Inter-and-Intra Domain Attention Relational Inference for Rack Temperature Prediction in Data Center

Fang Shen[1,2], Zhan Li[1(✉)], Bing Pan[1], Ziwei Zhang[3], Jialong Wang[1], Wendy Zhao[1], Xin Wang[3(✉)], and Wenwu Zhu[3(✉)]

[1] Alibaba Group, Hangzhou, China
{ziru.sf,zhan.li,diyuan.pb,quming.wjl,wendy.zhao}@alibaba-inc.com
[2] Alibaba Group, Bellevue, WA 98004, USA
[3] Tsinghua University, Beijing, China
{zwzhang,xin_wang,wwzhu}@tsinghua.edu.cn

Abstract. In a data center, predicting the rack temperature then generating alarms when an exception is detected can prevent server failure caused by high rack temperature. Each measuring point records the temperature of the rack over time, and each pair of measuring points may be associated with services or locations. Therefore, the rack temperature prediction problem can be modeled as a graph-based prediction problem. In this case, the prediction of the rack temperature depends not only on its own historical temperature but also on the temperature of racks having the same service or located near each other. Furthermore, the temperature of the rack is actually determined by various factors such as IT workloads and cold aisle temperature. Existing graph-based prediction methods do not consider the influence of these domains during the prediction, but only consider the temperature domain itself. To overcome this challenge, we propose an Inter-and-Intra domain Attention Relational Inference (I2A-RI) model: an unsupervised model that learns the relations between time series variables from different domains and utilizes the inferred interaction structure to achieve accurate dynamical predictions. Two attention modules, the guidance domain attention (GDA) module and the intra-domain attention (IDA) module, are proposed in I2A-RI, which encodes the inter-and-intra domain information to guide the learning procedure. Experiments on the real-world rack temperature dataset show that I2A-RI outperforms other state-of-the-art models since it takes the advantage of the ability to infer the potential interactions across domains. The benefits of the two proposed attention modules are also verified in the experiments.

Keywords: Data center · Rack temperature prediction · Relational inference · Graph neural network

A. Bhattacharya et al. (Eds.): DASFAA 2022, LNCS 13247, pp. 481–492, 2022.
https://doi.org/10.1007/978-3-031-00129-1_41

1 Introduction

In the data center intelligent operation and maintenance (O&M) system, monitoring the temperature of racks is of significant to prevent server downtime due to high temperature. By predicting the rack temperature, the O&M center can sense anomalies and generate alarms in advance, enabling onsite personnel to intervene to prevent accidents promptly. There are many racks in a large data center and the temperatures of different racks may be related to each other due to service or location proximity. The temperature of each rack is recorded over time, so the prediction of rack temperature can be modeled as a graph-based multivariate time series prediction problem. Many efforts have been made over the decades to model the multivariate time series, including statistical learning methods [2], deep neural networks (DNNs) [11] and graph neural networks (GNNs) [13,15]. Though statistical learning and DNNs have shown values in the area, our work focuses on GNNs since they take the graphs as inputs, allowing the complex relations and interactions between variables [14] to be naturally expressed in the model.

In recent years, prediction algorithms based on GNNs have been widely studied. Yu et al. [14] proposed the spatio-temporal graph convolutional network to capture both temporal and spatial dependencies for mid-and-long term traffic prediction. Wu et al. [13] proposed the multivariate time Series forecasting with GNN (MTGNN) model which constructs the graph from time series by learning the uni-directed relations then the temporal and spatial dependencies are captured by the dilated inception layer and the mix-hop propagation layer. However, these existing graph-based prediction methods only focus on the historical correlation information of the prediction domain itself. In this way, in the scenario of rack temperature prediction, only the historical temperature of its own rack and that of associated racks are considered. Nevertheless, the temperature of a rack in a data center is not only related to its historical temperature but also affected by factors of other domains, such as IT workloads and cold aisle temperature. In terms of IT workloads, the temperature of different racks and the workloads are a two-domain dynamic system, in which the potential interactions include some servers are running services in a sequence and tend to reach their peak workloads in a fixed order, or a few racks are close to each other and influence the temperature of one another more noticeable. The existing graph-based methods do not take these other domains' important factors into account, thus affecting the accuracy of prediction. It is encouraging to consider both intra-domain and inter-domain relationships in the complex system interact and achieve dynamical predictions.

