CDX-Net: Cross-Domain Multi-Feature Fusion Modeling via Deep Neural Networks for Multivariate Time Series Forecasting in AIOps

Jiajia Li[1,2], Ling Dai[1,2], Feng Tan[2], Hui Shen[3], Zikai Wang[2], Bin Sheng[1,2], Pengwei Hu[4]
[1]Department of Computer Science, Shanghai Jiao Tong University, Shanghai, China; [2]Shanghai Artificial Intelligence Research Institute, Shanghai, China
[3]Di-Matrix Corporation, Shanghai, China; [4]Merck China Innovation Hub, Shanghai, China

Abstract

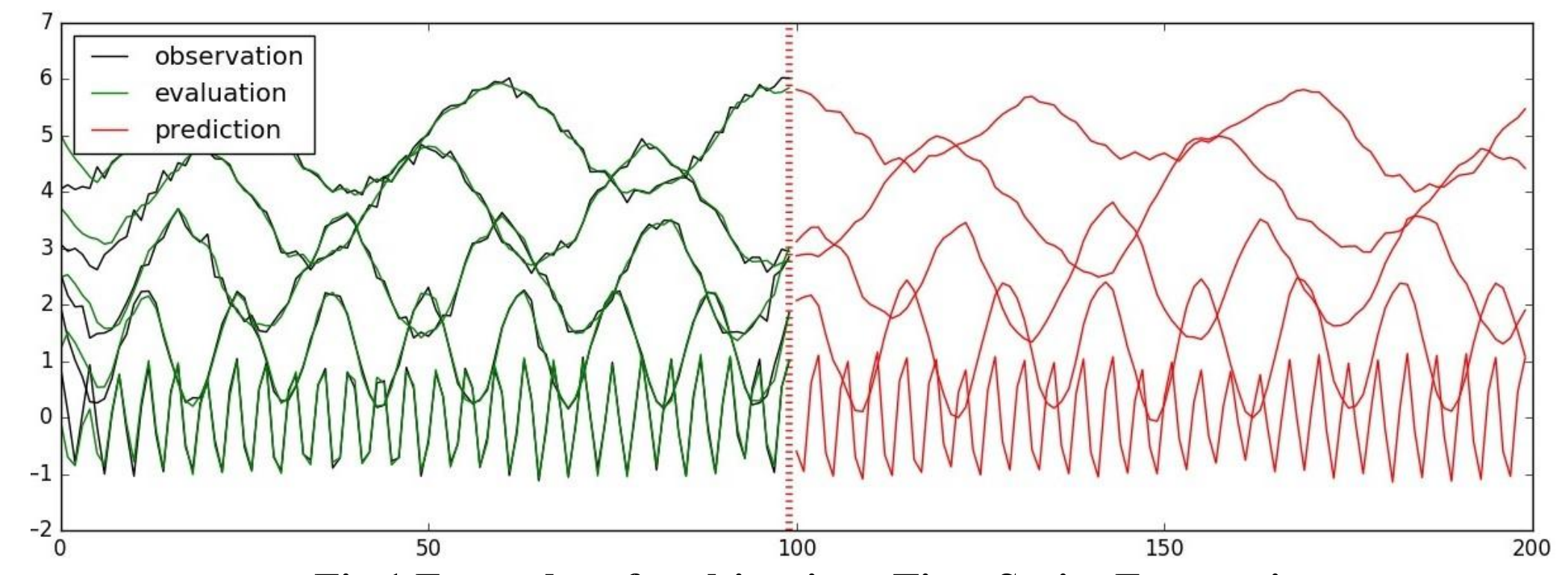
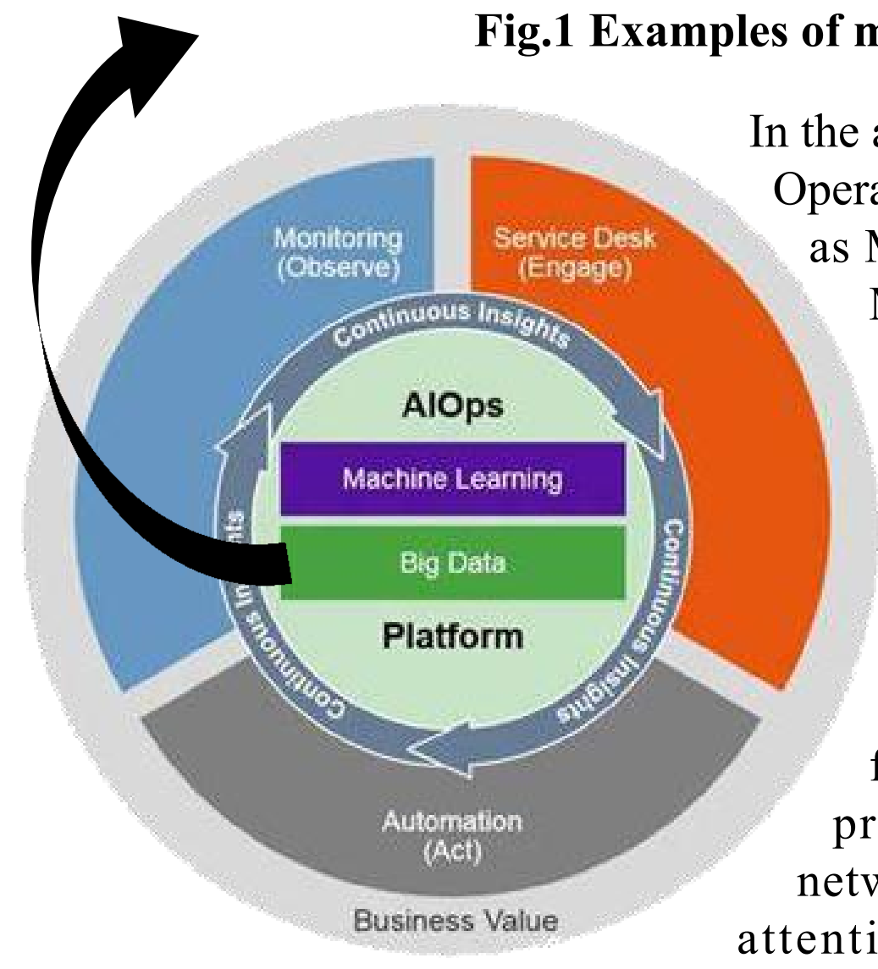
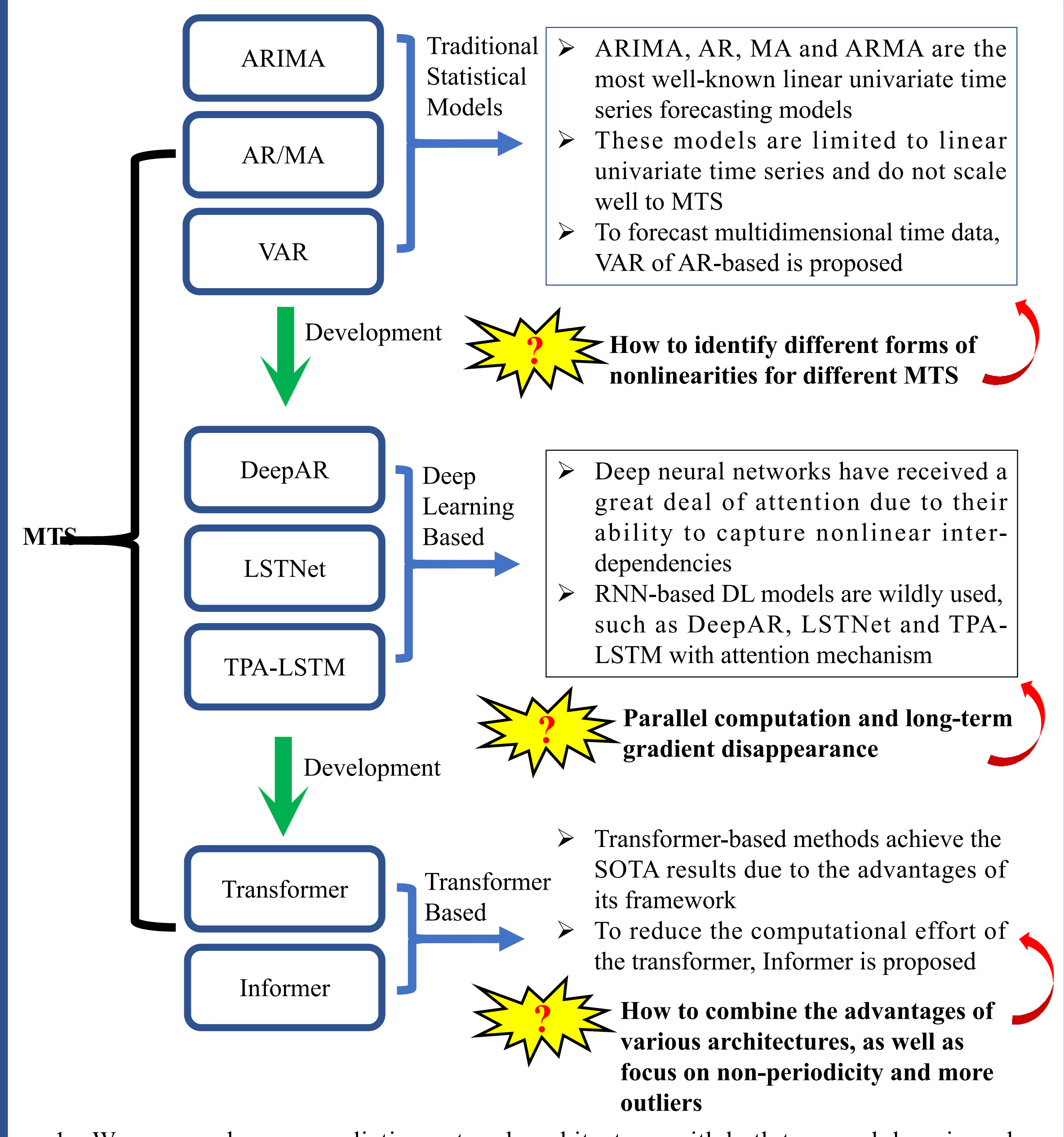


