SpatioTemporal Transformer-based Regressive Domain Adaptation for Remaining Useful Life Prediction

HyunYong Lee, Nac-Woo Kim, Jungi Lee, and Seok-Kap Ko
Honam Research Center (HRC)
Electronics and Telecommunications Research Institute (ETRI)
Gwangju, Korea
{hyunyonglee,nwkim,jungi,softgear}@etri.re.kr

***Abstract*—In realizing the accurate remaining useful life (RUL) that is quite important in many industrial areas, the data-driven domain adaptation (i.e., the regressive domain adaptation) has been widely used. In designing the effective regressive domain adaptation model, there are two main issues: model architecture and loss functions. In this paper, we first propose the spatiotemporal transformer-based model to effectively extract features. The proposed transformer considers the spatial relationships across multiple sensors and the temporal relationships of each sensor. We also discuss the usefulness of three loss functions in terms of domain adaptation.**

***Index Terms*—Remaining useful life, regressive domain adaptation, transformer, training loss.**

I. Introduction

Accurate prediction of remaining useful life (RUL) is a key factor to realize in-time maintenance for good operating conditions of target systems by avoiding unexpected system failures [1]. In realizing the RUL prediction through the data-driven method that has attracted attention [2], one practical challenging issue is the lack of labeled data. One solution widely used for this issue is domain adaptation [3]. With the domain adaption technique, a model is first trained using the labeled data of the source domain (e.g., the domain where it is relatively easy to acquire labeled data through repetitive experiments or simulations) and the trained model is applied to the target domain (i.e., the domain that we are interested in and the labeled data is not available). Assuming that the source domain and the target domain are likely to have similar features (e.g., operating conditions, weather conditions, and so on), one main issue for successful domain adaptation is to extract domain-invariant features for both the source domain and the target domain.

In this paper, to realize successful regressive domain adaptation for RUL prediction, we propose a spatiotemporal transformer-based model. With the spatiotemporal transformer, we try to consider the spatial relationships across multiple sensors and the temporal relationships of each sensor. We also discuss training losses to effectively train a model. In particular, we examine the three losses including RUL regression loss, domain classification loss, and reconstruction loss.

This work was supported by Electronics and Telecommunications Research Institute(ETRI) grant funded by the Korean government [23ZK1100, Honam region regional industry-based ICT convergence technology advancement support project].

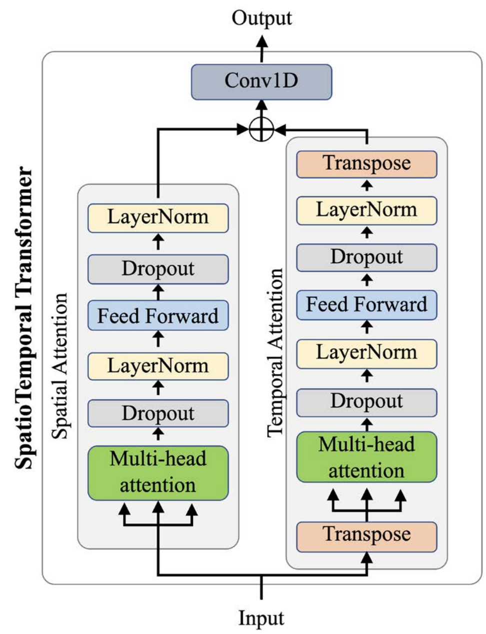

Fig. 1: Proposed SpatioTemporal Transformer block.

To verify our approach, we conduct experiments using the well-known aircraft engine dataset (i.e., C-MAPSS) [5]. RUL regression loss alone is also not meaningful because it does not consider the target domain data. With the RUL regression loss only, the RMSE of the source domain and the target domain is 37.27 and 50.36. We also found that the reconstruction loss of the AutoEncoder-style model is not meaningful. With the RUL regression loss and the reconstruction loss, the RMSE of the source domain and the target domain is 38.63 and 50.41. On the contrary, domain prediction loss plays an important role in extracting the domain-invariant. With the RUL regression