Thus, we propose a novel GNN model, inter-and-intra domain attention relational inference (I2A-RI), which addresses the problem of learning the latent relations among time series variables across domains. The model is in the form of a variational autoencoder (VAE) [6,8]: the encoder learns the implicit relations between variables and constructs multiple graphs in an unsupervised manner; while the decoder takes the constructed structure and the time series data for prediction. The graphs learned by the encoders are in the guidance domains or

the predictive domain. Accordingly, we propose two attention [12] modules to make these graphs interact within and between domains: the guidance domain attention (GDA) module and the in-domain attention (IDA) module. The GDA module extracts information from other domains to guide the relational inference in the prediction domain. The IDA module is used to capture the asynchronous interactions in the prediction stage. For example, when the servers' workload jumps up, it takes a while for the heat to be fully spread into the server room and influence the temperature. In summary, our contributions are as follows:

- We propose a novel GNN framework (I2A-RI) to infer the variables' relations in multivariate time series modeling. The model automatically learns relations then combines them with the time series data to perform predictions. To the best of our knowledge, no prior work studies multivariate time series modeling problems from a relational inference perspective with multiple graphs representing relations of variables from different domains.
- We define two types of domains: prediction domain and guidance domain. The prediction domain contains the variables for forecasting, and the guidance domain contains the variables that influence those in the prediction domain. We also introduce two attention modules (GDA and IDA) in the model: the GDA module extracts information from the guidance domain into the prediction domain to guide the relational reasoning. The IDA module captures the asynchronous interactions in the prediction stage.
- We show that I2A-RI outperforms the state-of-the-art approaches in forecasting the temperature measurements of a server room on the real operation data.

2 Preliminaries

In this paper, our task is to model the multivariate time series. Given the multivariate time series with historical T time steps $X = [\mathbf{x}_1^i, \mathbf{x}_2^i, \cdots, \mathbf{x}_T^i], i = 1, 2, \cdots, N$, where N is the number of variables. Our goal is to predict the future value $Y = [\mathbf{x}_M^i]$, where M means M steps away or the future sequence $Y = [\mathbf{x}_{T+1}^i, \mathbf{x}_{T+2}^i, \cdots, \mathbf{x}_{T+M}^i]$, where M represents M time steps in the future. We aim to find a function f that maps from X to Y. From the graph's perspective, each variable in a multivariate time series can be regarded as a node in the graph. Connections between nodes are represented by an edge category matrix, which is expected to be learned and cannot be obtained in advance. Some definitions used in this paper are given as follows:

Definition 1 Graph. A graph is denoted as $G = (V, E)$, where V represents the set of nodes and E represents the set of edges. The number of all nodes is denoted as N.

Definition 2 Prediction Domain. The domain expected to obtain the predicted value.

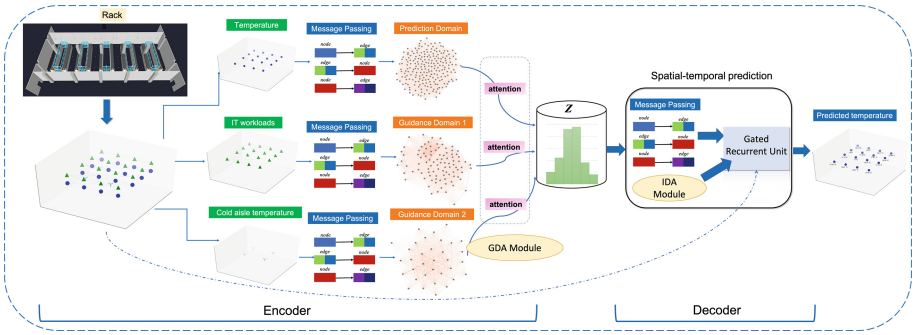


Fig. 1. I2A-RI architecture. The framework is composed of a multi-graph relational reasoning encoder and a spatial-temporal prediction decoder.

Definition 3 Guidance Domain. Factors affecting the prediction domain. For example, if we want to predict the future values of temperatures in data center operations, we need to consider the servers' workloads (guidance domain) and the cold aisle temperature (guidance domain) because the workloads are the indicator of the heat generated by the servers and the cold aisle temperature is the upstream measurement point of the rack temperature.

