Federated Generalized Face Presentation Attack Detection

Rui Shao, Member, IEEE, Pramuditha Perera, Member, IEEE, Pong C. Yuen, Senior Member, IEEE, and Vishal M. Patel, Senior Member, IEEE

Abstract—Face presentation attack detection (fPAD) plays a critical role in the modern face recognition pipeline. An fPAD model with good generalization can be obtained when it is trained with face images from different input distributions and different types of spoof attacks. In reality, training data (both real face images and spoof images) are not directly shared between data owners due to legal and privacy issues. In this article, with the motivation of circumventing this challenge, we propose a federated face presentation attack detection (FedPAD) framework that simultaneously takes advantage of rich fPAD information available at different data owners while preserving data privacy. In the proposed framework, each data owner (referred to as data centers) locally trains its own fPAD model. A server learns a global fPAD model by iteratively aggregating model updates from all data centers without accessing private data in each of them. Once the learned global model converges, it is used for fPAD inference. To equip the aggregated fPAD model in the server with better generalization ability to unseen attacks from users, following the basic idea of FedPAD, we further propose a federated generalized face presentation attack detection (FedGPAD) framework. A federated domain disentanglement strategy is introduced in FedGPAD, which treats each data center as one domain and decomposes the fPAD model into domain-invariant and domain-specific parts in each data center. Two parts disentangle the domain-invariant and domain-specific features from images in each local data center. A server learns a global fPAD model by only aggregating domain-invariant parts of the fPAD models from data centers, and thus, a more generalized fPAD model can be aggregated in server. We introduce the experimental setting to evaluate the proposed FedPAD and FedGPAD frameworks and carry out extensive experiments to provide various insights about federated learning for fPAD.

Index Terms—Face presentation attack detection (fPAD), federated learning (FL), generalization ability.

I. INTRODUCTION

RECENT advances in face recognition methods have prompted many real-world applications, such as automated teller machines (ATMs), mobile devices, and entrance guard systems, to deploy this technique as an authentication method. Wide usage of this technology is due to both high accuracy and convenience it provides. However, many recent works [3], [15], [20], [26]–[29], [36] have found that this technique is vulnerable to various face presentation attacks such as print attacks, video-replay attacks [2], [4], [15], [36], [44], and 3-D mask attacks [13], [14], [27], [28]. Therefore, developing face presentation attack detection (fPAD) methods that make current face recognition systems robust to face presentation attacks has become a topic of interest in the biometrics community.

In this article, we consider the deployment of an fPAD system in a real-world scenario. We identify two types of stakeholders in this scenario—data centers and users. Data centers are entities that design and collect fPAD datasets and propose fPAD solutions. Typically, data centers include research institutions and companies that carry out the research and development of fPAD. These entities have access to both real data and spoof data and therefore are able to train fPAD models. Different data centers may contain images of different identities and different types of spoof data. However, each data center has limited data availability. Real face images are obtained from a small set of identities and spoof attacks are likely to be from a few known types of attacks. Therefore, these fPAD models have poor generalization ability [26], [29] and are likely to be vulnerable to attacks unseen during training.

On the other hand, users are individuals or entities that make use of fPAD solutions. For example, when an fPAD algorithm is introduced in mobile devices, mobile device customers are identified as users of the fPAD system. Users have access only to real data collected from local devices. Due to the absence of spoof data, they cannot locally train fPAD models. Therefore, each user relies on a model developed by a data center for fPAD, as shown in Fig. 1 (top). Since data center models lack generalization ability, inferring with these models is likely to result in erroneous predictions.

It has been shown that utilizing real data from different input distributions and spoof data from different types of spoof attacks through domain generalization and meta-learning techniques can significantly improve the generalization ability of fPAD models [26], [29]. Therefore, the performance of fPAD models, shown in Fig. 1 (top), can be improved if data from all data centers can be exploited collaboratively. In reality, due to data sharing agreements and privacy policies, data centers are not allowed to share collected fPAD data
with each other. For example, when a data center collects face images from individuals using a social media platform, it is agreed not to share the collected data with third parties.

In this article, we present a framework called federated face presentation attack detection (FedPAD) based on the principles of federated learning (FL) targeting fPAD. The proposed method exploits information across all data centers while preserving data privacy. In the proposed framework, models trained at data centers are shared and aggregated, while training images are kept private in their respective data centers, thereby preserving privacy.

FL is a distributed and privacy-preserving machine learning technique. FL training paradigm defines two types of roles named server and client. Clients contain training data and the capacity to train a model. As shown in Fig. 1 (middle), each client trains its own model locally and uploads them to the server at the end of each training iteration. Server aggregates local updates and produces a global model. This global model is then shared with all clients, which will be used in their subsequent training iteration. This process is continued until the global model is converged. During the training process, data of each client are kept private. Collaborative FL training allows the global model to exploit rich local clients’ information while preserving data privacy.

In the context of FedPAD, both data centers and users can be identified as clients. However, roles of data centers and users are different from conventional clients found in FL. As shown in Fig. 1 (middle), in FL, all clients train models and carry out inference locally and clients in the testing are usually seen during the training. Therefore, most current FL works focus on exploiting the personalization issue, which aims to adapt the aggregated FL model to the seen clients in the testing. In contrast, in FedPAD, only data centers carry out local model training. Data centers share their models with the server and download the global model during training. On the other hand, users download the global model at the end of the training procedure and only carry out inference, as shown in Fig. 1 (bottom). Therefore, the downloaded FL model will inevitably encounter various unseen face presentation attacks from the users. As such, the proposed FedPAD focuses on exploring the generalization of FL model, which aims to generalize well to unseen attacks from users in the testing.

To equip the aggregated fPAD model in the server with better generalization ability to unseen attacks from users in the testing, based on the basic concept of FedPAD, a federated generalized face presentation attack detection (FedGPAD) framework is further proposed. Inspired by the current domain generalization methods which improve the generalization ability of models by learning domain-invariant features, we introduce a federated domain disentanglement strategy in FedGPAD. This strategy treats each data center as one source domain and decomposes the fPAD model in every data center into domain-invariant and domain-specific parts. Domain-invariant and domain-specific features can be disentangled from images by these two parts correspondingly. Considering the privacy issues, we carry out domain-invariant feature learning under the FL framework by conducting the communications of model parameters between domain-invariant parts of the fPAD models from the data centers and the server. Therefore, a global fPAD model with more domain-invariant information will be aggregated in the server in a privacy-preserving manner, which is able to generalize well to unseen attacks.