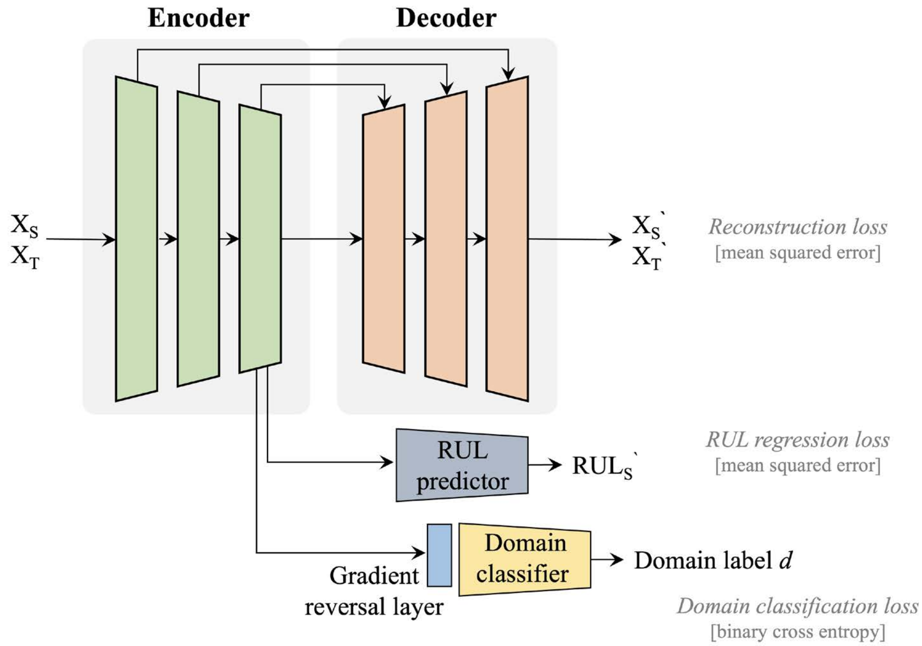

Fig. 2: Spatiotemporal transformer-based AutoEncoder model.

loss and the domain classification loss (with or without the reconstruction loss), the RMSE of the source domain and the target domain is 32.52 and 33.42

The rest of this paper is organized as follows. In Section II, we describe our proposed model and related loss terms. In Section III, we show the experimental results. In Section IV, we conclude this paper.

II. Proposed Approach

A. SpatioTemporal Transformer

The typical input data for RUL prediction is multivariate time series data (i.e., time series data of multiple sensors). One basic approach to extracting meaningful features from the multivariate time series is to consider both spatial relationships across sensors and temporal relationships across the time domain of each sensor. Observing that Transformer architecture [4] has shown promising results in various areas, we adopt Transformer architecture for our purpose. Adopting the transformer architecture, we first propose a SpatioTemporal Transformer (STT) block to be used as a building block of the AutoEncoder model described in Section II.B.

Fig. 1 shows our proposed STT block. STT block includes two sub-blocks. The first sub-block is for spatial attention. Given input data (e.g., in the form of batch_size x window_size x num_sensors), multi-head attention, dropout, layer normalization, feed-forward network, dropout, and layer normalization operations are applied sequentially. The second sub-block is for temporal attention. To examine the temporal relationship, the input data is first transposed into the form of batch_size x num_sensors x window_size. Then, multi-head attention, dropout, layer normalization, feed-forward network, dropout, and layer normalization operations are applied sequentially like in the case of the spatial attention sub-block. Again, the transpose operation is applied to transform the output shape into batch_size x window_size x num_sensors. Finally, the outputs of the spatial attention and the temporal attention sub-blocks are added. The added output is processed by 1D convolutional operation to produce a final output.

B. SpatioTemporal Transformer-based Model

Following the well-known domain adaptation technique [6], we propose a new feature extractor using STT as a basic building block. Fig. 2 shows our approach.

STT-based AutoEncoder. The first main component is the STT-based AutoEncoder. Encoder and Decoder have three layers. Each layer of encoder and decoder is the STT block. An encoder layer and a decoder layer are the same except for the parameters of the last 1D convolutional layer. The 1D convolutional layer of the encoder layer reduces the data dimension. On the contrary, the 1D convolutional layer of the decoder layer expands the data dimension. There are three residual connections between the corresponding encoder and decoder layers. AutoEncoder tries to extract useful features by trying to reconstruct the input from the compressed data.

RUL predictor. The second component of the proposed model is the RUL predictor which consists of sequential feed-forward layers. The objective of this component is to predict the RUL of the source domain data. The input to this component is the features of the source domain data and the training label is the corresponding RUL value.

Domain classifier. The third component of the proposed model is the domain classifier. The domain classifier is connected to the encoder part via the gradient reversal layer [6].

TABLE I: Performance of RUL prediction (in RMSE).

