MULTIPLE TEMPORAL CONTEXT EMBEDDING NETWORKS FOR UNSUPERVISED TIME SERIES ANOMALY DETECTION

Hanhui Li[⋆] *Xinggan Peng*[†] *Huiping Zhuang*[†] *Zhiping Lin*[†]

[⋆] Shenzhen Campus of Sun Yat-sen University, Shenzhen 518107, China
[†] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798

ABSTRACT

Unsupervised anomaly detection for time series signals is challenging, due to the imbalanced distribution of data and the lack of ground-truth labels. Current methods on this topic are mainly based on deep neural networks, which are optimized by heuristic constraints or empirical priors. However, various patterns of anomalous data, especially those lasting for varying periods, are hard to be captured by plain networks. To tackle this problem, we propose a multiple temporal context embedding method. The core of our method is to construct a unified representation of the multiple temporal contexts of data, which is achieved by learning a set of base features to reconstruct the hidden features within existing anomaly detection networks. The proposed method can be implemented as a convenient plug-in module, and be combined with various network architectures, such as autoencoders and graph neural networks. Extensive experiments on multiple datasets demonstrate that the proposed method can boost the performance of baseline networks significantly.

***Index Terms*—** Anomaly Detection, Unsupervised Learning, Temporal Context, Time Series Signal Processing

1. INTRODUCTION

Anomaly detection is an important research topic in the fields of signal processing [1], data mining [2], and artificial intelligence [3]. In real-life scenarios, compared with normal instances, anomalies are rare and hence it is difficult to collect and label the abnormal data [4]. Therefore, unsupervised anomaly detection has attracted increasing attention and has a wide range of applications [5, 6, 7].

Recently, extensive deep learning based methods have been proposed to tackle unsupervised anomaly detection, such as autoencoder [8, 9], recurrent neural networks (RNNs) [5, 10], graph neural networks (GNNs) [11], and generative adversarial networks (GANs) [12, 13]. Under the constraint of unsupervised learning, existing methods usually adopt heuristic optimization objectives, such as adversarial learning [13] and reconstructing inputs with their latent features [14]. However, due to complex and heterogeneous anomalous patterns, existing methods need to be tailored carefully to achieve satisfying performance. For instance, a large detection window tends to ignore point anomalies, because of the loss of details in feature extraction. On the other hand, a small detection window is insufficient in capturing group anomalies that are long sequences. Worse, anomalies may appear in the training data, and consequently they will disrupt the training process and result in degraded performance.

Therefore, in this paper, we propose to exploit the multiple contexts of time-series data to better tackle the anomaly detection problem. Specifically, we aim at finding a unified embedding of the hidden features extracted by multiple modules of a network, so that the embedded features can preserve and merge the important details in multiple hidden features. The embedding is achieved by representing the hidden features as the combination of a set of base features, which are designed to reduce the effects of anomalous data. The embedded features are then utilized in two ways: First, they are adopted as the refined representation of data. Second, they are used to generate auxiliary predictions, which can help in network training and inference. Furthermore, the proposed method can be implemented as a convenient plug-in for mainstream networks. We apply the proposed method into a convolutional autoencoder and a GNN of hybrid architectures, and our experiments validate that the proposed method improves the performance of the two baselines consistently.

In summary, this paper has the following main contributions: First, a multiple temporal context embedding method is proposed to refine the features of multivariate time series data. Second, the proposed method can generate auxiliary predictions to detect anomalies, and be implemented as a plug-in module for various network architectures. Finally, the proposed method is evaluated on three public datasets for anomaly detection, and achieves state-of-the-art performance.

2. RELATED WORK

Traditional anomaly detection methods [15, 16] are mainly based on hand-crafted features with well-designed classifiers,

This work was partially supported by the National Natural Science Foundation of China under Grant No. 61902088, No. 61976233 and No. 61936002. Huiping Zhuang is the corresponding author (Email: HUIPING001@e.ntu.edu.sg).

distance measures, or statistical models. Due to the advantages in end-to-end learning and handling high-dimensional data, deep learning based methods have become the popular solution for anomaly detection. In spite of the various network modules and architectures, representation learning that extracts important features from data, is the core of deep neural networks. RNNs, especially the long short-term memory (LSTM) network and its variants [1, 5, 7, 10, 17], are one of the major architectures for representation learning with time series data, because of their capacity for modeling temporal dependencies in data. Yet RNNs are sequential models, which means they are hard to process data in parallel. Hence, convolutions have also been adopted in existing methods [11, 18].

