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, Shanhai, China
[2]Shanghai Artificial Intelligence Research Institute, Shanghai, China
[3]Shanghai Dingmao Information Technology Inc., Shanghai, China
[4]Merck China Innovation Hub, Shanghai, China

ABSTRACT

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.

Index Terms— Multivariate Time Series, Deep Neural Network, Feature Fusion, Attention Mechanism, Time Series Prediction

1. INTRODUCTION

The main application scenarios of Artificial Intelligence for IT Operations (AIOps) are system bottleneck analysis, anomaly detection, and root cause detection. Datacenter-scale system is often service-oriented and contains hundreds of software and hardware components. These components are interconnected and need to respond to the needs of multiple applications. Hence, it generates a large amount of monitoring data, e.g. CPU/disk/memory workload data, which is usually modeled as Multivariate Time Series (MTS)[1].

In the typical time-series analysis, approaches tend to build a generative model that can recognize the stochastic mechanism and then predict the possible future values based on the observed data. These approaches range widely from conventional statistical models to deep learning networks. For example, the autoregressive integrated moving average model (ARIMA) is the most well-known model that aims at linear univariate time series. Furthermore, other autoregressive time series models, including autoregressive (AR), moving average (MA), and autoregressive moving average (ARMA) [1] are also proposed to deal with time series forecasting. Moreover, the machine learning-based method, e.g. linear support vector regression (SVR) [2], tries to consider the forecasting problem as a typical regression problem with time-varying parameters. However, most of these models are limited to linear univariate time series and do not scale well to MTS that involves multidimensional features.

To address the multidimensional time series, vector autoregression (VAR) [2], a generalization of AR-based models, has been proposed. On the other hand, MTS prediction also entails the modeling of nonlinearities, while neither AR-based nor VAR-based models can be adopted as their incapabilities to model non-linear coupling. Therefore, models based on kernel methods [3] and Gaussian processes [4] are proposed to address nonlinearities by assuming a predetermined non-linear form.

For the MTS, several deep learning models are proposed [5–13]. In LSTNet [5] , one-dimensional ordinary convolution is used to capture short-term local information, while GRU [6] and Skip-RNN for capturing long-term macroscopic information. In addition, the attention module is added to model the periodicity and the autoregressive process is added in the prediction part. The TPA-LSTM [7] improves the attention mechanism by focusing on the selection of key variables. Meanwhile, Transformer [8] and its variant Informer [9] has also achieved good results in MTS prediction.

[1]Thanks to the Science and Technology Commission of Shanghai Municipality (STCSM) for financial support (21511101200).

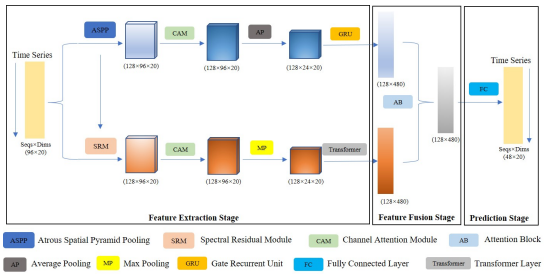

Fig. 1. Proposed CDX-Net architecture.

In [10], the authors down-sample the time series to a lower resolution and apply existing seasonal-trend decomposition algorithms to obtain rough estimates of trends and other seasonal components. STAM [11] aims for solving the MTS forecasting problem by fusing spatial and temporal features and using the attention mechanism. Inkit Padhi *et al*. [12] apply the BERT to MTS representation learning, which can be used for end-to-end pre-training and downstream tasks. For the inference, GPT Generator is used to handle the MTS forecasting problem.

However, MTS prediction with insignificant periodicity and spiky outliers still poses a challenge for these models. To this end, we introduce CDX-Net in this paper. Specifically, this paper's contributions include:

- A novel deep network model is proposed, which considers both temporal and spectral domain features of the time series, increasing its robustness and generalization.
- The channel attention module is added, which reflects the channel weight distribution comprehensively.
- The feature fusion method based on the attention mechanism is proposed.
- Extensive experiments on an aperiodic MTS dataset from a typical AIOps application validate CDX-Net's potential value in MTS prediction.

