Deep Federated Anomaly Detection for Multivariate Time Series Data

Wei Zhu[1], Dongjin Song[2], Yuncong Chen[3], Wei Cheng[3], Bo Zong[3],
Takehiko Mizoguchi[3], Cristian Lumezanu[3], Haifeng Chen[3], and Jiebo Luo[1]

[1]University of Rochester
[2]University of Connecticut
[3]NEC Labs America

Abstract—**Although many anomaly detection approaches have been developed for multivariate time series data, limited effort has been made in federated settings in which multivariate time series data are heterogeneously distributed among different edge devices while data sharing is prohibited. In this paper, we investigate the problem of federated unsupervised anomaly detection and present a Federated Exemplar-based Deep Neural Network (Fed-ExDNN) to conduct anomaly detection for multivariate time series data on different edge devices. Specifically, we first design an Exemplar-based Deep Neural network (ExDNN) for learning local time series representations based on their compatibility with an exemplar module which consists of hidden parameters learned to capture varieties of normal patterns on each edge device. Next, a constrained clustering mechanism (FedCC) is employed on the centralized server to align and aggregate the parameters of different local exemplar modules to obtain a unified global exemplar module. Finally, the global exemplar module is deployed together with a shared feature encoder to each edge device, and anomaly detection is conducted by examining the compatibility of testing data to the exemplar module. Fed-ExDNN captures local normal time series patterns with ExDNN and aggregates these patterns by FedCC, and thus can handle the heterogeneous data distributed over different edge devices simultaneously. Thoroughly empirical studies on six public datasets show that ExDNN and Fed-ExDNN can outperform state-of-the-art anomaly detection algorithms and federated learning techniques, respectively.**

Index Terms—**Federated Learning, Unsupervised Anomaly Detection, Representation Learning**

Anomaly detection in multivariate time series refers to identifying abnormal status in certain time steps of the time series data [1] [2]. Building an effective unsupervised anomaly detection algorithm, however, is challenging since it requires collecting and profiling as much as (normal) multivariate time series data so as to reduce potential false positives [3]–[7]. With the rapid development of 5G networks, multivariate time series data are increasingly collected in various types of Internet of Things (IoT) edge devices, *e.g.*, mobile phones, healthcare, wearable devices, *etc*. However, due to privacy concerns [8], regulations [9], and transmission load [10], directly transferring data from edge devices to a centralized server to train a unified anomaly detection model is usually infeasible or prohibited [9] [11]. Consequently, there is a huge demand to develop an anomaly detection algorithm that can collaboratively handle the multivariate time series data distributed on different edge devices while preserving privacy.

For this purpose, we resort to federated learning and aim to conduct privacy-preserving anomaly detection. Specifically, assuming the normal status of multivariate time series data consists of K different modes which are heterogeneously distributed over L different edge devices, we aim to learn a unified model that can not only preserve data privacy on each edge device but also identify all anomalous situations accurately. A notable challenge is that due to the environmental conditions or other external factors, the time series data collected on each edge device may only partially (*i.e.*, less than K modes) cover all different modes of normal status. In this case, simply training an anomaly detection model may involve many false positives. Taking wearable devices as an example, a mode can be “walking”, “sitting”, “running”, “bicycling”, or “standing”, while an anomaly can represent “falling down” which is unusual to happen. A notable issue with this setting is that due to environmental conditions or other external factors, the time series data collected on each edge device may only partially cover the entire normal space. In other words, they are often heterogeneously distributed among different edge devices. For instance, an elder person tends to have activities of “walking”, “sitting”, and “standing”, while a young person may prefer activities of “walking”, “running”, and “bicycling”. In this case, if we only train an anomaly detection model based on the data collected from the elder person’s edge device, we may falsely detect “running” and “bicycling” as anomalies (*i.e.*, “falling down”) since they are not considered in training.

Directly applying existing federated learning approaches to perform unsupervised anomaly detection in multivariate time series data often leads to inferior performance. This is because most existing anomaly detection methods assume that the entire normal space is covered by the training data [12], which may not be true with the current setting; another reason is that existing federated learning algorithms, *e.g.*, Federated Averaging (FedAvg) [11], are originally designed for supervised learning and may not be able to properly handle the unsupervised tasks with heterogeneous data distributed on edge devices. Simply combining federated learning and unsupervised anomaly detection algorithms could lead to a series of problems. One issue is that each locally trained model

978-1-6654-8045-1/22/$31.00 ©2022 IEEE

may only partially cover the entire normal space; the federated learning algorithms, *e.g.*, federated averaging operation, could produce a global model which may collapse to certain normal modes and even cover a certain part of the abnormal space [13]. As a result, many anomalies could be mistreated as normal status and vice versa. Another issue is that existing unsupervised anomaly detection methods often rely on the encoder-decoder framework [4] [5] and Generative Adversarial Network (GAN) [14] [15] to extract semantic representation, and the extra parameters brought by the decoder and generator may result in a heavy communication cost.

To address the aforementioned issues, in this paper, we present a Federated Exemplar-based Deep Neural Network (Fed-ExDNN) to perform federated anomaly detection with multivariate time series data. On the edge device side, we specifically designed an Exemplar-based Deep Neural Network (ExDNN) to perform anomaly detection. ExDNN can simultaneously learn local time series representations based on their compatibility with an exemplar module which consists of hidden parameters learned to capture varieties of normal patterns in the hidden feature space. On the server side, to cope with the heterogeneity of time series data on different edge devices, Fed-ExDNN employs a Federated Constrained Clustering (FedCC) technique to align and aggregate parameters of different local exemplar modules. Eventually, the updated global exemplar module, together with a shared feature encoder, will be sent back to edge devices, and the anomaly detection is conducted by measuring the compatibility of test data (with extracted features) to the global exemplar module.

The main contributions of this paper are summarized as follows:

- We formally investigate the problem of federated unsupervised anomaly detection (FedUAD) for multivariate time series data and develop Fed-ExDNN, which consists of ExDNN for local anomaly detection and FedCC for model aggregation to handle FedUAD.
- On the edge device side, we present an Exemplar-based Deep Neural Network (ExDNN), which can simultaneously learn local time series representations based on their compatibility with an exemplar module developed to capture potential normal patterns in the hidden feature space. On the server side, we develop FedCC to align and aggregate different local exemplar modules. ExDNN and FedCC work jointly to address the heterogeneous distribution among edge devices.
- Our empirical studies on six public multivariate time series datasets demonstrate the effectiveness of the proposed ExDNN and Fed-ExDNN.

I. Related Work

A. Unsupervised Anomaly Detection

Recently, deep learning-based anomaly detection methods have shown fruitful progress compared with traditional methods, including one class SVM [16], Isolated Forest [17], *etc.* In general, they can be categorized into four different types, *i.e.*, One-class classification-based methods, reconstruction-based approaches, contrastive learning-based techniques, and clustering-based methods.

