FADngs: Federated Learning for Anomaly Detection

Boyu Dong, Dong Chen, Yu Wu, Siliang Tang, *Member, IEEE*, and Yueting Zhuang, *Senior Member, IEEE*

***Abstract*— With the increasing demand for data privacy, federated learning (FL) has gained popularity for various applications. Most existing FL works focus on the classification task, overlooking those scenarios where anomaly detection may also require privacy-preserving. Traditional anomaly detection algorithms cannot be directly applied to the FL setting due to false and missing detection issues. Moreover, with common aggregation methods used in FL (e.g., averaging model parameters), the global model cannot keep the capacities of local models in discriminating anomalies deviating from local distributions, which further degrades the performance. For the aforementioned challenges, we propose Federated Anomaly Detection with Noisy Global Density Estimation, and Self-supervised Ensemble Distillation (FADngs). Specifically, FADngs aligns the knowledge of data distributions from each client by sharing processed density functions. Besides, FADngs trains local models in an improved contrastive learning way that learns more discriminative representations specific for anomaly detection based on the shared density functions. Furthermore, FADngs aggregates capacities by ensemble distillation, which distills the knowledge learned from different distributions to the global model. Our experiments demonstrate that the proposed method significantly outperforms state-of-the-art federated anomaly detection methods. We also empirically show that the shared density function is privacy-preserving. The code for the proposed method is provided for research purposes https://github.com/kanade00/Federated_Anomaly_detection.**

***Index Terms*— Anomaly detection, distributed learning, federated learning (FL), unsupervised learning.**

I. Introduction

FEDERATED learning (FL) [1] focuses on training models from distributed clients collaboratively without sharing their privacy-sensitive data. Most FL works were studied in the classification setting [1], [2], [3], [4], [5], [6]. However, in practice, there are also some privacy-preserving scenarios of anomaly detection that detect anomalies from normal samples, such as financial fraud detection, network intrusion detection, and disease detection. In this article, we focus on federated anomaly detection, where anomalies are data instances that do not belong to the local data distributions of any clients.

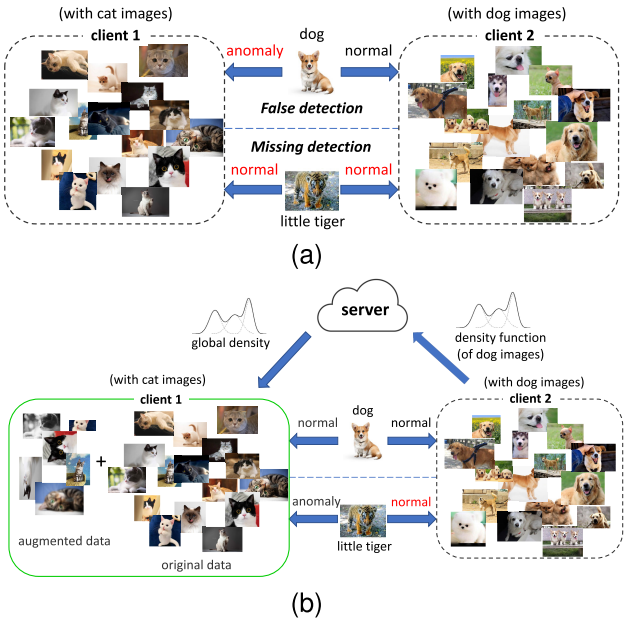

Fig. 1. False and missing detection in federated anomaly detection. (a) The scenario where cat and dog images are normal samples while images of other categories are anomalies. False detection is caused by non-IID data among clients, as client 1 holds cat images while client 2 holds dog images, which leads to different definitions of anomalies. Missing detection is caused by the limited data, as the local model cannot accurately learn normal distribution, which leads the model to regard a similar but anomaly sample as a normal sample. (b) Example of our solution to the aforementioned problems. For false detection, as client 1 gets the shared noisy density functions from client 2 and client 1, client 1 could correct the definition of anomaly. For missing detection, as local data are augmented according to the predicted results based on the shared noisy density functions, client 1 could learn normal distribution more accurately.

FL always faces the problems of *not identically and independently distributed* (non-IID) and limited data due to the diversity of client sources and limited resources of client devices (e.g., mobile phones), which results in false and missing detection issues. False detection issue is that the anomaly detection model predicts normal samples as anomalies, while missing detection means the model predicts anomalies as normal samples. Fig. 1(a) shows a detailed illustration of the false detection and missing detection. For **false detection**, the dog image (normal sample) in Fig. 1 may be predicted as different detection (anomaly/normal) results from different clients due to the diversity of local data distributions. However, from the perspective of global data distribution, the dog image should be treated as normal

Manuscript received 15 September 2022; revised 16 April 2023 and 30 October 2023; accepted 1 January 2024. This work was supported in part by the NSFC under Grant 62272411, in part by the Key Research and Development Projects in Zhejiang Province under Grant 2024C01106, in part by the National Key Research and Development Project of China under Grant 2018AAA0101900, and in part by Ant Group. *(Corresponding author: Siliang Tang.)*

Boyu Dong, Dong Chen, Siliang Tang, and Yueting Zhuang are with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China (e-mail: 22021016@zju.edu.cn; chendongcs@zju.edu.cn; siliang@zju.edu.cn; yzhuang@zju.edu.cn).

Yu Wu is with the School of Computer Science, Princeton University, Princeton, NJ 08540 USA (e-mail: yw5952@princeton.edu).

Digital Object Identifier 10.1109/TNNLS.2024.3350660

samples. For **missing detection**, the little tiger image (anomaly), which does not belong to any client, is detected as a normal sample due to the inaccurate representations learned from limited data.

Existing popular anomaly detection approaches, such as reconstruction-based methods [7], [8], [9], [10] usually require massive data to learn the discriminative representation of normal samples. However, this is impossible for FL, where each client only has very limited data (e.g., 100 data samples); thus missing detection is severe. Some prior works [11], [12], [13], [14], [15] proposed to learn more discriminative representations for anomaly detection by contrastive learning, which usually improves the diversity of training samples through data augmentation. Although these works make full use of data, the learned feature space is not suitable for anomaly detection. The contrastive optimization pushes any two samples to be far away from each other, while ignoring the difference between normal samples and anomalies. For a better anomaly detection task, we argue that the normal samples should have more similar representations (close to each other), and these should be far away from anomalies. Meanwhile, prior anomaly detection methods [16], [17], [18], [19], [20] mainly focus on centralized and IID data, which ignores the false detection issue in FL caused by non-IID data. In addition, non-IID data results in weight divergence among clients and degrades the performance of the aggregated model (i.e., the aggregated model cannot work with local distributions) [1]. For instance, if client 1 detects anomalies deviating from cats while client 2 detects anomalies deviating from dogs, the aggregated model that is aggregated by averaging parameters [21] may not detect anomalies deviating either from dogs or cats. Overall, it is necessary to propose a robust anomaly detection method for FL.

In this article, we tackle the false and missing detection issue by sharing noisy local density functions and learning more discriminative feature space. For false detection, we propose to construct a global density function set shared among all clients to align the knowledge of data distributions and the definition of anomalies. For privacy concerns, the shared density functions are estimated with noise that introduces samples belonging to other classes. As for the missing detection, we propose a new contrastive learning method that designs a specific optimization for anomaly detection, where normal-like samples will be pulled close while anomaly-like samples will be pushed away. Furthermore, to keep the capacities of local models within the global model (i.e., detect anomalies with specific local distributions), we propose to aggregate capacities rather than model parameters by ensemble distillation, where the capacities of detecting anomalies from the ensemble of local models will be used to guide the global model training.

This article proposes Federated Anomaly Detection with Noisy Global Density Estimation and Self-supervised Ensemble Distillation (FADngs) for federated anomaly detection learning. Specifically, to align the definition of anomalies and alleviate false detection while keeping privacy-preserving, FADngs performs weak clustering first, introducing different samples belonging to other clusters as random noise to ensure

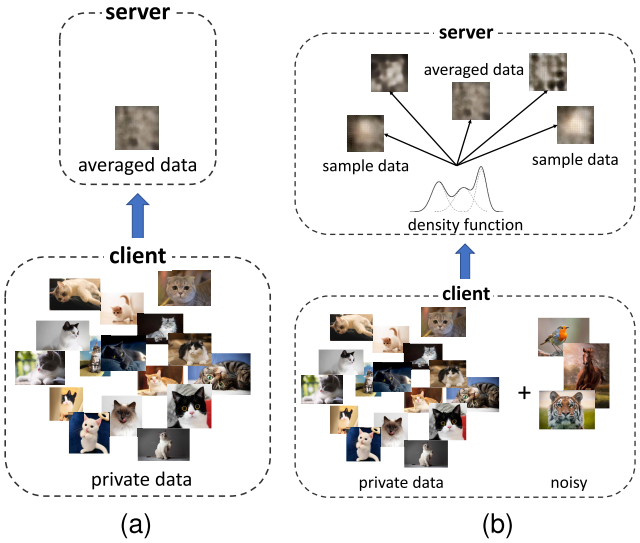

Fig. 2. Privacy concerns of our FL settings. Previous work like [22] and [23] transmit information about averaged local data to the server. As illustrated in (a), different from raw data, averaged data can hardly be identified; thus contains little information about privacy. As illustrated in (b), our work shares the noisy density functions in the representation space of clients, which is more difficult to expose the raw data compared to the aforementioned works. (a) Previous work. (b) FADngs.

the cluster's density function will not leak data privacy. The estimated noisy density function of each cluster will be sent to the server to construct a global density function set that will be shared among clients. Note that the noise introduced by samples are related to various classes, which will be neutralized in the global density set, as the global density set try to denote the distribution of all classes. With the shared global density functions, FADngs performs the selection to get predicted normal samples (i.e., samples that fit the shared noisy density functions) and anomalies (i.e., samples that do not fit the shared noisy density functions). Then, FADngs performs contrastive learning with augmented positive and negative samples, where positive samples are generated by the predicted normal samples with augmentations while negative samples are the predicted anomalies with augmentations. The training procedure calculates the similarity between data samples and gets the representation of positives closer while pushing the negatives away, which is meaningful for distinguishing anomalies. To keep the capacities of local models for the global model, FADngs aggregates capacities by ensemble distillation. In this distillation procedure, the local models are teachers and the global model is the student. FADngs distills the knowledge of detecting anomalies deviating from different distributions to the global model, which ensures the global model adapts well to various distributions.

As for the privacy concerns of FADngs passing noisy density functions, some recent methods like [22] and [23] have a similar setting to ours, which allows some meaningful information transmitted among the server and clients to integrate diverse data distributions from different clients. FedMix [22] allows each client to exchange its mashed (or averaged) data. FedProto [23] allows the clients and server to communicate the abstract class prototypes, where the

prototype is the mean representation of the observed samples. As illustrated in Fig. 2(a), the reconstructed averaged data can hardly be identified in relation to the original private data. Following these prior works, in FADngs, the clients share the noisy density functions in the low-dimension representation space. As illustrated in Fig. 2(b), it is difficult to sample and reconstruct some private information from these noisy density functions. Besides, we quantify data privacy by distance correlation, and we empirically show that the shared noisy density function will not leak data privacy.