Fig.1 Examples of multivariate Time Series Forecasting.



In the application of Artificial Intelligence for IT Operations (AIOps), monitoring data are usually modeled as MTS (Multivariate Time Series). The prediction of MTS has been widely studied and various models, including statistic algorithms and deep learning networks, have been proposed, which attempt to capture the multi-dimensional and non-linear features. To this end, this paper focuses on one important type of time series: aperiodic MTS. Our solution introduces a deep neural network named CDX-Net to describe and analyze aperiodic MTS from both temporal and spectral domains. We also propose the integration of the convolution neural network (CNN), recurrent neural network (RNN) and attention mechanism into the predictive model. The introduction of these modules can effectively improve the feature extraction and feature fusion procedures. We conduct performance evaluation on a real-world dataset from an AIOps application and the correlation between the predicted result and ground-truth is found to be significant. The proposed model is compared with several state-of-the-art baseline methods. Empirical results show that our model achieves better performance in most evaluation metrics while others can perform better under some particular settings.

Introduction



- ARIMA, AR, MA and ARMA are the most well-known linear univariate time series forecasting models
- These models are limited to linear univariate time series and do not scale well to MTS
- To forecast multidimensional time data, VAR of AR-based is proposed

How to identify different forms of nonlinearities for different MTS

- Deep neural networks have received a great deal of attention due to their ability to capture nonlinear inter-dependencies
- RNN-based DL models are wildly used, such as DeepAR, LSTNet and TPA-LSTM with attention mechanism

Parallel computation and long-term gradient disappearance

- Transformer-based methods achieve the SOTA results due to the advantages of its framework
- To reduce the computational effort of the transformer, Informer is proposed

How to combine the advantages of various architectures, as well as focus on non-periodicity and more outliers

1. We proposed a new prediction network architecture, with both temporal domain and spectral domain features of the time series, increasing the robustness and generalization of the model.
2. We improved the channel attention module reworked to reflect the channel weight distribution more comprehensively.
3. We designed the feature fusion method based on the attention mechanism. Finally, we applied the model to the AIOps and achieved significant results.