For one-class classification-based methods, Deep Support Vector Data Descriptor (SVDD) replaces the kernel used in OCSVM [16] with a deep neural network [3]. Context Vector Data Description (CVDD) generates multi-semantic contexts by multi-head attention mechanism [18]. Temporal Hierarchical One-Class (THOC) conducts multi-scale one class learning in a hierarchical manner [19]. Reconstruction-based approaches mainly rely on autoencoder framework to reconstruct the input and employ the reconstruction error to detect anomalies. For instance, LSTM Autoencoder (AE) [4] adopts LSTM to encode and decode multivariate time series (MTS) data. To better model inter-correlation between different time series, Multiscale Convolutional Recurrent Encoder-Decoder (MSCRED) is proposed to reconstruct system signature matrices by an attention-based convLSTM [5]. Memory Augmented Autoencoder (MemAE) augments autoencoder with an external memory [20]. BeatGAN [21] regularizes the reconstructed data by a generative adversarial network [14]. OmniAnomaly learns robust representation with stochastic variable connection and planar normalizing flow [22]. Unsupervised Anomaly Detection (USAD) imposes additional constraints on reconstruction by an additional decoder [23]. Li *et al.* conduct anomaly detection with hierarchical VAE and low dimensional embedding for anomaly detection [24]. More recently, contrastive learning-based techniques are becoming popular for time series anomaly detection. For instance, self-supervised contrastive predictive coding is proposed to handle anomaly points [25]. Cho *et al.* propose a masked contrastive method by using class-wise scale factor [26]. A unified contrastive anomaly detection framework is proposed by [27]. Carmona *et al.* perform time series anomaly detection by generating abnormal series with expertise knowledge [28]. Qiu *et al.* propose deterministic contrastive loss to enable the anomaly score to be consistent with training loss [29]. Clustering-based deep neural networks are also applied to anomaly detection. For instance, Zong *et al.* proposes Data Encoder Gaussian Mixture Model (DAGMM) for anomaly detection, and they conduct GMM on the feature space composed of reconstruction score and encoding of the autoencoder.

In this paper, we formally introduce the task - Federated Unsupervised Anomaly Detection (FedUAD). The heterogeneous data distribution on edge devices and data-free server-side model aggregation brings additional challenges to existing anomaly detection methods. We specially design Fed-ExDNN to overcome these problems.

B. Federated Learning

Federated learning has received increasing attention recently [9] [8] [30] [31]. One of the major concerns of federated learning is the heterogeneity problem [32]–[37]. Federated Averaging (FedAvg) is the most widely used algorithm for federated tasks [33]. Federated Proximal (FedProx) is proposed to alleviate the heterogeneity challenge [32]. Sattler *et* *al.* [38] propose to hierarchically cluster the locally learned models. Xie *et al.* [39] propose to maintain multi-global models and assign user's gradient to different global models. Liang *et al.* [40] propose to learn local representations on each device and an overall global model across devices. Federated Matching Average (FedMA) learns to cluster the local models before averaging the weights [13]. Yu *et al.* ([41]) propose FedAwS where there is only one class on each device. Fedfast is proposed to handle a federated recommendation system [42]. FedDF conducts data-free knowledge distillation on the server side to aggregate local models [43]. Fallah *et. al.* [44] applies meta-learning [45] on client updates. However, these methods are all proposed to handle supervised federated learning tasks. The unsupervised setting of UAD makes the heterogeneity problem much more challenging, and we find that the proposed Fed-ExDNN has clear advantages over conventional federated learning methods for FedUAD.

More recently, federated anomaly detection has drawn increasing attention [46]–[50]. Nguyen *et al.* propose to apply federated anomaly detection for IoT devices [47]. Federated anomaly detection is also used to address the IoT security attacks [51]. Liu *et al.* propose an on-device method for industrial IoT anomaly detection. Zhao *et al.* propose a multi-task network for federated anomaly detection [48]. Compared to existing works, our proposed Fed-ExDNN focuses on unsupervised anomaly detection and employs an effective method to address the heterogeneity problem.

II. Federated Exemplar-based Deep Neural Network

In this section, we present a Federated Exemplar-based Deep Neural Network (Fed-ExDNN) to perform federated unsupervised anomaly detection. Fed-ExDNN consists of ExDNN for local anomaly detection and Federated Constrained Clustering (FedCC) for model aggregation.

We find that combining existing federated learning and anomaly detection approaches to handle the task often leads to inferior performance. This should be attributed to the fact that most existing anomaly detection methods are developed based on the assumption that the entire normal space is covered by the training data [12], and this assumption does not hold when we conduct training on edge devices with heterogeneously distributed data, *i.e.*, not all normal patterns can be accessed during the training stage on each edge device. On the other hand, existing federated learning algorithms, *e.g.*, Federated Averaging (FedAvg) [11], are originally designed for supervised classification tasks, and they may not properly aggregate the unsupervised anomaly detection models trained on edge devices. To this end, we present Fed-ExDNN, which consists of a clustering-based anomaly detection method ExDNN and a federated aggregation method FedCC to align and aggregate different local exemplar modules.

We brief the basic training procedure as follows: assuming there are L edge devices, the l-th local device learns a device-specific model, which includes an embedding network $f^l(\cdot;\theta^l)$ for feature encoding and an exemplar module in which a set of K local exemplars $\mathbf{C}^l = \{\mathbf{c}_1^l, \cdots, \mathbf{c}_K^l\} \in \mathbb{R}^{d\times K}$ is learned to capture the potential normal patterns in the hidden feature space. The local model is trained based on the time series data collected on the l-th local device for unsupervised anomaly detection. The central server aggregates local models from different devices to construct a global model. The embedding network (for feature encoding) of the global model $g(\cdot;\bar{\theta})$ is obtained by Federated Averaging, and the global exemplar module with K learnable exemplars $\mathbf{U} = \{\mathbf{u}_1, \cdots, \mathbf{u}_K\}$ is obtained by aggregating and align all local exemplar modules with the proposed FedCC. Finally, the server sends the global model to different edge devices to update their local models. Fed-ExDNN jointly adopts FedCC and ExDNN to enable the global model to capture entire normal patterns even when the data are heterogeneously distributed among edge devices. Please refer to Fig. 1 for a detailed illustration.

A. Local Device: ExDNN

Exemplar-based Deep Neural Network (ExDNN), as a clustering-based anomaly detection method, is naturally suitable for handling heterogeneous data. Compared with existing clustering-based methods [2], [52] that adopt Gaussian Mixture Model (GMM) as the clustering objective, the ExDNN is developed based on an advanced clustering algorithm [53], [54]. Moreover, we propose Deep Relation Preserving (DRP) to learn the representation of multivariate time series data in an unsupervised manner. We'd like to emphasize that ExDNN conducts clustering and representation learning simultaneously to mutually boost their performance for better anomaly detection. The details of ExDNN are shown in Fig. 2. In the following, the algorithm described is for one particular local device, so we omit the superscript l for brevity.

For a specific local device with n multivariate time series segments $\{\mathbf{X}^i\}_{i=1}^n \in \mathbb{R}^{m\times t}$, where m denotes the number of time series, and t is the length of a segment, our method learns the optimal embedding network $f(\cdot;\theta)$ and an exemplar module with a learnable parameter $\mathbf{C} \in \mathbb{R}^{d\times K}$. In this paper, we use LSTMs to encode temporal dynamics in the multivariate time series. The exemplar module $\mathbf{C}$ is implemented by a fully connected layer and is jointly trained with the embedding network parameter θ in an end-to-end manner.