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

1) This article is the first to study the FL technique for the task of fPAD. Compared to the most existing FL works exploiting the personalization issue, this article studies the generalization issue and thus proposes a FedPAD framework to develop a generalized fPAD model in a data privacy-preserving way.

2) To equip the fPAD model with better generalization ability to unseen attacks from users under the FL framework, we further propose a FedGPAD framework. A federated domain disentanglement strategy is introduced in this framework, contributing to a global fPAD model with more domain-invariant information in the server.

3) An experimental setting is defined for the FL framework for the task of fPAD. Extensive experiments are carried out to show the effectiveness of the proposed framework. Various issues and insights about FL for fPAD are also discussed.

II. RELATED WORK

A. Face Presentation Attack Detection

Current fPAD methods can be categorized under single-domain and multidomain approaches. The single-domain approach focuses on extracting discriminative cues between
real and spoof samples from a single dataset, which can be further divided into appearance-based methods and temporal-based methods. Appearance-based methods focus on extracting various discriminative appearance cues for detecting face presentation attacks. Multiscale LBP [20] and color textures [3] methods are two typical texture-based methods that extract various LBP descriptors in various color spaces for the differentiation between real/spoof samples. Image distortion analysis (IDA) [36] aims to detect the surface distortions as the discriminative cue. Moreover, Yu et al. [42] proposed central difference convolution (CDC) to capture intrinsic detailed patterns via aggregating both intensity and gradient information. Then, a bilateral convolutional network [39] is proposed to capture intrinsic material-based patterns via aggregating multilevel bilateral macro and micro information. On the other hand, temporal-based methods extract different discriminative temporal cues through multiple frames between real and spoof samples. Various dynamic textures are exploited in [23], [27], and [28] to extract discriminative facial motions. Remote photoplethysmography (rPPG) signals are exploited by Liu et al. [13], [14] to capture discriminative heartbeat information from real and spoof videos. Liu et al. [15] learned a convolution neural network–recurrent neural network (CNN-RNN) model to estimate different face depths and rPPG signals between the real and spoof samples. Spatiotemporal information of fPAD is also exploited in [35] by using the spatiotemporal propagation module (STPM). Gonzalez-Soler et al. [8] explored a feature space based on Fisher vectors. Compact binarized statistical image feature (BSIF) histograms are calculated to search this feature space. In such a feature space, semantic feature subsets from known samples can be found to enhance the detection of unknown attacks. The method proposed in [31] exploits a sequence of 3-D face scans to capture the plausibility of facial behavior. Temporal curvature change is further used to measure facial behavior with a compact feature representation.

Various fPAD datasets are introduced recently that explore different characteristics and scenarios of face presentation attacks. Treating each dataset as one domain, the multidomain approach is proposed to improve the generalization ability of the fPAD model to unseen attacks. Recent work [26] casts fPAD as a domain generalization problem and proposes a multiadversarial discriminative deep domain generalization framework to search generalized differentiation cues in a shared and discriminative feature space among multiple fPAD datasets. Liu et al. [16] treated fPAD as a zero-shot problem and proposed a deep tree network to partition the spoof samples into multiple subgroups of attacks. Qin et al. [24] and Shao et al. [29] addressed fPAD based on a meta-learning framework and enabled the model to learn to generalize well through simulated train–test splits among multiple datasets. Moreover, Wang et al. [33] proposed a disentangled representation learning (DR-Net) and multidomain learning (MD-Net) to learn the disentangled representation learning for cross-domain presentation attack detection. Lately, Yu et al. [41] exploited the neural architecture search (NAS) for fPAD and proposed a cross-domain/type-aware meta-learning to form a task-aware network. These multidomain methods have access to data from multiple datasets or multiple spoof subgroups that enable them to obtain generalized models. In this article, we study the scenario in which each data center contains data from a single domain. Due to data privacy issues, we assume that they do not have access to data from other data centers and thus aim to exploit multidomain information in a privacy-preserving manner.

B. Federated Learning

FL is a decentralized machine learning approach that enables multiple local clients to collaboratively learn a global model with the help of a server while preserving data privacy of local clients. Federated averaging (FedAvg) [21], one of the fundamental frameworks for FL, learns a global model by averaging model parameters from local clients. FedProx [25] and agnostic federated learning (AFL) [22] are two variants of FedAvg, which aim to address the bias issue of the learned global model toward different clients. These two methods achieve better models by adding a proximal term to the cost functions and optimizing a centralized distribution mixed with client distributions. More recently, federated matched averaging (FedMA) [34] algorithm is proposed to improve FedAvg with a coordinate-wise averaging of weights, which exploits permutation invariance of the neurons of a network. In addition, several personalized variants of FL have been proposed to exploit the issue of personalization of FL [5], [7], which aims to adapt the FL model to each local client during testing. However, these local clients during testing are all seen during training. In a practical scenario of fPAD, many unseen face presentation attacks will be presented to users during testing. Therefore, this article attempts to address the issue of generalization of FL, with the objective of improving the generalization ability of fPAD under the FL framework.

C. Disentanglement Learning

The objective of disentanglement learning is to factorize the data into distinct informative factors of variations [18] so that complex data can be better represented by the disentangled features. In facial recognition, DR-GAN [32] disentangles a facial image into identity and pose factors, which can be used for pose-invariant face recognition. The method proposed in [37] disentangles a facial image into different factors to encode distinct facial attributes. Moreover, Zhang et al. [45] disentangled an input gait video into different representations of appearance, canonical, and pose features for gait recognition. An image synthesis method [6] utilizes U-Net and variational autoencoder (VAE) to disentangles an image into appearance and shape.

Recently, some works have also exploited the disentangle-ment learning for fPAD [17], [43]. In particular, Liu et al. [17] designed an adversarial learning framework to disentangle the spoof traces from input faces for fPAD and further synthesized realistic new spoof faces. Zhang et al. [43] proposed to disentangle the liveness features and content features from facial images, and the liveness features are further used for fPAD. There are some key differences between the above two fPAD methods and the proposed FedGPAD framework in this article. First, compared with disentangling spoof traces [17]
scenarios of face presentation attacks. The corresponding own fPAD datasets designed for different characteristics and as Algorithm 1. Suppose that $K$.