2. THE MODEL

2.1. Overview

In CDX-Net, there contains three stages: feature extraction, feature fusion and prediction stage. In this following, we will 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. Other modules will be briefly explained.

2.2. Atrous Spatial Pyramid Pooling Module

Atrous Spatial Pyramid Pooling Module (ASPP) [14] is proposed in computer vision and mainly serves two roles: 1. to expand the receptive domain and reduce the computational complexity in deep networks; 2. to capture multi-scale contextual information. In MTS prediction, after normalizing the multi-dimensional data, the input time series become similar to the image. Therefore, using the ASPP module can not only get the dependent features between different dimensions but also obtain the features of different steps.

2.3. Spectral Residual Module

In MTS prediction, several models can have good performance for data with highly significant periods, but the opposite is not true for data without significant periods, such as anomalies with more saliency. In the paper [15], by analyzing the spectrum of input data, it can obtain the residual spectrum in the frequency domain. Then, it can filter the residual spectrum and construct the saliency map in the spatial domain. In doing so, it is possible to eliminate the periodic fraction of the time series to derive outliers. Inspired by this, we designed the Spectral Residual Module (SRM) whose structural feature is shown in Fig. 2 to quickly filter the features at the outliers. Let $I(x)$ represents the input feature map in

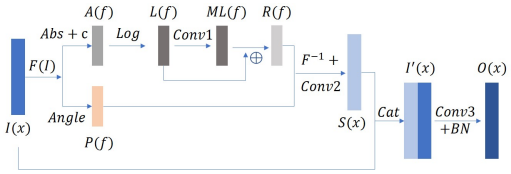

Fig. 2. The structure of proposed spectral residual module (SRM).

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 followings: $L(f) = Log(A(f))$, 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 3×3, 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 $BN()$:

$$S(x) = ReLU[Conv2(F^{-1}([exp(R(f) + P(f))]^2))] \quad (1)$$

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 1×1 convolution $Conv3()$ with batch normalization to squeeze the channel number of $I^{'}(x)$ from $2C$ to C, which is consistent of the input's.

2.4. Channel Attention Module

There are some methods for obtaining the channel attention, such as SEnet [16] and CAnet [17]. In our paper, an improved channel attention module (CAM) described by Fig. 3 is proposed.

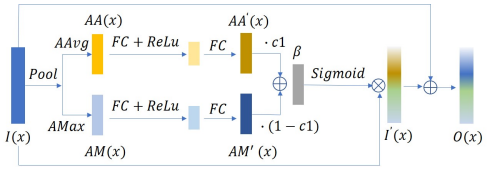

Fig. 3. The structure of proposed channel attention module (CAM).

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 $C \times 1 \times 1$.

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. 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. 1.

2.5. Attention Block

The attention block (AB) is motivated by the non-local network [18]. As shown in Fig. 4, we improve this module on two fronts. Firstly, we denote $p1$ and $p2$ as the outputs with a shape of (128×480) from the two branches, respectively. Then, we use three parallel 1×1 convolutional layers named $Conv1$, $Conv2$ and $Conv3$ to reduce the dimension of $p1$ and $p2$. The outputs are then θ, ϕ and g, respectively, which have the same shape of 128×240.

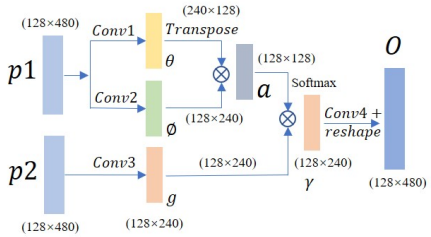

Fig. 4. The structure of attention block (AB).

The a means an attention coefficient map, which is obtained from:

$$a = Softmax(\theta^T \cdot \phi) \tag{2}$$

where T represents matrix transpose operation. a has a shape of 128×128. $Softmax$ 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: $\gamma = g \cdot a$ and its shape is 128×240. Due to γ is the feature map reduced dimension and its dimension number is half of the input. We use a 1×1 convolutional layer with batch normalization to expand the dimension number and reshape the feature map O to 128×240.

3. EXPERIMENTS

3.1. Dataset

To validate our model, we collected the monitoring data of an AIOps system as a dataset[2]. We randomly divide the dataset into training set, validation set and testing set at the ratio of 6:2:2.