Specifically, our proposed method is motivated by Deep Embedding Clustering [54] with the following objective:

$$\min_{\theta,\mathbf{C}} \frac{1}{n}\sum_{i=1}^{n} KL(\mathbf{p}_i \| \mathbf{q}_i) \tag{1}$$

where $KL(\cdot\|\cdot)$ is the Kullback–Leibler divergence and Eq. (1) encourages the exemplars to be close to the training samples on the embedding space, and each learned exemplar could then capture a specific pattern of the normal data just like a clustering center in K-means. Specifically, $\mathbf{q}_i \in R^K$ is the cluster indicator vector for the i-th segment where q_{ij} is the probability of assigning the i-th data to the j-th exemplar. This probability is computed as

$$q_{ij} = \frac{\exp(\gamma_1 s(f(\mathbf{X}^i), \mathbf{c}_j))}{\sum_{k=1}^{K}\exp(\gamma_1 s(f(\mathbf{X}^i), \mathbf{c}_k))}, \tag{2}$$

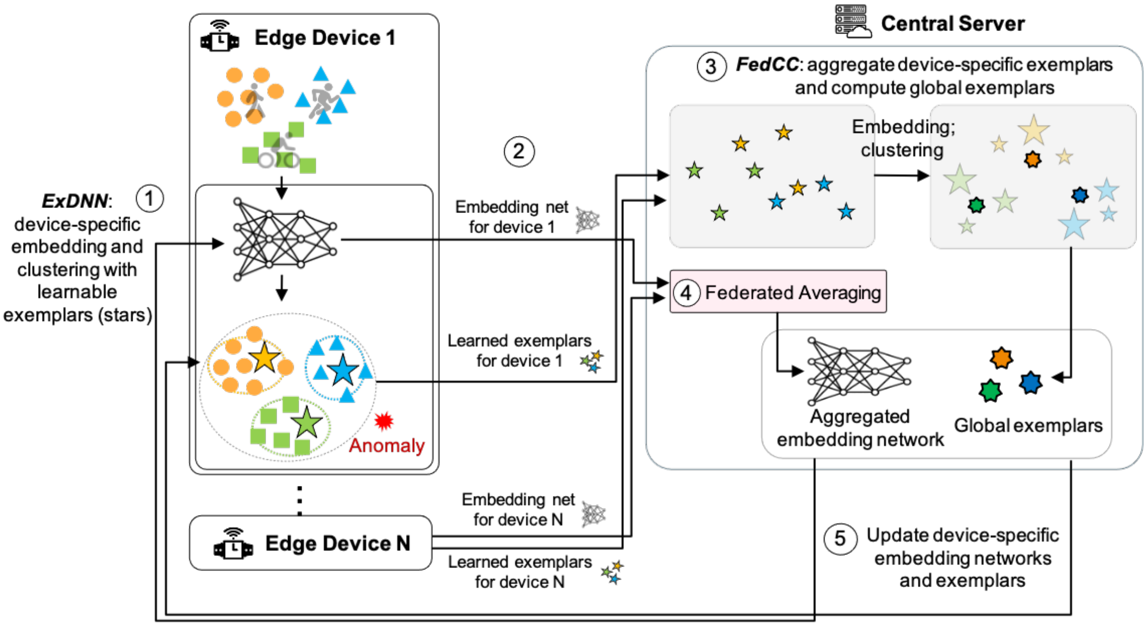

Central Server
3 FedCC: aggregate device-specific exemplars and compute global exemplars
Embedding; clustering
4 Federated Averaging
Aggregated embedding network
Global exemplars
Edge Device 1
ExDNN: 1 device-specific embedding and clustering with learnable exemplars (stars)
2
Embedding net for device 1
Learned exemplars for device 1
Anomaly
Edge Device N
Embedding net for device N
Learned exemplars for device N
5 Update device-specific embedding networks and exemplars

Fig. 1: Block diagram of the proposed Fed-ExDNN, which contains five steps for each communication round.

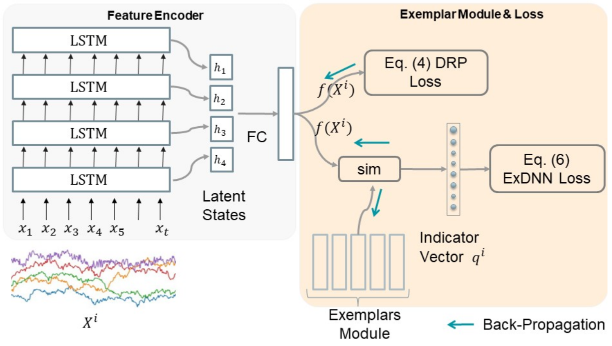


Fig. 2: The time series segment X^i is processed by a 4-layer LSTM followed by a fully connected embedding layer. The exemplars module is updated by gradient descent and is implemented with a fully connected layer.

where s is the cosine similarity $s(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^\mathsf{T}\mathbf{b}}{\|\mathbf{a}\|_2\|\mathbf{b}\|_2}$, γ_1 is a learnable scale factor, and $\mathbf{C} = \{\mathbf{c}_1, \cdots, \mathbf{c}_K\} \in \mathbb{R}^{d\times K}$ denote local exemplars. We should highlight that the local exemplars are the same as the weights of a fully connected layer and their privacy concern could be addressed by well-studied approaches, *e.g.*, differential privacy [8]. Following [54], we raise $\mathbf{q}_i$ to the second power and normalize it by the size of clusters to obtain $\mathbf{p}_i$ as

$$p_{ij} = \frac{q_{ij}^2 / \sum_{i'=1}^{n} q_{i'j}}{\sum_{j'} (q_{ij'}^2 / \sum_{i'=1}^{n} q_{i'j'})} \tag{3}$$

However, the exemplar learned by optimizing Eq. (1) may converge to patterns with few samples and even noisy data, and we then adopt the balanced loss to alleviate the problem as

$$\min_{\theta, \mathbf{C}} -\boldsymbol{\alpha}^\mathsf{T} \log \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{q}_i \right) \tag{4}$$

$\boldsymbol{\alpha} \in \mathbb{R}^K$ is a prior distribution over the exemplars, and this term encourages the cluster sizes on edge devices to match the prior [53]. We set $\boldsymbol{\alpha} = \frac{1}{K}\mathbf{1}$, *i.e.*, uniform distribution. The effectiveness of the balancing term on handling contaminated data is shown in our ablation studies.

Representation learning is critical for deep neural network training, and DEC is often trained with auto-encoder initialization [54], which may lead to privacy concerns and computational costs, especially in federated learning settings. Furthermore, we propose Deep Relative Preserving (DRP) that encourages the latent space to preserve the local similarity induced by the original feature [55], [56]:

$$M(\mathbf{X}^i) = \min_{\theta} \log \left(1 + \sum_{\substack{p \in \mathcal{P}_i \\ n \notin \mathcal{P}_i}} \exp(\gamma_2 (s_{in} - s_{ip})) \right) \tag{5}$$

where $\mathcal{P}_i$ is the set of nearest neighbors of the i-th example. γ_2 is a learnable scale factor. s_{ij} is shorthand for $s(f(\mathbf{X}^i), f(\mathbf{X}^j))$. Eq. (5) encourages the similarity of positive pairs to be larger than that of negative pairs [57]. Moreover, to avoid the computation and storage cost of the KNN graph, we approximate KNN by the samples within each minibatch.