3 I2A-RI Model

The I2A-RI architecture is presented in Fig. 1. As illustrated in the figure, the whole framework of I2A-RI is based on VAE and mainly consists of two components: the multi-graph relational reasoning encoders and a spatial-temporal prediction decoder. The multi-graph relational reasoning encoder inputs data from multiple domains, and each domain forms a separate graph. The graphs of different domains are aggregated by a cross-domain attention layer. Note that although the encoder extracts features for multiple graphs, it only outputs one graph that integrates information of all domains. In summary, given historical time series $\mathbf{X}$, the encoder returns a factorized distribution $q_\psi(\mathbf{Z}|\mathbf{X})$ of the discrete relation type z_{ij} between nodes v_i and v_j. The decoder takes the learned graph from the encoder and the historical data to perform the spatial-temporal prediction:

$$p_\eta(\mathbf{X}|\mathbf{Z}) = \prod_{t=1}^{T} p_\eta(\mathbf{x}^{t+1}|\mathbf{x}^t, \ldots, \mathbf{x}^1, \mathbf{Z}) \tag{1}$$

The details of these two parts, including cross-domain and intra-domain attention mechanisms, will be covered next. Note that our proposal is implemented based on neural relational inference (NRI), and we focus on the improved part here. For more details please refer to [7].

3.1 Multi-graph Relational Reasoning Encoder

The purpose of the encoder is to learn the relationship type z_{ij} between each pair of nodes from historical time series data. Since we don't have a graph to start with, we first perform GNN with a fully-connected graph without self-loops. In the application of rack temperature prediction, the encoder has a total of three inputs, one for the prediction domain, that is, the temperature to be predicted, and the other two for the guidance domain, that is, IT workloads and cold aisle temperature. Then three fully-connected graphs are used to extract the features of the three domains respectively.

Any domain in the encoder contains three message passing operations. For one single domain, given the time series of each node $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$, the message passing operations in the encoder are performed as follows:

$$h_j^1 = fc(\mathbf{x}_j) \tag{2}$$

$$n \longrightarrow e : h_{i,j}^1 = f_{n2e}^1(h_i^1 || h_j^1) \tag{3}$$

$$e \longrightarrow n : h_j^2 = f_{e2n}^1(\sum_{i \neq j} h_{i,j}^1) \tag{4}$$

$$n \longrightarrow e : h_{i,j}^2 = f_{n2e}^2(h_i^2 || h_j^2) \tag{5}$$

where $fc(\cdot)$ denotes the fully connected networks and $\cdot||\cdot$ denotes the concatenation of two feature vectors.

After the last layer of node-to-edge operations, we need to consolidate the information for all the domains. We propose the guidance domain attention (GDA) layer to achieve this goal.

Guidance Domain Attention. For any domain k, assume that the dimension of the edge feature output by the last layer of node-to-edge is F, the weight vector $W_k \in \mathbf{R}^F$ is applied to each edge feature $h_{i,j}$ then obtain the relation coefficient $Rel_{i,j}^k$.

$$Rel_{i,j}^k = W_k^T \cdot h_{i,j}^2 \tag{6}$$

$Rel_{i,j}^k$ represents the importance score between nodes j and i. For better comparison, the relation coefficients are normalized across all edges by employing the softmax function:

$$a_{i,j}^k = softmax(Rel_{i,j}^k) = \frac{exp(Rel_{i,j}^k)}{\sum_{s \in Neigh(i)} exp(Rel_{i,s}^k)} \tag{7}$$

To summary, the whole process can be described as,

$$a_{i,j}^k = \frac{exp(LeakyReLU(W_k^T \cdot h_{i,j}^2)}{\sum_{s \in Neigh(i)} exp(LeakyReLU(W_k^T \cdot h_{i,s}^2)} \tag{8}$$

where LeakyReLU(·) [12] is used as the nonlinearity function and it can produce either positive or negative relationship coefficients. $a^k_{i,j}$ is the final output attention coefficients for domain k. Finally, the guidance domain features are aggregated to aid the prediction domain:

$$\tilde{h}^2_{i,j} = \sigma(\frac{1}{K}\sum_{k=1}^{K} a^k_{i,j}\mathbf{W}^k_{agg}h^2_{i,j}) \tag{9}$$

where $\mathbf{W}^k_{agg}$ is the learned weight vector applied to $h^2_{i,j}$. Then there is a residual connection and an output layer after the GDA layer.

