodic structure (top left in Fig. 4). In contrast, the data of channel 36 has the low value of the saliency mask and reflects a relatively acylic structure (top right in Fig. 4). Furthermore, to prove the periodicity of our selected a channels, we map the corresponding feature to a frequency domain by a fast fourier transform procedure (FFT) [Nussbaumer, 1981] and show the frequency result below the data of the two channels.

Quantitative Comparison

Finally, similar as in [Ismail *et al.*, 2020], we present a quantitative metric to evaluate the interpretability for time series forecasting and compare with various baselines.

Specifically, for each testing example X, we use a given interpretation method to generate the feature importance map, and select the top k features. Then, we add the random Gaussian noise on the selected features, and feed the perturbed version into the pretrained model for forecasting. Given a testing set, we calculate the corresponding CORR value. We change the value of $k = [10\%, ..., 90\%]$, and obtain the decreasing CORR curve with different k (i.e., different perturbation levels). Finally, we calculate the area under the CORR curve as the quantitative metric and denote it by AUCORR. The lower AUCORR values mean that we select the more important features ßand the interpretation method is more effective.

We compare with a wide range of baselines including Grad [Baehrens *et al.*, 2010], DeepLift [Shrikumar *et al.*, 2017], feature ablation [Suresh *et al.*, 2017], Occlu-

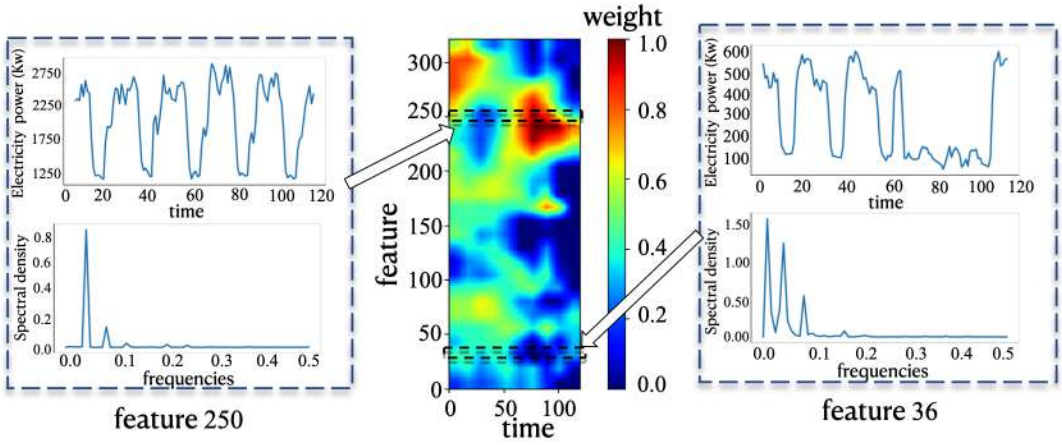

Figure 4: Saliency visualization on the electricity dataset with horizon $\tau = 6$ and $T = 120$. Note that feature 250 corresponding to higher salient region is often periodic, while feature 36 corresponding to weaker salient region is acyclic. Best viewed in color.

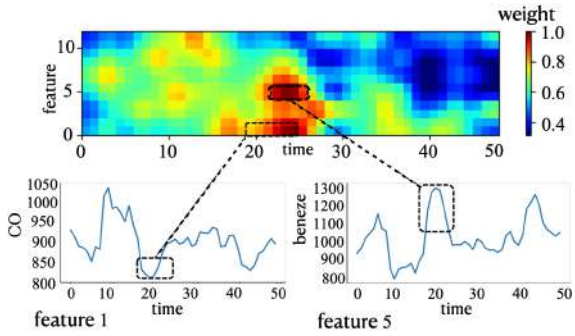

Figure 5: Correspond interpretable region in temporal saliency map. The series saliency is visualized on the air quality data with forecasting horizon $\tau = 6$ (Best viewed in color).