We could perform anomaly detection by jointly optimizing Eq. (1), Eq. (4), and Eq. (5) as

$$\min_{\theta, \mathbf{C}} \frac{1}{n} \sum_{i=1}^{n} \left(KL(\mathbf{p}_i \| \mathbf{q}_i) + M(\mathbf{X}^i) \right) - \boldsymbol{\alpha}^\mathsf{T} \log \left(\frac{1}{n} \sum_{i=1}^{n} \mathbf{q}_i \right), \tag{6}$$

and the *anomaly score* for sample $\mathbf{X}$ is calculated by the negative cosine similarity between the samples and its nearest exemplars as

$$\text{Score}(\mathbf{x}) = -\max_{j}\ s(f(\mathbf{X}), \mathbf{c}_j). \tag{7}$$

However, Eq. (6) only forces the relative similarity between the sample and its nearest exemplar to be larger than the similarity between the sample and other exemplars [54]. After training, we actually have a little guarantee on the numeric value of the anomaly score. Therefore, although Eq. (6) may be effective for clustering, it leads to sub-optimal performance for anomaly detection in our experiments. To alleviate the problem, we introduce an absolute term to directly optimize the numeric value of the anomaly score, and our final objective for anomaly detection on a local device becomes:

$$\begin{aligned}\min_{\theta,\mathbf{C}} \frac{1}{n}\sum_{i=1}^{n}\big(KL(\mathbf{p}_i\|\mathbf{q}_i) + M(\mathbf{X}^i)\big) - \boldsymbol{\alpha}^{\mathsf{T}}\log\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{q}_i\right) \\ +\frac{1}{n}\sum_{i=1}^{n}\log\big(1+\exp(-\gamma_3(s(f(\mathbf{X}^i),\bar{\mathbf{c}}_i)-m)\big).\end{aligned} \tag{8}$$

The last term of Eq. (8) [1] is the proposed absolute score loss. Since the max operation in Eq. (7) is non-differentiable, our absolute score term maximizes the cosine similarity between the i-th segment and a soft approximation of the nearest exemplar $\bar{\mathbf{c}}_i = \sum_{j=1}^{K} q_{ij}\mathbf{c}_j$, where $\mathbf{q}_i$ is the soft indicator vector for the i-th sample. γ_3 is a learnable scale factor and $m > 0$ is the margin.We adopt softplus operation to make the proposed term have a similar scale as other terms. Since the distribution of pairwise cosine similarity of two random high-dimensional unit vectors approaches a zero mean Gaussian, it is thus necessary to have $s(f(\mathbf{X}^i), \bar{\mathbf{c}}_i)$ larger than a positive margin m. We highlight that although the ExDNN in Eq. (8) contains several hyper-parameters, most of them are fixed during the empirical studies in this paper besides the number of exemplars K.

After certain rounds of training on the l-th local device, the parameters of the embedding network θ^l and the set of exemplars $\{\mathbf{c}_1^l, \cdots, \mathbf{c}_K^l\}$ are all uploaded to a central server.

B. Central Server: FedCC for Exemplars Aggregation

The central server aggregates the local models uploaded from edge devices to obtain a global feature encoding network g and a global exemplar module $\mathbf{U} = \{\mathbf{u}_1, \cdots, \mathbf{u}_K\}$ that capture the heterogeneous data distribution on all edge devices. The feature encoding network could be aggregated by existing federated learning methods, *e.g.*, federated averaging (FedAvg) [11] and Federated Proximal (FedProx) [32]. However, due to heterogeneous data on edge devices, the local exemplar module, even with the same initialization, may significantly deviate from each other to better fit the local data. Even worse, due to the cost of transmission, the local training is desired to be longer to reduce the communication load between servers and clients. As a result, the alignment between the updated local exemplar module and the previous global exemplar module may not hold as the training on local devices proceeds.

Most of the existing federated learning methods, *e.g.*, FedAvg and FedProx, element-wisely average the local exemplars based on the assumption that all local exemplar modules and the global exemplar module are still well-aligned during the training. For supervised tasks, the alignment could be regularized by the ground truth label. For the proposed Federated Unsupervised Anomaly Detection (FedUAD) task, however, no label is available. Therefore, vanilla federated learning methods will lead to suboptimal performance in practice. Aggregating the exemplar modules by K-means seems to be a reasonable choice [13]. However, K-means may also result in misalignment since its objective and the representation are decoupled, which makes it impossible to adjust the representation so as to mitigate the issue. In this paper, we propose an innovative approach, namely Federated Constrained Clustering (FedCC) to address the heterogeneity problem. The basic idea is to first learn a projection function h that could align the local exemplars to discover and enhance the alignment, and then conduct clustering on the learned embedding space to obtain the global exemplar module. The effectiveness of FedCC is verified in the experiment section.

From L edge devices each with exemplar module consisting of K learnable exemplars, the central server receives a total of $N = LK$ local exemplars which is denoted as $\{\mathbf{c}_1^1, \cdots, \mathbf{c}_K^1, \mathbf{c}_1^2, \cdots,\ \mathbf{c}_K^2, \cdots, \mathbf{c}_1^L \cdots \mathbf{c}_K^L\}$. The proposed FedCC is formulated as:

$$\begin{aligned}\min_{\phi,\{\mathbf{v}_1,\cdots,\mathbf{v}_K\}} -\frac{1}{N}\sum_{i=1}^{K}\sum_{l=1}^{L}\mathbf{p}_{il}^{\mathsf{T}}\log\mathbf{q}_{il} \\ -\mathbf{1}^{\mathsf{T}}\log\left(\frac{1}{N}\sum_{i=1}^{K}\sum_{l=1}^{L}\mathbf{p}_{il}\right) + \frac{1}{N}\sum_{i=1}^{K}\sum_{l=1}^{L}R(\mathbf{c}_i^l),\end{aligned} \tag{9}$$

where ϕ denotes the parameters of the projection network h, and $\{\mathbf{v}_1, \cdots, \mathbf{v}_K\}$ are the latent cluster centers in the output space of h. The first two terms in Eq. (9) are for clustering, similar to the first and third terms in Eq. (8). $\mathbf{q}_{il}$ are defined similarly to Eq. (2):

$$q_{il,j} = \frac{\exp(\gamma_4 \cdot s(h(\mathbf{c}_i^l), \mathbf{v}_j))}{\sum_{k=1}^{K}\exp(\gamma_4 \cdot s(h(\mathbf{c}_i^l), \mathbf{v}_k))}, \tag{10}$$

and $\mathbf{p}_{il}$ is defined based on $\mathbf{q}_{il}$ in the same way as Eq. (3), and γ_4 is the scale factor. We further introduce the constraints R to encourage the learned projection h to give similar embedding for exemplars that have the same initialization (stars with same color in Fig. 1).

$$R(\mathbf{c}_i^l) = \log\left(1 + \sum_{m=1}^{L}\sum_{j=1}^{K}\exp(\gamma_5\mathbf{e}_{ij}s(h(\mathbf{c}_i^l), h(\mathbf{c}_j^m)))\right), \tag{11}$$

[1] Here we slightly abuse the symbol of cosine similarity and $s(f(\mathbf{X}^i), \bar{\mathbf{c}}_i)$ is calculated by the dot product between $f(\mathbf{X}^i)^T$ and $\bar{\mathbf{c}}$ with normalized $f(\mathbf{X}^i)$ and $\mathbf{c}_j$.

TABLE I: The detailed statistics of six multivariate time series datasets.

