Federated Test-Time Adaptive Face Presentation Attack Detection with Dual-Phase Privacy Preservation

Rui Shao[1] Bochao Zhang[1] Pong C. Yuen[1] Vishal M. Patel[2]
[1] Department of Computer Science, Hong Kong Baptist University, Hong Kong
[2] Department of Electrical and Computer Engineering, Johns Hopkins University, USA

Abstract— **Face presentation attack detection (fPAD) plays a critical role in the modern face recognition pipeline. The generalization ability of face presentation attack detection models to unseen attacks has become a key issue for real-world deployment, which can be improved when models are trained with face images from different input distributions and different types of spoof attacks. In reality, due to legal and privacy issues, training data (both real face images and spoof images) are not allowed to be directly shared between different data sources. In this paper, to circumvent this challenge, we propose a Federated Test-Time Adaptive Face Presentation Attack Detection with Dual-Phase Privacy Preservation framework, with the aim of enhancing the generalization ability of fPAD models in both training and testing phase while preserving data privacy. In the training phase, the proposed framework exploits the federated learning technique, which simultaneously takes advantage of rich fPAD information available at different data sources by aggregating model updates from them without accessing their private data. To further boost the generalization ability, in the testing phase, we explore test-time adaptation by minimizing the entropy of fPAD model prediction on the testing data, which alleviates the domain gap between training and testing data and thus reduces the generalization error of a fPAD model. We introduce the experimental setting to evaluate the proposed framework and carry out extensive experiments to provide various insights about the proposed method for fPAD.**

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 [13], [3], [29], [21], [11], [20], [23], [24], [26] have found that this technique is vulnerable to various face presentation attacks such as print attacks, video-replay attacks [2], [31], [4], [29], [11] and 3D mask attacks [9], [10]. 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 paper, we consider the deployment of a fPAD system in the 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 [20], [24] and are likely to be vulnerable against attacks unseen during training.

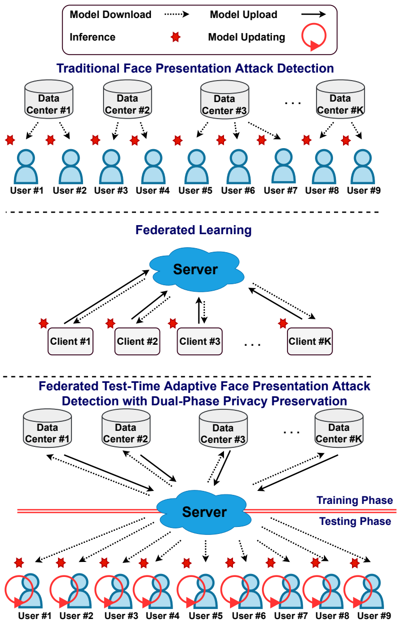

Fig. 1. Comparison between fPAD (top), traditional federated learning (middle) and the proposed framework (bottom).

On the other hand, *users* are individuals or entities that make use of fPAD solutions. For example, when a 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 Figure 1 (top). Since *data center* models lack generalization ability, inferencing with these models are 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-

978-1-6654-3176-7/21/$31.00 ©2021 IEEE

learning techniques can significantly improve the generalization ability of fPAD models [20], [24]. Therefore, the performance of fPAD models, shown in Figure 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 collected data with third parties.

In this paper, we present a framework called Federated Test-Time Adaptive Face Presentation Attack Detection with Dual-Phase Privacy Preservation based on the principles of Federated Learning (FL) and test-time adaptation targeting fPAD. Federate learning is a distributed and privacy preserving machine learning technique [14], [8], [27], [17], [15], [22]. 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 is kept private. Collaborative FL training allows the global model to exploit rich local clients information while preserving data privacy.

In the context of FL for fPAD, 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. In FL, all *clients* train models and carry out inference locally. In contrast, in FL for fPAD, only *data centers* carry out local model training. *Data centers* share their models with the *server* and download the global model during the training phase. On the other hand, *users* download the global model at the end of the training procedure and carry out inference in the testing phase as shown in Figure 1 (bottom).