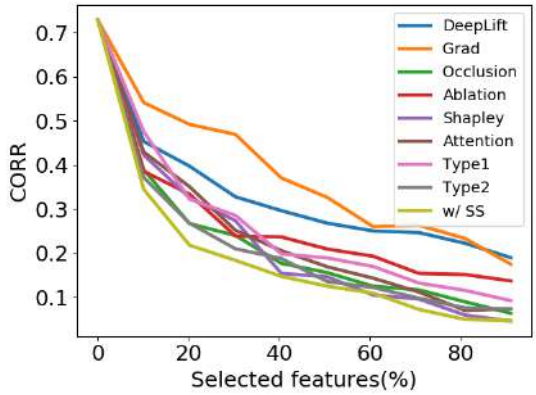

Figure 6: CORR curves of Self-Attention encoder for industry data with horizon $\tau = 6$. Best performance in boldface

sion [Zeiler and Fergus, 2014], Shapely sampling [Castro *et al.*, 2009] and attention [Petersen and Posner, 2012]. Also, to examine the effect of regularizers in interpretation, we consider the following variants of our method:

- **w/ SS**: The model in Fig. 2 interpreted with both ℓ_r and ℓ_m regularizers as in Eq. (8);
- **Type1**: The model in Fig. 2 interpreted by minimizing the loss L_2 only (i.e., without ℓ_r and ℓ_m);
- **Type2**: The model in Fig. 2 interpreted by minimizing the loss L_2 with ℓ_r regularizer (i.e., without ℓ_m).

Table 2 and Fig. 6 show the results. We can see that the series saliency map by our method can obtain the best quantitative interpretation results compared to those by other interpretation results (e.g., DeepLift and attention). Moreover, the two regularization terms ℓ_m and ℓ_r are useful to improve the interpretation results. In general, although we can't guarantee any selected future values will have the representative interpretation in qualitative analysis (e.g., Fig. 5), the performance of our method is quantitatively superior to other competitors.

4 Conclusions

We present a novel scheme of series saliency to boost both accuracy and interpretability for multivariate time series forecasting. By extracting series images from sliding windows of the time series, we design series saliency as a mixup approach with a learnable mask defined on the series images and their perturbed versions. Series saliency acts as an adaptive data augmentation method for training deep models, and meanwhile by slightly changing the objective, it can be optimized to find a mask for interpretable forecasting in both feature and time dimensions. Experimental results show the superiority of series saliency over various baselines.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (No. 2017YFA0700904,No.2020AAA0104304), NSFC Projects (Nos. 61620106010, 62076147, U19A2081, U19B2034, U1811461), Beijing Academy of Artificial Intelligence (BAAI), Tsinghua-Huawei Joint Research Program, a grant

Methods	Industry	Air Quality	Electricity
Grad	0.214 ± 0.007	0.297 ± 0.007	0.199 ± 0.008
DeepLift	0.211 ± 0.008	0.241 ± 0.006	0.174 ± 0.003
Ablation	0.204 ± 0.008	0.225 ± 0.006	0.213 ± 0.009
Occlusion	0.124 ± 0.004	0.221 ± 0.011	0.142 ± 0.005
Shapley	0.145 ± 0.006	0.211 ± 0.010	0.171 ± 0.005
Attention	0.141 ± 0.007	0.203 ± 0.007	0.139 ± 0.003
Type1	0.131 ± 0.005	0.205 ± 0.008	0.143 ± 0.004
Type2	0.122 ± 0.002	0.201 ± 0.005	0.135 ± 0.001
w/ SS	$\mathbf{0.117} \pm 0.002$	$\mathbf{0.192} \pm 0.003$	$\mathbf{0.131} \pm 0.002$

Table 2: AUCORR values of various methods on interpreting self attention encoder on the industry, air quality and electricity datasets.

from Tsinghua Institute for Guo Qiang, Tiangong Institute for Intelligent Computing, and the NVIDIA NVAIL Program with GPU/DGX Acceleration.

References

[Assaf *et al.*, 2019] Roy Assaf, Ioana Giurgiu, Frank Bagehorn, and Anika Schumann. Mtex-cnn: Multivariate time series explanations for predictions with convolutional neural networks. In *IEEE International Conference on Data Mining (ICDM)*, pages 952–957. IEEE, 2019.

[Baehrens *et al.*, 2010] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11:1803–1831, 2010.

[Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473*, 2014.

[Box and Jenkins, 1976] George EP Box and Gwilym M Jenkins. Time series analysis, forecasting and control. *Holden-Day*, 1976.

[Castelvecchi, 2016] Davide Castelvecchi. Can we open the black box of ai? *Nature News*, 538(7623):20, 2016.

[Castro *et al.*, 2009] Javier Castro, Daniel Gómez, and Juan Tejada. Polynomial calculation of the shapley value based on sampling. *Computers & Operations Research*, 36(5):1726–1730, 2009.