Methods

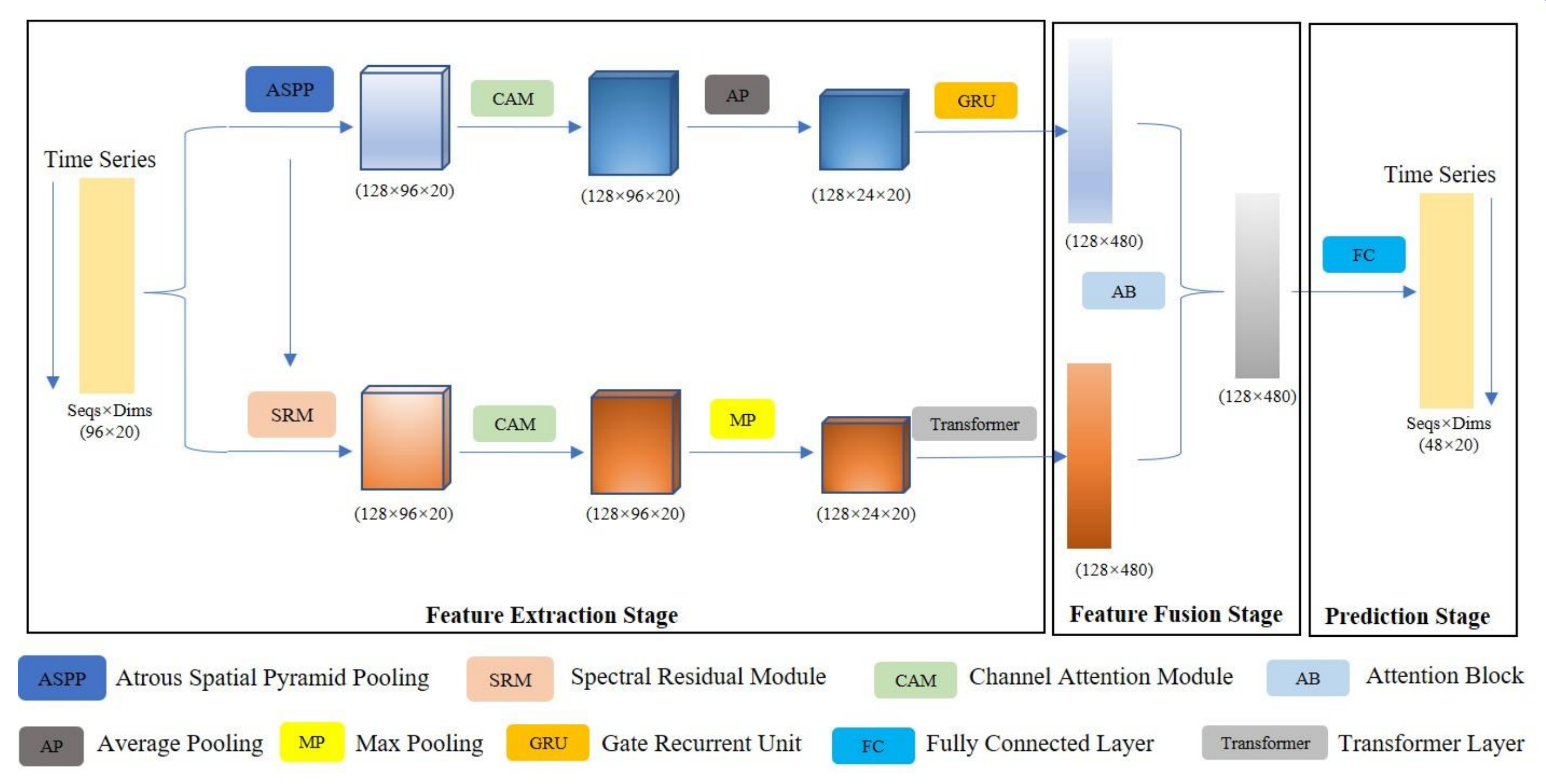
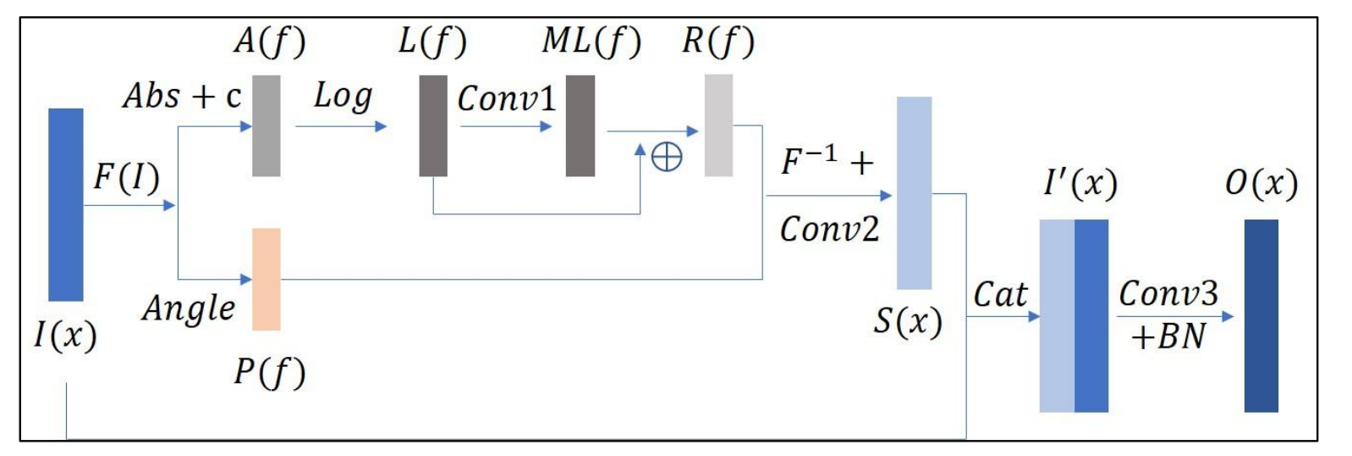


Fig.2 Proposed CDX-Net architecture.