The proposed method gets promising results compared to the recent federated anomaly detection method, FedDetect [24], and other anomaly detection algorithms run in federated settings. Moreover, related experiments show that the proposed methods alleviate false and missing detection effectively while maintaining privacy-preserving.

The main contributions of this article can be summarized as follows.

1) We are the first to formulate the false and missing detection issues for federated anomaly detection.
2) We propose to align the definition of anomalies among clients by sharing noisy density functions for false detection. We also aggregate the capacities of local models to enhance the global model detection anomalies with specific local distributions.
3) We propose to improve learned representation by augmenting data with the shared noisy density functions for missing detection, where the representations of all predicted normal samples are optimized to be closer while anomalies are pushed away.
4) The proposed FADngs relatively improves the baseline FedDetect [24] by about 28.83% and 26.40% in terms of area under the receiver operating characteristic curve (AUROC) and area under the precision-recall curve (AUPR), respectively.

II. Related Works

A. Anomaly Detection

We roughly divide anomaly detection into classification-based, reconstruction-based, and density-based approaches. Classification-based methods [20], [25], [26], [27], [28], such as one-class support vector machine (SVM) [25] are always well explainable. However, these traditional algorithms are not suitable for high-dimensional data. Nguyen and Vien [25] combine one-class SVM with deep learning techniques to address the above problem but still cannot get satisfactory results. Multilayer one-class classification for anomaly detection (MOCCA) [20] optimizes each layer's feature space for the anomaly detection task during the model training, but is not suitable for non-IID data. Reconstruction-based anomaly methods [9], [10], [29], [30], [31], try to learn the normal distribution by reconstructing input data, but this kind of method usually requires lots of data, which is impossible in FL. Density-based methods [32], [33], [34], [35] also require lots of data, as sample covariance is not an accurate estimator of covariance when the number of samples is less than the dimension of feature space [36].

To learn discriminative representation for federated anomaly detection, we propose to train the local model with contrastive learning. And the learned feature space will be used for the density-based method. There are some similar works [11], [12], [13], [14], [37], [38], [39], [40] that focus on self-supervised learning, particularly contrastive learning. Reference [37] is the first work to achieve anomaly detection by augmented samples with contrastive learning. Hendrycks et al. [11] find that contrastive learning can drastically improve anomaly detection on difficult, near-distribution anomalies. However, these works are not suitable for distributed learning, as there is limited data on each client. Simple contrastive learning (SimCLR) [41] uses image transformations to create different augmented views of the same data example. The augmented views are considered positive samples and other data examples are considered negative samples. While Tack et al. [12] propose that some samples with distributionally-shifted augmentations could also be considered negatives. This "hard" augmentations can help the model to learn a new task of discriminating normal and anomaly samples, which results in improved performance. Han et al. [14] propose a semi-supervised approach based on [12], which leverages a new energy function to circumvent the possible weakness of data contamination.

Inspired by the effectiveness of [12], we propose a new contrastive learning method that designs a specific optimization for anomaly detection, where the training process will bring all normal samples' representations closer, which is meaningful for recognizing anomalies.

Moreover, for limited data, some methods like [40], [42], and [43] generate anomaly-like samples for improving model training through data augmentation. Ramírez Rivera et al. [42] synthesize anomalies on the outskirts of the training data from a two-level hierarchical latent space, which relies on high-quality data to generate reliable anomalies. Liu et al. [43] identify anomalies from the data itself, where anomalies are discriminated by exploring representative neighbors and leveraging a tailored graph clustering technique. But [43] has high time complexity and is not suitable for FL with limited resources. Our method also identifies anomalies like [43] but is performed in a more simple way, and uses them as negative samples in contrastive learning.

B. Federated Learning

Non-IID problem [1] is the key challenge of FL. Some works [1], [44], [45], [46], [47], [48], [49] focus on improving the aggregation method rather than directly averaging model parameters in [21]. Knowledge distillation [50] is considered as a way of updating the global model in these methods: FedDF [47] proposes to use ensemble distillation [50], [51] for robust model aggregation, which allows for heterogeneous client models and data, and is robust to the choices of neural architectures; FedBE [48] incorporate Bayesian model ensemble into FL, and summarize model ensemble into a single global model by knowledge distillation.

Other works [22], [23], [52], [53], [54], [55], [56], [57] aim at improving local training in the clients, FedProx [54] proposes to add a proximal term to the objective that helps to improve the local training; FedMix [22] proposes a framework that clients send and receive averaged local data, and use

a new augmentation algorithm based on this information; FedProto [23] proposes a framework that the clients and server communicate the abstract class prototypes, which are the mean representations transformed from the observed samples belonging to the same class. The framework aggregates the local prototypes from the clients and then sends the global prototypes back to all clients to regularize the local training. Similarly, our method aggregates the local density functions which represent the local data distributions from the clients, and then sends the global density function back to all clients to improve training and detection.

C. Federated Learning for Anomaly Detection

Currently, there are few works on federated anomaly detection. Zhao et al. [58] propose a classification-based method for network anomaly detection in FL, which tackles the data scarcity problem by training three tasks simultaneously. However, this method is not applicable to unsupervised tasks in practice. Khan et al. [59] and Zhang et al. [60] study federated anomaly detection with reconstruction-based method that is totally unsupervised like our work. Specifically, they learn a threshold for the trained global model, if one input data sample achieves a reconstruction error above the threshold will be detected as an abnormal data sample. For this important task in practice, our work focuses on the unsupervised anomaly detection method and minimizes the risk of privacy leakage.

III. Method

In this section, we present FADngs, which consists of six stages: 1) train the model with vanilla contrastive loss; 2) cluster with early stopping based on the representation extracted by the trained model and estimate the noisy density function for each cluster; 3) share local noisy density functions to construct the global density function set; 4) predict anomalies based on the shared global density functions, and augment data as positive and negative samples, then train local models with an improved contrastive loss; 5) aggregate the capacities of local models into the global model with self-supervised ensemble distillation (SED); and 6) detecting anomalies with the global model and global density functions. The critical steps contain Noisy Global Density Estimation (NGDE, stages 2 and 3), Contrastive Representation for Anomaly Detection (CRAD, stage 4) and SED (stage 5) are presented in Figs. 3–5, respectively. Note that the model is mainly trained in two stages: CRAD and SED.

A. Noisy Global Density Estimation

To address the false detection issue, we propose to share the noisy density functions of each client in a low-dimensional representation space for the false detection problem, which is proven to be privacy-preserving in experiments. First, each client gets low-dimensional representations $z(x)$ of all the training data by the current global model $f(\cdot)$. Then, we perform clustering with early stopping to introduce noise (samples that should belong to other classes in local data) into each cluster, and we call this process weak clustering. Assume that there are C clients in total, we partition the features into K_c clusters for the cth client. For each cluster, we model the Gaussian density function as follows:

$$\hat{\mu}_k = \frac{1}{N_k}\sum_{i=1}^{N_k} z(x_i) \tag{1}$$

$$\hat{\Sigma}_k = \frac{1}{N_k}\sum_{i=1}^{N_k}\big(z(x_i)-\hat{\mu}_k\big)\big(z(x_i)-\hat{\mu}_k\big)^\top \tag{2}$$

where $\hat{\mu}_k$ and $\hat{\Sigma}_k$ are the mean and covariance of the Gaussian density function for the kth cluster, N_k is the number of features in the kth cluster. Note that $z(x)$ is computed by $z(x) = h(f(x))$, where $h(\cdot)$ is an additional projection layer after extracting the features.

Since the number of samples N_k is usually small due to the limited resources of clients, the estimation of covariance $\hat{\Sigma}_k$ will become unstable. Thus, we additionally use Shrinkage covariance estimation [61] to improve the accuracy of $\hat{\Sigma}_k$, where Shrinkage is defined as a linear combination of empirically estimated $\hat{\Sigma}$ and the identity matrix I_D

$$\hat{\Sigma}_{\text{shrink}} = (1-\rho)\hat{\Sigma} + \rho\frac{\text{tr}\big(\hat{\Sigma}\big)}{D} I_D \tag{3}$$

where D is the dimension of the feature space and I_D is the regularization term to reduce the variance of the result while introducing some bias in turn. ρ is shrinkage intensity that regulates the impact of I_D on the final matrix.

Note that the clustered density functions differ from true density functions corresponding to raw data because of the introduced noise. That is, the clustering process is incomplete by early stopping, so each cluster will have little data that should belong to other clusters, which can be defined as

$$\hat{\mu}_k = \hat{\mu}_k^t + \hat{\mu}_k^n \tag{4}$$

$$\hat{\Sigma}_k = \frac{1}{N_k}\left(\sum_{i=1}^{N_n}\big(z(x_i)-\hat{\mu}_k\big)\big(z(x_i)-\hat{\mu}_k\big)^\top + \sum_{i=1}^{N_t}\big(z(x_i)-\hat{\mu}_k\big)\big(z(x_i)-\hat{\mu}_k\big)^\top\right) \tag{5}$$

where $\hat{\mu}_k^t$ is the mean of examples that belongs to the class of largest proportion in the kth cluster, $\hat{\mu}_k^n$ is the mean of rest examples that can be regarded as noise, covariance is defined in the same way. Moreover, NGDE can be integrated with various privacy-preserving techniques to introduce additional noise and further enhance security. Thus, the proposed NGDE only shares a fuzzy distribution that can roughly show the data distribution in the feature space, minimizing the risk of privacy leakage.