Although domain gaps between training and testing data can be alleviated by the above federated learning process carried out in the training phase, fPAD models are still hard to generalize well to unseen attacks with significant domain variations in the real-world deployment. To further improve the generalization ability of fPAD model, we incorporate test-time adaptation into our framework by exploiting hints provided by the testing data. Specifically, such hints can be unveiled from the entropy of model predictions on the testing data and generalization error of fPAD models can be reduced via minimization of entropy of model predictions. As illustrated in Figure 1 (bottom), after downloading the trained fPAD models from the training phase, *users* further adapt the fPAD model in the testing phase with the help of entropy minimization of model predictions before the final classification. Note that test-time adaptation is sensitive to the initial parameters of a pre-trained model. Federated learning can produce a well pre-trained model with privacy preservation which provides a good starting point for the following test-time adaptation. Two phases are compatible with each other and together improve the generalization ability of fPAD without accessing private training data from multiple data centers.

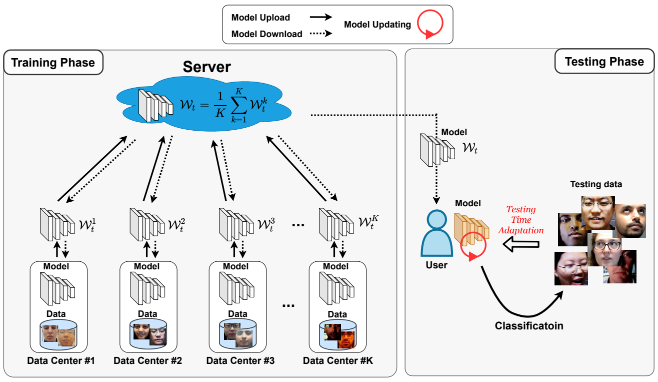

Fig. 2. Overview of the proposed framework. In the training phase, through several rounds of communications between *data centers* and *server*, the collaborated trained global fPAD model parameterized by $\mathcal{W}_t$ can be obtained in a data privacy preserving way. In the testing phase, users download this global model from the server to their device and further carry out testing time adaptation given the testing data at hand. The final classification for the testing data can be carried out based on the updated fPAD model.

II. Related Work

A. Face Presentation Attack Detection

Current fPAD methods can be categorized under single-domain and multi-domain approaches. Single-domain approaches focus 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. Multi-scale LBP [13] and color textures [3] methods are two texture-based methods that extract various LBP descriptors in various color spaces for the differentiation between real/spoof samples. Image distortion analysis [29] aims to detect the surface distortions as the discriminative cue.

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 [16], [23], [21] to extract discriminative facial motions. rPPG signals are exploited by Liu *et al.* [10], [9] to capture discriminative heartbeat information from real and spoof videos. [11] learns a CNN-RNN model to estimate different face depth and rPPG signals between the real and spoof samples.

Various fPAD datasets are introduced recently that explore different characteristics and scenarios of face presentation attacks. Multi-domain approach is proposed in order to improve the generalization ability of the fPAD model to unseen attacks. Recent work [20] casts fPAD as a domain generalization problem and proposes a multi-adversarial discriminative deep domain generalization framework to search generalized differentiation cues in a shared and discriminative feature space among multiple fPAD datasets. [12] treats fPAD as a zero-shot problem and proposes a Deep Tree Network to partition the spoof samples into multiple subgroups of attacks. [24] addresses fPAD with a meta-learning framework and enables the model learn to generalize well through simulated train-test splits among multiple datasets. These multi-domain approaches have access to data from multiple datasets or multiple spoof sub-groups that enable them to obtain generalized models. In this paper, 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*. This paper aims to exploit multi-domain information in a privacy preserving manner.

B. Federated Learning

Federated learning 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) [14], one of the fundamental frameworks for FL, learns a global model by averaging model parameters from local clients. FedProx [17] and Agnostic Federated Learning (AFL) [15] are two variants of FedAvg which aim to address the bias issue of the learned global model towards different clients. These two methods achieve better models by adding proximal term to the cost functions and optimizing a centralized distribution mixed with client distributions, respectively. A recent work FedPAD [25] also exploits federated learning for the task of fPAD. However, when encountering unseen attacks with large domain gap, the performance is still degraded. Comparatively, the proposed method integrates the testing-time adaptation with federated learning, which further facilities the generalization ability of a fPAD model during testing.