**Algorithm 1 FedPAD**

**Input:** $K$ Data Centers have $K$ fPAD datasets $D^1, D^2, \ldots, D^K$

1. **Server aggregates:**
2. initialize $W_0$
3. for each round $t = 0, 1, 2, \ldots$ do
4. for each data center $k = 1, 2, \ldots, K$ in parallel do
5. $W^k_t = \text{DataCenterUpdate}(k, W_t)$
6. end
7. $W_t = \text{Eq.}(3)$
8. **Download** $W_t$ to Data Centers
9. end
10. **Users Download** $W_t$
11. **DataCenterUpdate**$(k, W_t)$:
12. for each local epoch $1, 2, \ldots, L$ do
13. $\ell(W^k_t) = \text{Eq.}(1)$
14. $W^k_t \leftarrow W^k_t - \eta \nabla \ell(W^k_t)$
15. end
16. **Upload** $W^k_t$ to Server

or liveness features [43] within a single source domain, FedGPAD focuses on disentangling the domain-invariant and domain-specific features with the objective of improving the generalization ability to unseen face presentation attacks with multisource domain (data centers) data. Second, the above two fPAD methods carry out disentanglement learning without considering the privacy issue. In contrast, the proposed federated domain disentanglement in FedGPAD is carried out under the FL framework, which is the first to conduct the domain-invariant feature learning in a privacy-preserving way.

### III. PROPOSED METHOD

#### A. Federated Face Presentation Attack Detection

The proposed FedPAD framework is summarized in Fig. 2 and Algorithm 1. Suppose that $K$ data enters collect their own fPAD datasets designed for different characteristics and scenarios of face presentation attacks. The corresponding collected fPAD datasets are denoted as $D^1, D^2, \ldots, D^K$ with data provided with image and label pairs denoted as $x$ and $y$, respectively. $y$ are the ground truth with binary class labels ($y = 0/1$ are labels of spoof/real samples). Based on the collected fPAD data, each data center can train its own fPAD model by iteratively minimizing the cross-entropy loss as follows:

$$\ell(W^k_t) = \sum_{(x, y) \sim D^k} y \log \mathcal{F}^k(x) + (1 - y) \log (1 - \mathcal{F}^k(x))$$

(1)

where the fPAD model $\mathcal{F}^k$ of the $k$th data center is parameterized by $\mathcal{W}^k$ $(k = 1, 2, 3, \ldots, K)$. After optimization with several local epochs via

$$\mathcal{W}^k \leftarrow \mathcal{W}^k - \eta \nabla \ell(W^k_t)$$

each data center can obtain the trained fPAD model with the updated model parameters.

It should be noted that the dataset corresponding to each data center is from a specific input distribution and it only contains a finite set of known types of spoof attack data. When a model is trained using this data, it focuses on addressing the characteristics and scenarios of face presentation attacks prevalent in the corresponding dataset. However, a model trained from a specific data center will not generalize well to unseen face presentation attacks. It is a well-known fact that diverse fPAD training data contribute to a better generalized fPAD model. A straightforward solution is to collect and combine all the data from $K$ data centers denoted as $D = \{D^1 \cup D^2 \cup \ldots \cup D^K\}$ to train an fPAD model. It has been shown that domain generalization and meta-learning-based fPAD methods can further improve the generalization ability with the above combined multidata $D$ [26], [29]. However, when sharing data between different data centers are prohibited due to the privacy issue, this naive solution is not practical.

To circumvent this limitation and enable various data centers to collaboratively train an fPAD model, we propose the FedPAD framework. Instead of accessing private fPAD data of each data center, the proposed FedPAD framework introduces a server to exploit the fPAD information of all data centers by aggregating the above model updates ($\mathcal{W}^1, \mathcal{W}^2, \ldots, \mathcal{W}^K$) of all data centers. Inspired by the FedAvg [21] algorithm, in the proposed framework, the server carries out the aggregation of model updates via calculating the average of updated parameters ($\mathcal{W}^1, \mathcal{W}^2, \ldots, \mathcal{W}^K$) in all data centers as follows:

$$\mathcal{W} = \frac{1}{K} \sum_{k=1}^{K} \mathcal{W}^k.$$ (2)

After the aggregation, the server produces a global fPAD model parameterized by $\mathcal{W}$ that exploits the fPAD information of various data centers without accessing the private fPAD data. We can further extend the above aggregation process into $t$ rounds. The server distributes the aggregated model $\mathcal{W}$ to every data center as the initial model parameters for the next-round updating of local parameters. Thus, data centers can obtain the $t$th round updated parameters denoted as ($\mathcal{W}_{t1}, \mathcal{W}_{t2}, \ldots, \mathcal{W}_{tK}$). The $t$th aggregation in the server can be
To overcome this limitation of the FedPAD framework, we further propose an improved version called FedGPAD framework, with the objective of exploiting an fPAD model with better generalization ability under the FL framework. Fig. 3 and Algorithm 2 give an overview of FedGPAD. Many existing domain generalization methods assume that by learning domain-invariant features among multiple source domains, the model is more able to generalize well to the unseen target domain during testing [26]. Inspired by this idea, we treat every data center as a source domain and aim to learn an fPAD model focusing on extracting the domain-invariant features in the server. However, due to the privacy-preserving issues required by FL, current domain-invariant feature learning-based domain generalization methods cannot be adapted as they require to get access to all source domain data from all local data centers simultaneously. To address this issue, we propose a novel federated domain disentanglement strategy in the FedGPAD framework, aiming to carry out domain-invariant feature learning in a privacy-preserving way. Federated domain disentanglement is composed of two iterative steps, i.e., local domain disentanglement learning and domain-invariant model parameters communications. Specifically, local domain disentanglement learning decomposes the fPAD model in each data center into domain-invariant part and domain-specific part, which disentangles the domain-invariant and domain-specific features from images, respectively. At this stage, the domain-invariant information extracted by the domain-invariant part of the fPAD model in each data center is still restricted by the corresponding local data distribution. To further learn domain-invariant information among all the data centers, every data center communicates with the server by uploading its domain-invariant part of the fPAD model to the server for model aggregation. In this way, the domain-invariant information from all the data centers will be summarized by the aggregated model parameters in the server, and thus, domain-invariant features among all the data centers can be learned by the aggregated model. In the next communication rounds, the aggregated domain-invariant model in the server will be downloaded to all the data centers, which will continually participate and optimize the local domain disentanglement learning in the local data centers. Therefore, the current domain-invariant feature learning is substituted by the iterative steps carried out in the federated domain disentanglement in a privacy-preserving manner.