The extracted features are then utilized to calculate anomaly scores based on forecasting models or reconstruction models [11]. Forecasting models predict the arriving data and use the prediction errors as anomaly scores, such as the LSTM-NDT method [5]. The reconstruction models try to restore the input data based on the extracted features, and typical examples of these methods are autoencoders [8, 9, 14] and GANs [2, 12, 13]. These methods assume that the abnormal inputs cannot be restored well, and hence the anomaly scores can be estimated based on the reconstruction errors. Interested readers can refer to [4, 19] for the comprehensive survey on deep learning methods for anomaly detection.

In short, feature representation plays an important role in unsupervised anomaly detection. Thus, we focus on improving the features of time series data via exploring their temporal contexts. In this way, our method is independent from a particular module for representation learning and anomaly score estimation, and hence it differs from previous methods.

3. THE PROPOSED METHOD

In this section, we present the details of our multiple temporal context embedding (MTCE) method. We begin by formulating the MTCE method in Section 3.1, and then present the usage of our method for anomaly detection, with two existing methods as the baselines in Section 3.2.

3.1. Multiple Temporal Context Embedding

The key idea of our MTCE method is to merge and refine the hidden features that encode the temporal contexts of data. Formally, assume our target network consists of L hidden layers and their outputs are denoted as $H = \{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_L\}$, respectively. For the convenience of discussion, we also assume that the prediction of the network is generated by a module f with $\mathbf{h}_L$ as its input. The goal of our MTCE method is to generated the refined $\mathbf{h}_L$ based on H (denoted as $\mathbf{h}_L^*$), and use $f(\mathbf{h}_L^*)$ as the auxiliary prediction to help detection.

Considering that the features in H are from different feature spaces, we first apply L feature transformations to project them into the same space. The transformations can be implemented by multilayer perceptrons (MLPs) or convolutional layers. After that, we have $N = \sum_{l=1}^{L} N_l$ feature vectors together, where N_l is the length of $\mathbf{h}_l$ after the transformation. Let $\mathbf{x} \in \mathbb{R}^C$ denote one of these feature vectors and C is the number of channels. Since these features are embedded into the same space, we can calculate the similarity among them as follows:

$$s_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j), \tag{1}$$

where $k(\cdot,\cdot)$ can be any similarity measures, such as the radial basis function. In this paper, we simply use the dot product, i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$. Note that anomalous data are rare and different from normal data. To identify anomalies and reduce their effects, for each $\mathbf{x}$, we propose to calculate the sum of the similarity between it and other feature vectors, so that the lower the sum is, the more likely the feature vector is anomalous. We consider the top-N^* feature vectors with the largest sum of similarities as our bases for feature reconstruction. We propose to reconstruct $\mathbf{x}$ as follows:

$$\mathbf{x}^* = \mathbf{x} + g(\sum_{n=1}^{N^*} k(\mathbf{x}, \mathbf{x}_n)\mathbf{x}_n), \tag{2}$$

where $g(\cdot)$ is another feature transformation. $\forall \mathbf{x} \in \mathbf{h}_L$, We apply Eq. (2) to obtain the embedded features $\mathbf{h}_L^*$.

3.2. Anomaly Detection with Embedded Features

The proposed MTCE method can be applied to various networks, as it does not depend on a specific modules or architecture. Here we use the convolutional autoencoder (Conv-AE) [18] and the MTAD method [11] as the baselines to demonstrate the usage of the proposed method, as shown in Fig. 1.

Conv-AE with MTCE Utilizing Conv-AE to detect anomalies consisting of two major steps. First, an encoder is used to convert the input signals into the hidden features (i.e., $\mathbf{h_L}$ in our case). After that, a decoder uses the hidden features to restore the input signals. Formally, let $\mathbf{y}$ denote the input signals, training Conv-AE can be done by minimizing the following lost function:

$$L_{AE} = \|\mathbf{y} - f(\mathbf{h}_L)\|_2, \tag{3}$$

where $\|\cdot\|_2$ is the L_2 norm. Note that Conv-AE is trained via end-to-end learning but we omit the function of the encoder for the conciseness of expression. Conv-AE assumes that the anomalous data cannot be reconstructed well, therefore, if $\|\mathbf{y} - f(\mathbf{h}_L)\|_2$ is larger than a predefined threshold, $\mathbf{y}$ is considered anomalous.