After the aforementioned process, the cth client will send the server calculated density functions of K_c clusters. For simplicity, in experiments, we set K_c to 1 for each client due to the limited data of each client in FL. Then the server aggregates all the received density functions and gets a mixture global density functions of the whole data, which is presented in Fig. 3(a). As for the number of density functions M the server received, it depends on the number of clusters each

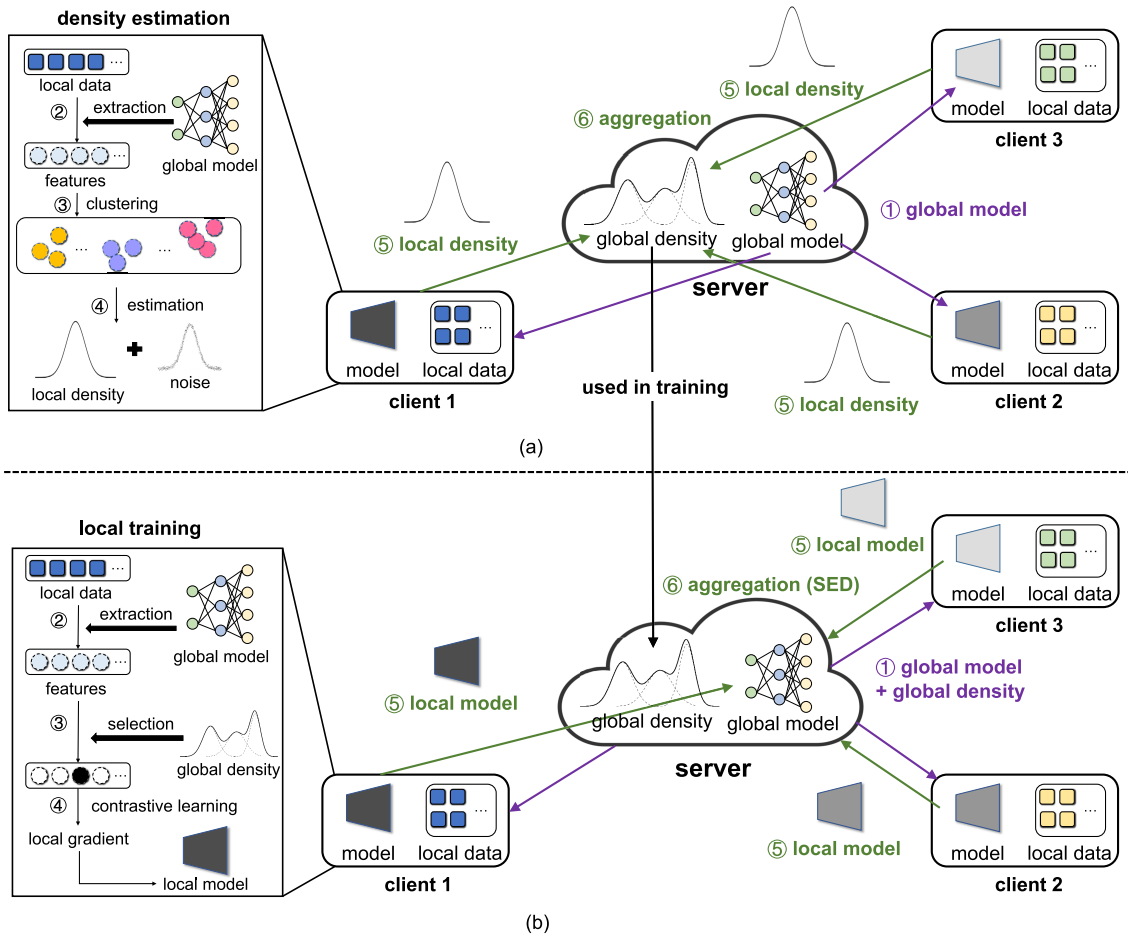

Fig. 3. Overview of FADngs that includes NGDE and CRAD. (a) NGDE contains the following steps: 1) transmit global model to the clients; 2) extract the features of local data; 3) perform weak cluster that with early stopping to keep privacy; 4) estimate noisy density functions; 5) transmit local density functions to the server; and 6) construct the global density function in the server. (b) CRAD contains the following steps: 1) transmit global model and global density functions to each client; 2) extract features from local data; 3) predict normal samples and anomalies with the shared global density functions; 4) augment data with the predicted results and learn features by contrastive learning; 5) transmit local model to the server; and 6) update the global model in the server.

client holds, i.e., $M = \sum_{c=1}^{C} K_c$. Thus, the server holds mean set $\{\hat{\mu}_k\}_{k=1}^{M}$ and covariance set $\{\hat{\Sigma}_k\}_{k=1}^{M}$ after estimation.

B. Score Function for Detection

Based on the trained model and the global density functions, we define a score function for detecting whether a given sample x is an anomaly or not. Specifically, after getting the low-dimensional representation $z(x)$ with the current global model, we calculate Mahalanobis distance [62] between $z(x)$ and noisy density functions as anomaly score

$$s(x) = \min_{1\le k\le M} \big(z(x) - \hat{\mu}_k\big)^{\top} \hat{\Sigma}_k^{-1} \big(z(x) - \hat{\mu}_k\big). \tag{6}$$

With this score function, we can predict whether a sample is an anomaly or not according to the distance between it and the closest cluster. Note that although the density function does not expose the true data distribution, it still provides a vague distribution range in the testing phase.

C. Contrastive Representation for Anomaly Detection

To address the missing detection issue caused by limited data, we propose to learn more discriminative feature space for anomaly detection with the proposed contrastive method that takes full advantage of limited data to construct a feature space for anomaly detection. Contrastive learning aims to learn representations by contrasting positive pairs against negative pairs. Furthermore, we use image transformations to create different augmented views of the same data example like SimCLR [41]. In particular, the augmented views are considered as positive, and other samples are considered as negative. For a sample x with its positive samples $\{x_+\}$ and negative samples $\{x_-\}$, the model $f(\cdot)$ is optimized to pull the feature of x close to its positives while pushing away from negatives. The contrastive loss is defined as follows:

$$\mathcal{L}_{\text{con}}(x, \{x_+\}, \{x_-\}) = -\frac{1}{|\{x_+\}|} \log \frac{\sum_{x'\in\{x_+\}} \exp\big(\text{sim}\big(z(x), z\big(x'\big)\big)/\tau}{\sum_{x'\in\{x_+\}\cup\{x_-\}} \exp(\text{sim}(z(x), z(x'))/\tau} \tag{7}$$

where $\text{sim}(z, z') = z^{\top} z'/\|z\|\|z'\|$ denotes the dot product between l_2 normalized z and z', τ denotes a temperature parameter and $|\{x_+\}|$ denotes the cardinality of the set $\{x_+\}$.

For the vanilla SimCLR that learns with contrastive loss, the set of negative samples $\{x_-\}$ consists of all the other instances in one batch when training. However, this way cannot apply to anomaly detection, as all normal samples should be pulled closer, while all anomalies should be pushed

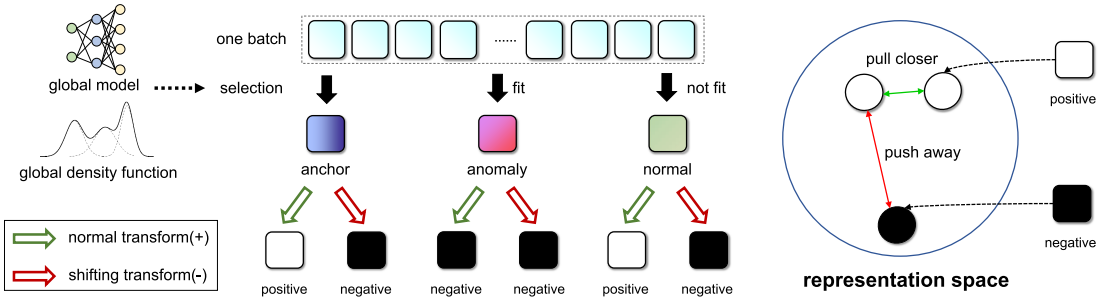

Fig. 4. Details of generating positive and negative samples in CRAD. In a batch, each sample will be inferred according to the global density functions with (7) at first. Then, all the samples will be augmented in two ways: normal transformations and shifting transformations. For the predicted normal samples, the augmented samples generated by normal transformations are considered positives, while other augmented samples generated by shifting transformations are negatives. As for the predicted abnormal samples, all the augmented samples generated are only negatives. Note that only the predicted normal samples are used as anchor images when learning feature space with contrastive loss.

away from normal samples. Recent work [12] has shown that samples that are augmented with distribution-shifting augmentations as negatives can improve anomaly detection significantly, but instances except the anchor are still regarded as negatives. Nevertheless, we argue that samples that are more likely to be normal should be pulled closed, which is away from negative samples. Thus, we propose CRAD. CRAD first performs global selection with the global density to get predicted normal and anomaly samples with (6). Then in the contrastive procedure, the predicted normal samples will be augmented as positive and negative samples according to the type of augmentations, while the predicted anomalies will be augmented as negative samples only. The training process gets the representations of all predicted normal samples closer, which is specific for anomaly detection. Note that the imbalance between normal samples and anomalies will not lead to an imbalance in generating positive and negative samples, as the number of normal samples is significantly greater than that of anomalies. In other words, anomalies do not have a significant impact on the quantity of negative samples. More specifically, we define a set $\mathcal{T}$ consisting of different transformations, and the augmented samples generated by transformations $\mathcal{T}_-$ such as rotation are all considered as negatives. Other augmented samples of predicted normal samples generated by transformations $\mathcal{T}_+$ such as crop, grayscale, and color jitter are considered as positive samples. We called the transformations from $\mathcal{T}_-$ as shifting transformations and that from $\mathcal{T}_+$ as normal transformations. The illustration of the proposed CRDA is presented in Fig. 4.

During the training step, for a given batch $\mathcal{B} := \{x_i\}_{i=1}^{B}$, the predicted abnormal samples set $\mathcal{A}$ is defined as

$$\mathcal{A} := \{x_i | s(x_i) > \delta\}, \forall x_i \in \mathcal{B} \tag{8}$$

where δ is the threshold for detection, which is typically chosen so that a high fraction of training data (e.g., 95%) is classified as normal data. The set $\{x_+\}$ and $\{x_-\}$ for a given instance x is defined as

$$\begin{aligned} \{x_+\} &:= \{T_+(x_i) | s(x_i) \leq \delta\} \\ \{x_-\} &:= \{T_+(x_i) | s(x_i) > \delta\} \cup \{T_-(x_i)\} \\ &\forall x_i \in \mathcal{B} \quad \forall T_- \in \mathcal{T}_- \quad \forall T_+ \in \mathcal{T}_+. \end{aligned} \tag{9}$$

Note that only the predicted normal samples that do not belong to $\mathcal{A}$ is used as anchor images in contrastive loss.

As shown in Fig. 3 (right), each client updates the local model with its privacy data, and the local models will be sent to the server after fixed epochs to get a new global model.

D. Self-Supervised Ensemble Distillation for Capacity Aggregation

To further address the missing detection issue, we propose a new model aggregation strategy based on ensemble learning [51] and knowledge distillation [50] to make the global model keep the capacities of local models which are trained with CRAD, where data samples for distillation are from a public dataset.

With the local model parameters $\{\theta_c, \forall c \in S_t\}$ from the activated set of clients S_t in each communication round, traditional FL framework, like FedAvg [21] takes an average for aggregation, which is computed as follows:

$$\theta \leftarrow \sum_{c \in S_t} \frac{n_c}{\sum_{c \in S_t} n_c} \theta_c \tag{10}$$

where n_c is the number of samples in the client c.

To make the aggregated global model keep the local models' capacities, we propose SED to perform an ensemble distillation procedure, where the local models are teachers (distillation source) and the global model is the student (distillation target). As we consider that the knowledge of distinguishing between normal samples and anomalies is most important for anomaly detection, the knowledge in SED is related to the similarity among data samples (i.e., the similarities between normal and anomaly samples must be low while the similarities among normal samples must be high). Specifically, as illustrated in Fig. 5, each local model maintains an instance queue on the server. Then, all models will output features with extractors with the same inputs, and the similarity between output features and instance queue will be calculated. SED aligns these similarity score distributions between the global model and the ensemble of local models, which make the global model gain the capacity to detect anomalies with specific distributions.