As shown in Fig. 3, we first carry out the first step: local domain disentanglement learning. In data center $k$, we try to disentangle domain-invariant features and domain-specific features by feeding the input data into domain-invariant feature extractor and domain-specific feature extractor, respectively.
data centers and server, the collaboratively trained global fPAD model parameterized by $\mathcal{W}^{k}_{L}$ is purified with domain-invariant information. Therefore, we can carry out the second step—domain-invariant feature learning in local data center disentanglement learning as follows:

$$\mathcal{L}_{\text{Diff}}(\mathcal{W}^{k(EI)}, \mathcal{W}^{k(ES)}) = \sum_{x \sim \mathcal{D}} \left\| (Z^k_I)^T (Z^k_S) \right\|_F^2$$  \hspace{1cm} (7)

where $\| \cdot \|_F$ is the squared Frobenius norm. The difference loss encourages orthogonality between the domain-invariant and domain-specific features, and thus, they can be better disentangled.

The overall loss function of the local domain disentanglement learning in local data center $k$ is given as follows:

$$\mathcal{L}(\mathcal{W}^{k(EI)}, \mathcal{W}^{k(ES)}, \mathcal{W}^{k(Dec)}, \mathcal{W}^{k(C)}) = \mathcal{L}_{\text{Cls}}(\mathcal{W}^{k(EI)}, \mathcal{W}^{k(Dec)}) + \mathcal{L}_{\text{Dep}}(\mathcal{W}^{k(EI)}, \mathcal{W}^{k(Dec)}) + \mathcal{L}_{\text{Rec}}(\mathcal{W}^{k(EI)}, \mathcal{W}^{k(Dec)})$$

Note that the domain-invariant part (i.e. domain-invariant feature extractors, classifiers, and depth estimators) of fPAD models is purified with domain-invariant information. Therefore, we can carry out the second step—domain-invariant model parameters communication, that is, we communicate the parameters of the domain-invariant part of fPAD model in every local data center with the server. Specifically, at the communication round $t$, we update model parameters of the domain-invariant part of the fPAD models after the local domain disentanglement learning as follows:

$$\mathcal{W}^{k(EI)}_t \leftarrow \mathcal{W}^{k(EI)}_t - \eta \nabla \left( \mathcal{L}(\mathcal{W}^{k(EI)}_t, \mathcal{W}^{k(Dec)}_t), \mathcal{W}^{k(C)}_t \right)$$

$$\mathcal{W}^{k(Dec)}_t \leftarrow \mathcal{W}^{k(Dec)}_t - \eta \nabla \left( \mathcal{L}_{\text{Rec}}(\mathcal{W}^{k(EI)}_t, \mathcal{W}^{k(Dec)}_t) \right)$$

$$\mathcal{W}^{k(C)}_t \leftarrow \mathcal{W}^{k(C)}_t - \eta \nabla \left( \mathcal{L}_{\text{Cls}}(\mathcal{W}^{k(EI)}_t, \mathcal{W}^{k(C)}_t) \right)$$

Moreover, since domain-invariant and domain-specific encoders should encode different aspects of the input data, to better disentangle the two types of features, we further impose a soft subspace orthogonality constraint between the domain-invariant and domain-specific features via a difference loss as follows:

$$\mathcal{L}_{\text{Diff}}(\mathcal{W}^{k(EI)}, \mathcal{W}^{k(ES)}) = \sum_{x \sim \mathcal{D}} \left\| (Z^k_I)^T (Z^k_S) \right\|_F^2$$  \hspace{1cm} (7)

where $\| \cdot \|_F$ is the squared Frobenius norm. The difference loss encourages orthogonality between the domain-invariant and domain-specific features, and thus, they can be better disentangled.

The overall loss function of the local domain disentanglement learning in local data center $k$ is given as follows:

$$\mathcal{L}(\mathcal{W}^{k(EI)}, \mathcal{W}^{k(ES)}, \mathcal{W}^{k(Dec)}, \mathcal{W}^{k(C)}) = \mathcal{L}_{\text{Cls}}(\mathcal{W}^{k(EI)}, \mathcal{W}^{k(Dec)}) + \mathcal{L}_{\text{Dep}}(\mathcal{W}^{k(EI)}, \mathcal{W}^{k(Dec)}) + \mathcal{L}_{\text{Rec}}(\mathcal{W}^{k(EI)}, \mathcal{W}^{k(Dec)})$$

Note that the domain-invariant part (i.e. domain-invariant feature extractors, classifiers, and depth estimators) of fPAD models is purified with domain-invariant information. Therefore, we can carry out the second step—domain-invariant model parameters communication, that is, we communicate the parameters of the domain-invariant part of fPAD model in every local data center with the server. Specifically, at the communication round $t$, we update model parameters of the domain-invariant part of the fPAD models after the local domain disentanglement learning as follows:

$$\mathcal{W}^{k(EI)}_t \leftarrow \mathcal{W}^{k(EI)}_t - \eta \nabla \left( \mathcal{L}(\mathcal{W}^{k(EI)}_t, \mathcal{W}^{k(Dec)}_t), \mathcal{W}^{k(C)}_t \right)$$

$$\mathcal{W}^{k(Dec)}_t \leftarrow \mathcal{W}^{k(Dec)}_t - \eta \nabla \left( \mathcal{L}_{\text{Rec}}(\mathcal{W}^{k(EI)}_t, \mathcal{W}^{k(Dec)}_t) \right)$$

$$\mathcal{W}^{k(C)}_t \leftarrow \mathcal{W}^{k(C)}_t - \eta \nabla \left( \mathcal{L}_{\text{Cls}}(\mathcal{W}^{k(EI)}_t, \mathcal{W}^{k(C)}_t) \right)$$
We upload the above updated model parameters of the domain-invariant part to the server for model parameter aggregation by calculating the average of the updated parameters in all data centers as follows:

\[
W^{(EI)}_t = \frac{1}{K} \sum_{k=1}^{K} W^{(EI)}_{t,k},
\]

\[
W^{(C)}_t = \frac{1}{K} \sum_{k=1}^{K} W^{(C)}_{t,k},
\]

\[
W^{(Dep)}_t = \frac{1}{K} \sum_{k=1}^{K} W^{(Dep)}_{t,k}.
\]

The server downloads these aggregated domain-invariant parts of the fPAD models \(W^{(EI)}_t, W^{(C)}_t,\) and \(W^{(Dep)}_t\) to every data center as the initial model parameters for the next round of local model optimization. The downloaded domain-invariant parts of the fPAD models contain the updated domain-invariant information, which can be reversely participate and optimize the local domain disentanglement learning carried out in each local data center.