The encoder finally returns a distribution $q_\psi(\mathbf{z}_{ij}|\mathbf{x}) = softmax(f_{enc}(\mathbf{x})_{ij})$, where $f_{enc}(\mathbf{x})$ denotes all operations performed on the fully-connected graph in the encoder. Refer to [7] for details about VAE's reparametrization trick and the way to handle discrete variables.

3.2 Spatial-Temporal Prediction Decoder

The components of the decoder are the same with [7]. We only focus on the improvements in this section. Due to the influence of other domains, the prediction domain needs a certain amount of time to deal with these changes. Moreover, time series are frequently coherent to themselves and most of them can not change instantaneously. Therefore we design the IDA module to take the advantage of the information carried in its history. To predict the value of $\mathbf{x}^{t+1}$, not only the value of $\mathbf{x}^t$, but also the earlier observations such as $\mathbf{x}^{t-n}, \ldots \mathbf{x}^{t-2}, \mathbf{x}^{t-1}$ are considered. Thus, the input of the IDA module contains two parts: x^t and $[\mathbf{x}^{t-n}, \ldots \mathbf{x}^{t-2}, \mathbf{x}^{t-1}]$. In this paper, n is set to 3. Suppose that the shapes of the two input tensors are: $U_1 = [Batch, N, 1, channels]$ and $U_2 = [Batch, N, 3, channels]$, where N is the number of nodes in a graph. The two tensors are then input to two 1d convolution layers ($\alpha(\cdot)$ and $\beta(\cdot)$) in order to transform them into the same space. The dot-product is adopted to calculate the similarity between the two transformed tensors:

$$Similarity(U_1, U_2) = \alpha(U_1)^T\beta(U_2) \tag{10}$$

U_2 is also input into another function $\theta(\cdot)$. The final vector output by the IDA module is calculated as,

$$IDA = Similarity(U_1, U_2)\theta(U_2) \tag{11}$$

For simplicity, we consider $\theta(\cdot)$ in the form of a linear transformation: $\theta(U_2) = W_\theta U_2$, where W_θ is a learnable weight matrix. The details are described in Fig. 2, where the initial input channel is set to 256.

The decoder's inputs include the learned graph and the historical data of different domains. In general, GNN with the message passing operator is applied to capture the spatial feature, and GRU is used to capture the temporal feature. The cross-domain feature, the value at the current time step x^t_j, the output of

the IDA module, and the hidden state of the previous time step x^{t-1} are fed into GRU:

$$h_j^{t+1} = GRU([\tilde{h}_{i,j}^2, \mathbf{x}_j^t], [h_j^{t-1}, IDA_j]) \tag{12}$$

Noted that we only learned the changes of x_j^t:

$$\mu_j^{t+1} = \mathbf{x}_j^t + fc(h_j^{t+1}) \tag{13}$$

And

$$p(\mathbf{x}^{t+1}|\mathbf{x}^t, \mathbf{Z}) = N(\mu^{t+1}, \sigma^2 I) \tag{14}$$

The loss function of the whole framework is defined as the ELBO [7]:

$$Loss = E[\log p_\eta(\mathbf{X}|\mathbf{Z})] - KL[q_\psi(\mathbf{Z}|\mathbf{X})||p_\eta(\mathbf{Z})] \tag{15}$$

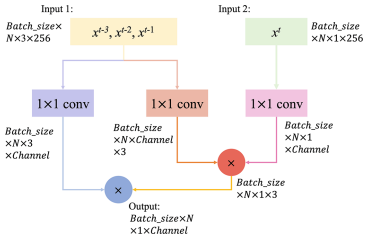
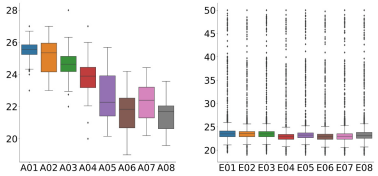


Fig. 2. The architecture of the IDA module.



(a) normal (b) abnormal

Fig. 3. The distributions of the two different data conditions.

4 Experiments

We create the RATEDC (Rack TEmperature data of the Data Center) dataset for the rack temperature prediction task.