Specifically, for each activated client c, we maintain a contrastive instance queue $D_c = [d_c^1, d_c^2, \ldots, d_c^B]$ to store data samples' representations output from the client teacher models. And D is progressively updated under the "first-in first-out" strategy as distillation proceeds. Specifically, in one distillation

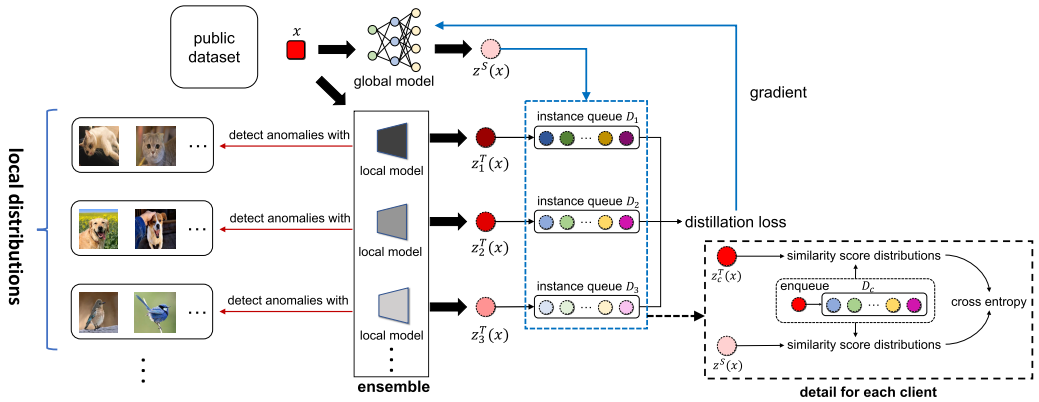

Fig. 5. Details of SED. We propose to aggregate the capacities of local models into the global model by SED strategy. Specifically, for each activated client, we maintain a contrastive instance queue to store data samples' representations output from the local model. The distillation objective is the cross-entropy between the similarity score distributions computed by local models and the global model. The instance queue is dynamically updated so that current samples' representations will be added to the queue, and the earliest elements will be removed.

step, we get a new batch of data samples and compute their representations, then add the current representations to one end of the queue and remove the same number of samples' representations from the other end of the queue. For a given data instance x, we extract the feature $z_c^T(x)$ computed by the current teacher model, then calculate the similarity scores $P_c(x; D_c, \theta_c)$ between $z_c^T(x)$ and the elements stored in the current queue D_c

$$P_c(x; D_c, \theta_c) = \left[p_c^1, p_c^2, \ldots, p_c^B\right]$$
$$p_c^j = \frac{\exp\left(\mathrm{sim}\left(z_c^T(x), d_c^j\right)/\tau\right.}{\sum_{d_c \in D_c} \exp\left(\mathrm{sim}\left(z_c^T(x), d_c\right)/\tau\right.} \tag{11}$$

where $\mathrm{sim}(z, d) = z^\top d/\|z\|\|d\|$.

Similarly, we extract the feature $z^S(x)$ computed by the current student model, then calculate the similarity scores $Q_c(x; D_c, \theta)$

$$Q_c(x; D_c, \theta) = \left[q_c^1, q_c^2, \ldots, q_c^B\right]$$
$$q_c^j = \frac{\exp\left(\mathrm{sim}\left(z^S(x), d_c^j\right)/\tau\right.}{\sum_{d_c \in D_c} \exp\left(\mathrm{sim}\left(z^S(x), d_c\right)/\tau\right.}. \tag{12}$$

As for distillation, we minimize the cross-entropy between the similarity score distributions computed by the student and the ensemble of teachers. Since we have multiple local models learned from diverse data distributions, we take an ensemble to get more comprehensive knowledge for distillation. Specifically, for the data samples x from a given batch $\mathcal{B}$, we calculate similarity scores $P_c(x; D_c, \theta_c)$ and $Q_c(x; D_c, \theta)$ for all the clients with instance queue D_c, the global model is updated as follows:

$$\theta \leftarrow \arg\min_\theta \sum_{c \in S_t} w_c \sum_{x \in \mathcal{B}} -P_c(x; D_c, \theta_c) \log Q_c(x; D_c, \theta). \tag{13}$$

The setting of w_c is related to the dynamic weighting strategy we adopted due to the different capacities of the teacher models. Here, we leverage each teacher model to get the anomaly scores of all the samples from the public dataset and take the mean score of each sample (the scores will be normalized). w_c for each teacher is computed based on the Pearson correlation [63] between its corresponding anomaly score set and mean score set.

For the concern of privacy-preserving, the public dataset used in distillation should be unrelated to the private data on the clients. As in [47], our framework uses unlabeled datasets from other domains or synthetic data from a pretrained generator.

The whole training algorithm procedure is presented in Algorithm 1.

E. Theoretical Analysis of Privacy-Preserving

We further perform theoretical analysis from the entropy perspective to show that introducing noise in NGDE can preserve privacy. We first present a lemma

$$a = \sum_{i=1}^n a_i, \quad b = \sum_{i=1}^n b_i, \quad \sum_{i=1}^n a_i \log_2 \frac{a_i}{b_i} \geq a \log_2 \frac{a}{b} \tag{14}$$

where a, b are samples from different modalities, n is the number of samples in one modality.

Let $\mathbb{Z}_C$ and $\mathbb{Z}_N$ be the variables of clean samples and noisy samples in the latent space, respectively. We use entropy to represent the uncertainty in a sample. The higher the uncertainty, the less likely the private content of the sample will be exposed. $H(\mathbb{Z}_N)$ is the entropy of noisy sample and $H(\mathbb{Z}_C)$ is the entropy of clean sample. Besides, let $H(\mathbb{Z}_C) = H(\mathbb{Z}_N \mid \mathbb{Y})$ to represent that the entropy of $\mathbb{Z}_N$ gets similar values to that of $\mathbb{Z}_C$ after getting rid of noise variable, $\mathbb{Y}$. Now we show that the proposed method is privacy-preserving, as the entropy of $H(\mathbb{Z}_N)$ is higher than that of $H(\mathbb{Z}_C)$

$$\begin{aligned} &H(\mathbb{Z}_N) - H(\mathbb{Z}_N \mid \mathbb{Y}) \\ &\quad = \sum_{z_n \in \mathbb{Z}_N} -p(z_n) \log_2 p(z_n) \\ &\qquad - \sum_{z_n \in \mathbb{Z}_N} \sum_{y \in \mathbb{Y}} -p(z_n, y) \log_2 p(z_n \mid y) \\ &\quad = \sum_{z_n \in \mathbb{Z}_N} \sum_{y \in \mathbb{Y}} p(z_n, y) \log_2 \frac{p(z_n, y)}{p(z_n) p(y)}. \end{aligned} \tag{15}$$

Algorithm 1 Training Algorithm for Federated Anomaly Detection

Input: C clients indexed by c, datasets X splited in $\{x_c\}_{c=1}^{C}$, public dataset X_D, number of clusters K, local epoch E_l, global distillation epoch E_g and total global rounds T
Output: final global model parameters θ
Server executes:
initialize parameters θ_0
for $t = 1$ **to** T **do**
  $S_t \leftarrow$ (random set S of clients)
  foreach *client* $c \in S_t$ **do**
    $\{\hat{\mu}_k^c\}_{k=1}^K, \{\hat{\Sigma}_k^c\}_{k=1}^K \leftarrow$ `GetDensityFunctions`(c, θ_{t-1})
  aggregates all the local density functions and get the global density function $\{\hat{\mu}\}$, $\{\hat{\Sigma}\}$
  foreach *client* $c \in S_t$ **do**
    $\theta_t^c \leftarrow$ `LocalUpdate`$(c,\ \theta_{t-1},\ \{\hat{\mu}\},\ \{\hat{\Sigma}\})$
  $\theta_t \leftarrow$ `Aggregate`$(\{\theta_t^c, \forall c \in S_t\})$
return θ_T
Function `GetDensityFunctions`(c, θ)**:**
  foreach *batch* $x \in$ *local dataset* X_c **do**
    compute low-dimensional representation z
  partition the representations in K clusters with clustering algorithm
  estimate $\{\hat{\mu}_k\}_{k=1}^K$ and $\{\hat{\Sigma}_k\}_{k=1}^K$ for each cluster using Eq. 1, Eq. 2 and Eq. 3
  introduce noise for $\{\hat{\mu}_k\}_{k=1}^K$ and $\{\hat{\Sigma}_k\}_{k=1}^K$ using Eq. 4 and Eq. 5
  return $\{\hat{\mu}_k\}_{k=1}^K$, $\{\hat{\Sigma}_k\}_{k=1}^K$
Function `LocalUpdate`$(c, \theta_{t-1},\ \{\hat{\mu}\},\ \{\hat{\Sigma}\})$**:**
  initialize local parameters $\theta_{t,0}^c \leftarrow \theta_{t-1}$
  for $i = 1$ **to** E_l **do**
    foreach *batch* $x \in$ *local dataset* X_c **do**
      predict anomalies using Eq. 6 with the density function $\{\hat{\mu}\}$ and $\{\hat{\Sigma}\}$
    foreach *batch* $x \in$ *local dataset* X_c **do**
      update model parameters $\theta_{t,i-1}^c$ using Eq. 7 with negative samples $\{x_-\}$ calculated by Eq. 9
  return θ_{t,E_l}^c
Function `Aggregate`$(\{\theta_t^c, \forall c \in S_t\})$**:**
  initialize parameters $\theta_{t,0} \leftarrow \sum_{c \in S_t} \frac{n_c}{\sum_{c \in S_t} n_c} \theta_t^c$
  foreach *client* $c \in S_t$ **do**
    initialize instances queue D_c randomly
  for $i = 1$ **to** E_g **do**
    foreach *batch* $x \in$ *public dataset* X_D **do**
      foreach *client* $c \in S_t$ **do**
        calculate $P_c(x; D_c, \theta_t^c)$ and $Q_c(x; D_c, \theta_{t,i-1})$ using Eq. 11 and Eq. 12
        Update instances queue D_c with current batch
      Update global model parameters $\theta_{t,i-1}$ using Eq. 13
  return θ_{t,E_g}

According to Lemma 14, (15) can be further written as

$$\sum_{z_n \in \mathbb{Z}_N} \sum_{y \in \mathbb{Y}} p(z_n, y) \log_2 \frac{p(z_n, y)}{p(z_n)p(y)} \geq \left[\sum_{z_n \in \mathbb{Z}_N} \sum_{y \in \mathbb{Y}} p(z_n, y) \right] \log_2 \frac{\sum_{z_n \in \mathbb{Z}_N} \sum_{y \in \mathbb{Y}} p(z_n, y)}{\sum_{z_n \in \mathbb{Z}_N} \sum_{y \in \mathbb{Y}} p(z_n)p(y)} = 0 \quad (16)$$

Note that the above inequality takes the equal sign iff $z_n \in \mathbb{Z}_N$ and $\mathbb{Y}$ are independent, i.e., $p(z_n, y) = p(z_n)p(y)$. However, according to [64], images usually contain noise, $\mathbb{Z}$ and Y are dependent. Thus, we get $H(\mathbb{X}) > H(\mathbb{X} \mid \mathbb{Y})$.