After \(t\) rounds of communications between data centers and the server, the trained global fPAD model parameterized by \(W^{(EI)}_t, W^{(C)}_t,\) and \(W^{(Dep)}_t\) can be obtained without compromising the private data of the individual data centers. Once training is converged, users will download the trained domain-invariant feature extractor and classifier \(W^{(EI)}_t\) and \(W^{(C)}_t\) from the server to their devices to carry out fPAD locally.

IV. EXPERIMENTS

This article focuses on exploiting FL to deal with 2-D face presentation attacks. To evaluate the performance of the proposed FL frameworks, we carry out experiments using five 2-D fPAD datasets. In this section, we first describe the datasets, testing protocols evaluation metrics and implementation details. Then, we report various experimental results based on multiple fPAD datasets. Discussions and analysis about the results are provided along with various insights about FL for fPAD.

A. Experimental Settings

1) Datasets: We evaluate our method using the following five publicly available 2-D fPAD datasets that contain print, video-replay attacks:

- Oulu-NPU [2] (O for short);
- CASIA-MFSD [44] (C for short);
- Idiap Replay-Attack [4] (I for short);
- MSU-MFSD [36] (M for short);
- SiW [15] (S for short);

The OULU-NPU dataset [2] is composed of 4950 real access and attack videos, which are displayed with Dell UltraSharp 1905FP display with 1280 \(\times\) 1024 resolution and 2015 Macbook 13-in laptop with Retina display of 2560 \(\times\) 1600 resolution. We extract total 42,248 frames from videos in the OULU-NPU dataset for experiments. The CASIA-MFSD dataset [44] consists of 600 videos with various imaging qualities: low, normal, and high. We collect in total 54,126 frames from videos in the CASIA-MFSD dataset for evaluation. The Idiap Replay-Attack dataset [4] contains 1200 videos displayed on the screen of an iPhone 3GS and on the screen of an iPad with a resolution of 1024 \(\times\) 168; 59,000 frames from videos in the Idiap Replay-Attack dataset are extracted for experiments. The MSU-MFSD dataset [36] consists of 440 videos that are replayed on an iPad Air screen and iPhone 5S screen with a resolution of 1920 \(\times\) 1080. Totally, 38,891 frames from videos in the MSU-MFSD dataset are captured for evaluation. The SiW dataset [15] provides 4620 videos displayed with two print and four replay devices (summarized in Table I). We extract in total 89,540 frames from videos in the SiW dataset for experiments.

Table I shows the variations in these datasets. Sample images from these datasets are shown in Fig. 4. From Table I and Fig. 4, it can be seen that different fPAD datasets exploit different characteristics and scenarios of face presentation attacks (i.e., different attack types, display materials and resolution, illumination, and background). Therefore, significant domain shifts exist among these datasets.

2) Protocol: The testing protocol used in this article is designed to evaluate the generalization ability of fPAD models under the FL framework. Therefore, the performance of a trained model is evaluated against a dataset that has not been observed during training. In particular, we choose one dataset at a time to emulate the role of users and consider all other datasets as data centers. Real images and spoof images of data centers are used to train an fPAD model. The trained model is tested by considering the dataset that emulates the role of users. We evaluate the performance of the model by considering how well the model is able to differentiate between real and spoof images belonging to each user.

3) Evaluation Metrics: Half total error rate (HTER) [1] (half of the summation of false acceptance rate and false rejection rate), equal error rate (EER), and area under curve (AUC) are used as evaluation metrics in our experiments, which are three most widely used metrics for the cross-dataset/cross-domain evaluations. Following [16], in the absence of a development set, thresholds required for calculating evaluation metrics are determined based on the data in all data centers.
Fig. 4. Sample images corresponding to real and attacked faces from CASIA-MFSD [44], Idiap Replay-Attack [4], MSU-MFSD [36], Oulu-NPU [2], and SiW [15] datasets.

| Layer | Chan./Stri. | Out.Size |
|-------|-------------|----------|
| Input |             |          |
| Conv-1 | 64/1         | 256      |
| Conv-2 | 128/1        | 256      |
| Conv-3 | 196/1        | 256      |
| Conv-4 | 128/1        | 256      |
| Pool-1 | 32/2         | 128      |
| Conv-5 | 128/1        | 128      |
| Conv-6 | 196/1        | 128      |
| Conv-7 | 128/1        | 128      |
| Pool-2 | 32/2         | 64       |
| Conv-8 | 128/1        | 64       |
| Conv-9 | 196/1        | 64       |
| Conv-10 | 128/1       | 64       |
| Pool-3 | 32/2         | 32       |
| Conv-11 | 128/1      | 32       |
| Pool-4 | 32/2         | 16       |
| Conv-12 | 256/1       | 16       |
| Pool-5 | 32/2         | 8        |
| Conv-13 | 512/1       | 8        |

Decoder

| Layer | Chan./Stri. | Out.Size |
|-------|-------------|----------|
| Input |             |          |
| Conv-1 | 1/1         | 512x4x4  |
| Deconv-1 | 512/2   | 8        |
| Deconv-2 | 256/2      | 16       |
| Deconv-3 | 128/2    | 32       |
| Deconv-4 | 64/2      | 64       |
| Deconv-5 | 32/2     | 128      |
| Deconv-6 | 64/2     | 256      |

Classifier

| Layer | Chan./Stri. | Out.Size |
|-------|-------------|----------|
| Input |             |          |
| Conv-1 | 1/1         | 512      |
| Conv-2 | 1/1         | 1        |

Depth Estimator

| Layer | Chan./Stri. | Out.Size |
|-------|-------------|----------|
| Input |             |          |
| Pool-1 | 1/1        |          |
| Pool-2 | 32/2       |          |
| Pool-3 | 32/2       |          |
| Conv-1 | 128/1      | 32       |
| Conv-2 | 64/1       | 32       |
| Conv-3 | 1/1        | 32       |

4) Implementation Details: Our deep network is implemented on the platform of PyTorch. For the network in the FedPAD framework, we directly adopt Resnet-18 [9] as the structure of fPAD models. For the network in the FedGPAD framework, the structure details of domain-invariant/specific feature extractor, decoder, classifier, and depth estimator are shown in Table II. Each convolutional layer in the feature extractor, classifier, and depth estimator is followed by a batch normalization layer and a rectified linear unit (ReLU) activation function, with a kernel size of 3 × 3. Each deconvolutional layer in the decoder is followed by a batch normalization layer and a LeakyReLU activation function, with kernel size 4 × 4.