[Chen *et al.*, 1990] Sheng Chen, Stephen A Billings, and PM Grant. Non-linear system identification using neural networks. *International journal of control*, 51(6):1191–1214, 1990.

[Chung *et al.*, 2014] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv:1412.3555*, 2014.

[Dabkowski and Gal, 2017] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *NeurIPS*, pages 6967–6976, 2017.

[Gamboa, 2017] John Cristian Borges Gamboa. Deep learning for time-series analysis. *arXiv:1701.01887*, 2017.

[Goodfellow *et al.*, 2016] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[Harvey, 1990] Andrew C Harvey. *Forecasting, structural time series models and the Kalman filter*. Cambridge university press, 1990.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[Hyndman and Athanasopoulos, 2018] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[Ismail *et al.*, 2020] Aya Abdelsalam Ismail, Mohamed Gunady, Héctor Corrada Bravo, and Soheil Feizi. Benchmarking deep learning interpretability in time series predictions. *arXiv:2010.13924*, 2020.

[Iwana and Uchida, 2020] Brian Kenji Iwana and Seiichi Uchida. An empirical survey of data augmentation for time series classification with neural networks. *arXiv preprint arXiv:2007.15951*, 2020.

[Lai *et al.*, 2018] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *International Conference on Information Retrieval*, pages 95–104, 2018.

[Lundberg and Lee, 2017] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *NIPS*, pages 4765–4774, 2017.

[Mitrea *et al.*, 2009] CA Mitrea, CKM Lee, and Z Wu. A comparison between neural networks and traditional forecasting methods: A case study. *International journal of engineering business management*, 1:11, 2009.

[Nussbaumer, 1981] Henri J Nussbaumer. The fast fourier transform. In *Fast Fourier Transform and Convolution Algorithms*, pages 80–111. Springer, 1981.

[Petersen and Posner, 2012] Steven E Petersen and Michael I Posner. The attention system of the human brain: 20 years after. *Annual review of neuroscience*, 35:73–89, 2012.

[Ribeiro *et al.*, 2016] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. In *SIGKDD*, pages 1135–1144, 2016.

[Samek *et al.*, 2019] Wojciech Samek, Grégoire M., Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. *Explainable AI: interpreting, explaining and visualizing deep learning*, volume 11700. Springer Nature, 2019.

[Serrano and Smith, 2019] Sofia Serrano and Noah A Smith. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, 2019.

[Shih *et al.*, 2019] Shun-Yao Shih, Fan-Keng Sun, and Hung-yi Lee. Temporal pattern attention for multivariate time series forecasting. *Machine Learning*, 108(8-9):1421–1441, 2019.

[Shrikumar *et al.*, 2017] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *arXiv:1704.02685*, 2017.

[Suresh *et al.*, 2017] Harini Suresh, Nathan Hunt, Alistair Johnson, Leo Anthony Celi, Peter Szolovits, and Marzyeh Ghassemi. Clinical intervention prediction and understanding using deep networks. *arXiv:1705.08498*, 2017.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[Zeiler and Fergus, 2014] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. Springer, 2014.

A Details of Datasets

Electricity The UCI electricity load diagrams dataset [Mitrea *et al.*, 2009] contains 370 customer power consumption per unit time. There is no missing value in this dataset, recording the power consumption per 15 minutes (KWH) from 2011 to 2014. Each column of the data represents a customer (370 columns in total), each row represents a quarter (140,256 rows in total), and all time labels are subject to Portuguese time.

Air-Quality The Air-Quality dataset [UCI, 2006] contains 9,358 instances of hourly averaged responses from an array of 5 metal chemical sensors embedded in Air quality Device. Data was recorded from March 2004 to February 2005 (one year). The device was located on the field in a significantly polluted area.

Industry data The data of Hangseng Stock Composite Index (HSCI) and another eleven industry stock indices are obtained from the Wind platform. Eleven industry stock indices include consumer good manufacturing, consumer service, energy, finance, industry, information technology, integrated industry, raw material, real estate, and utilities. The dataset covers the time period from September 2006 up to September 2019.

Datasets	T	D	L
Electricity	26,304	321	1 hour
Air-Quality	9,358	12	1 hour
Industry Stock Composite Index	3,205	12	1 day

Table 3: Dataset statistics, where T is length of time series or data size, D is the number of variables, and L is the sample rate. The three datasets are representative (from easy to difficult). The sample rate ranges from hour to day, and the dimension is from 12 to 321.