IV. Experiments

In our experiments, we aim to: 1) show the effectiveness of the proposed FADngs compared to state-of-the-art methods for federated anomaly detection; 2) validate the effectiveness of NGDE, CRAD and SED, respectively; 3) verify the proposed methods alleviate false and missing detection effectively; and 4) verify the proposed method is privacy-preserving.

Setup: We conduct experiments on CIFAR-10 [65], CIFAR-100 [65], SVHN [66] and Places365 [67]. For the setting of FL, we set ten clients, and each client contains 1000 data instances in the training set. To simulate the non-IID situation, each client has a high proportion of data from a single class (e.g., most samples of one client are pictures of dogs while most samples of another client are pictures of cats). Moreover, most of the data in the training set are from CIFAR-10 regarded as normal samples, while to simulate real-world scenarios the training data usually contains unlabeled anomalies as noisy instances, there is a small percentage (1%) of data from other datasets as anomaly-contaminated data.

In each experiment, we use ResNet-34 [68] as feature extractors trained with stochastic gradient descent for 1000 rounds. For local training with CRAD, it takes 1 epoch in each communication round. Learning rate is set to be 0.01 with cosine decay. Weight decay and batch size are set to $1e^{-4}$ and 32, respectively. The temperature parameter is set to 0.5 in (7). We set K_c to be 1 for each client in clustering, the shrinkage intensity ρ in (3) to be 0.1, δ in (8) to be that 95% of the training data is classified as normal. For the distillation procedure in SED, the global model is trained with stochastic gradient descent for 5 epochs, with a 0.03 learning rate in each communication round. The size of the queue is all set to 65 536. All aforementioned hyperparameters are decided by grid search. We use synthetic data from a pretrained generator as a public dataset. To evaluate the performance of the methods, we use AUROC [69] and AUPR as evaluation metrics. Besides, to test whether the proposed methods can effectively alleviate false and missing detection, we define new metrics $\texttt{False}_{\texttt{error}}$ (FE) and $\texttt{Missing}_{\texttt{error}}$ (ME) as

$$\begin{aligned} \texttt{False}_{\texttt{error}}(\text{FE}) &= \frac{\text{FP}}{\text{TP}+\text{FP}} \\ \texttt{Missing}_{\texttt{error}}(\text{ME}) &= \frac{\text{FN}}{\text{TP}+\text{FN}} \end{aligned} \tag{17}$$

where TP denotes anomaly samples are detected correctly, FP denotes normal samples are detected as anomaly samples, and FN denotes anomaly samples are detected as normal samples. Note that lower values of FE and ME represent better model performance. For data augmentations in CRAD, the augmented samples with shifting transformations $\mathcal{T}_-$ (e.g., rotation 90°, 180°, 270°) are regarded as anomalies, while the augmented samples with transformations $\mathcal{T}_+$ (e.g., crop, grayscale, and color jitter) are regarded as normal samples. Furthermore, the model is trained using SimCLR [41] at first on the same dataset before conducting the proposed method to stabilize the training optimization.

A. Compare With State-of-the-Art Methods

In Table I, we compare three mainstream anomaly detection methods that are reconstruction-based methods (e.g., auto-encoder (AE) [30], FedDetect [24]), classification-based methods (e.g., one class SVM (OCSVM) [25], deep support vector data description (Deep-SVDD) [26] and Panda [28]), and contrastive learning methods (e.g., contrasting shifted instances (CSI) [12], Elsa [14], and our method FADngs). Contrastive learning methods get the best results for all metrics, which illustrates that they are effective for the limited data problem in federated anomaly detection. Moreover, the results in Table I show that the proposed method significantly outperforms all the state-of-the-art methods. In fact, most previous methods seem not to be effective for federated anomaly detection as the results are close to the random guess performance (50%) on some datasets due to the non-IID and limited data among clients. In contrast, our method achieves 77.8%, 96.6%, 83.8% on AUROC and 75.1%, 95.3%, 82.2% on AUPR, respectively, which shows that the proposed FADngs is effective for anomaly detection in federated settings. Besides, the results of FE and ME of FADngs are much better than those of other algorithms, which illustrates the proposed method effectively alleviates false and missing detection issues. Meanwhile, compared to the prior contrastive learning method (CSI) [12], the results of FADngs are on average 18.13% and 16.30% higher for AUROC and AUPR, respectively, which shows the importance of the combination of NGDE, CRAD, and SED. Especially compared with the most recent federated anomaly detection algorithm, FedDetect [24], which learns AE and specific threshold, our method outperforms it up to 28.83% and 26.40% relatively of AUROC and AUPR, respectively. Moreover, we also compare different anomaly detection methods when training data contains no anomalies in Table II. Our method also outperforms other methods and shows robustness to contaminated data as the results in Table I.

To better demonstrate the effect of the proposed methods, FADngs, on false and missing detection, we use t-distributed stochastic neighbor embedding (t-SNE) [70] to visualize the representations of samples in feature space. For false detection, we illustrate the effectiveness of NGDE in Fig. 6. At first, we visualize the representations of normal samples learned by our method from different clients (to be more clear, we mainly show the results from three clients). Fig. 6(a) shows that the points are quite separated when they are from different clients and Fig. 6(b) demonstrates density estimation results. As illustrated in Fig. 6(d), samples from other clients are easily detected as anomalies incorrectly, as most of the points are red. In contrast, with sharing density functions of NGDE that align the definition of anomalies, most normal samples are detected correctly in Fig. 6(c). More specifically, the result in Fig. 6(d) only uses the local density functions of client 1, and we could find most of the points detected as normal are located in the bottom right, which is the region closer to the density functions of client 1. Furthermore, the result in Fig. 6(c) demonstrates a lower rate of false detection, as only the points far away from three density functions are likely to be detected as anomalies. As for missing detection, we compare the representations learned by FADngs and CSI in Fig. 7. We visualize the representations from different clients in different colors (where red points denote anomalies) in Fig. 7(a) and (c) learned by FADngs and CSI, respectively. Specifically, we only visualize anomalies in Fig. 7(b) and (d) detected by FADngs and CSI, respectively, where black points denote missing detection (note that, in fact, in Fig. 7(b) and (d), all points are anomalies, while some points are detected as normal samples. Thus, they are represented by black points). It can be seen that even the best baseline, CSI, which gets the highest results for AUROC and AUPR in Table I (compared

TABLE I

COMPARISON OF THREE DIFFERENT CATEGORIES OF FEDERATED ANOMALY DETECTION. RB DENOTES RECONSTRUCTION-BASED METHODS (E.G., AE [30], FEDDETECT [24]), CB DENOTES CLASSIFICATION-BASED METHODS (E.G., OCSVM [25], DEEP-SVDD [26], AND PANDA [28]), AND CL DENOTES CONTRASTIVE LEARNING METHOD (E.G., CSI [12], ELSA [14], AND OUR METHOD FADNGS W/O SED, FADNGS). FADNGS W/O SED DENOTES USING OUR METHOD WITHOUT SED BUT SIMPLY AVERAGING MODEL PARAMETERS LIKE FEDAVG [21]. NOTE THAT ALL BASELINE METHODS ARE COMBINED WITH FEDAVG [21] TO RUN FEDERATED EXPERIMENTS

Normal Dataset	Anomaly Dataset	Method		FE(%) ↓	ME(%) ↓	AUROC(%) ↑	AUPR(%) ↑
CIFAR-10	CIFAR-100	RB	AE [30]	$46.6_{\pm 0.9}$	$49.0_{\pm 1.8}$	$55.6_{\pm 1.3}$	$56.1_{\pm 2.6}$
			FedDetect [24]	$45.6_{\pm 1.0}$	$54.8_{\pm 0.7}$	$55.3_{\pm 0.8}$	$56.4_{\pm 1.4}$
		CB	OCSVM [25]	$47.3_{\pm 1.6}$	$46.7_{\pm 3.1}$	$53.8_{\pm 1.2}$	$53.6_{\pm 0.7}$
			Deep-SVDD [26]	$47.3_{\pm 0.8}$	$45.2_{\pm 0.8}$	$54.5_{\pm 1.0}$	$55.3_{\pm 0.9}$
			Panda [28]	$46.2_{\pm 0.1}$	$45.1_{\pm 1.1}$	$55.1_{\pm 0.1}$	$54.3_{\pm 0.0}$
		CL	CSI [12]	$43.2_{\pm 0.9}$	$40.9_{\pm 3.2}$	$60.6_{\pm 1.3}$	$60.2_{\pm 1.1}$
			Elsa [14]	$45.4_{\pm 1.0}$	$39.9_{\pm 4.7}$	$57.0_{\pm 1.8}$	$56.2_{\pm 2.2}$
			FADngs w/o SED	$32.9_{\pm 1.7}$	$30.0_{\pm 2.9}$	$73.1_{\pm 2.1}$	$69.7_{\pm 1.7}$
			FADngs	$\mathbf{30.4_{\pm 1.2}}$	$\mathbf{25.0_{\pm 0.6}}$	$\mathbf{77.8_{\pm 0.9}}$	$\mathbf{75.1_{\pm 0.9}}$
	SVHN	RB	AE [30]	$45.0_{\pm 1.3}$	$39.5_{\pm 1.4}$	$60.3_{\pm 1.4}$	$62.9_{\pm 1.3}$
			FedDetect [24]	$42.3_{\pm 2.1}$	$48.4_{\pm 4.6}$	$60.8_{\pm 3.2}$	$64.0_{\pm 3.3}$
		CB	OCSVM [25]	$34.9_{\pm 0.7}$	$37.5_{\pm 0.4}$	$68.2_{\pm 1.7}$	$60.7_{\pm 1.4}$
			Deep-SVDD [26]	$40.9_{\pm 0.6}$	$35.2_{\pm 0.8}$	$65.8_{\pm 0.6}$	$65.1_{\pm 0.6}$
			Panda [28]	$34.7_{\pm 1.6}$	$34.2_{\pm 4.2}$	$72.2_{\pm 1.0}$	$69.3_{\pm 0.9}$
		CL	CSI [12]	$22.9_{\pm 1.6}$	$25.7_{\pm 2.3}$	$83.4_{\pm 0.7}$	$83.3_{\pm 0.8}$
			Elsa [14]	$29.0_{\pm 1.7}$	$27.4_{\pm 2.1}$	$78.0_{\pm 0.9}$	$71.4_{\pm 1.4}$
			FADngs w/o SED	$14.9_{\pm 1.9}$	$8.5_{\pm 2.6}$	$94.7_{\pm 1.3}$	$94.0_{\pm 1.0}$
			FADngs	$\mathbf{9.4_{\pm 3.7}}$	$\mathbf{6.6_{\pm 3.0}}$	$\mathbf{96.6_{\pm 1.7}}$	$\mathbf{95.3_{\pm 1.8}}$
	Places365	RB	AE [30]	$46.2_{\pm 0.4}$	$40.8_{\pm 0.5}$	$55.2_{\pm 1.3}$	$52.4_{\pm 0.4}$
			FedDetect [24]	$47.1_{\pm 1.2}$	$53.8_{\pm 1.2}$	$55.6_{\pm 1.7}$	$53.0_{\pm 3.1}$
		CB	OCSVM [25]	$48.3_{\pm 1.5}$	$47.7_{\pm 3.3}$	$52.4_{\pm 2.1}$	$51.6_{\pm 1.7}$
			Deep-SVDD [26]	$48.1_{\pm 0.1}$	$48.1_{\pm 3.9}$	$52.4_{\pm 0.2}$	$51.2_{\pm 0.1}$
			Panda [28]	$47.7_{\pm 0.3}$	$47.5_{\pm 5.4}$	$52.8_{\pm 0.1}$	$51.4_{\pm 0.1}$
		CL	CSI [12]	$43.5_{\pm 0.8}$	$42.4_{\pm 5.2}$	$59.8_{\pm 0.6}$	$60.1_{\pm 0.9}$
			Elsa [14]	$47.0_{\pm 2.1}$	$47.5_{\pm 2.7}$	$54.1_{\pm 0.3}$	$53.7_{\pm 2.2}$
			FADngs w/o SED	$32.4_{\pm 0.7}$	$32.4_{\pm 2.8}$	$72.4_{\pm 0.4}$	$71.8_{\pm 0.8}$
			FADngs	$\mathbf{25.6_{\pm 1.2}}$	$\mathbf{20.5_{\pm 2.0}}$	$\mathbf{83.8_{\pm 0.8}}$	$\mathbf{82.2_{\pm 0.9}}$