To apply our MTCE method into Conv-AE, we simply modify the above loss function as follows:

$$L'_{AE} = \|\mathbf{y} - f(\mathbf{h}_L)\|_2 + \lambda \|\mathbf{y} - f(\mathbf{h}_L^*)\|_2, \tag{4}$$

where λ is a trade-off factor. The criterion of anomaly is adjusted accordingly, namely, if $\|\mathbf{y} - f(\mathbf{h}_L)\|_2 +$

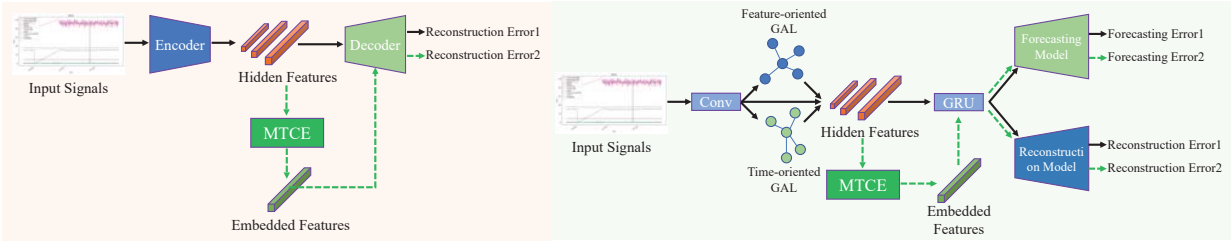

Fig. 1. Demonstrate of the utilization of our MTCE method with various existing networks (left: convolutional autoencoder [18], right: MTAD [11]). The workflows related to the proposed method are marked by green dash lines.

$\tau \|\mathbf{y} - f(\mathbf{h}_L^*)\|_2$ is larger than a threshold, then $\mathbf{y}$ is abnormal, where τ is also a trade-off factor.

MTAD with MTCE Unlike Conv-AE that is composed of convolutions, MTAD is a more complex network of hybrid architectures, including convolutions, graph attention layers (GALs), and gated recurrent units (GRUs). Hence, MTAD can be considered as a good example to validate that the proposed method can be combined with various architectures.

MTAD adopts a convolutional layer, followed by two parallel GALs for feature extraction. The concatenation of the features from these three layers is fed into the GRU and then processed by a forecasting model and a reconstruction model. To utilize the MTCE method with MTAD, we consider the concatenated features as $\mathbf{h}_L$. The objective function of MTAD can be summarized as follows:

$$L_{MTAD} = L_{For}(\mathbf{y}, \mathbf{h}_L) + L_{Rec}(\mathbf{y}, \mathbf{h}_L), \tag{5}$$

where the L_{For} term is calculated based on the prediction of the forecasting model, and the L_{Rec} term is based on that of the reconstruction model. Due to the page limitation, we omit the detailed definitions of these two terms, and focus on how to incorporate our embedded features into the MTAD framework. Similar to Conv-AE with MTCE, we extend Eq. (5) as follows:

$$\begin{aligned} L'_{MTAD} =& L_{For}(\mathbf{y}, \mathbf{h}_L) + \lambda L_{For}(\mathbf{y}, \mathbf{h}_L^*) \\ &+ L_{Rec}(\mathbf{y}, \mathbf{h}_L) + \lambda L_{Rec}(\mathbf{y}, \mathbf{h}_L^*). \end{aligned} \tag{6}$$

Accordingly, the anomaly score of MTAD is modified. Let $e(\mathbf{y}, \mathbf{h}_L)$ denote the original anomaly score estimated by MTAD, then our corresponding estimation is given by $e(\mathbf{y}, \mathbf{h}_L) + \tau e(\mathbf{y}, \mathbf{h}_L^*)$.

4. EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the proposed MTCE method. We first report the experimental setup, and then report the results on public datasets. An ablation study is also conducted to provide the comprehensive analysis of our method.

Table 1. Statistics of datasets for evaluation.

	Dims	Train	Test	Anomalies (%)
SMAP	25	135,183	427,617	0.1279
MSL	55	58,317	73,729	0.1053
SKAB	8	13,600	37,459	0.3535

4.1. Setup

Our experiments are conducted on three public time series datasets, including SMAP, MSL [5], and SKAB [18]. The dimensions of data, training/test sizes, and anomaly ratios of these datasets are listed in Table 1. We adopt the evaluation metrics used by these datasets, including $F_1 = \frac{2*\text{precision}*\text{recall}}{\text{precision}+\text{recall}}$, precision $= \frac{\text{TP}}{\text{TP+FP}}$, recall $= \frac{\text{TP}}{\text{TP+FN}}$, false alarm rate FAR $= \frac{\text{FP}}{\text{FP+TN}}$, and missing alarm rate MAR $= \frac{\text{FN}}{\text{FN+TP}}$. TP, TN, FP, and FN denote the number of true positive, true negative, false positive, and false negative, respectively. Results of the state-of-the-arts are quoted from [2, 11] and [18].