Dataset	# train - val - test	# dim (m)	# Length (t)
2D Gesture	8171 - 876 - 2044	2	80
SWaT	47420 - 11198 - 33594	51	100
ECG5000	292 - 1125 - 3375	1	140
HAR Laying	4559 - 1676 - 2947	9	128
AerobicDigits	6600 - n/a - 2200	13	20
UWave	1600 - n/a - 2879	3	40

where $\mathbf{e}_{ij} = 1$ if $i = j$ and c_i^l is K nearest neighbor of c_j^m, and $\mathbf{e}_{ij} = -1$ otherwise. γ_5 is a learnable scale factor. Finally, the global exemplar module $\{\mathbf{u}_1, \cdots, \mathbf{u}_K\}$ can be obtained based on the clustering indicator matrix:

$$\mathbf{u}_z = \frac{1}{\sum_{i=1}^{K} \sum_{l=1}^{L} q_{il,z}} \sum_{i=1}^{K} \sum_{l=1}^{L} q_{il,z} \mathbf{c}_i^l. \tag{12}$$

After obtaining the global exemplar module and the averaged embedding network (for feature encoding), they will be sent back to each edge device for the next round of learning.

III. Experiments

In this section, we verify the superiority of ExDNN and Fed-ExDNN for anomaly detection.

A. Datasets and Evaluation Metrics

We conduct experiments on six publicly available multivariate time series datasets, including 2D Gesture [58], ECG5000 [59], SWaT [60], HAR Laying [61], UWave [62], and ArabicDigits [63]. The details of these datasets and train/validation/test partitions are summarized in Table I.

For 2D Gesture and SWaT, we use a sliding window with stride 1 to partition them. Moreover, since their training sets only have normal data, we select a portion from the original testing data to construct the validation set for hyperparameter tuning. We downsample the time series of SWaT by 10 following [19]. The length of segments of AerobicDigit and Uwave varies and we resize their segments to 20 and 40, respectively. For 2D Gesture, ECG5000, SWaT, and HAR Laying, we set hyperparameters based on grid search over the validation set. We report the AUC, F1, Precision, and Recall linked to the best F1 score on the validation set following [23]. For AerobicDigit and UWave, following the setting similar to [3], we do not construct a validation set, and only report average AUC by iteratively treating each class as the anomaly case. All experiments are run three times.

B. ExDNN for Anomaly Detection

We first conduct experiments to show the effectiveness of the proposed ExDNN for unsupervised anomaly detection with multivariate time series (MTS) data.

1) Comparison Methods and Experimental Settings: We compare the proposed ExDNN with seven deep learning methods, including LSTM-AutoEncoder (LSTM-AE) [4], BeatGAN [21], Memory Augmented AutoEncoder (MemAE) [4], USAD [23], Deep SVDD [3], Contextual SVDD (CVDD) [18], and Deep Autoencoding Gaussian Mixture Model (DAGMM) [2]. Among them, LSTM AE, BeatGAN, MemAE, and USAD are reconstruction-based methods, while Deep SVDD and CVDD are one class-based method. DAGMM jointly uses auto-encoder reconstruction and clustering for anomaly detection.

All deep learning methods are implemented with the same feature encoder shown in Fig. 2, which is composed of a 4-layer LSTM with the hidden dimension set as 8. The decoder for reconstruction-based methods is with the same structure as the encoder. We fix the batch size as 128 and fixed the learning rate set as 0.005 for all methods. We conduct an exhaustive grid search for the deep learning methods to find the optimal parameters for each dataset. For ExDNN, we fix the hyperparameter as $\gamma_1 = 2$, $\gamma_2 = 10$, $\gamma_3 = 10$, $m = 0.5$, and the number of nearest neighbors in DRP as 10, unless otherwise stated. We search the number of clusters from $\{8, 16, 32, 64, 128\}$. Since AerobicDigit and UWave do not have a validation set, we set the number of exemplars as 32 and 64 respectively. All experiments are conducted on a server with 4 Nvidia GTX 2080 Ti graphics cards.

2) Effectiveness of ExDNN: The anomaly detection results on MTS datasets are shown in Table II. According to the results, several interesting points are summarized as follows. First, either reconstruction or one-class-based methods cannot handle all different scenarios. By imposing additional restrictions on the latent space learned by auto-encoder, BeatGAN, MemAE, and USAD could potentially boost the performance of LSTM AE on homogeneous datasets, especially on HAR Laying, but are of little use and even also degrade the performance on more heterogeneous datasets, *e.g.*, SWaT and AerobicDigits. By contrast, Deep SVDD works much better for homogeneous datasets, *e.g.* HAR Laying, but also suffers from the under-fitting problem with a lower recall score. CVDD could largely alleviate the under-fitting problem of SVDD through multi-context learning. Second, the proposed ExDNN, although designed for anomaly detection with heterogeneous normal samples, could handle the over-fitting and under-fitting problem by tuning the number of exemplars and generally achieves superior performance on all datasets. ExDNN outperforms compared methods, especially on datasets with heterogeneous normal cases. For example, ExDNN achieves 4.58% improvements (regarding averaged AUC) on AerobicDigit compared with the second-best method CVDD. This is because ExDNN not only extracts superior representations by DRP but also conducts effective deep clustering to generate representative exemplars for anomaly detection.

3) Ablation Studies: In this section, we conduct ablation studies on 2D Gesture and AerobicDigit. For AerobicDigit, only averaged AUC is reported. We first study the effectiveness

TABLE II: Anomaly detection performance on six MTS datasets. The best methods are highlighted in bold.

Dataset	Metric	LSTM AE	BeatGAN	MemAE	USAD	Deep SVDD	CVDD	DAGMM	ExDNN(Ours)
2D Gesture	AUC	80.19±4.32	81.87±3.84	80.88±0.40	80.15±3.99	73.58±2.73	82.64±0.05	74.74±8.51	**88.37**±0.82
	F1	58.57±3.59	60.81±4.05	62.77±1.14	61.58±3.36	54.42±3.83	62.54±2.47	56.06±9.12	**71.13**±2.31
	Prec	50.28±2.05	58.10±8.94	57.84±3.85	55.84±0.73	49.98±8.83	62.88±5.16	47.54±8.69	66.96±5.74
	Rec	70.66±8.15	65.83±7.05	69.05±2.86	68.79±7.37	63.38±9.15	62.48±1.25	71.82±1.47	76.40±2.39
SWaT	AUC	91.87±0.85	91.39±0.30	91.29±0.47	90.69±0.33	89.51±8.02	**94.70**±0.94	90.02±0.99	91.61±0.78
	F1	81.89±1.99	80.65±1.70	81.52±0.52	80.25±0.48	79.86±5.55	83.34±1.63	80.07±3.50	**87.34**±0.61
	Prec	72.56±4.63	70.15±4.25	71.35±1.42	69.35±1.28	69.19±4.78	71.71±2.39	70.15±5.71	85.10±0.72
	Rec	94.44±3.00	95.31±3.16	95.19±2.40	95.26±1.05	94.46±6.95	99.51±0.15	93.28±1.64	89.71±0.60
ECG5000	AUC	95.04±0.63	94.85±0.61	94.88±1.24	93.73±0.82	90.35±5.40	97.38±0.25	89.14±0.99	**98.45**±1.05
	F1	92.09±1.57	92.31±0.16	92.16±0.99	91.69±1.48	85.91±7.74	91.20±0.26	88.00±1.43	**92.86**±2.64
	Prec	88.36±3.32	87.95±0.51	88.05±2.28	86.53±2.66	89.58±6.12	90.69±0.92	79.55±2.20	91.29±3.64
	Rec	96.24±0.85	97.12±0.88	96.74±0.85	97.57±1.07	82.57±5.43	91.76±1.31	98.45±0.50	94.55±1.80
HAR Laying	AUC	59.73±4.76	95.30±2.97	95.20±2.38	69.98±1.94	**100**±0	94.60±5.90	62.82±1.92	99.99±0
	F1	42.89±0.24	81.00±0.13	83.73±2.95	50.19±3.31	99.14±1.15	72.06±4.54	33.69±3.31	**99.35**±0.20
	Prec	27.30±0.19	73.97±0.10	77.39±5.84	34.70±1.62	100±0	64.51±6.73	25.47±1.62	98.71±0.39
	Rec	100±0	91.56±0.10	91.81±5.16	97.95±6.96	98.32±2.24	84.67±5.92	53.32±6.96	100±0
Aerobic Digit	Avg AUC	79.70	77.27	77.67	77.80	74.21	83.12	81.52	**87.70**
UWave	Avg AUC	84.33	86.95	85.35	86.61	70.32	81.11	73.20	**88.72**

TABLE III: Performance of ExDNN variants. The number of exemplars is set as 32. The best results are highlighted in bold.