TABLE II

EXPERIMENTS WITH *No Anomaly-Contaminated* DATASETS. OTHER SETTINGS ARE THE SAME AS TABLE I. WE USE CIFAR-10 FOR NORMAL DATASET AND CIFAR-100 FOR ANOMALY DATASET

Method	FE(%) ↓	ME(%) ↓	AUROC(%) ↑	AUPR(%) ↑
CSI [12]	$42.6_{\pm 1.4}$	$39.5_{\pm 1.6}$	$60.8_{\pm 0.9}$	$60.4_{\pm 0.8}$
Elsa [14]	$44.6_{\pm 1.3}$	$40.3_{\pm 1.3}$	$57.8_{\pm 2.0}$	$57.4_{\pm 1.9}$
FADngs w/o SED	$\mathbf{32.9_{\pm 0.7}}$	$\mathbf{29.7_{\pm 2.0}}$	$\mathbf{73.4_{\pm 0.6}}$	$\mathbf{70.1_{\pm 1.2}}$
FADngs	$\mathbf{30.0_{\pm 0.9}}$	$\mathbf{24.1_{\pm 0.7}}$	$\mathbf{78.5_{\pm 0.4}}$	$\mathbf{75.7_{\pm 0.3}}$

to other baselines), learns poor representations that the points of different colors are mixed and are hardly distinguished. Furthermore, the result in Fig. 7(b) demonstrates a lower rate of missing detection compared with Fig. 7(d), as our method learns more discriminative representations and has more knowledge about global data distributions by sharing noisy density functions.

B. Ablation Experiments

In this section, we perform ablation experiments to show the effectiveness of different modules in FADngs.

To show the effectiveness of NGDE for false and missing detection, we conduct ablation experiments and present results in Table III. The first line uses local models without sharing model and density functions. The second line uses the global model without sharing noisy density functions. The results of the first line are much lower than those of the second and the third line, which denotes that non-IID data causes trained local models cannot adapt to other clients. Furthermore, the proposed FADngs far outperforms the second line method by sharing noisy density functions, which illustrate that this module can effectively alleviate false and missing detection.

To validate the effectiveness of the proposed CRAD for learning more discriminative representation with limited data, we compare our method with some representation learning methods in Table IV, where all algorithms share noisy density functions. It illustrates that the reconstruct-based method using AE [30], the most common method in traditional anomaly detection, is ineffective in learning the representations with limited data for FL. Meanwhile, some contrastive learning methods like CSI [12] are shown to significantly help to learn discriminative features, and the proposed method outperforms the CSI [12] 6.80% AUROC and 5.30% AUPR, which shows that the specific augmented way in CRAD is significant for

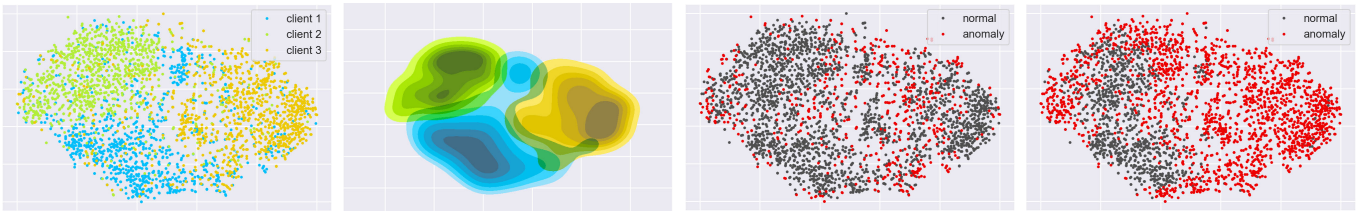

Fig. 6. T-SNE visualizations of false detection problem. (a) Representations of data samples in different clients. (b) Kernel density estimation of representations in different clients. (c) Detection results with global density functions. (d) Detection results with local density functions in client 1. Note that all the samples used in the above figures are real normal samples.

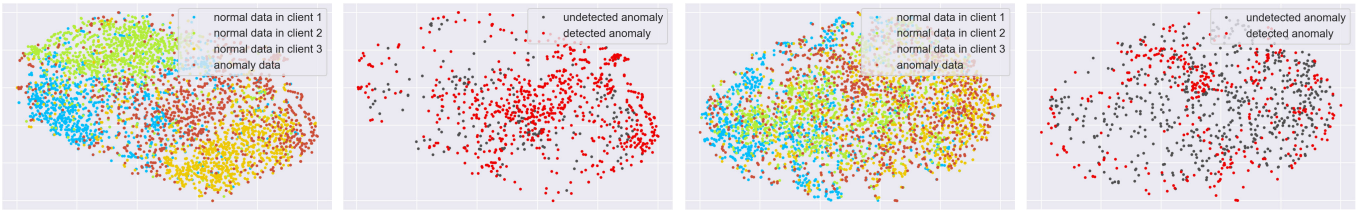

Fig. 7. T-SNE visualizations of missing detection problem. (a) Representations learned by FADngs of normal samples in different clients and anomalies. (b) Detection results for anomalies of FADngs. (c) Representations learned by baseline method (CSI [12]) of normal samples in different clients and anomalies. (d) Detection results for anomalies of baseline method (CSI [12]). Note that red points in (a) and (c) are real anomalies, while those in (b) and (d) are predicted anomalies.

TABLE III

ABLATION EXPERIMENTS ABOUT NGDE. WE COMPARE OUR WORKS WITH THE METHOD THAT ONLY USES LOCAL MODELS, AND THE METHOD ONLY USES LOCAL DENSITY FUNCTIONS. WE PERFORM THE SAME TRAINING AND DETECTION STEPS FOR ALL EXPERIMENTS. MORE SPECIFICALLY, THE METHOD IN THE FIRST LINE DOES NOT SHARE ANYTHING, WHILE THE METHOD IN THE SECOND LINE ONLY SHARES MODEL PARAMETERS TO THE SERVER. ALL THE EXPERIMENT AGGREGATING MODELS BY SIMPLY AVERAGING MODEL PARAMETERS LIKE FEDAVG [21]. WE USE CIFAR-10 FOR NORMAL DATASET AND CIFAR-100 FOR ANOMALY DATASET

Method	FE(%) ↓	ME(%) ↓	AUROC(%) ↑	AUPR(%) ↑
using local models	$42.3_{\pm 0.3}$	$37.2_{\pm 1.2}$	$61.0_{\pm 0.3}$	$57.8_{\pm 0.3}$
using local density functions	$37.8_{\pm 2.9}$	$34.4_{\pm 1.4}$	$66.4_{\pm 2.8}$	$63.6_{\pm 2.7}$
FADngs w/o SED	$\mathbf{32.9_{\pm 1.7}}$	$\mathbf{30.0_{\pm 2.9}}$	$\mathbf{73.1_{\pm 2.1}}$	$\mathbf{69.7_{\pm 1.7}}$

TABLE IV

ABLATION EXPERIMENTS ABOUT CRAD. THE RESULTS ABOUT COMPARISON ON THE EFFECTIVENESS OF REPRESENTATIONS LEARNED. ALL ALGORITHMS IN THIS TABLE ARE COMBINING WITH NGDE, WHICH SHARES NOISY DENSITY FUNCTIONS IN THE TRAINING AND DETECTION PROCESS. SUPERVISED CE DENOTES SUPERVISED METHOD USING CROSS-ENTROPY LOSS FUNCTION. CSI [12] DENOTES USING CONTRASTIVE LEARNING WITHOUT GLOBAL SELECTION, WHERE ALL THE OTHER SAMPLES ARE CONSIDERED AS NEGATIVES. OUR METHOD USES CRAD. ALL THE EXPERIMENT AGGREGATING MODELS BY SIMPLY AVERAGING MODEL PARAMETERS LIKE FEDAVG [21]. WE USE CIFAR-10 FOR NORMAL DATASET AND CIFAR-100 FOR ANOMALY DATASET

Training Method	FE(%) ↓	ME(%) ↓	AUROC(%) ↑	AUPR(%) ↑
AE [30]	$47.2_{\pm 0.3}$	$41.3_{\pm 4.2}$	$54.3_{\pm 0.4}$	$54.1_{\pm 0.6}$
Supervised CE	$43.8_{\pm 0.7}$	$38.8_{\pm 2.5}$	$59.6_{\pm 1.0}$	$56.8_{\pm 1.0}$
SimCLR [41]	$47.0_{\pm 0.3}$	$42.8_{\pm 1.2}$	$54.2_{\pm 0.3}$	$53.0_{\pm 0.9}$
CSI [12]	$38.8_{\pm 1.1}$	$35.7_{\pm 4.8}$	$66.3_{\pm 0.6}$	$64.4_{\pm 0.6}$
CRAD	$\mathbf{32.9_{\pm 1.7}}$	$\mathbf{30.0_{\pm 2.9}}$	$\mathbf{73.1_{\pm 2.1}}$	$\mathbf{69.7_{\pm 1.7}}$

anomaly detection (where generate positive and negative samples according to the predict results). Besides, the proposed method further outperforms the supervised method using cross entropy loss. It is due to that the supervised method mainly learns to distinguish samples from different classes, while the proposed method learns some common representations of normal samples from different classes, and could more clearly distinguish normal samples from anomalies.