In CDX-Net, there contains three stages: feature extraction, feature fusion and prediction stage. In our paper, we mainly introduce the innovative modules. That are, the Spectral Residual Module (SRM), and Channel Attention Module (CAM) for feature extraction and Attention Block (AB) for feature fusion.



$$\begin{cases} A(f) = Abs(F(f)) + c & (1) \\ P(f) = Angle(F(f)) & (2) \\ L(f) = Log(A(f)) & (3) \\ S(x) = ReLU[Conv2(F^{-1}([exp(R(f) + P(f))]^2))] & (4) \\ I'(x) = Cat[S(x), I(x), dim = 0] & (5) \end{cases}$$

Fig.3 The structure of proposed spectral residual module (SRM)

Let $I(x)$ represents the input feature map in our model. The $F(I)$ is obtained by using discrete Fourier transform of $I(x)$, where the amplitude and phase spectra are denoted as $Abs()$ and $Angle()$. To avoid problems with the logarithmic spectrum, we add a constant c, which we set to e^{-10}. The logarithmic spectrum of input feature map is derived as Eq.3, where $L(f)$ has the same shape as the input $I(x)$. We use the mean convolutional operation $Conv1()$ to obtain the averaged spectrum $ML(f)$. The convolution kernel size is defined as 3x3, the stride is 1 and the padding is 1. The channel number of the input feature map is C. The spectral residual $R(f)$ is the difference between $L(f)$ and $ML(f)$. At this point, we have obtained the spectral residual of the input feature map, which is in the frequency domain. Then, we use the residual spectrum $R(f)$ and the phase spectrum $P(f)$ to calculate the final saliency representation in the spatial domain by inverse Fourier transform F^{-1} and Gaussian smoothing filtering $Conv2()$ followed by $ReLU()$ and batch normalization $BN()$(Eq.5). For maintaining integrity of input features, we derive $I'(x)$ by concatenating $S(x)$ with $I(x)$ according to the direction of channel dimension. Finally, we use a 1x1 convolution $Conv3()$ with batch normalization to squeeze the channel number of $I'(x)$ from 2C to C, which is consistent of the input's.

$$\begin{cases} \beta = AA'(x) \cdot c1 + AM'(x) \cdot (1 - c1) & (6) \\ O(x) = I(x) \cdot Sigmoid(\beta) + I(x) & (7) \end{cases}$$

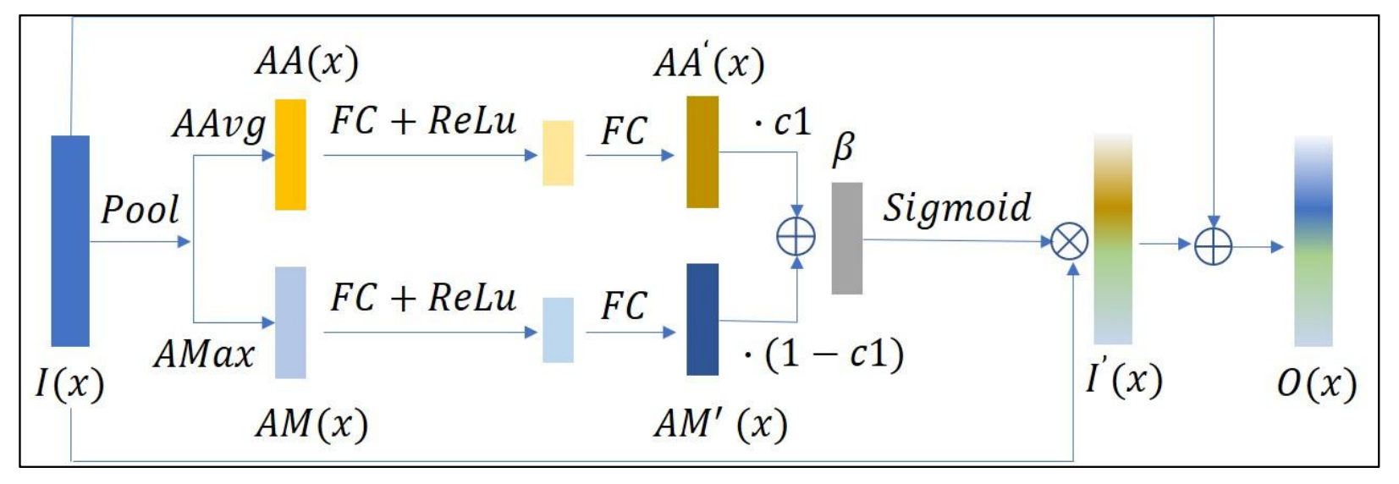


Fig.4 The structure of proposed channel attention module (CAM).

Firstly, let $I(x)$ represent the input feature map with C channels. To obtain the global pooling results, we have two paths where $AAvg()$ and $AMax()$ denote a global average pooling operation and a global maximal pooling operation, respectively. $AA(x)$and $AM(x)$ are then produced and they have the same shape of Cx1x1.