B Metric Description

We choose two widely used metrics to measure the performance on multivariate time series forecasting datasets. The first one is the root relative squared error (RSE), which is the scaled version of the widely used Root Mean Square Error (RMSE) to remove the influence of data scale:

$$RSE = \frac{\sqrt{\sum_{t=t_0}^{t_1}\sum_{i=1}^{n}(y_{t,i} - \hat{y_{t,i}})^2}}{\sqrt{\sum_{t=t_0}^{t_1}\sum_{i=1}^{n}(\hat{y}_{t,i} - \overline{\hat{y}_{t_0:t_1,1:n}})^2}} \tag{9}$$

The second metric is the empirical correlation coefficient (CORR), and the CORR is the measure of correlation between actual and forecast variables in time series.

$$\frac{1}{n}\sum_{i=1}^{n}\frac{(y_{t,i} - \overline{y_{t_0:\hat{t}_{1,i}}})(\hat{y}_{t,i} - \overline{\hat{y}_{t_0:t_{1,i}}})}{\sqrt{\sum_{t=t_0}^{t_1}(y_{t,i} - \overline{y_{t_0:t_{1:i}}})^2 \sum_{t=t_0}^{t_1}(\hat{y}_{t,i} - \overline{\hat{y}_{t_0:t_{1:i}}})^2}} \tag{10}$$

where y and $\hat{y}$ are the ground truth and the predicted value, respectively. $\overline{y}$ denotes the mean of set y values within a given set (e.g., testing set). For RSE, the lower the better, whereas for CORR, the higher the better. These datasets and the metrics of RSE and CORR have been widely used in many papers about time series forecasting.

C Detailed Experimental Settings

C.1 Base module

The four state-of-art deep learning methods for comparison are CNN, GRU, LSTNet and Self-Attention encoder, as detailed below:

- *CNN*: Convolutional nerual network [Goodfellow *et al.*, 2016] was designed to ensure only past information for forecasting. We use 7-layer CNN to do time series forecasting.
- *GRU + Attention*: GRU [Chung *et al.*, 2014] had been used in time series forecasting and combined with attention to improve interpretability.
- *LSTNet*: LSTNet [Lai *et al.*, 2018] is a model using CNN and RNN for multivariate time series forecasting. The architecture uses convolutional network and the Recurrent Neural Network (RNN) to extract short-term local dependency patterns among variables and to discover long-term patterns for time series trends.
- *Self-Attention encoder*: Transformer-based [Vaswani *et al.*, 2017] architecture has been modified for forecasting. We design a variant of Transformer with encoder-only structure which consists of L blocks of multi-head Self-Attention layers and position-wise feed forward layers.

C.2 Hyperparameters

Since the model structure is universal for all methods, we adjust the same optimal hyperparameters on the training data. Firstly, we use the Adam algorithm for the optimization with learning rate 10^{-4} and weigth decay 10^{-3}. For data preprocessing, we scale the data into the range $[0, 1]$ by batch normalization to avoid extreme values and improve the computation stability. We set $\lambda_1 = 10^{-3}$ and $\lambda_2 = 10^{-3}$ in both L_1 and L_2. We select the batch size according to the size of each dataset (either 64 or 128).

D Forecasting Visualization and Interpretation Features on More Datasets

Because the Self-Attention encoder obtains the best performance, we evaluate the forecasting results of Self-Attention encoder visually in series saliency on all the three datasets. As shown in Fig.s 7,8,9, our method gives accurate forecasts. The proposed model clearly yields better forecasts around the flat line after the peak and in the valley.