Table 1. Results on our dataset using $CORR$, MAE and MSE as metrics.

Metrics	CORR	MAE	MSE	Para
TPA-LSTM [7]	0.8182	0.1997	0.3340	23.50M
LSTNet [5]	0.8410	0.1973	**0.28796**	22.45M
Informer [9]	0.8426	0.1924	0.2905	62.97M
Ours	**0.8443**	**0.1859**	0.28797	20.65M

3.2. Experimental Settings

All methods are implemented in the Pytorch framework. We use Stochastic Gradient Descent (SGD) for training with initial learning rate 10^{-3}, weight decay $5e^{-4}$, momentum 0.9, batch size 32, and iteration 30 epochs. The learning rate is decayed by 0.5 every 5 epochs. The training is implemented

[2]https://github.com/Torchlight-ljj/AIOPSdataset.

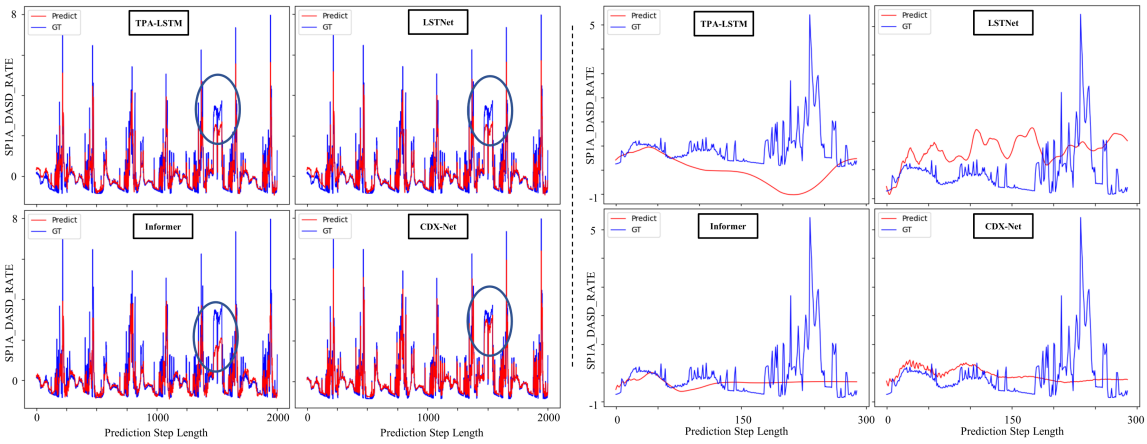

Fig. 5. A comparison between the prediction results and the true value from different models in different cases. GT means the ground truth.

Table 2. Results on our dataset using $CORR$, MAE, MSE as metrics.

Pred_steps	**12**			**48**		
Metrics	**CORR**	**MAE**	**MSE**	**CORR**	**MAE**	**MSE**
TPA-LSTM [7]	0.1043	0.6459	0.7134	0.2847	1.3669	2.9709
LSTNet [5]	0.2469	0.1474	0.0424	0.3895	0.3761	0.3066
Informer [9]	0.2625	0.3329	0.3140	0.4903	0.4301	0.4169
Ours	**0.6550**	**0.0992**	**0.0199**	**0.7830**	**0.2157**	**0.0842**
Pred_steps	**96**			**144**		
Metrics	**CORR**	**MAE**	**MSE**	**CORR**	**MAE**	**MSE**
TPA-LSTM [7]	0.1340	2.0184	6.8042	0.0729	2.4342	9.7157
LSTNet [5]	0.4841	0.4333	0.3505	0.2914	0.6089	0.7189
Informer [9]	0.3587	0.4470	0.3971	0.3302	0.4623	0.3954
Ours	**0.5578**	**0.3182**	**0.2114**	**0.4387**	**0.3441**	**0.2275**

on one NVIDIA Geforce GTX 3090 Ti GPU. We use the MSE loss for the training of each network and use the best-performing model on the validation set among all the epochs for testing. We use 5-fold cross-validation for the final evaluation.