As for SED, we show the capacities that detect anomalies with local distributions of averaged and SED model in Fig. 8. The capacities of the models are presented by how separate the distributions of normal samples and anomalies are. To better show the capacities with local distributions, the normal samples and anomalies are only from the corresponding clients. It can be seen that the vanilla averaged model cannot preserve the capacities of local models, as two distributions (box) have a larger overlap and the difference between the medians of normal samples and anomalies (red dotted line) are smaller than local models with local distributions. In contrast, the SED model has a narrower overlap, and the difference is larger than the vanilla averaged model. The performance of the SED model is closer to or even better than that of the local model with local distributions, which shows that aggregated model by SED can maintain the capacities of local models effectively.

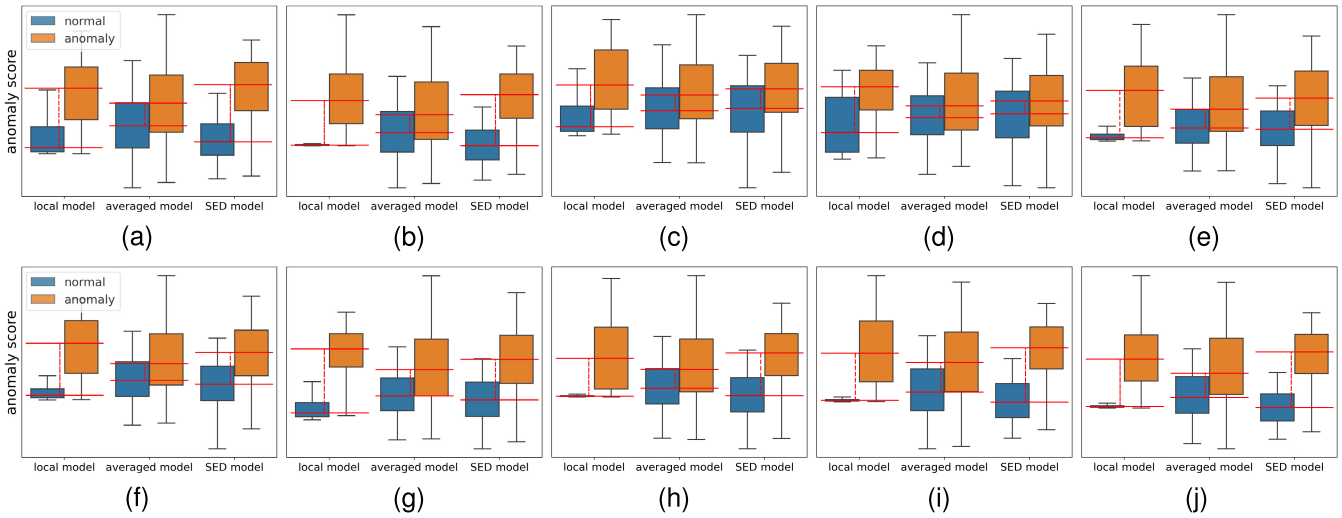

Fig. 8. Capacities of different aggregation methods to detect anomalies with various local distributions. Note that the normal samples and anomalies are only from the corresponding clients, and the boxes denote distributions of predicted scores for real normal samples and anomalies, respectively. The detection model must make the overlap of the two boxes as small as possible. The solid red line in the figure denotes the median, and the length of the red dotted line denotes the difference between the median of the normal sample and the median of the abnormal sample. The larger the difference, the easier it is for the model to distinguish normal samples and anomalies correctly. (a) Client 1. (b) Client 2. (c) Client 3. (d) Client 4. (e) Client 5. (f) Client 6. (g) Client 7. (h) Client 8. (i) Client 9. (j) Client 10.

TABLE V

ABLATION EXPERIMENTS ABOUT NUMBER OF LOCAL CLIENT TRAINING EPOCH E_l IN EACH COMMUNICATION ROUND, WHICH IS EQUIVALENT TO THE COMMUNICATION FREQUENCY

E_l	FE(%) ↓	ME(%) ↓	AUROC(%) ↑	AUPR(%) ↑
1	$\mathbf{30.4}_{\pm\mathbf{1.2}}$	$\mathbf{25.0}_{\pm\mathbf{0.6}}$	$\mathbf{77.8}_{\pm\mathbf{0.9}}$	$\mathbf{75.1}_{\pm\mathbf{0.9}}$
2	$32.2_{\pm0.8}$	$26.9_{\pm1.5}$	$74.8_{\pm0.9}$	$73.0_{\pm2.1}$
5	$32.3_{\pm0.6}$	$27.0_{\pm0.7}$	$74.5_{\pm0.7}$	$71.8_{\pm0.9}$
10	$33.1_{\pm0.3}$	$27.9_{\pm1.3}$	$73.2_{\pm0.3}$	$70.5_{\pm0.6}$

TABLE VI

ABLATION EXPERIMENTS ABOUT QUEUE SIZE IN SED

Queue Size	FE(%) ↓	ME(%) ↓	AUROC(%) ↑	AUPR(%) ↑
256	$30.8_{\pm0.4}$	$25.7_{\pm0.9}$	$77.1_{\pm0.4}$	$73.9_{\pm0.4}$
1024	$31.0_{\pm0.5}$	$25.6_{\pm0.8}$	$77.0_{\pm0.6}$	$74.3_{\pm0.5}$
4096	$30.4_{\pm0.4}$	$25.4_{\pm1.2}$	$77.7_{\pm0.3}$	$74.9_{\pm0.3}$
16384	$30.5_{\pm0.3}$	$25.3_{\pm0.3}$	$77.5_{\pm0.5}$	$74.7_{\pm0.4}$
65536	$30.4_{\pm1.2}$	$25.0_{\pm0.6}$	$77.8_{\pm0.9}$	$75.1_{\pm0.9}$

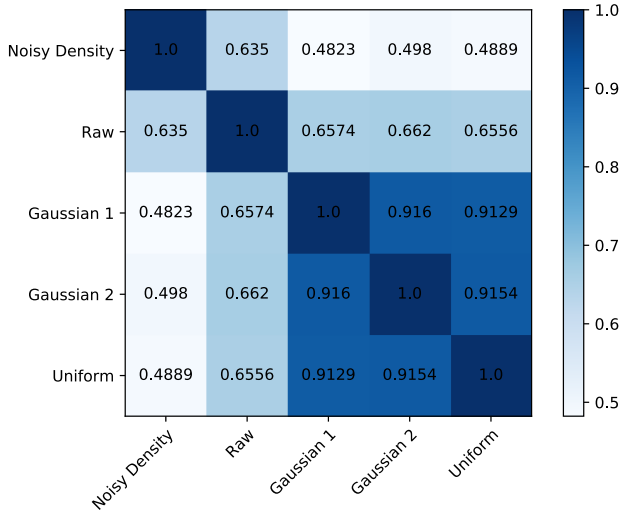

Fig. 9. Heatmap of distance correlation between different distributions. Gaussian 1 denotes a Gaussian distribution with mean 1, variance 10. Gaussian 2 denotes a Gaussian distribution with mean 0, variance 1. Uniform denotes a uniform distribution in [0, 1).

C. Hyperparameter Analysis

In this section, we perform ablation experiments across different hyperparameters in FADngs. We use CIFAR-10 for the normal dataset and CIFAR-100 for the anomaly dataset.

As for the number of local client training epoch E_l in each communication round, we present different results with different E_l in Table V. All the settings have the same number of total training epochs but have different frequencies of communication and update of the global model. It can be seen that higher frequency means better performance. Thus, we set $E_l = 1$ in our main experiments, which means that we aggregate the local models after 1 epoch.

We also evaluate the impact of hyperparameters in our model. For the size of stored instance queue D_c in SED, we set it to 256, 1024, 4096, 16 384, and 65 536 in our experiments. As illustrated in Table VI, larger queue size seems to have better performance, but their performance is very similar. We set it to 65 536 in our main experiments for the best performance.

D. Privacy Analysis

One of the key points of FL is to preserve the privacy of clients. In this section, we will show that sharing noisy density functions is privacy-preserving.

Data privacy is related to the correlation between shared information and raw data (e.g., if X and Y are highly correlated, we can infer X with Y, and vice versa. However, if the correlation between X and Y is low, it is hard to infer X with Y). Thus, we use distance correlation to quantify the

risk of privacy leakage [71]. Distance correlation is a measure of dependence between two paired variables of arbitrary, not necessarily equal, dimension [72], [73], which can be defined as

$$\mathrm{dCor}(X, Y) = \frac{\mathrm{dCov}^2(X, Y)}{\sqrt{\mathrm{dVar}(X)\,\mathrm{dVar}(Y)}} \tag{18}$$

where X, Y are the variables corresponding to the shared density functions and raw data, respectively. dVar is the distance standard deviation. dCov^2 is the distance covariance, which is

$$\mathrm{dCov}^2(X, Y) = \frac{1}{c_p c_q} \int_{\mathbb{R}^{p+q}} \frac{|\varphi_{X,Y}(s,t) - \varphi_X(s)\varphi_Y(t)|^2}{|s|_p^{1+p}|t|_q^{1+q}} dtds \tag{19}$$

where $\varphi_{X,Y}(s,t)$, $\varphi_X(s)$, $\varphi_Y(t)$ are the characteristic functions of (X, Y), X, Y, respectively, p, q denote the Euclidean dimension of X and Y, and thus of s and t, and c_p, c_q are constants. Note that $0 \leq \mathrm{dCor} \leq 1$, and $\mathrm{dCor} = 0$ if and only if the variables are independent.

As illustrated in Fig. 9, the distance correlation between the shared noisy density functions and raw distribution is 0.6350, which is similar to that between random Gaussian distributions and raw distribution 0.6574, 0.662. Thus, for inferring raw data, the shared information is similar to random Gaussian distributions, which shows that sharing noisy density functions is privacy-preserving.

V. Conclusion

This article proposes a federated anomaly detection algorithm with NGDE, CRAD, and SED, which far outperforms existing federated anomaly detection algorithms. Specifically, we propose to perform clustering with introduced noise and estimate the density function of each cluster. Then we share noisy density functions to align local distributions. Moreover, we prove that sharing noisy density functions is privacy-preserving. In addition, the proposed CRAD learns discriminative features for anomaly detection with contrastive learning based on the shared density functions. Besides, the proposed SED aggregated capacities instead of parameters which alleviates false detection effectively. Extensive experiments validate the effectiveness of the proposed methods, which significantly outperform various baselines [74].

References

[1] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," 2018, *arXiv:1806.00582*.

[2] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," 2020, *arXiv:2003.13461*.

[3] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "FedBN: Federated learning on non-IID features via local batch normalization," 2021, *arXiv:2102.07623*.

[4] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

[5] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3710–3722, Aug. 2021.

[6] J. Xu, W. Du, Y. Jin, W. He, and R. Cheng, "Ternary compression for communication-efficient federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 3, pp. 1162–1176, Mar. 2022.

[7] T.-N. Nguyen, S. Roy, and J. Meunier, "SmithNet: Strictness on motion-texture coherence for anomaly detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2287–2300, Jun. 2022.

[8] K. Zhou et al., "Memorizing structure-texture correspondence for image anomaly detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2335–2349, Jun. 2022.