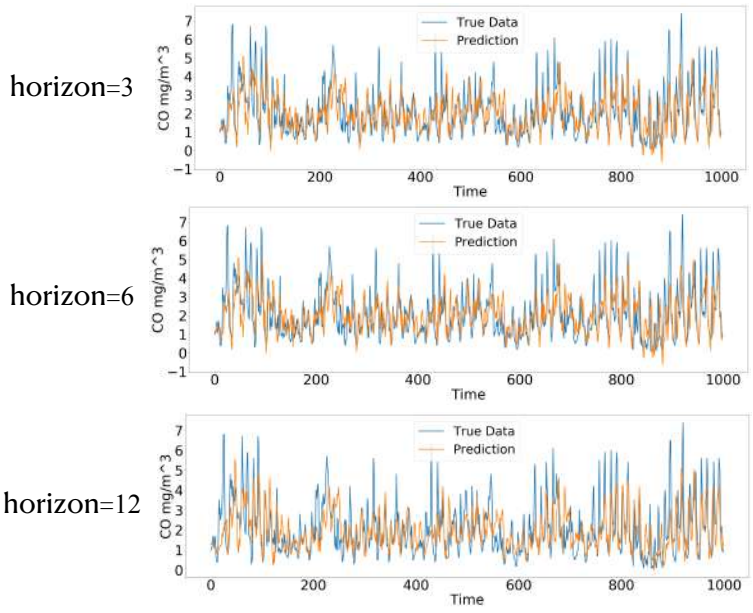

Figure 7: Prediction results for self-attention encoder in series saliency on air quality with horizon $\tau = \{3, 6, 12\}$ and window size $T = 64$. The feature is the true hourly averaged concentration CO in mg/m^3

Because the Self-Attention encoder obtains the best performance, we visualize the series saliency on other datasets, including the air quality and industry stock composite index, for horizon $\tau = 6$. As shown in Fig. 10 and Fig. 11, we could analyze the series saliency and obtain the corresponding feature importance.

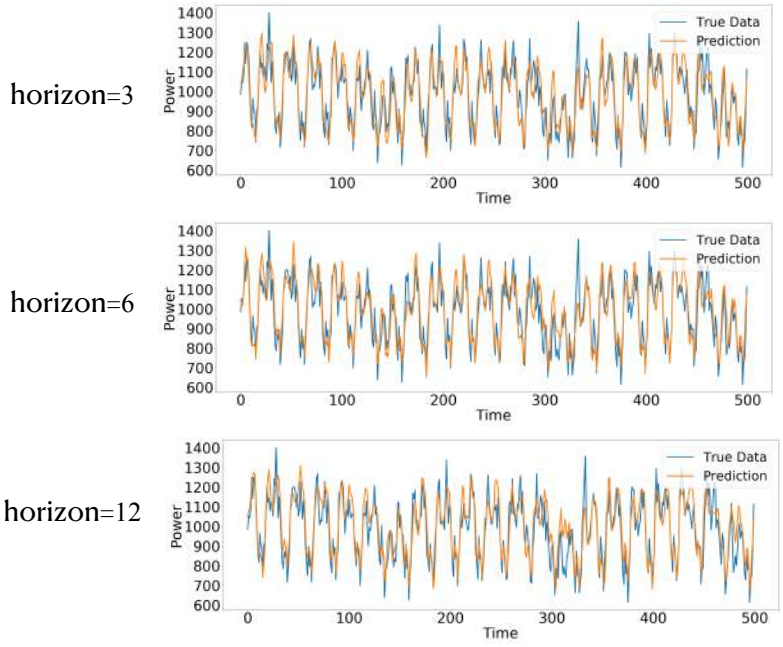

Figure 8: Prediction results for Self-Attention encoder in series saliency on electricity with horizon $\tau = \{3, 6, 12\}$ and window size $T = 168$. The feature is power consumption of No.7 powerplant. The model learned the periodicity of electricity data.

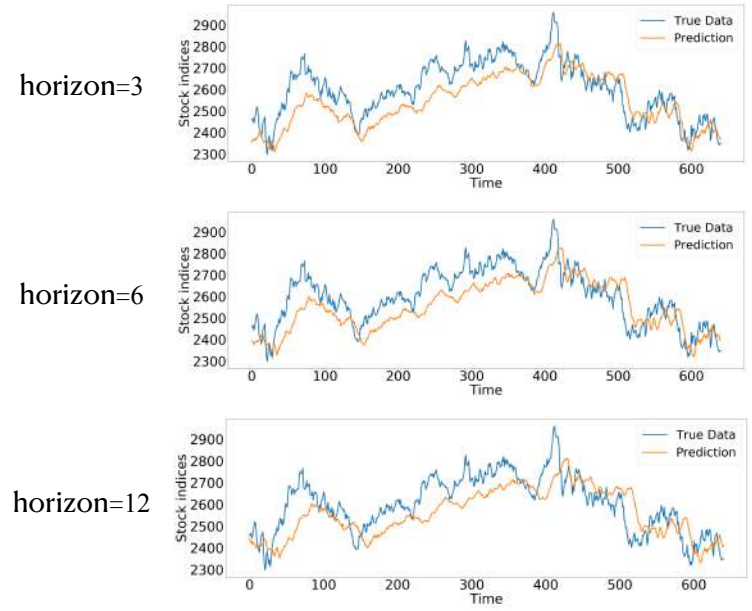

Figure 9: Prediction results for Self-Attention encoder in series saliency on industry stock indicies with $\tau = \{3, 6, 12\}$ and window size $T = 168$. The feature is No.1 stock indicies of Hangseng Stock Composite Index.

