4A-06

Time-delayed multivariate time-series prediction: a technical extension

Huy Quang Ung, Hao Niu, Habault Guillaume, Roberto Legaspi, Atsunori Minamikawa

KDDI Research, Inc., Saitama, Japan.

I. Introduction

A real-time monitoring and time-series collecting system often encounters disruption due to common issues such as hardware/software failure or network disconnection, leading to the problem of missing or time-delayed data. The performance of online time-series prediction models could degrade due to inputs that have missing values. A simple conventional approach to solve this problem is to replace the missing values with zeros or interpolate them. More recent studies utilized neural network to perform complex interpolation [1]. A new approach leveraging time-delayed complete data to enhance the time-series representation learning is introduced as enhanced representation learning (ERL), which demonstrated efficacy against previous methods to address this missing value problem [2]. The main idea is to learn representation alignment of complete and incomplete data using the concept of contrastive learning. Given a prediction model consisting of a representation block (encoder) and a prediction block (decoder), a training pipeline of ERL consists of three steps (as visualized in Fig. 1a): (1) learn the representation of complete data, (2) train the encoder f_a^r to align the representation vectors of complete and incomplete data, (3) use inputs with missing values to finetune the decoder f_c^p to obtain f_a^p.

In this paper, we investigate the technical extensions of ERL to enhance its performance while reducing the training time and memory consumption. Firstly, we merge steps (2) and (3) to force the prediction model to achieve a global optimal and reduce the training time. Secondly, we investigate the effect of different mini-batch sizes in learning representation alignment. Using less the batch-size directly reduces the memory consumption during training phase.

II. Methodology

We consider time-delayed complete multivariate time-series (MTS) data X_c and corresponding incomplete MTS X_m as inputs for MTS prediction models, where X_c and $X_m \in \mathbb{R}^{H\times N}$; H and N are the number of time slots and the number of variables, respectively. Y and $\hat{Y} \in \mathbb{R}^{L\times N}$ are truth and predicted values, where L is the number of prediction time slots.

1. ERL

We briefly summarize the main idea of ERL [2]. This method originally consists of three-step training.

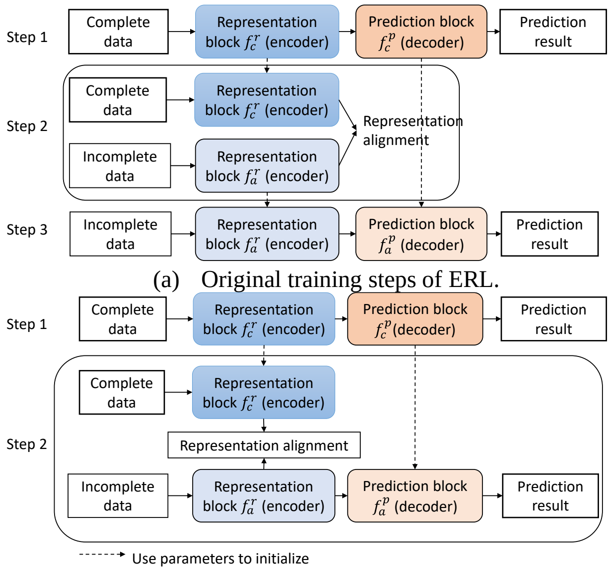
- Step 1: train $f_c^r: X_c \to R_c$ and $f_c^p: R_c \to \hat{Y}$ using mean square error loss (MSE), i.e., $\mathcal{L}_{MSE}(Y, \hat{Y})$.
- Step 2: train $f_a^p: X_a \to R_a$ so that $f_a^r(X_a)$ is similar to $f_c^r(X_c)$, using a loss function as follows:

$$\mathcal{L}_{align}(R_a, R_c) = \mathcal{L}_{dis}(R_a, R_c) + \alpha\mathcal{L}_{CL}(R_a, R_c),$$

where $\mathcal{L}_{dis}$ and $\mathcal{L}_{CL}$ are a distance-based alignment loss and a contrastive learning-based alignment loss, respectively; $0 < \alpha \leq 1$ is a hyper-parameter. Distance-based alignment loss, a loss to force R_a and R_c to be close to each other, is computed as follows: $\mathcal{L}_{dis}(R_a, R_c) = \left\|R_a - R_c\right\|_2$. Besides, contrastive learning-based alignment loss accounts for sample differences in a mini-batch, where we need to set up positive and negative pairs. For simplicity, we want R_a^i to be close as possible to R_c^i (i is the index in the mini-batch) so that they are a positive pair. By contrast, R_a^i and R_a^j or R_c^j $(i \neq j)$ are negative pairs. The loss is computed as follows:

$$\mathcal{L}_{CL}(R_a, R_c) = -\log\frac{\exp\left(R_a^i, R_c^i\right)}{\sum_j\left(\exp\left(R_a^i, R_a^j\right) + \exp\left(R_a^i, R_c^j\right)\right)}$$

- Step 3: train $f_a^p: R_a \to \hat{Y}$ with weight initialization of f_c^p by the MSE loss.



(b) Our proposed training steps for Ext_ERL.

Figure 1. Training steps of ERL and our Ext_ERL.