Methods	2D Gesture		AerobicDigit
	AUC	F1	Avg AUC
ExDNN w/ AE	68.83±3.17	61.35±4.36	83.57
ExDNN w/o clus	85.96±1.69	67.11±2.04	68.57
ExDNN w/o bal	85.01±1.02	64.46±1.05	83.67
ExDNN w/o abs	83.26±1.34	63.93±2.43	85.94
ExDNN	**86.31**±1.47	**67.34**±1.98	**87.70**

TABLE IV: Performance of ExDNN with different numbers of exemplars. Best results are highlighted in bold.

K	2D Gesture		AerobicDigit
	AUC	F1	Avg AUC
8	82.34±1.60	62.30±3.47	75.51
16	84.94±2.88	66.15±3.63	84.02
32	86.31±3.55	67.34±4.67	87.70
64	87.34±0.72	68.41±1.35	**88.62**
128	**88.37**±0.82	**71.13**±2.31	88.61

of different components of the proposed ExDNN. The results are shown in Table III. We denote ExDNN without the cluster term in Eq. (8) as ExDNN w/o clus, ExDNN without the balanced term in Eq. (8) as ExDNN w/o bal, and ExDNN without the absolute term in Eq. (8) as ExDNN w/o abs. Moreover, to validate the effectiveness of the proposed DRP for representation, we replace DRP with a pretrained network by autoencoder as ExDNN w/ AE. The experiments are conducted with 32 exemplars for 2D Gesture and AerobicDigit. Based on the results, we could conclude that the cluster term brought from DEC [54] is essential for the success of our method for AerobicDigit whose normal cases are heterogeneous. The proposed absolute term in Eq. (8) could consistently boost the anomaly detection performance on different datasets. The balanced term could also improve performance and stabilize the training process. Comparing ExDNN with ExDNN w/ AE, DRP is a better choice than the autoencoder for ExDNN.

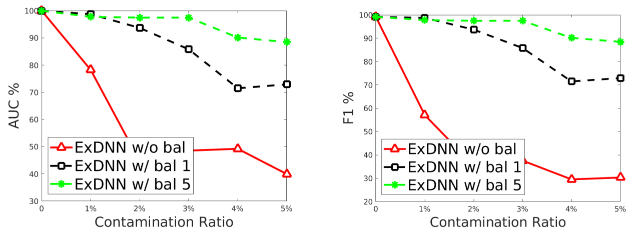
4) Influence of the number of exemplars: We study the influence of the number of exemplars, and we detail the results in Table IV. The results indicate that increasing the number of exemplars could improve the performance of ExDNN for heterogeneous normal cases in general.

5) Contamincation Study: ExDNN is developed based on the assumption that the training set only contains normal data. To study the performance of ExDNN on contaminated data, we inject different percentages of abnormal data into the training set. We vary this percentage from 1% to 5% and conduct experiments on HAR Laying. To validate the effectiveness of the balanced term for handling noisy samples, we vary the weight of the balanced term from $\{0, 1, 5\}$, and these variants of ExDNN are denoted as ExDNN w/o bal, ExDNN w/ bal 1, and ExDNN w/ bal 5 respectively. According to the experiments shown in Fig. 3, the balanced term can significantly boost the robustness of ExDNN. Although it is better to set a large weight for the balanced term, the default setting of ExDNN (ExDNN w/ bal 1) is already robust to abnormal sample contamination, and is sufficient for most real-life applications.

6) Training Time: We compare the training time of reconstruction-based (LSTM AE), one class-based (Deep SVDD), and clustering-based (ExDNN) deep anomaly detection methods. There is no need to conduct experiments with other deep learning methods as they are developed based on either LSTM AE or Deep SVDD. The results are shown in Table VI. LSTM AE is more computational expensive than ExDNN and Deep SVDD for the decoder process, and the auto-encoder pertaining required by Deep SVDD will also significantly slow down its training process. ExDNN replaces the auto-encoder with DRP for representation learning which is effective and brings negligible additional computation cost.

TABLE V: Federated anomaly detection performance on six MTS datasets. Best performance is highlighted in bold.

Dataset		FedAvgAE	FedProxAE	FedAvgSVDD	FedProxSVDD	FedAvgEx	FedProxEx	FedKmsEx	Fed-ExDNN
2D Gesture	AUC	80.64±0.89	81.29±1.14	70.22±4.61	77.64±4.45	82.81±1.37	82.79±0.27	83.57±2.70	**85.24**±1.20
	F1	57.92±1.32	58.13±1.28	51.86±5.74	56.58±2.60	63.12±2.24	63.97±0.93	62.56±1.47	**64.75**±2.41
	Prec	50.55±2.63	50.17±1.65	45.00±8.12	53.85±2.67	57.39±2.65	58.42±1.79	59.29±0.54	59.88±1.34
	Rec	65.44±1.69	66.15±0.84	69.44±9.54	59.64±2.99	72.99±3.25	71.08±4.93	66.25±2.61	74.82±3.11
SWaT	AUC	90.19±0.17	90.31±0.46	**91.57**±2.49	90.53±5.25	86.20±2.96	88.47±1.37	84.41±1.23	89.27±1.91
	F1	76.94±0.08	76.87±0.04	79.19±2.32	79.76±4.27	80.38±0.35	80.95±0.81	80.58±0.08	**83.61**±0.58
	Prec	62.52±1.10	62.44±0.05	65.79±3.33	66.75±6.08	70.68±1.86	72.17±1.88	70.63±0.09	72.57±0.25
	Rec	100±0	100±0	99.63±0.27	99.54±0.11	93.35±2.89	92.22±0.97	93.81±0.39	94.90±0.17
ECG5000	AUC	94.01±0.62	95.88±1.87	94.81±4.71	95.47±4.01	97.39±0.79	97.36±0.80	96.50±2.03	**98.17**±0.53
	F1	89.30±0.20	90.66±2.24	91.83±4.95	92.67±4.47	92.61±1.49	92.61±1.60	91.08±4.35	**93.86**±0.48
	Prec	82.34±0.62	84.15±4.27	87.47±9.55	88.33±9.02	93.83±2.39	93.94±2.39	92.31±2.99	93.49±0.78
	Rec	97.83±1.02	98.43±0.50	97.68±2.11	98.14±1.21	91.45±1.48	91.36±1.69	89.93±5.64	94.00±0.46
HAR Laying	AUC	89.74±1.31	88.74±0.21	99.66±0.09	99.55±0.06	99.43±0.56	99.71±0.22	99.99±0.01	**99.99**±0.01
	F1	69.52±2.15	67.63±0.56	98.35±0.54	97.38±1.33	97.39±0.16	96.95±0.47	97.87±0.15	**98.99**±0.08
	Prec	55.13±2.70	52.80±0.57	96.77±1.05	95.09±2.37	97.51±2.30	99.41±0.59	97.30±1.75	98.08±0.27
	Rec	94.23±0.19	94.04±0.37	100±0	99.81±0.19	97.39±2.61	94.60±0.37	98.51±1.49	99.91±0.09
AerobicDigit	Avg AUC	63.76	65.57	66.97	68.52	67.84	70.18	72.86	**79.53**
UWave	Avg AUC	81.74	84.06	80.27	81.43	81.17	81.59	79.65	**86.77**



(a) AUC

(b) F1

Fig. 3: Anomaly detection performance on the contaminated HAR Laying dataset.