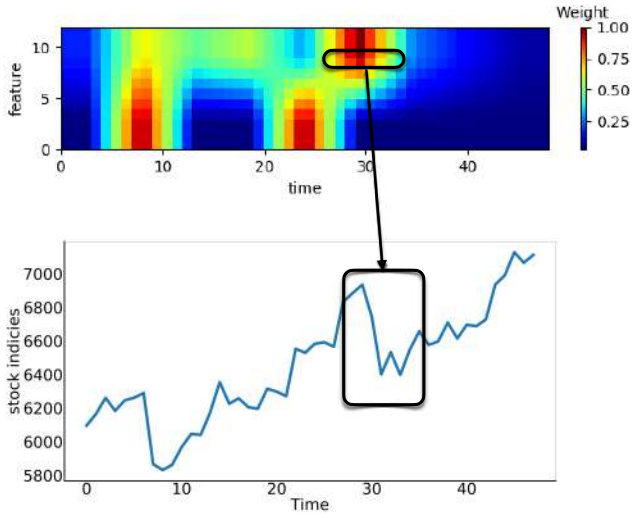

Figure 10: Series saliency for Self-Attention encoder on industry stock indicies with horizon $\tau = 6$ and window size $T = 64$. The highlighted area corresponds to the dramatic change in the industry stock indices. The phenomenon shows that the stock indices influences the forecasting greatly.

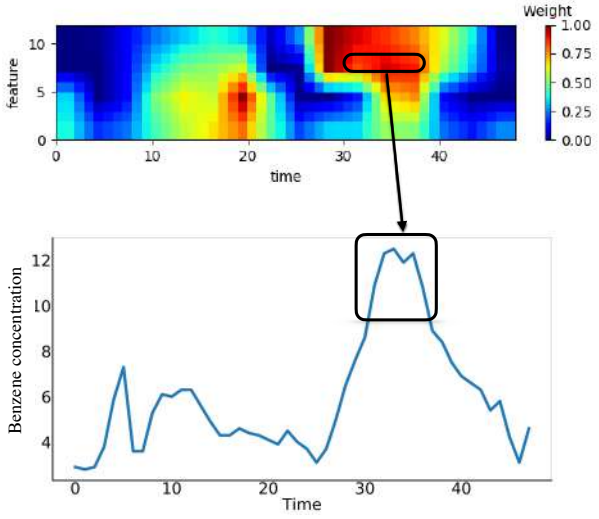

Figure 11: Series saliency for Self-Attention encoder on air quality with horizon $\tau = 6$ and window size $T = 64$. The highlighted area corresponds that benzene increases sharply in air quality. As you known, benzene is harmful, which impacts air quality greatly.

E Empirical results

The full set of results (including RSE and CORR) is shown in Table 4 and Table 5. We ran the experiments for 10 times and present the full results (mean and standard deviations).