Implementation Details Our MTCE method is implemented with PyTorch and all experiments are conducted with a GTX 1080 Ti graphics card. The number of base features $N^* = 0.5N$, where N is the number of all hidden feature vectors. On SMAP and MSL, the auxiliary scaling factors are set to $\lambda = \tau = 1$, while on SKAB, $\lambda = 2$ and $\tau = 1$. All features transformations within the MTCE module are implemented by single-layer 1D convolutions of size 1×1, with 64 kernels on SMAP and MSL, and with 16 kernels on SKAB. We adopt the implementation of Conv-AE from `https://github.com/waico/SKAB`, and that of MTAD from `https://github.com/ML4ITS/mtad-gat-pytorch`, with their default parameter settings and experimental protocols. Our code is available at `https://github.com/hanhuili/MTCE-AnomalyDetection`.

4.2. Results

Table 2 and Table 3 report the results of our MTCE method with MTAD as the baseline on SMAP and MSL respectively. Results marked with $*$ are obtained in our own experiments. From the results we can see that our method improves the per-

Table 2. Comparison of the proposed method against other methods on the SMAP dataset.

Method	$F_1 \uparrow$	Precision ↑	Recall ↑
DAGMM [9]	0.7105	0.5845	0.9058
LSTM-VAE [7]	0.7298	0.8551	0.6366
GAN-Li [12]	0.7579	0.6710	0.8706
KitNet [14]	0.8014	0.7725	0.8327
MAD-GAN [13]	0.8131	0.8049	0.8214
OmniAnomaly [17]	0.8434	0.7416	0.9776
USAD [2]	0.8634	0.7697	0.9831
LSTM-NDT [5]	0.8905	0.8965	0.8846
MTAD [11]	0.9013	0.8906	0.9123
MTAD*	0.9043	0.8806	0.9293
MTAD* + MTCE	**0.9469**	**0.8996**	**0.9994**

Table 3. Comparison of the proposed method against other methods on the MSL dataset.

Method	$F_1 \uparrow$	Precision ↑	Recall ↑
LSTM-NDT [5]	0.5640	0.5934	0.5374
LSTM-VAE [7]	0.6780	0.5257	0.9546
DAGMM [9]	0.7007	0.5412	**0.9934**
KitNet [14]	0.7031	0.6312	0.7936
GAN-Li [12]	0.7823	0.7102	0.8706
MAD-GAN [13]	0.8747	0.8517	0.8991
OmniAnomaly [17]	0.8989	0.8867	0.9117
MTAD [11]	0.9084	0.8754	0.9440
USAD [2]	0.9272	0.8810	0.9786
MTAD*	0.9498	0.9546	0.9451
MTAD* + MTCE	**0.9621**	**0.9661**	0.9582

formance of the baseline consistently on both datasets. Particularly, the performance gains of the proposed method on the SMAP dataset are notable, with the increase of 4.26% in F_1 score and 7.01% in recall rate. Furthermore, our method obtains the highest performance on SMAP with all three metrics. On MSL, the recall of DAGMM is higher than that of ours, yet its F_1 score and precision rate are lower than those of ours (70.07% vs. 96.21%, and 54.12% vs. 96.61%), and hence the performance of our method is more balanced.

Table 4 reports our results on the SKAB dataset with Conv-AE as the baseline. Our method also boosts the performance of the baseline with all three metrics. We notice that the performance gains of our method with Conv-AE are less than those with MTAD. This is reasonable because MTAD is more complex and robust than Conv-AE. In fact, Conv-AE only adopts two convolutional layers as its encoder, and hence its representation ability is weaker than that of MTAD.

In the proposed method, the embedded features are used for feature refinement and predicting auxiliary predictions. We conduct an ablation study to understand the effects of these two strategies. We consider two variants, i.e., MTCE with only the original prediction (denoted as MTCE w/ Orig.) and MTCE with only the auxiliary prediction (denoted as MTCE w/ Aux.). The former variant can validate the advantages of feature refinement while the latter is for analyzing the auxiliary prediction. The experimental results are reported in Table 5. We can see that both strategies help to improve the F_1 score, and the feature refinement mainly helps in recall while the auxiliary prediction helps in precision, and combing both strategies allows us to get the most balanced performance. As for the complexity of our MTCE module, the numbers of parameters of Conv-AE and MTAD are $12,648$ and $361,327$, while those of the augmented methods are $13,720$ and $379,058$, which are ignorable. In summary, our experimental results validate that the MTCE module is an attractive lightweight module.