TABLE VI: Running time (second) per epoch of three different types of anomaly detection methods. Deep SVDD requires extra epochs for auto-encoder pretraining.

Methods	2D Gesture	SWAT	ECG5000	HAR Laying
ExDNN	3.61	10.87	0.12	2.05
Deep SVDD	**3.41**	**10.11**	**0.10**	**1.87**
LSTM AE	5.53	17.26	0.19	3.24

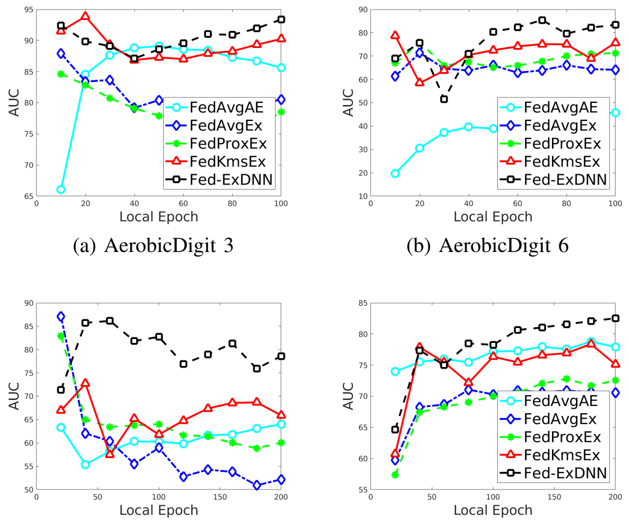
C. Fed-ExDNN for Federated Anomaly Detection

We conduct experiments to validate the superiority of Fed-ExDNN for FedUAD.

1) Comparison Methods and Experimental Settings: The experiments are conducted on all six MTS datasets. To simulate federated settings, for 2D Gesture, SWaT, and ECG5000, we sequentially partition the data into L different parts and assign them to L different edge devices. For HAR Laying, we assign the samples from each subject to an edge device and discard samples of two random activities for each subject. For AerobicDigit and UWave, the training set on each edge device is constructed by randomly selecting 900 and 300 samples from 3 different classes.

We implement several federated anomaly detection baselines. We aggregate the local models trained by LSTM AE (Deep SVDD) by Federated Average (FedAvg) and Federated Proximal (FedProx) as FedAvgAE (FedAvgSVDD) and FedProxAE (FedProxSVDD), respectively. Moreover, to justify the motivation of FedCC, we propose several variants of ExDNN as follows: we apply FedAvg and FedProx on the proposed ExDNN as FedAvgEx and FedProxEx, respectively; we also adopt Kmeans to aggregate the exemplars as a direct counterpart for FedCC termed as FedKmeans. For FedCC and FedKmeans, we adopt FedAvg to aggregate the feature encoder network. The hyperparameters of ExDNN for local training are described in the previous section and we search γ_4 and γ_5 from $\{1, 5, 10\}$ for FedCC. For FedProx, we search hyperparameters from $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. We conduct federated learning for 5 communication rounds which is sufficient for the performance of all federated anomaly detection methods to converge. We implement the network for FedCC with a three-layer multi-layer perceptron with ReLU as an activation function. The batch size for FedCC is set as 256 and the learning rate is 0.005. We initialize the global exemplars with kmeans++ and train FedCC for 500 steps. All methods are implemented in Pysyft [64] and Pytorch [65]

2) Results: The results of FedUAD are shown in Table V. According to the results, overall, the variants of Fed-ExDNN outperform federated anomaly detection baselines. This should be attributed to the fact that ExDNN explicitly takes the heterogeneous data on edge devices into consideration. Moreover, FedKmsEx and Fed-ExDNN outperform FedAvgEx and FedProxEx, since FedKmsEx and Fed-ExDNN could handle the deviation of exemplars. Finally, the proposed Fed-ExDNN performs better than other variants of Federated ExDNN since FedCC can simultaneously learn to align and aggregate local exemplars. Fig. 4 shows the federated learning results for each communication round, and we could conclude that the proposed Fed-ExDNN consistently outperforms its counterparts.



(a) AerobicDigit 3

(b) AerobicDigit 6

(c) Uwave 4

(d) Uwave 5

Fig. 4: We show the federated learning results on AerobicDigit and Uwave datasets with different communication rounds.

IV. Conclusions

In this paper, we developed the Federated Exemplar-based Deep Neural Network (Fed-ExDNN) to perform federated anomaly detection with multivariate time series data. We first investigated the problem of federated unsupervised anomaly detection with multivariate time series data. Then, we developed an Exemplar-based Deep Neural Network (ExDNN) to learn local time series representations based on their compatibility with an exemplar module that can capture varieties of normal patterns. Meanwhile, we also introduced a constrained clustering mechanism to align and aggregate the parameters of local exemplar modules to obtain a unified global exemplar module. Finally, the updated embedding network (for feature encoding) along with the global exemplar module is sent back to edge devices and the anomaly detection is conducted by comparing to those learned global exemplars.

V. Acknowledgement

Wei Zhu and Jiebo Luo were supported in part by NIH grant P50NS108676. Dongjin Song was supported by UConn Research Excellence Program (REP), Award, NSF CNS-2154191, and NEC Laboratories America, Inc.

References

[1] J. Kim and C. D. Scott, "Robust kernel density estimation," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 2529–2565, 2012.

[2] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International Conference on Learning Representations*, 2018.

[3] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International conference on machine learning*, 2018, pp. 4393–4402.

[4] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proceedings*, vol. 89. Presses universitaires de Louvain, 2015, pp. 89–94.

[5] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. 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*, vol. 33, 2019, pp. 1409–1416.

[6] Y. Jiao, K. Yang, D. Song, and D. Tao, "Timeautoad: Autonomous anomaly detection with self-supervised contrastive loss for multivariate time series," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 3, pp. 1604–1619, 2022.

[7] J. Ni, Z. Chen, W. Cheng, B. Zong, D. Song, Y. Liu, X. Zhang, and H. Chen, "Interpreting convolutional sequence model by learning local prototypes with adaptation regularization," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 1366–1375.

[8] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[9] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[10] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan *et al.*, "Towards federated learning at scale: System design," *arXiv preprint arXiv:1902.01046*, 2019.

[11] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[12] G. Pang, C. Shen, L. Cao, and A. v. d. Hengel, "Deep learning for anomaly detection: A review," *arXiv preprint arXiv:2007.02500*, 2020.

[13] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," *arXiv preprint arXiv:2002.06440*, 2020.