	Air Quality			Industry			Electricity		
Methods	$\tau = 3$	$\tau = 6$	$\tau = 12$	$\tau = 3$	$\tau = 6$	$\tau = 12$	$\tau = 3$	$\tau = 6$	$\tau = 12$
CNN	0.775 ± 0.003	0.701 ± 0.001	0.636 ± 0.001	0.919 ± 0.022	0.919 ± 0.019	0.841 ± 0.008	0.883 ± 0.004	0.871 ± 0.002	0.866 ± 0.004
GRU	0.804 ± 0.003	0.712 ± 0.002	0.639 ± 0.003	0.953 ± 0.003	0.936 ± 0.013	0.904 ± 0.011	0.878 ± 0.001	0.877 ± 0.003	0.867 ± 0.002
LSTNet	0.777 ± 0.001	0.708 ± 0.004	0.623 ± 0.004	0.949 ± 0.004	0.934 ± 0.003	0.876 ± 0.011	0.922 ± 0.004	0.913 ± 0.002	0.906 ± 0.002
Self-Attention	0.813 ± 0.002	0.722 ± 0.003	0.643 ± 0.003	0.961 ± 0.002	0.942 ± 0.005	0.905 ± 0.009	0.919 ± 0.007	0.907 ± 0.001	0.902 ± 0.003
CNN w/ Type1	0.765 ± 0.002	0.712 ± 0.001	0.637 ± 0.003	0.923 ± 0.006	0.921 ± 0.002	0.835 ± 0.001	0.889 ± 0.002	0.882 ± 0.007	0.876 ± 0.001
GRU w/ Type1	0.807 ± 0.004	0.713 ± 0.005	0.642 ± 0.011	0.959 ± 0.009	0.926 ± 0.002	0.905 ± 0.012	0.883 ± 0.001	0.879 ± 0.006	0.869 ± 0.003
LSTNet w/ Type1	0.788 ± 0.001	0.683 ± 0.002	0.643 ± 0.002	0.952 ± 0.007	0.931 ± 0.001	0.879 ± 0.006	0.922 ± 0.001	0.912 ± 0.005	0.906 ± 0.008
SA w/ Type1	0.814 ± 0.001	0.717 ± 0.005	0.639 ± 0.004	0.963 ± 0.003	0.949 ± 0.003	0.907 ± 0.001	0.919 ± 0.007	0.908 ± 0.004	0.903 ± 0.003
CNN w/ Type2	0.764 ± 0.001	0.721 ± 0.002	0.639 ± 0.003	0.944 ± 0.005	0.924 ± 0.003	0.864 ± 0.007	0.891 ± 0.001	0.891 ± 0.003	0.882 ± 0.006
GRU w/ Type2	0.808 ± 0.004	0.710 ± 0.001	0.641 ± 0.003	0.957 ± 0.008	0.939 ± 0.007	0.911 ± 0.003	0.899 ± 0.002	0.883 ± 0.007	0.877 ± 0.005
LSTNet w/ Type2	0.791 ± 0.003	0.704 ± 0.001	0.645 ± 0.004	0.956 ± 0.007	0.934 ± 0.008	0.887 ± 0.002	0.924 ± 0.003	0.916 ± 0.003	0.904 ± 0.008
SA w/ Type2	0.814 ± 0.001	0.729 ± 0.006	0.646 ± 0.005	0.954 ± 0.002	0.953 ± 0.003	0.914 ± 0.002	0.926 ± 0.003	0.911 ± 0.001	0.903 ± 0.003
CNN w/ SS	0.779 ± 0.005	0.723 ± 0.009	0.641 ± 0.007	0.941 ± 0.006	0.927 ± 0.004	0.881 ± 0.001	0.898 ± 0.004	0.893 ± 0.002	0.892 ± 0.007
GRU w/ SS	0.809 ± 0.003	0.716 ± 0.012	0.649 ± 0.003	0.955 ± 0.001	0.935 ± 0.002	0.912 ± 0.003	0.905 ± 0.004	0.889 ± 0.008	0.878 ± 0.003
LSTNet w/ SS	0.794 ± 0.008	0.724 ± 0.002	0.641 ± 0.003	0.959 ± 0.004	0.938 ± 0.001	0.901 ± 0.002	$\mathbf{0.928} \pm 0.003$	$\mathbf{0.918} \pm 0.003$	0.907 ± 0.001
SA w/ SS	$\mathbf{0.819} \pm 0.003$	$\mathbf{0.732} \pm 0.009$	$\mathbf{0.658} \pm 0.001$	$\mathbf{0.965} \pm 0.003$	$\mathbf{0.955} \pm 0.016$	$\mathbf{0.916} \pm 0.004$	0.923 ± 0.003	0.915 ± 0.001	$\mathbf{0.911} \pm 0.002$

Table 4: Empirical CORR results of air quality, industry and electricity when horizon $\tau = \{3, 6, 12\}$. Best performance in boldface. We report the mean and standard deviations in ten runs. For CORR, the higher the better.