Each path needs to pass a multiple layer perception (MLP), which is used to obtain the channel attention coefficient. Each MLP has one fully connected FC layer and one $ReLU()$ layer that is followed by another FC layer. The channel attention coefficients for these two paths are $AA'(x)$ and $AM'(x)$, respectively. Furthermore, to better balance the attention coefficients of the two paths, we use a learnable parameter c1 to weight and sum them separately to obtain the final attention coefficient β, shown in Eq.6. The coefficient β is fed into a $Sigmoid$ function, whose output multiplies the input feature map to obtain $I'(x)$. For the purpose of benefiting the training, we use a residual connection adding $I'(x)$ to $I(x)$. Finally, the output $O(x)$ is obtained. In our CDX-Net, the CAM is equipped in the two paths at the feature extraction stage, as shown in Fig. 2.

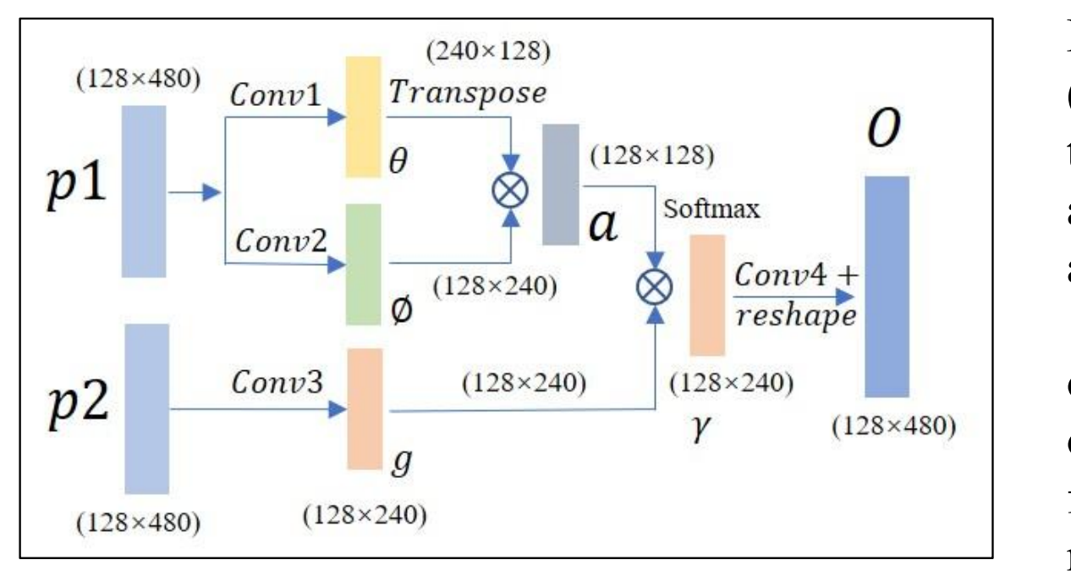


Fig.5 The structure of proposed channel attention module (AB).

$$\begin{cases} a = Softmax(\theta^T \cdot \phi) & (8) \\ \gamma = \bar{g} \cdot a & (9) \end{cases}$$

Firstly, we denote $p1$ and $p2$ as the outputs with a shape of (128x480) from the two branches, respectively. Then, we use three parallel 1x1 convolutional layers named $Conv1$, $Conv2$ and $Conv3$ to reduce the dimension of $p1$ and $p2$. The outputs are then θ, $\emptyset$ and g, respectively, which have the same shape of 128x240. The a means an attention coefficient map, which is obtained from Eq.8, where T represents matrix transpose operation. a has a shape of 128x128. $Softmax$ is a row-wise function, which can sum the value of each row of the feature map to 1.0. Intuitively, a reflects the interrelationship between individual points and the sum of all points. The following feature map γ is calculated by Eq.9 and its shape is 128x240. Because γ is the feature map reduced dimension and its dimension number is half of the input. We use a 1x1 convolutional layer with batch normalization to expand the dimension number and reshape the feature map O to 128x240.

Experiments

Dataset

Table1. Excluding the data column, the overall properties of the dataset, where L is the length of the time series, D is the dimension number of time series, I is the sampling spacing, S is size of the dataset in bytes and Mean and Var are the dataset's mean and variance, respectively.

Features	Whole	Training	Validation	Testing
L	101583	60949	20317	20317
D	20	20	20	20
I	5 mins	5 mins	5 mins	5 mins
S	13309 KB	7957 KB	2665 KB	2687 KB
Mean	857.2778	804.7236	866.8692	1005.3438
Var	2678.4322	2498.7410	2670.6132	3158.5293

To validate our model, we collected the log data of the system in the year 2015 as a dataset, whose distribution and characteristics are shown in the Table 1 and our dataset has a total of 20 dimensions.