3.3. Evaluation Metrics

For the evaluation metrics of the model, we use $CORR$, MAE and MSE, where $CORR$ means the empirical correlation coefficient, MAE is Mean Absolute Error and MSE represents Mean Squared Error. We compare CDX-Net with the three most dominant current deep models, including TPA-LSTM, LSTNet and Informer.

4. RESULTS

Firstly, in the MTS prediction task, we need to set the sizes of window, label and horizon, which represent the length of the input data, the length of the label and the distance step between window and label, respectively. In this work, we set window=96, label=1 and horizon=1. Experiments consist of two cases. In the first case, we compare the true and predicted values of each data in the test set, and summarize all the test results; in the second case, we select a random section of data as input in the test set, predict N steps in an iterative form, and compare the predicted and true values.

From the Tab. 1, we can find that for single-step prediction, each model achieves good results from the overall trend. However, when we move the gaze to the local position, our model predicts better when certain non-periodic and insignificant outliers appear in the time series, as can be seen by referring to the left parts of the Fig. 5 marked with ovals.

From Tab. 2, since the prediction errors accumulate gradually, the accuracy of each model gradually decreases as the number of prediction steps increases. When the number of steps reaches 48, all of the models reach the relative optimum and the visualized results display on the right side of Fig. 5. However, the performance of our model exceeds the baseline models. It can thereby be shown that CDX-Net provides better learning capability for the long- and short-term characteristics of historical data.

5. CONCLUSION

In this paper, we focus on MTS forecasting and propose a novel deep neural network named CDX-Net which includes a series of improved modules, ASPP module, SRM, CAM, GRU, transformer and AB module. On a dataset from a typical AIOps application, experiments strongly support this conclusion and show that the proposed model achieves state-of-the-art results.

6. REFERENCES

[1] M. Valipour, M. E. Banihabib, and S. M. R. Behbahani, "Comparison of the arma, arima, and the autoregressive artificial neural network models in forecasting the monthly inflow of dez dam reservoir," *Journal of Hydrology*, vol. 476, pp. 433–441, 2013.

[2] L. Cao and F. Tay, "Support vector machine with adaptive parameters in financial time series forecasting," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1506–1518, 2003.

[3] S. Chen, X. Wang, and C. Harris, "Narx-based nonlinear system identification using orthogonal least squares basis hunting," *IEEE Transactions on Control Systems and Technology*, vol. 16, no. 1, pp. 78–84, 2008.

[4] R. Frigola and C. E. Rasmussen, "Integrated preprocessing for bayesian nonlinear system identification with gaussian processes," in *52nd IEEE Conference on Decision and Control*, 2013, pp. 5371–5376.

[5] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long- and short-term temporal patterns with deep neural networks," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 95–104.

[6] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[7] S.-Y. Shih, F.-K. Sun, and H. yi Lee, "Temporal pattern attention for multivariate time series forecasting," *Machine Learning*, vol. 108, no. 8, pp. 1421–1441, 2019.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.

[9] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *AAAI*, 2020, pp. 11 106–11 115.

[10] L. Yang, Q. Wen, B. Yang, and L. Sun, "A robust and efficient multi-scale seasonal-trend decomposition," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5085–5089.

[11] T. Gangopadhyay, S. Y. Tan, Z. Jiang, R. Meng, and S. Sarkar, "Spatiotemporal attention for multivariate time series prediction and interpretation," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 3560–3564.

[12] I. Padhi, Y. Schiff, I. Melnyk, M. Rigotti, Y. Mroueh, P. Dognin, J. Ross, R. Nair, and E. Altman, "Tabular transformers for modeling multivariate time series," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 3565–3569.

[13] A. Abdulaal and T. Lancewicki, "Real-time synchronization in neural networks for multivariate time series anomaly detection," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 3570–3574.

[14] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[15] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[16] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 2011–2023, 2018.

[17] R. Gu, G. Wang, T. Song, R. Huang, M. Aertsen, J. Deprest, S. Ourselin, T. Vercauteren, and S. Zhang, "Ca-net: Comprehensive attention convolutional neural networks for explainable medical image segmentation," *IEEE Transactions on Medical Imaging*, vol. 40, no. 2, pp. 699–711, 2021.

[18] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.