	Air Quality			Industry			Electricity		
Methods	$\tau = 3$	$\tau = 6$	$\tau = 12$	$\tau = 3$	$\tau = 6$	$\tau = 12$	$\tau = 3$	$\tau = 6$	$\tau = 12$
CNN	0.309 ± 0.001	0.362 ± 0.007	0.405 ± 0.002	0.162 ± 0.003	0.168 ± 0.006	0.208 ± 0.008	0.101 ± 0.004	0.105 ± 0.006	0.107 ± 0.004
GRU	0.312 ± 0.011	0.357 ± 0.012	0.397 ± 0.002	0.202 ± 0.006	0.192 ± 0.009	0.222 ± 0.001	0.119 ± 0.007	0.125 ± 0.005	0.132 ± 0.004
LSTNet	0.327 ± 0.003	0.362 ± 0.006	0.407 ± 0.008	0.181 ± 0.010	0.191 ± 0.005	0.228 ± 0.004	0.089 ± 0.001	0.098 ± 0.003	0.104 ± 0.006
Self-Attention	0.301 ± 0.002	0.353 ± 0.003	0.387 ± 0.008	0.147 ± 0.006	0.172 ± 0.007	0.186 ± 0.001	0.088 ± 0.008	0.095 ± 0.007	0.101 ± 0.010
CNN w/ Type1	0.303 ± 0.006	0.361 ± 0.002	0.401 ± 0.002	0.168 ± 0.005	0.161 ± 0.004	0.199 ± 0.005	0.099 ± 0.003	0.101 ± 0.005	0.103 ± 0.006
GRU w/ Type1	0.307 ± 0.009	0.342 ± 0.002	0.386 ± 0.005	0.185 ± 0.003	0.184 ± 0.004	0.214 ± 0.006	0.113 ± 0.009	0.119 ± 0.004	0.128 ± 0.009
LSTNet w/ Type1	0.314 ± 0.008	0.361 ± 0.008	0.399 ± 0.006	0.180 ± 0.007	0.189 ± 0.003	0.225 ± 0.002	0.088 ± 0.003	0.092 ± 0.005	0.099 ± 0.006
SA w/ Type1	0.298 ± 0.004	0.351 ± 0.002	0.366 ± 0.005	0.134 ± 0.006	0.169 ± 0.001	0.181 ± 0.002	0.086 ± 0.003	0.096 ± 0.005	0.097 ± 0.005
CNN w/ Type2	0.299 ± 0.003	0.355 ± 0.005	0.398 ± 0.004	0.158 ± 0.008	0.164 ± 0.006	0.196 ± 0.009	0.093 ± 0.003	0.099 ± 0.004	0.101 ± 0.001
GRU w/ Type2	0.301 ± 0.003	0.337 ± 0.002	0.356 ± 0.001	0.179 ± 0.004	0.187 ± 0.007	0.205 ± 0.006	0.109 ± 0.008	0.115 ± 0.003	0.119 ± 0.002
LSTNet w/ Type2	0.299 ± 0.007	0.356 ± 0.002	0.389 ± 0.001	0.178 ± 0.008	0.193 ± 0.005	0.222 ± 0.001	0.087 ± 0.011	0.093 ± 0.006	0.098 ± 0.005
SA w/ Type2	0.291 ± 0.001	0.348 ± 0.003	0.362 ± 0.006	0.125 ± 0.006	0.166 ± 0.001	0.178 ± 0.002	0.085 ± 0.002	0.091 ± 0.002	0.094 ± 0.001
CNN w/ SS	0.298 ± 0.007	0.351 ± 0.004	0.392 ± 0.001	0.153 ± 0.004	0.166 ± 0.002	0.191 ± 0.003	0.090 ± 0.002	0.103 ± 0.001	0.102 ± 0.004
GRU w/ SS	0.292 ± 0.004	$\mathbf{0.323} \pm 0.002$	$\mathbf{0.342} \pm 0.006$	0.177 ± 0.009	0.176 ± 0.007	0.199 ± 0.002	0.106 ± 0.008	0.112 ± 0.005	0.112 ± 0.009
LSTNet w/ SS	0.297 ± 0.009	0.349 ± 0.004	0.387 ± 0.000	0.173 ± 0.001	0.179 ± 0.002	0.218 ± 0.005	0.086 ± 0.004	0.094 ± 0.008	0.094 ± 0.006
SA w/ SS	$\mathbf{0.288} \pm 0.003$	0.341 ± 0.003	0.351 ± 0.003	$\mathbf{0.119} \pm 0.003$	$\mathbf{0.159} \pm 0.003$	$\mathbf{0.174} \pm 0.003$	$\mathbf{0.081} \pm 0.002$	$\mathbf{0.089} \pm 0.001$	$\mathbf{0.091} \pm 0.002$

Table 5: Empirical RSE results of air quality, industry and electricity when horizon $\tau = \{3, 6, 12\}$. Best performance in boldface. We report the mean and standard deviations in ten runs. For RSE, the lower the better.