Settings

PyTorch + + NVIDIA.

- **Optimizer:** Stochastic Gradient Descent (SGD)
- **Initial LR:** 0.001
- **Weight decay:** 0.0005
- **Momentum:** 0.9
- **Batch size:** 32
- **Epoch:** 30
- **LR decay:** decayed by 0.5 every 5 epochs
- **Loss Function:** MSE loss
- **Window Length:** 96
- **Label Length** :1
- **Horizon Length:** 1

Metrics

For the evaluation metrics of the model, we use CORR, MAE and MSE. They are defined as below:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |\hat{y}_i - y_i| ,$$

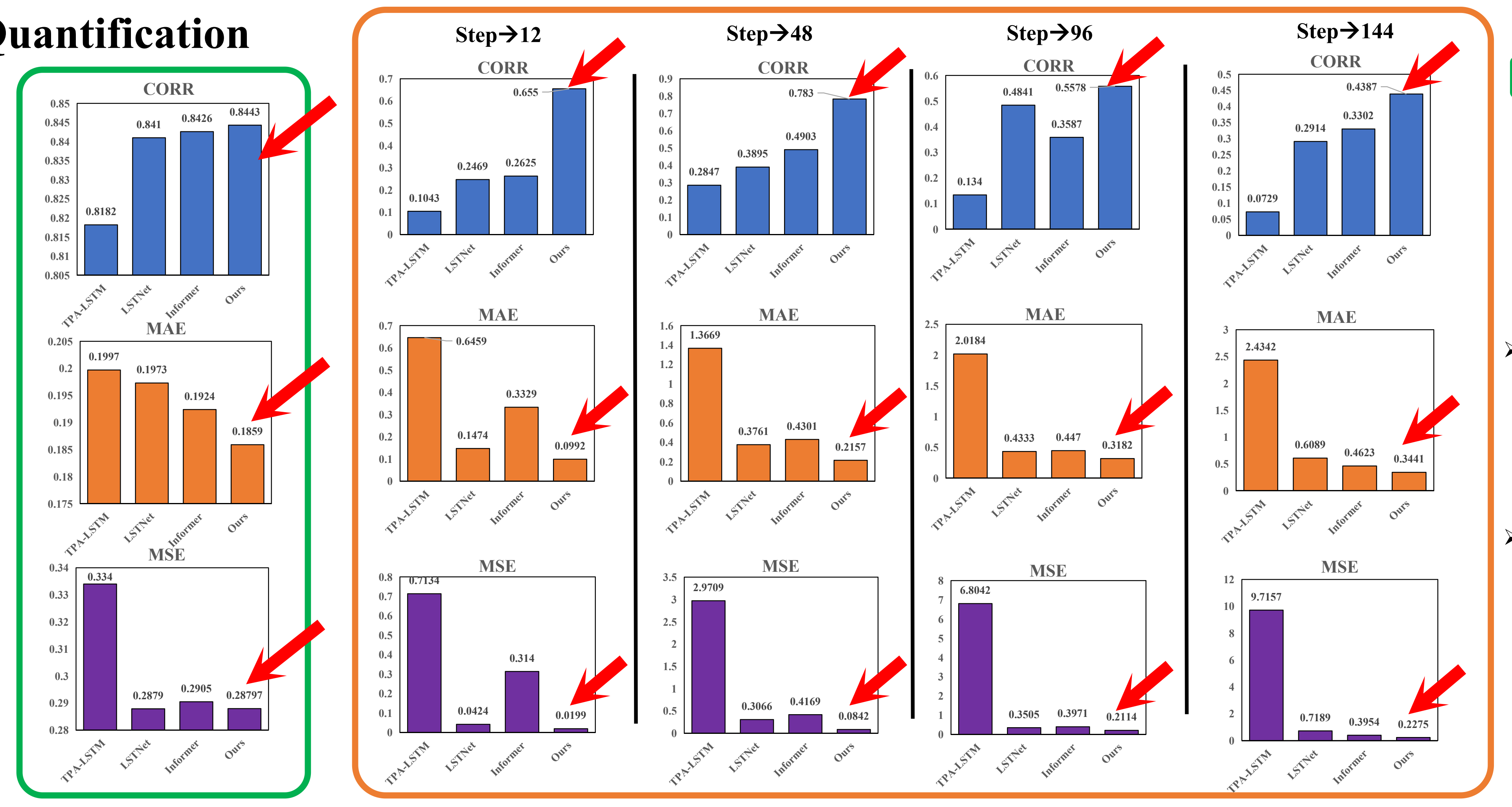
$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad \text{and}$$

$$\text{CORR} = \frac{\sum_{i=1}^{n} (y_i - \text{mean}(y))(\hat{y}_i - \text{mean}(\hat{y}))}{\sqrt{\sum_{i=1}^{n} (y_i - \text{mean}(y))^2}\sqrt{\sum_{i=1}^{n} (\hat{y}_i - \text{mean}(\hat{y}))^2}}$$