The size of input image is 256 × 256 × 6, where we extract the RGB and HSV channels of each input image. Adam optimizer [10] is used for the optimization. The learning rate is set as 1e−2. The batch size is 64 per data center for FedPAD and 16 for FedGPAD.

B. Experimental Results

In this section, we demonstrate the practicality and generalization ability of the proposed framework in a real-world scenario. We first compare the performance of the proposed framework with models trained on data from a single data center. As mentioned above, due to the restriction of data privacy in practice, data cannot be shared among different data centers. In this case, users will directly obtain a trained model from one of the data centers. We report the performance of this baseline in Table III under the label Single. For different choices of user datasets (from O, C, I, M), we report the performance when the model is trained from the remaining datasets independently.

Rather than obtaining a trained model from a single data center, it is possible for users to obtain multiple trained models from several data centers and fuse their prediction scores during inference, which is also privacy preserving. In this case, we fuse the prediction scores of the trained model from various data centers by calculating the average. The results of this baseline are shown in Table III denoted as fused. Note that it is impractical for users to carry out feature-level fusion because a classifier is not allowed to be trained based on the fused features through access to any real/spoof data during inference. According to Table III, fusing scores obtained from different data centers improves the fPAD performance on average. However, this would require higher inference time and computation complexity (of order 3 for the case considered in this experiment).

On the other hand, Table III shows that the average values of all evaluation metrics of the proposed FedPAD framework outperform both baselines. This demonstrates that the proposed FedPAD method is more effective in exploiting fPAD information from multiple data centers. This is because the proposed framework actively combines fPAD information across data centers during training as opposed to the fused baseline. As a result, it is able to generalize better to unseen/novel spoof attacks. Moreover, we further tabulate the performance of the improved FedGPAD framework in Table III and further plot
Table III

Comparison with Models Trained by Data From Single Data Center and Various Data Centers

| Methods | Data Centers | User | HTER (%) | EER (%) | AUC (%) | Avg. HTER | Avg. EER | Avg. AUC |
|---------|--------------|------|----------|---------|---------|-----------|----------|---------|
| Single  | O            | M    | 41.29    | 37.42   | 67.93   | 36.43     | 34.31    | 70.36   |
|         | C            | M    | 27.09    | 24.69   | 82.91   |           |          |         |
|         | I            | M    | 49.05    | 20.04   | 85.89   |           |          |         |
|         | O            | C    | 31.33    | 34.73   | 73.19   |           |          |         |
|         | M            | C    | 39.80    | 40.67   | 66.58   |           |          |         |
|         | I            | C    | 49.25    | 47.11   | 55.41   |           |          |         |
|         | O            | I    | 42.21    | 43.05   | 54.16   |           |          |         |
|         | M            | I    | 48.50    | 33.70   | 66.29   |           |          |         |
|         | O            | O    | 29.80    | 24.12   | 84.86   |           |          |         |
|         | M            | O    | 33.97    | 21.24   | 84.33   |           |          |         |
|         | I            | O    | 46.95    | 35.16   | 71.58   |           |          |         |
| Fused   | O&C&I        | M    | 34.42    | 23.26   | 81.67   | 35.75     | 31.29    | 73.89   |
|         | O&M&I        | C    | 38.32    | 38.31   | 67.93   |           |          |         |
|         | O&C&M        | I    | 42.21    | 41.36   | 59.72   |           |          |         |
|         | I&C&M        | O    | 28.04    | 22.24   | 86.24   |           |          |         |
| FedPAD  | O&C&I        | M    | 19.45    | 17.43   | 90.24   | 32.17     | 28.84    | 76.51   |
|         | O&M&I        | C    | 42.27    | 36.95   | 70.49   |           |          |         |
|         | O&C&M        | I    | 32.53    | 26.54   | 73.58   |           |          |         |
|         | I&C&M        | O    | 34.44    | 34.45   | 71.74   |           |          |         |
| FedGPAD | O&C&I        | M    | 12.73    | 13.36   | 91.25   | 18.59     | 17.48    | 89.25   |
|         | O&M&I        | C    | 28.69    | 27.55   | 80.58   |           |          |         |
|         | O&C&M        | I    | 10.97    | 11.11   | 94.34   |           |          |         |
|         | I&C&M        | O    | 21.95    | 17.91   | 89.85   |           |          |         |
| All     | O&C&I        | M    | 21.80    | 17.18   | 90.96   | 27.26     | 25.09    | 80.42   |
|         | O&M&I        | C    | 29.46    | 31.54   | 76.29   |           |          |         |
|         | O&C&M        | I    | 30.57    | 25.71   | 72.21   |           |          |         |
|         | I&C&M        | O    | 27.22    | 25.91   | 82.21   |           |          |         |

Fig. 5. ROC curves of models trained by data from single data center and various data centers.

Their corresponding receiver operating characteristic (ROC) curves in four settings corresponding to fPAD datasets in Fig. 5. The results in Table III and Fig. 5 show that the proposed FedGPAD outperforms the other baselines and significantly improves the performance compared to FedPAD. This demonstrates that the strategy of federated domain disentanglement is extremely effective for the improvement of generalization ability of fPAD under the FL framework, and thus, a more optimal fPAD model can be obtained based on FedGPAD.

Moreover, we further consider the case where a model is trained with data from all available data centers, which is denoted as All in Table III. Note that this baseline violates the assumption of preserving data privacy and, therefore, is not a valid comparison for FedPAD. Nevertheless, it indicates the upper bound of performance for the proposed FedPAD framework. From Table III, it can be seen that the proposed FedPAD framework is only 3.9% worse than the upper bound in terms of AUC. Specifically, the proposed FedGPAD can even outperform this baseline All by about 9% improvement in HTER and AUC. This further shows that the proposed framework FedGPAD is able to obtain a generalized fPAD model in a privacy-preserving manner without sacrificing the fPAD performance.