[9] V. Zavrtanik, M. Kristan, and D. Skocaj, "DRÆM—A discriminatively trained reconstruction embedding for surface anomaly detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8330–8339.

[10] D. Li, Q. Tao, J. Liu, and H. Wang, "Center-aware adversarial autoencoder for anomaly detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2480–2493, Jun. 2022.

[11] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve model robustness and uncertainty," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 15663–15674.

[12] J. Tack, S. Mo, J. Jeong, and J. Shin, "CSI: Novelty detection via contrastive learning on distributionally shifted instances," in *Proc. 34th Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 11839–11852.

[13] V. Sehwag, M. Chiang, and P. Mittal, "Ssd: A unified framework for self-supervised outlier detection," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[14] S. Han, H. Song, S. Lee, S. Park, and M. Cha, "ELSA: Energy-based learning for semi-supervised anomaly detection," in *Proc. 32nd Brit. Mach. Vis. Conf.*, 2021.

[15] Y. Liu, Z. Li, S. Pan, C. Gong, C. Zhou, and G. Karypis, "Anomaly detection on attributed networks via contrastive self-supervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2378–2392, Jun. 2022.

[16] Z. Yang, T. Zhang, I. S. Bozchalooi, and E. Darve, "Memory-augmented generative adversarial networks for anomaly detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2324–2334, Jun. 2022.

[17] Y. Zhou, X. Song, Y. Zhang, F. Liu, C. Zhu, and L. Liu, "Feature encoding with autoencoders for weakly supervised anomaly detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2454–2465, Jun. 2022.

[18] D. Macêdo, T. I. Ren, C. Zanchettin, A. L. I. Oliveira, and T. Ludermir, "Entropic out-of-distribution detection: Seamless detection of unknown examples," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2350–2364, Jun. 2022.

[19] Q. Xie, P. Zhang, B. Yu, and J. Choi, "Semisupervised training of deep generative models for high-dimensional anomaly detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2444–2453, Jun. 2022.

[20] F. V. Massoli, F. Falchi, A. Kantarci, S. Akti, H. K. Ekenel, and G. Amato, "MOCCA: Multilayer one-class classification for anomaly detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2313–2323, Jun. 2022.

[21] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist. (AISTATS)* (Proceedings of Machine Learning Research), vol. 54. Fort Lauderdale, FL, USA: W&CP, 2017, pp. 1273–1282.

[22] T. Yoon, S. Shin, S. J. Hwang, and E. Yang, "FedMix: Approximation of mixup under mean augmented federated learning," in *Proc. 9th Int. Conf. Learn. Represent. (ICLR)*, 2021.

[23] Y. Tan et al., "Fedproto: Federated prototype learning across heterogeneous clients," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 8, 2022, pp. 8432–8440.

[24] M. Wen, R. Xie, K. Lu, L. Wang, and K. Zhang, "Feddetect: A novel privacy-preserving federated learning framework for energy theft detection in smart grid," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 6069–6080, Apr. 2022.

[25] M.-N. Nguyen and N. A. Vien, "Scalable and interpretable one-class svms with deep learning and random Fourier features," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Cham, Switzerland: Springer, 2018, pp. 157–172.

[26] L. Ruff et al., "Deep one-class classification," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4393–4402.

[27] L. Ruff, "Deep semi-supervised anomaly detection," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[28] T. Reiss, N. Cohen, L. Bergman, and Y. Hoshen, "PANDA: Adapting pretrained features for anomaly detection and segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2805–2813.

[29] D. Gong et al., "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1705–1714.

[30] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 665–674.

[31] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," in *Proc. Wireless Telecommun. Symp. (WTS)*, Apr. 2018, pp. 1–5.

[32] B. Zong et al., "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[33] Y. Du and I. Mordatch, "Implicit generation and modeling with energy based models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 3608–3618.

[34] J. Ren et al., "Likelihood ratios for out-of-distribution detection," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 14707–14718.

[35] W. Grathwohl, K.-C. Wang, J.-H. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky, "Your classifier is secretly an energy based model and you should treat it like one," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[36] C. Stein, "Estimation of a covariance matrix," in *Proc. 39th Annu. Meeting IMS*, Atlanta, GA, USA, 1975.

[37] I. Golan and R. El-Yaniv, "Deep anomaly detection using geometric transformations," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 9781–9791.

[38] L. Bergman and Y. Hoshen, "Classification-based anomaly detection for general data," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[39] S. Mohseni, M. Pitale, J. Yadawa, and Z. Wang, "Self-supervised learning for generalizable out-of-distribution detection," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 4, pp. 5216–5223.

[40] X. Du, Z. Wang, M. Cai, and Y. Li, "VOS: Learning what you don't know by virtual outlier synthesis," in *Proc. Int. Conf. Learn. Represent.*, 2021.

[41] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[42] A. Ramírez Rivera, A. Khan, I. E. I. Bekkouch, and T. S. Sheikh, "Anomaly detection based on zero-shot outlier synthesis and hierarchical feature distillation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 1, pp. 281–291, Jan. 2022.

[43] H. Liu, X. Xu, E. Li, S. Zhang, and X. Li, "Anomaly detection with representative neighbors," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 6, pp. 2831–2841, Oct. 2023.

[44] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[45] S. J. Reddi et al., "Adaptive federated optimization," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[46] D. Li and J. Wang, "FedMD: Heterogenous federated learning via model distillation," 2019, *arXiv:1910.03581*.

[47] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *Proc. NIPS*, vol. 33, 2020, pp. 2351–2363.

[48] H.-Y. Chen and W.-L. Chao, "FedBE: Making Bayesian model ensemble applicable to federated learning," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[49] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12878–12889.

[50] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[51] C. Bucilua, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 535–541.

[52] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-IID private data," 2018, *arXiv:1811.11479*.

[53] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5132–5143.

[54] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, vol. 2, 2020, pp. 429–450.

[55] P. Pu Liang et al., "Think locally, act globally: Federated learning with local and global representations," 2020, *arXiv:2001.01523*.

[56] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 3557–3568.

[57] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10708–10717.

[58] Y. Zhao, J. Chen, D. Wu, J. Teng, and S. Yu, "Multi-task network anomaly detection using federated learning," in *Proc. 10th Int. Symp. Inf. Commun. Technol.*, 2019, pp. 273–279.

[59] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for Internet of Things: Recent advances, taxonomy, and open challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1759–1799, 3rd Quart., 2021.

[60] T. Zhang, C. He, T. Ma, L. Gao, M. Ma, and S. Avestimehr, "Federated learning for Internet of Things," in *Proc. 19th ACM Conf. Embedded Networked Sensor Syst.*, 2021, pp. 413–419.

[61] O. Ledoit and M. Wolf, "Honey, I shrunk the sample covariance matrix," *J. Portfolio Manage.*, vol. 30, no. 4, pp. 110–119, Jul. 2004.

[62] P. C. Mahalanobis, "On the generalized distance in statistics," Nat. Inst. Sci. India, India, pp. 49–5, 1936, vol. 2.

[63] J. L. Rodgers and W. A. Nicewander, "Thirteen ways to look at the correlation coefficient," *Amer. Statistician*, vol. 42, no. 1, p. 59, Feb. 1988.

[64] D. Chen, L. Wu, S. Tang, X. Yun, B. Long, and Y. Zhuang, "Robust meta-learning with sampling noise and label noise via Eigen-reptile," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 3662–3678.

[65] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. TR-2009, 2009.

[66] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2011.

[67] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, Jun. 2018.

[68] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[69] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, Jul. 1997.

[70] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

[71] G. J. Székely and M. L. Rizzo, "Partial distance correlation with methods for dissimilarities," *Ann. Statist.*, vol. 42, no. 6, pp. 2382–2412, Dec. 2014.

[72] Z. Zhou, "Measuring nonlinear dependence in time-series, a distance correlation approach," *J. Time Ser. Anal.*, vol. 33, no. 3, pp. 438–457, May 2012.

[73] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing dependence by correlation of distances," *Ann. Statist.*, vol. 35, no. 6, pp. 2769–2794, Dec. 2007.

[74] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

Boyu Dong received the B.E. degree from Zhejiang University, Hangzhou, China, in 2020, where he is currently pursuing the master's degree with the College of Computer Science and Technology.

His research interests include machine learning and federated learning.

Dong Chen received the B.E. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2019. He is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China.

His research interests include machine learning and federated learning.

Yu Wu received the Ph.D. degree from the University of Technology Sydney, Sydney, NSW, Australia, in 2021.

He is currently a Post-Doctoral Fellow with Princeton University, Princeton, NJ, USA. His research interests include multimodal perception and video understanding.

Dr. Wu was a recipient of the Google Ph.D. Fellowship 2020.

Siliang Tang (Member, IEEE) received the Ph.D. degree from the National University of Ireland, Maynooth, Ireland, in 2012.

He is currently a Full Professor with the College of Computer Science, Zhejiang University, Hangzhou, China. So far, he has authored more than 100 papers in top-tier scientific conferences, such as Association for the Advancement of Artificial Intelligence (AAAI), International Joint Conference on Artificial Intelligence (IJCAI), Conference on Neural Information Processing Systems (NIPS), International Conference on Machine Learning (ICML), Knowledge Discovery and Data Mining (KDD), Conference on Computer Vision and Pattern Recognition (CVPR), International Conference on Computer Vision (ICCV), Association for Computational Linguistics (ACL), Special Interest Group on Multimedia (SIGMM), Special Interest Group on Information Retrieval (SIGIR), and IEEE journals, such as IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS (TVCG), IEEE TRANSACTIONS ON MULTIMEDIA (TMM), IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT), and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS). His research interests include machine learning, computer vision, and multimodal data analysis.

Dr. Tang has been serving as the Area Chair/Senior PC Member or a PC Member for conferences, such as NIPS, ICML, KDD, AAAI, IJCAI, CVPR, ACL, Conference on Empirical Methods in Natural Language Processing (EMNLP), North American Chapter of the Association for Computational Linguistics (NAACL), and SIGMM and a reviewer for journals, such as IEEE TIP, IEEE TMM, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS (TSMC), *ACM Computing Surveys*, and *Nature-Scientific Reports*.

Yueting Zhuang (Senior Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in computer science from Zhejiang University, Hangzhou, China, in 1986, 1989, and 1998, respectively.

From February 1997 to August 1998, he was a Visiting Scholar with the University of Illinois at Urbana–Champaign, Champaign, IL, USA. He has served as the Dean of the College of Computer Science, Zhejiang University, from 2009 to 2017, where he was the Director of the Institute of Artificial Intelligence from 2006 to 2015 and is currently a Full Professor with the College of Computer Science and the Director of the MOE-Digital Library Engineering Research Center. He has authored over 600 papers with 12 000 citations by Google Scholar. His research interests mainly include data mining, computer vision, and cross-media computing.

Dr. Zhuang has been performing his duties by serving as a reviewer for the prestigious IEEE journals, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), the IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), the IEEE and TRANSACTIONS ON MULTIMEDIA (TMM). He has been participating in the IEEE conferences regularly and working as session chairs.