, where y and $\hat{y}$ are ground-truth signals and system prediction signals, respectively. Furthermore, we set $y = \{y_1, y_2, ...y_n\}$ and $\hat{y} = \{\hat{y}_1, \hat{y}_2, ...\hat{y}_n\}$and n means the number of the samples.

Results

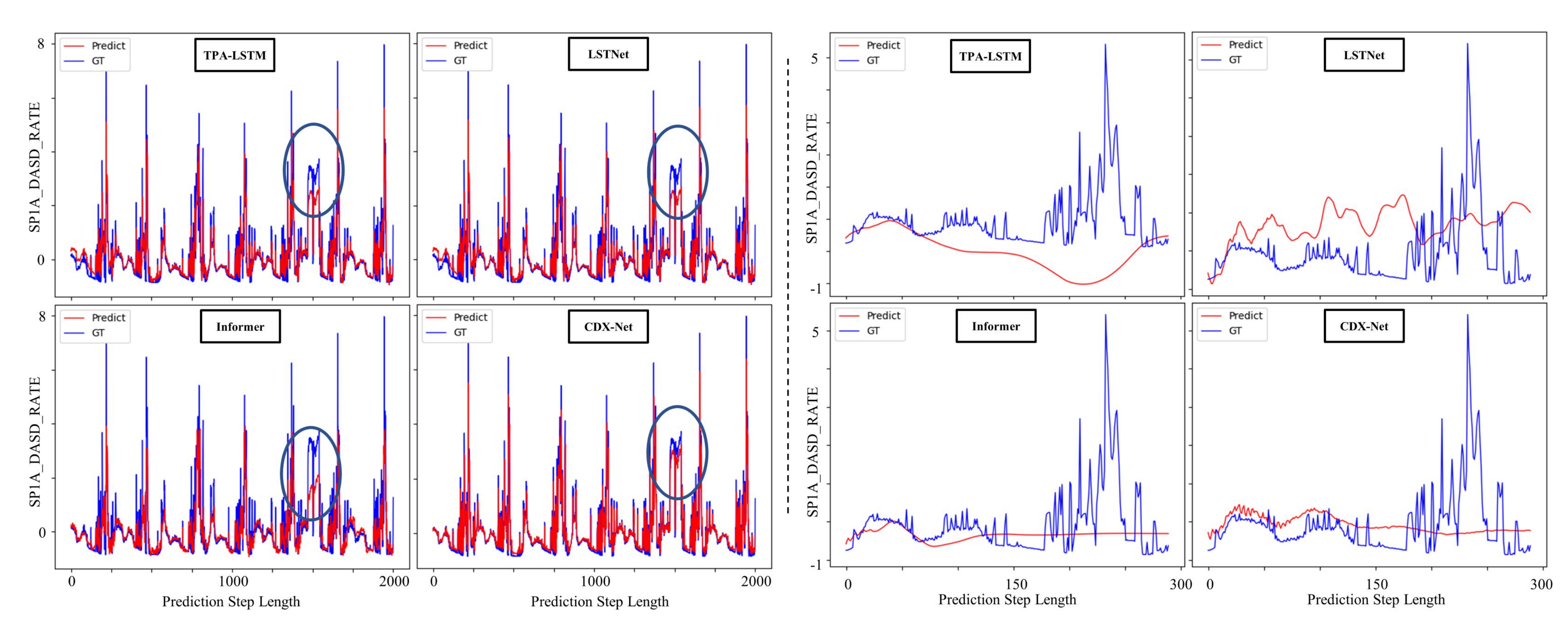
Quantification



- 🟩 Forecasting by iterating the single-step
- 🟧 Forecasting by iterating the step-by-step
- ⬅ Best performance

- **For single-step prediction, each model achieves good results from the overall trend**
- **For step-by-step prediction, our model always outperforms the SOTA models, especially, when the number of prediction step reaches over 48**

Visualization



- Ground truth
- Predicts

- **Move the gaze to the local position, our model predicts better when certain non-periodic and outliers (marked with ovals) appear in the time series**
- **CDX-Net provides better learning capability for the long- and short-term characteristics of historical data**

Fig.7 A comparison between the prediction results and the true value from different models in different cases.