C. Comparison With State-of-the-Art fPAD Methods

Although all the current state-of-the-art fPAD methods exploiting the generalization ability of fPAD do not consider the privacy issue, the comparison with them can still be a useful indicator for the proposed method. Therefore, we further compare the performance of our method with the following state-of-the-art fPAD methods: multiscale LBP (MS_LBP) [20]; binary CNN [38]; IDA [36]; Color Texture [3]; LBPTOP [23]; Auxiliary(Depth Only) [15]; MMD-AAE [11]; MADDG [26]; DR-MD-Net [33]; RFMeta [29]; and NAS-Baseline, NAS-Baseline w/D-Meta, NAS-FAS, NAS-FAS w/D-Meta [41], and DC-CDN [40].
We tabulate the results of comparison in Table IV. From Table IV, it can be seen that even without getting access to all the source domain data from data centers, the proposed FedGPAD can still reach very close average performance in terms of the metrics of HTER and AUC to the state-of-the-art method NAS-FAS w/D-Meta [41], with only about 4% degradation in HTER and AUC. Specifically, Table IV shows that the proposed FedGPAD can achieve the best performance in some scenarios, such as the best HTER results in O&C&M to I and the second-best HTER performance in O&C&I to M. Note that the method NAS-FAS w/D-Meta [41] not only requires data from all the source domains without privacy preserving but also leverages very complicated techniques such as NAS, and cross-domain/type meta-learning, which are all extremely time consuming and difficult to train. In contrast, the proposed FedGPAD framework can be easily implemented and trained. These results further demonstrate that the proposed FedGPAD framework can achieve comparable performance to the state-of-the-art fPAD methods in a privacy-preserving way.

D. Comparison With State-of-the-Art FL Methods

This section further compares the proposed method with some state-of-the-art FL methods: FedAvg [21] and FedMA [34]. We tabulate the results of comparison in Table V. From experimental results tabulated in Table V, it can be seen that FedMA improves the performance compared with vanilla FL method FedAvg for the task of fPAD as it adopts more advanced network parameters averaging mechanism. Furthermore, the proposed FedGPAD outperforms current FL methods such as [21] and [34]. This is because compared with existing FL methods, the federated domain disentanglement strategy proposed in this article is more able to disentangle the domain-invariant information in the server under the FL framework. This results in a more generalized global fPAD model to deal with various unseen face presentation attacks.

E. Ablation Study

We first carry out an ablation study corresponding to the loss in 8, which is composed of four loss components ($L_{\text{Diff}}$, $L_{\text{Rec}}$, $L_{\text{Dep}}$, and $L_{\text{Cls}}$). $L_{\text{Cls}}$ is necessary for classification, and thus, the ablation study is conducted in terms of the other three losses and the results in the setting of O&M&I to C are tabulated in Table VI. From Table VI, it can be seen that removing any loss component in the optimization will significantly degrade the performance of the whole framework, which demonstrates that all the added loss components complement each other and all of them contribute to the better overall performance together.
Since federated domain disentanglement is the key strategy proposed in the FedGPAD framework, one more ablation study corresponding to this key strategy is conducted in all four settings and the results are shown in Table VII. Table VII shows that the average performance of all the three metrics is significantly improved by the strategy of federated domain disentanglement. This further demonstrates the effectiveness of the federated domain disentanglement in the FedGPAD framework.

F. Comparison of Different Number of Data Centers

In this section, we investigate the importance of having more data centers during training for both FedPAD and FedGPAD frameworks. Different data centers exploit different characteristics and scenarios of face presentation attacks. Therefore, we expect that aggregating fPAD models from more data centers will enable the aggregated model to summarize more knowledge from distinct characteristics and scenarios of face presentation attacks and thus learn domain-invariant features with better generalization to unseen attacks. To verify this point, we increase the number of data centers in the proposed FedPAD and FedGPAD frameworks and report the results in Tables VIII and IX and Fig. 6. The experiments are carried out using five datasets (O, M, I, C, and S). In Fig. 6 and Table VIII, we select dataset C as the data presented to the user and the remaining datasets as the data centers for training the fPAD model with our frameworks. We increase the number of data centers from 2 to 4. As shown in Fig. 6 and Table IX, another experiment is carried out with a different combination of the same five datasets by altering the user as S. From the curve in Fig. 6 and Tables VIII and IX, it can be seen that evaluation results improve when the number of data centers increases in both FedPAD and FedGPAD frameworks. This demonstrates that increasing the number of data centers in the proposed FedPAD framework can improve the performance. Moreover, FedGPAD can achieve better performance when the number of data centers goes from 2 to 4, which can further demonstrate the superiority of the proposed FedGPAD framework.

G. Generalization Ability to Various 2-D Spoof Attacks

In reality, due to limited resources, one data center may only have limited types of 2-D attacks. However, various 2-D attacks may appear to the users. Assume that one data center collects one particular type of 2-D attack such as print attack or video-replay attack. As shown in Table X, first, we select real faces and print attacks from dataset I and real faces and video-replay attacks from dataset O to train an fPAD model and evaluate them on dataset M (containing both print attacks and video-replay attacks). In both considered cases, as shown in Table X, the corresponding trained models cannot generalize well to dataset M, which contains the additional types of 2-D attacks compared to dataset I and O. This tendency can be alleviated when the prediction scores of two independently trained models on both types of attacks are fused, as shown in Table X. Comparatively, the FedPAD method obtains a performance gain of 4.71% in HTER and 3.3% in AUC compared to score fusion. Furthermore, FedGPAD improves the EER and HTER performances and achieves comparable AUC results compared to FedPAD. This experiment demonstrates that carrying out FedPAD and FedGPAD frameworks among data centers with different types of 2-D spoof attacks can improve the generalization ability of the trained fPAD model to various 2-D spoof attacks.