Table 4. Comparison of the proposed method against other methods on the SKAB dataset.

Method	$F_1 \uparrow$	FAR ↓	MAR ↓
Isolation Forest [15]	0.40	**0.0686**	0.7209
Autoencoder [8]	0.45	0.0756	0.6657
LSTM-VAE [18]	0.56	0.0913	0.5503
MSCRED [3]	0.64	0.1356	0.4116
LSTM-AE [18]	0.68	0.1424	0.3556
MSET [16]	0.73	0.2082	0.2008
Conv-AE [18]	0.79	0.1369	0.1777
Conv-AE*	0.8157	0.1455	0.1279
Conv-AE* + MTCE	**0.8258**	0.1422	**0.1137**

Table 5. Ablation study on the SMAP dataset.

Method	$F_1 \uparrow$	Precision ↑	Recall ↑
Baseline	0.9043	0.8806	0.9293
MTCE w/ Orig.	0.9222	0.8768	0.9726
MTCE w/ Aux.	0.9084	0.9155	0.9013
MTCE	0.9469	0.8996	0.9994

5. CONCLUSION

In this paper, we have proposed a multiple temporal context embedding method to tackle the anomaly detection problem for time series data. Our embedding method merges the hidden features of various network modules, and adopts a set of base features to refine the hidden features. Furthermore, auxiliary predictions are generated by the refined features to help to detect anomalies. We have combined our method with a simple convolutional autoencoder, and a more complicated MTAD with multiple hybrid modules. We have evaluated the proposed method on three datasets, including SMAP, MSL, and SKAB. Our results demonstrate that the proposed method does bring the considerable performance gains to the baseline networks, and outperforms other cutting-edge methods.

6. REFERENCES

[1] Yann Cherdo, Paul de Kerret, and Renaud Pawlak, "Training lstm for unsupervised anomaly detection

without a priori knowledge," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 4297–4301.

[2] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga, "Usad: Unsupervised anomaly detection on multivariate time series," in *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3395–3404.

[3] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V Chawla, "A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 1409–1416.

[4] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel, "Deep learning for anomaly detection: A review," *ACM Computing Surveys*, vol. 54, no. 2, pp. 1–38, 2021.

[5] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 387–395.

[6] Julian von Schleinitz, Michael Graf, Wolfgang Trutschnig, and Andreas Schröder, "Vasp: An autoencoder-based approach for multivariate anomaly detection and robust time series prediction with application in motorsport," *Engineering Applications of Artificial Intelligence*, vol. 104, pp. 104354, 2021.

[7] Daehyung Park, Yuuna Hoshi, and Charles C Kemp, "A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1544–1551, 2018.

[8] Jinghui Chen, Saket Sathe, Charu Aggarwal, and Deepak Turaga, "Outlier detection with autoencoder ensembles," in *Proceedings of the SIAM international conference on data mining*, 2017, pp. 90–98.

[9] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International Conference on Learning Representations*, 2018.

[10] Shixiang Zhu, Henry Shaowu Yuchi, and Yao Xie, "Adversarial anomaly detection for marked spatio-temporal streaming data," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 8921–8925.

[11] Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang, "Multivariate time-series anomaly detection via graph attention network," in *IEEE International Conference on Data Mining*, 2020, pp. 841–850.

[12] Dan Li, Dacheng Chen, Jonathan Goh, and See-kiong Ng, "Anomaly detection with generative adversarial networks for multivariate time series," *arXiv preprint arXiv:1809.04758*, 2018.

[13] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng, "Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks," in *International Conference on Artificial Neural Networks*, 2019, pp. 703–716.

[14] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," *arXiv preprint arXiv:1802.09089*, 2018.

[15] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 1, pp. 1–39, 2012.

[16] Nela Zavaljevski and Kenny C Gross, "Sensor fault detection in nuclear power plants using multivariate state estimation technique and support vector machines.," Tech. Rep., 2000.

[17] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2828–2837.

[18] Iurii D. Katser and Vyacheslav O. Kozitsin, "Skoltech anomaly benchmark," `https://www.kaggle.com/dsv/1693952`, 2020.

[19] Astha Garg, Wenyu Zhang, Jules Samaran, Ramasamy Savitha, and Chuan-Sheng Foo, "An evaluation of anomaly detection and diagnosis in multivariate time series," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.