4.1 RATEDC

The RATEDC dataset contains temperature, IT workloads, and cold aisle temperature data of 63 racks for one year. The data are collected every 2.5 min. A rack's temperature is a critical metric in data center daily operations: high temperature increases the equipment failure rate dramatically [4]. The workloads of a rack are the sum of the workloads of the servers in the rack. The workloads indicate how much heat will be generated. The cold aisle temperature is upstream of the rack temperature, thus they are closely related. The future rack temperatures are affected by various factors, such as the historical rack temperatures, other racks in the same server room, IT workloads of the rack itself,

and the cold aisle temperature. Therefore, we set the temperature data as the prediction domain while the IT workload data and the cold aisle temperature data as the guidance domains. Since it takes time to exchange the heat in the air, the prediction targets are all racks' temperatures 10 min ahead.

4.2 Experiment Setup

The data preprocessing steps are as follows:

- Filter out abnormal data. The abnormal data are determined based on industry knowledge and data distribution. For example, the data is abnormal when the cold aisle temperature rises but the rack temperature does not, or the cold aisle temperature rises to 30 °C. According to the data distribution, the data points far from the distribution center are filtered out. Figure 3 shows the distribution of the normal and abnormal temperature. The horizontal coordinate represents the rack name. For example, A01 indicates the first rack in column A.
- Screening for fluctuating temperatures as training, validation, and test sets for the reason that we are only interested in predicting fluctuating temperatures than near-constant temperatures.
- Exponential smoothing is used to smooth the time series to further filter out the noise. The smoothing constant is set to 0.9.
- Min-max normalization.

In the experiments, the dimension of the weight vector in the attention layer is set to 256. Other network parameters are the same as those of the NRI. Adam [5] with the learning rate of 0.001, decayed by the factor of 0.5 every 100 epochs, is used as the optimizer. The maximum number of epochs is 500. The batch size is set to 16 and each batch has 48 time points for the RATEDC dataset. The reported results are averaged after 5 runs. Both of the datasets are split into training (80%), validation (10%), and testing sets (10%). The Mean Square Error (MSE) is used to evaluate the performance of the models.

4.3 Performance Comparison

To further study the effectiveness of the model, we compare I2A-RI with other advanced prediction algorithms as follows:

- ⋆ **VAR:** vector autoregression [9]
- ⋆ **ARIMA:** The auto-regressive integrated moving average [1]
- ⋆ **GRU:** Gated Recurrent Unit [3]
- ⋆ **TPA-LSTM:** A temporal pattern attention LSTM for multivariate time series forecasting [11]
- ⋆ **DARNN:** Dual-stage attention-based recurrent neural network [10]
- ⋆ **STGCN:** The spatio-temporal graph convolutional network [14]
- ⋆ **MTGNN:** The multivariate time Series forecasting with GNN [13]
- ⋆ **NRI:** Neural relational inference [7].

In Table 1, we present the performance of I2A-RI compared with VAR, ARIMA, GRU, TPA-LSTM, DARNN, STGCN, MTGNN, and NRI for the RAT-EDC datasets. The VAR model is a statistical method that represents a group of time-dependent variables as linear functions of their own past values and the past values of all other variables. The ARIMA model needs to transform the non-stationary time series into stationary time series first, then the predicted values depending on the past values, and the present and past values of the random error term. GRU, TPA-LSTM, and DARNN are deep learning models that can utilize the latent inter-dependencies among variables for prediction. STGCN, MTGNN, and NRI are graph-based prediction methods by modeling the relationships between variables as the graph to help make better predictions.

In the experiments, we divided the data into two categories: one is abnormal temperature caused by failure, and the other is normal data. In the data with the abnormal occurrence, we also evaluate the predicted data by two criteria: All conditions and $delta > 1\,^{\circ}\mathrm{C}$. The former calculates MSE on all of the actual and predicted target values in the testing set, while the latter calculates MSE only when there was $1\,^{\circ}\mathrm{C}$ or more temperature increase in 2.5 min. The "$delta > 1\,^{\circ}\mathrm{C}$" criterion is added because we want to catch a more significant temperature change in data center operations.