2. Our extension of ERL

We perform steps 2 and 3 simultaneously to force the prediction model to achieve global optimal. Fig. 1b presents an overview of our extension for ERL (denoted as Ext_ERL) consisting of two-step training.

Step 1 in our Ext_ERL is the same as in ERL. In the step 2, we use the loss function below to train f_a^r and f_a^p:

$$\mathcal{L}(R_a, R_c) = \mathcal{L}_{dis}(R_a, R_c) + \alpha\mathcal{L}_{CL}(R_a, R_c) + \beta\mathcal{L}_{MSE}(Y, \hat{Y})$$

where α and β are hyper-parameters.

III. Experiments

1. Experimental settings

Table 1. Prediction errors on the testing set of incomplete data.

Missing ratio	0.2		0.5		0.8	
Method/Error	RMSE	MAE	RMSE	MAE	RMSE	MAE
Informer'	5.83	3.08	6.02	3.20	6.48	3.38
Informer*	5.72	3.00	6.07	3.18	6.47	3.36
ERL	5.68	2.96	**5.97**	3.12	**6.43**	**3.30**
Ext_ERL	**5.63**	**2.91**	6.08	**3.11**	6.47	3.33

We evaluated Ext_ERL on the PeMs-Bay dataset. We chronologically divided the dataset into training, validation, and testing sets with a ratio of 6:2:2. The model looked back 24 hours to predict 12 hours ahead. The granularity of this dataset is 5-minute interval. We use the Informer [3] as prediction model (same as [2]). The settings of Informer are also the same as [2]. α and β are both set to 1. The mini-batch size is 16. We performed our experiments on a Linux server with a Tesla P100 16Gb GPU.

Originally, the PeMs-Bay dataset did not have missing values. We simply simulated for the missing value that randomly occurs in the dataset with a pre-defined missing ratio, and we made this to happen simultaneously to all variables.

We compared our Ext_ERL to the original ERL and the top baselines in [2], which are the Informer* and the Informer'. Informer is not originally designed to handle the missing value. Therefore, Informer* and Informer' are Informers in which zero-imputation and linear-imputation were applied to the input before going into the Informers, respectively.

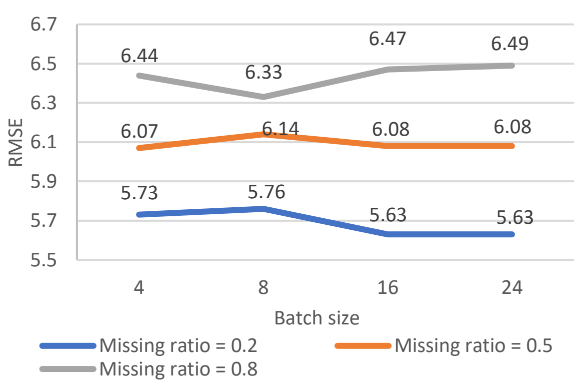


Figure 2. Performance of Ext_ERL with varying the batch-size.

2. Results

Table 1 presents the model prediction errors with incomplete input data. The overall missing ratio are 0.2, 0.5 and 0.8. The error in each experiment is the average error value of three executions with different seeds. We observed that Ext_ERL achieves a better performance than ERL in several cases. Ext_ERL performs better than Informer' and Informer* in most of cases. Ext_ERL seems to perform worse compared to ERL when we increase the missing ratio. The training time of Ext_ERL is 40% faster than ERL (shown in Table 2).

3. Effect of different mini-batch sizes

Table 2. Comparison of training time between ERL and Ext_ERL.

	Training time (minutes)
ERL	53
Ext_ERL	33

We evaluated the effect of different mini-batch sizes by increasing the size in {4, 8, 16, 24}. Due to memory limitation, we could not use larger batch-sizes.

Fig. 2 presents the RMSE values of Ext_ERL in this experiment. With the missing ratio of 0.2, increasing the batch-size to 16 and 24 achieves better performance. With the missing ratio of 0.5 and 0.8, using small batch-sizes (i.e., 4 and 8 respectively), however, performs better than using larger ones.

IV. Conclusion and future work

In this paper, we conducted extensive experiments for ERL to address time-delayed multivariate time-series prediction. We firstly reduced the number of training steps in ERL by merging steps 2 and 3. Results showed that Ext_ERL achieved better performance than ERL in several cases while reducing the training time. Secondly, we evaluated the effect of different mini-batch sizes in representation alignment learning based on contrastive learning. We will consider complex definitions of positive and negative pairs as future work.

Acknowledgement

This research and development include the results of "Research and Development for AI optimization technology through safe data coordination" (JPMI00316) of the Ministry of Internal Affairs and Communications.

References

[1] X. Miao, Y. Wu, J. Wang, Y. Gao, X. Mao and J. Yin, "Generative Semi-supervised Learning for Multivariate Time Series Imputation," in *Proc. of the AAAI*, 2021.

[2] H. Niu, H. Guillaume, R. Legaspi, C. Meng, D. Cao, S. Wada, C. Ono and Y. Liu, "Time-delayed Multivariate Time Series Predictions," in *Proc. of the 2023 SIAM International Conference on Data Mining (SDM)*, Minnesota, U.S., 2023.

[3] 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 *Proc. of the AAAI*, Vancouver, Canada, 2021.