Loss	Encoder only		Encoder + Decoder	
	Source(FD001)	Target(FD003)	Source(FD001)	Target(FD003)
RUL	36.51 (7.93)	49.83 (6.76)	37.27 (8.10)	50.36 (6.27)
RUL, Domain	32.33 (5.27)	33.44 (5.28)	32.52 (4.63)	33.42 (5.08)
RUL, Recon.	N/A		38.63 (8.30)	50.41 (7.14)
RUL, Domain, Recon.	N/A		32.58 (5.11)	33.10 (4.82)

The objective of this component is to classify the domain of the given features. The input to this component is the features of the source and target domain data and the training label is the true domain value of the input data.

C. Loss Terms

To extract the domain-invariant features, we examine three loss terms.

RUL regression loss. In building an RUL predictor that will be used for the target domain, RUL predictor needs to be accurate first for the source domain data. For this, an RUL predictor is trained with the extracted features of the source domain data and the true RUL value. The mean squared error is applied for (the true RUL value, the predicted RUL value) as RUL regression loss.

Domain classification loss. The domain classifier is connected to the AutoEncoder via the gradient reversal layer. The role of the gradient reversal layer is to reverse the gradient when applying the back-propagation process. In other words, the domain classifier tries to correctly classify the input data while the Encoder tries to fool the domain classifier by generating the domain-invariant features. Actually, the domain classification loss is the key part to produce the domain-invariant features [6]. The binary cross entropy is applied for (the true domain label, the predicted class probabilities) as the domain classification loss.

Reconstruction loss. Using the AutoEncoder model, we can apply the reconstruction loss. The rationale for using the reconstruction loss is to further improve the ability in producing the domain-invariant feature. An AutoEncoder tries to map the given input data into the same feature space to extract the meaningful features to be used as a seed for the reconstruction of the original input data. The mean squared error is applied for (the input data, the reconstructed data) as the reconstruction loss.

III. Experimental Study

A. Experiment Setting

To verify the feasibility of our method, we conduct a series of experiments using C-MAPSS dataset [5]. The C-MAPSS dataset contains information about the degradation of the aircraft engine. The target aircraft engine includes various sub-components such as high pressure compressor, low pressure compressor, fan, combustor, low pressure turbine, and high pressure turbine. The dataset includes multiple time series data of each operational setting. Each time series data include 21 onboard sensor readings and operation settings including measuring temperature, pressure, and speed. The entire dataset contains four sub-datasets, i.e., about four operational conditions and fault conditions. Among the four datasets, we choose FD001 as the source domain and FD003 as the target domain. FD001 has 1 operational condition and 1 fault condition. FD003 has 1 operational condition and 2 fault conditions. The number of train data (i.e., the number of engines) is 100 and the number of test data is 100 for both FD001 and FD003. We run each experiment N times and show the average.

B. Results

Table I shows the experiment results. The numbers inside the parenthesis indicate the standard deviation. The results first show that using the encoder only is enough. Using the AutoEncoder-style model with the reconstruction loss does not improve the prediction accuracy. Among the three losses we examined, the key loss is the domain classification loss. The domain classification loss together with the gradient reversal layer is the key player in enabling domain-invariant feature for the regressive domain adaptation. The RMSE of the target domain decreases from 50.36 to 33.42. Another interesting observation is that using the domain classification loss improves the RUL prediction performance of the source domain (i.e., from 37.27 to 32.52). It seems that the domain-invariant features of the target domain data may be augmented data to the source domain.

IV. Conclusion

In this paper, we propose the spatiotemporal Transformer-based model and examine the three losses with the model. We found that using the Encoder only is enough and the domain classification loss is enough to improve the prediction accuracy of both the source domain and the target domain.

References

[1] L. Liao, "Discovering prognostic features using genetic programming in remaining useful life prediction," *IEEE Transactions on Industrial Electronics*, vol.61, p.2464-2472, 2014.

[2] Y. Lei, N. Li, L. Guo, and N. Li, "Machinery health prognostics: A systematic review from data acquisition to RUL prediction," *Mechanical Systems and Signal Processing*, vol.104, p.799-834, 2018.

[3] HyunYong Lee, Nac-Woo Kim, Jun-Gi Lee, and Byung-Tak Lee, "Towards uncertainty-aware remaining useful life prediction via domain adaptation," in *Proc. of ICTC*, 2022.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I Polosukhin, "Attention is all you need," in *Proc. of Advances in Neural Information Processing Systems*, 2017.

[5] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *Proc. of International Conference on Prognostics Health Management*, 2008.

[6] Y. Ganin, et. al., "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol.7, pp.1-35, 2016.