As depicted in Table 1, I2A-RI achieves the best performance over other methods. I2A-RI reduces the MSE of the second-best model (STGCN) by 2.88% with the criterion "All" and 4.30% with the criterion "$delta > 1\,^{\circ}\mathrm{C}$".

Table 1. Performance comparison (MSE) among different approaches.

	Abnormal		Normal
	All	$delta > 1\,^{\circ}\mathrm{C}$	All
VAR	0.0334 ± 0.001	0.4290 ± 0.02	2.58e−03 ± 1.17e−04
ARIMA	0.0354 ± 0.003	0.4016 ±0.02	8.90e−04 ± 1.16e−05
GRU	0.0291 ± 0.002	0.3591 ± 0.04	8.50e−04 ± 1.91e−05
TPA-LSTM	0.0270 ± 0.006	0.3271 ± 0.05	1.30e−03 ± 1.29e−05
DARNN	0.0254 ± 0.005	0.3267 ± 0.04	7.92e−04 ± 1.03e−06
STGCN	0.0243 ± 0.001	0.2019 ± 0.00	4.89e−04 ± 1.39e−06
MTGNN	0.0246 ± 0.001	0.2579 ± 0.02	4.59e−04 ± 1.14e−06
NRI	0.0263 ± 0.006	0.3132 ± 0.04	5.09e−04 ± 2.11e−06
I2A-RI	**0.0236 ± 0.002**	**0.1932 ± 0.02**	**4.54e−04 ± 1.01e−06**

4.4 Ablation Study

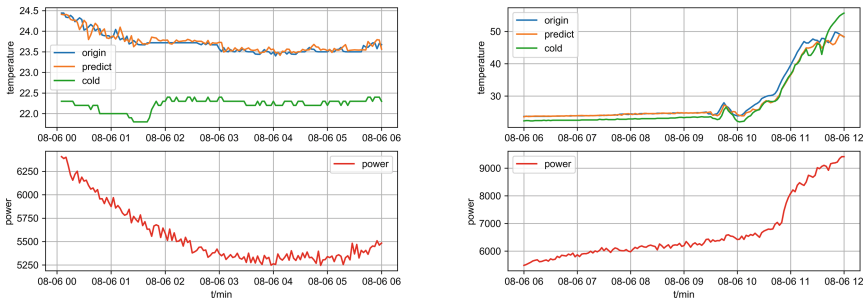
In this section, we aim to verify the effect of the two import modules: the GDA module and the IDA module. The methods without these two modules are denoted as follows:

- ⋆ + **GDA:** I2A-RI without IDA.
- ⋆ + **IDA:** I2A-RI without GDA.
- ⋆ **-GDA-IDA:** I2A-RI without either GDA or IDA

The performance of the three variants are shown in Table 2. The results show that both the GDA and IDA can improve the model's performance, which verifies the correctness of our conjecture about inter-and-intra domain attention.

Table 2. Effects of GDA and IDA modules (MSE).

	Abnormal		Normal
	All	*delta* >1 °C	All
-GDA-IDA	0.0263 ± 0.006	0.3132 ± 0.04	5.09e−04 ± 2.11e−06
+GDA	0.0240 ± 0.001	0.2511 ± 0.01	4.88e−04 ± 1.95e−06
+IDA	0.0226 ± 0.001	0.2700 ± 0.02	4.90e−04 ± 2.98e−06

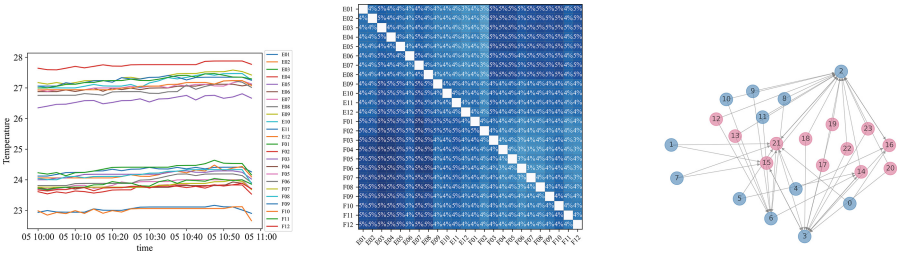


(a) in-distribution prediction (b) out-of-distribution prediction

Fig. 4. Examples of the ground-truth and predicted time series.