[14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[15] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International conference on information processing in medical imaging*. Springer, 2017, pp. 146–157.

[16] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, "Support vector method for novelty detection," in *Advances in neural information processing systems*, 2000, pp. 582–588.

[17] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*. IEEE, 2008, pp. 413–422.

[18] L. Ruff, Y. Zemlyanskiy, R. Vandermeulen, T. Schnake, and M. Kloft, "Self-attentive, multi-context one-class classification for unsupervised anomaly detection on text," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 4061–4071.

[19] L. Shen, Z. Li, and J. Kwok, "Timeseries anomaly detection using temporal hierarchical one-class network," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[20] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. v. d. Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1705–1714.

[21] B. Zhou, S. Liu, B. Hooi, X. Cheng, and J. Ye, "Beatgan: Anomalous rhythm detection using adversarially generated time series." in *IJCAI*, 2019, pp. 4433–4439.

[22] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2828–2837.

[23] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "Usad: Unsupervised anomaly detection on multivariate time series," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3395–3404.

[24] Z. Li, Y. Zhao, J. Han, Y. Su, R. Jiao, X. Wen, and D. Pei, "Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding," in *Proceedings of the 27th ACM*

SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 3220–3230.

[25] S. Deldari, D. V. Smith, H. Xue, and F. D. Salim, "Time series change point detection with self-supervised contrastive predictive coding," in *Proceedings of the Web Conference 2021*, 2021, pp. 3124–3135.

[26] H. Cho, J. Seol, and S.-g. Lee, "Masked contrastive learning for anomaly detection," *arXiv preprint arXiv:2105.08793*, 2021.

[27] V. Sehwag, M. Chiang, and P. Mittal, "Ssd: A unified framework for self-supervised outlier detection," *arXiv preprint arXiv:2103.12051*, 2021.

[28] C. U. Carmona, F.-X. Aubet, V. Flunkert, and J. Gasthaus, "Neural contextual anomaly detection for time series," *arXiv preprint arXiv:2107.07702*, 2021.

[29] C. Qiu, T. Pfrommer, M. Kloft, S. Mandt, and M. Rudolph, "Neural transformation learning for deep anomaly detection beyond images," *arXiv preprint arXiv:2103.16440*, 2021.

[30] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[31] B. Liu, C. Tan, J. Wang, T. Zeng, H. Shan, H. Yao, H. Heng, P. Dai, L. Bo, and Y. Chen, "Fedlearn-algo: A flexible open-source privacy-preserving machine learning platform," *arXiv preprint arXiv:2107.04129*, 2021. [Online]. Available: https://arxiv.org/abs/2107.04129

[32] A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith, "On the convergence of federated optimization in heterogeneous networks," *arXiv preprint arXiv:1812.06127*, vol. 3, 2018.

[33] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[34] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5132–5143.

[35] A. Reisizadeh, F. Farnia, R. Pedarsani, and A. Jadbabaie, "Robust federated learning: The case of affine distribution shifts," *arXiv preprint arXiv:2006.08907*, 2020.

[36] C. T. Dinh, N. H. Tran, and T. D. Nguyen, "Personalized federated learning with moreau envelopes," *arXiv preprint arXiv:2006.08848*, 2020.

[37] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," *arXiv preprint arXiv:2105.10056*, 2021.

[38] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multi-task optimization under privacy constraints," *arXiv preprint arXiv:1910.01991*, 2019.

[39] M. Xie, G. Long, T. Shen, T. Zhou, X. Wang, and J. Jiang, "Multi-center federated learning," *arXiv preprint arXiv:2005.01026*, 2020.

[40] P. P. Liang, T. Liu, L. Ziyin, R. Salakhutdinov, and L.-P. Morency, "Think locally, act globally: Federated learning with local and global representations," *arXiv preprint arXiv:2001.01523*, 2020.

[41] F. X. Yu, A. S. Rawat, A. K. Menon, and S. Kumar, "Federated learning with only positive labels," *arXiv:2004.10342*, 2020.

[42] K. Muhammad, Q. Wang, D. O'Reilly-Morgan, E. Tragos, B. Smyth, N. Hurley, J. Geraci, and A. Lawlor, "Fedfast: Going beyond average for faster training of federated recommender systems," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1234–1242.

[43] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," *arXiv preprint arXiv:2006.07242*, 2020.

[44] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[45] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1126–1135.

[46] S. Singh, S. Bhardwaj, H. Pandey, and G. Beniwal, "Anomaly detection using federated learning," in *Proceedings of International Conference on Artificial Intelligence and Applications*. Springer, 2021, pp. 141–148.

[47] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "Dïot: A federated self-learning anomaly detection system for iot," in *2019 IEEE 39th International conference on distributed computing systems (ICDCS)*. IEEE, 2019, pp. 756–767.

[48] Y. Zhao, J. Chen, D. Wu, J. Teng, and S. Yu, "Multi-task network anomaly detection using federated learning," in *Proceedings of the tenth international symposium on information and communication technology*, 2019, pp. 273–279.

[49] R. A. Sater and A. B. Hamza, "A federated learning approach to anomaly detection in smart buildings," *ACM Transactions on Internet of Things*, vol. 2, no. 4, pp. 1–23, 2021.

[50] X. Wang, S. Garg, H. Lin, J. Hu, G. Kaddoum, M. J. Piran, and M. S. Hossain, "Towards accurate anomaly detection in industrial internet-of-things using hierarchical federated learning," *IEEE Internet of Things Journal*, 2021.

[51] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated learning-based anomaly detection for iot security attacks," *IEEE Internet of Things Journal*, 2021.

[52] W. Liao, Y. Guo, X. Chen, and P. Li, "A unified unsupervised gaussian mixture variational autoencoder for high dimensional outlier detection," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 1208–1217.

[53] W. Hu, T. Miyato, S. Tokui, E. Matsumoto, and M. Sugiyama, "Learning discrete representations via information maximizing self-augmented training," *arXiv preprint arXiv:1702.08720*, 2017.

[54] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International conference on machine learning*, 2016, pp. 478–487.

[55] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[56] C. K. Chui and H. N. Mhaskar, "Deep nets for local manifold learning," *Frontiers in Applied Mathematics and Statistics*, vol. 4, p. 12, 2018.

[57] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei, "Circle loss: A unified perspective of pair similarity optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6398–6407.

[58] E. Keogh, J. Lin, and A. Fu, "Hot sax: Efficiently finding the most unusual time series subsequence," in *Fifth IEEE International Conference on Data Mining (ICDM'05)*. Ieee, 2005, pp. 8–pp.

[59] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[60] A. P. Mathur and N. O. Tippenhauer, "Swat: A water treatment testbed for research and training on ics security," in *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*. IEEE, 2016, pp. 31–36.

[61] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz *et al.*, "A public domain dataset for human activity recognition using smartphones." in *Esann*, vol. 3, 2013, p. 3.

[62] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uwave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 657–675, 2009.

[63] N. Hammami and M. Sellam, "Tree distribution classifier for automatic spoken arabic digit recognition," in *2009 International Conference for Internet Technology and Secured Transactions,(ICITST)*. IEEE, 2009, pp. 1–4.

[64] C. He, S. Li, J. So, X. Zeng, M. Zhang, H. Wang, X. Wang, P. Vepakomma, A. Singh, H. Qiu *et al.*, "Fedml: A research library and benchmark for federated machine learning," *arXiv preprint arXiv:2007.13518*, 2020.

[65] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.