H. Visualization of Latent Features

To better understand the proposed algorithm, in Figs. 7 and 8, we visualize the features extracted from the fPAD...
TABLE X

| Methods  | Data Centers | User       | HTER (%) | ERR (%) | AUC (%) |
|----------|--------------|------------|----------|---------|---------|
| Single   | I (Print)    | M (Print, Video) | 38.82    | 33.63   | 72.46   |
|          | O (Video)    | M (Print, Video) | 35.75    | 28.55   | 78.86   |
| Fused    | I (Print) & O (video) | M (Print, Video) | 35.22    | 25.56   | 81.54   |
| FedPAD   | I (Print) & O (video) | M (Print, Video) | 30.51    | 26.10   | 84.82   |
| FedGPAD  | I (Print) & O (video) | M (Print, Video) | **28.19** | **23.01** | **84.75** |

Fig. 7. tSNE visualization of features in the setting of O&C&M to I, where data centers are O, C, and M, and data from dataset I is presented to the user. For baseline All and the proposed FedPAD framework, src1_real/fake, src2_real/fake, and src3_real/fake mean the features extracted by the fPAD model in the server when the training data of real and spoof faces are from data centers O, C, and M, respectively, while tgt_real/fake denotes the features extracted by fPAD model in the server when the testing data of real and spoof faces are from dataset I. For the proposed FedGPAD framework, the features are extracted by the domain-invariant feature extractor in the server. (a) All. (b) FedPAD. (c) FedGPAD.

Fig. 8. tSNE visualization of domain-invariant features in the setting of O&C&I to M. (a) All. (b) FedPAD. (c) FedGPAD.

models in baseline All and FedPAD and the domain-invariant feature extractor in the FedGPAD framework, respectively, using t-distributed stochastic neighbor embedding (tSNE) visualization [19]. From Figs. 7 and 8, it can be seen that in baseline All and the proposed FedPAD framework based on the vanilla FL, features between real and spoofing faces are not well separated, and feature distributions between data centers and users are not aligned. In contrast, in the proposed FedGPAD framework, the generated domain-invariant features can be well separated between real and spoof faces from both data centers and users and a better class boundary between real and spoof faces can be achieved by FedGPAD for data from both data centers and the user. Moreover, we can clearly observe that the feature distributions between data centers and users can be well aligned. This means that the domain-invariant feature extractor in the proposed FedGPAD framework can truly extract invariant features between data centers during training and users during testing. This means that these features can generalize well to the unseen attacks presented to the users.

V. CONCLUSION

In this article, we presented FedPAD, an FL-based framework targeting application of fPAD with the objective of obtaining generalized fPAD models while preserving data privacy. Through communications between data centers and the server, a global fPAD model is obtained by iteratively aggregating the model updates from various data centers. Local private data are not accessed during this process. To equip the fPAD model with better generalization ability to unseen attacks from users, we further propose a FedGPAD framework. A federated domain disentanglement strategy is exploited in this framework, which disentangles the domain-invariant part of fPAD models among the local data centers and uses them to communicate with server for model aggregation. As such, a global fPAD model with better domain-invariant information will be aggregated in the server, which is more able to generalize well the unseen attacks in the users. Extensive experiments are carried out to demonstrate the effectiveness of the proposed framework, which provides various insights regarding FL for fPAD.
Pramuditha Perera (Member, IEEE) received the bachelor’s degree in electrical and electronic engineering from the University of Peradeniya, Peradeniya, Sri Lanka, in 2014, the master’s degree in electrical and computer engineering from Rutgers University, New Brunswick, NJ, USA, in 2018, and the Ph.D. degree in electrical and computer engineering from Johns Hopkins University, Baltimore, MD, USA, in 2020. He is currently an Applied Scientist with the AWS AI Labs, New York, NY, USA. His research interests include computer vision and machine learning with applications in biometrics. Dr. Perera’s work received the Best Student Paper Award at the International Association of Pattern Recognition (IAPR) International Conference on Pattern Recognition (ICPR) 2018.

Pong C. Yuen (Senior Member, IEEE) joined Hong Kong Baptist University, Hong Kong, in 1993, where he was the Head of the Department of Computer Science from 2011 to 2017. He was a Visiting Scholar/Professor with The University of Sydney, Sydney, NSW, USA; the University of Maryland at College Park, College Park, MD, USA; INRIA Rhone Alpes, Montbonnot-Saint-Martin, France; and ETH Zürich, Zürich, Switzerland. He is currently a Chair Professor with the Department of Computer Science and the Associate Dean of Science Faculty, Hong Kong Baptist University. Dr. Yuen is a fellow of the International Association of Pattern Recognition (IAPR). He received the first-prize and second-prize Natural Science Awards from the Guangdong Province and the Ministry of Education, China, and the Outstanding Editorial Board Service Award. He has been actively involved in many international conferences and professional communities. He has served as the General/Program/Track Co-Chair for many conferences and workshops, including International Conference on Pattern Recognition (ICPR), BTAS, WIFS, and IICB. He has been serving as the Director for the IAPR/IEEE Winter School on Biometrics since 2017. He has also served as the Vice President (Technical Activities) of the IEEE Biometrics Council from 2017 to 2019. He is an Editorial Board Member of Pattern Recognition. He served as an Associate Editor for IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (TIFS). He has been serving as an Associate Editor for IEEE TIFS since 2021.

Vishal M. Patel (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Maryland, College Park, MD, USA, in 2010. He is currently an Associate Professor with the Department of Electrical and Computer Engineering (ECE), Johns Hopkins University, Baltimore, MD, USA. Prior to joining Hopkins, he was an A. Walter Tyson Assistant Professor with the Department of Electrical and Computer Engineering, Rutgers University, New Brunswick, NJ, USA, and a member of the research faculty at the University of Maryland Institute for Advanced Computer Studies (UMIACS), College Park. Dr. Patel has received a number of awards, including the 2021 IEEE Signal Processing Society (SPS) Pierre-Simon Laplace Early Career Technical Achievement Award; the 2021 NSF CAREER Award; the 2021 International Association of Pattern Recognition (IAPR) Young Biometrics Investigator Award (YBIA); the 2016 ONR Young Investigator Award; the 2016 Jimmy Lin Award for Invention; the A. Walter Tyson Assistant Professorship Award; Best Paper Awards at IEEE AVSS 2017 and 2019, IEEE BTAS 2015, IAPR ICB 2018, and IEEE ICIP 2021; and two Best Student Paper Awards at IAPR International Conference on Pattern Recognition (ICPR) 2018. He is an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and Pattern Recognition journal. He serves on the Machine Learning for Signal Processing (MLSP) Committee of the IEEE Signal Processing Society. He serves as the Vice President of conferences for the IEEE Biometrics Council.