4.5 Visualization

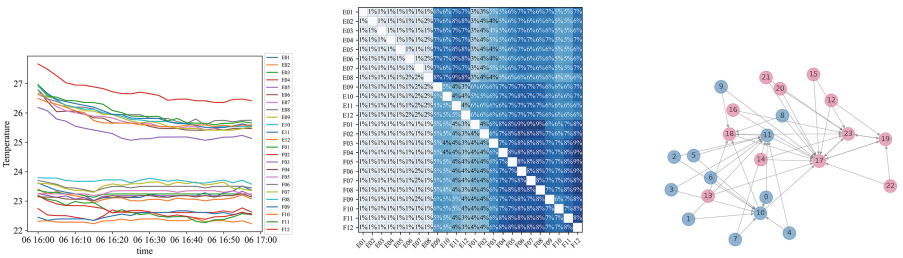
Additionally, we provide ground-truth and predicted time series trends of several selected variables from the dataset, as shown in Fig. 4. As can be seen from the



(a) Temperature trend (b) Adjacency matrix (c) Learned graph

Fig. 5. The adjacency matrix of the learned graph (same trend).

figures, when the predicted points are within the distribution of the training set, the model can predict them accurately; when the predicted points are beyond the distribution range, the model cannot capture this trend. We present the adjacency matrix and the learned graph of two columns of racks sharing the same cold aisle in Fig. 5 and 6.



(a) Temperature trend (b) Adjacency matrix (c) Learned graph

Fig. 6. The adjacency matrix of the learned graph (different trend).

As can be seen from Fig. 5, when the temperature curves of different racks change in a similar way and the trend is consistent, the adjacency matrix presents similar relationship weights. As can be seen from Fig. 6, when the temperature curves of different racks fluctuate in different ways, the relationship between different racks is obviously different in the adjacency matrix. In the learned graph, each rack shows only the other two racks with which they are most closely associated. They may be responsible for the same business, thus these learned relationships may help troubleshoot when faults occur.

5 Conclusions

In the motivation of modeling interactions between time series from different domains, we propose the I2A-RI model to utilize the information of the guidance

domain to learn a more accurate graph for prediction in this paper. Our results strongly prove that I2A-RI can learn underlying relations from data of different domains. With the learned structure, I2A-RI outperforms other state-of-the-art models for the RATEDC dataset. In the future, to better predict the values out of the data distribution, we will introduce more prior knowledge, including the pre-known dynamic and static information and the latest events.

Acknowledgements. This work is supported by Alibaba Post-doctoral Research Station and the National Key Research and Development Program of China No. 2020AAA0106300 and National Natural Science Foundation of China No. 62102222.

References

1. Adhikari, R., Agrawal, R.K.: An introductory study on time series modeling and forecasting. arXiv preprint arXiv:1302.6613 (2013)
2. Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: Time Series Analysis: Forecasting and Control. Wiley, Hoboken (2015)
3. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
4. El-Sayed, N., Stefanovici, I.A., Amvrosiadis, G., Hwang, A.A., Schroeder, B.: Temperature management in data centers: why some (might) like it hot. In: Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems, pp. 163–174 (2012)
5. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2014)
6. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
7. Kipf, T., Fetaya, E., Wang, K.C., Welling, M., Zemel, R.: Neural relational inference for interacting systems. arXiv preprint arXiv:1802.04687 (2018)
8. Kipf, T.N., Welling, M.: Variational graph auto-encoders. arXiv preprint arXiv:1611.07308 (2016)
9. Lütkepohl, H.: Vector autoregressive models. In: Handbook of Research Methods and Applications in Empirical Macroeconomics. Edward Elgar Publishing (2013)
10. Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., Cottrell, G.: A dual-stage attention-based recurrent neural network for time series prediction. arXiv preprint arXiv:1704.02971 (2017)
11. Shih, S.Y., Sun, F.K., Lee, H.Y.: Temporal pattern attention for multivariate time series forecasting. Mach. Learn. **108**(8–9), 1421–1441 (2019)
12. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
13. Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., Zhang, C.: Connecting the dots: multivariate time series forecasting with graph neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 753–763 (2020)
14. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875 (2017)
15. Zhou, J., et al.: Graph neural networks: a review of methods and applications. arXiv preprint arXiv:1812.08434 (2018)