III. Proposed Method

A. Training Time Federated Learning

The proposed Test-Time Adaptive Face Presentation Attack Detection 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 $\mathcal{D}^1, \mathcal{D}^2, ..., \mathcal{D}^K$ with data provided with image and label pairs denoted as x and y. y denotes the ground-truth with binary class labels (y= 0/1 are the 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:

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

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

$$\mathcal{W}^k \leftarrow \mathcal{W}^k - \eta\nabla\mathcal{L}(\mathcal{W}^k),$$

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

Algorithm 1 Federated Test-Time Adaptive Face Presentation Attack Detection with Dual-Phase Privacy Preservation

```
Require:
  Input: $K$ Data Centers have $K$ fPAD datasets $\mathcal{D}^1, \mathcal{D}^2, ..., \mathcal{D}^K$, Testing data presented to user $\mathcal{U}$
  Initialization: $K$ Data Centers have $K$ fPAD models $\mathcal{F}^1, \mathcal{F}^2, ..., \mathcal{F}^K$ parameterized by $\mathcal{W}_0^1, \mathcal{W}_0^2, ..., \mathcal{W}_0^K$. $L$ is the number of local epochs. $\eta$ is the learning rate. $t$ is the federated learning rounds
  Training Phase:
  Server aggregates:
  initialize $\mathcal{W}_0$
  for each round $t$ = 0, 1, 2,... do
    for each data center $k$ = 1, 2,..., $K$ in parallel do
      $W_t^k \leftarrow$ DataCenterUpdate$(k, \mathcal{W}_t)$
    end for
    $\mathcal{W}_t = \frac{1}{K}\sum_{k=1}^{K} \mathcal{W}_t^k$
    Download $W_t$ to Data Centers
  end for
  Testing Phase:
  Users Download model $\mathcal{F}_t$ parameterized by $\mathcal{W}_t$
  $\mathcal{H}(\mathcal{F}_t(x)) = \sum_{x\sim\mathcal{U}} \mathcal{F}_t(x)\log\mathcal{F}_t(x) + (1-\mathcal{F}_t(x))\log(1-\mathcal{F}_t(x))$
  $\mathcal{W}_{t(\gamma,\beta)} \leftarrow \mathcal{W}_{t(\gamma,\beta)} - \eta\nabla\mathcal{H}(\mathcal{F}_t(x))$
  Users make final classification

  DataCenterUpdate$(k, \mathcal{W})$:
  for each local epoch $i$ = 1, 2,..., $L$ do
    $\mathcal{L}(\mathcal{W}^k) = \sum_{(x,y)\sim\mathcal{D}^k} y\log\mathcal{F}^k(x) + (1-y)\log(1-\mathcal{F}^k(x))$
    $\mathcal{W}^k \leftarrow \mathcal{W}^k - \eta\nabla\mathcal{L}(\mathcal{W}^k)$
  end for
  Upload $W^k$ to Server
```

It should be noted that 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 contributes to a better generalized fPAD model. A straightforward solution is to collect and combine all the data from K data centers denoted as $\mathcal{D} = \{\mathcal{D}^1 \cup \mathcal{D}^2 \cup ... \cup \mathcal{D}^K\}$ to train a 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 multi-domain data $\mathcal{D}$ [20], [24]. 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 a fPAD model, we propose the Test-Time Adaptive Face Presentation Attack Detection framework. Instead of accessing private fPAD data of each *data center*, the proposed framework introduces a *server* to exploit the fPAD information of all data centers by aggregating the above model updates $(\mathcal{W}^1, \mathcal{W}^2, ..., \mathcal{W}^K)$ of all data centers. Inspired by the Federated Averaging [14] algorithm, in the proposed framework, server carries out the aggregation of model updates via calculating the average of updated parameters $(\mathcal{W}^1, \mathcal{W}^2, ..., \mathcal{W}^K)$ in all data centers as $\mathcal{W} = \frac{1}{K}\sum\limits_{k=1}^{K} \mathcal{W}^k$.

After the aggregation, 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. 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 $(\mathcal{W}_t^1, \mathcal{W}_t^2, ..., \mathcal{W}_t^K)$. The t-th aggregation in the server can be carried out as $\mathcal{W}_t = \frac{1}{K}\sum\limits_{k=1}^{K} \mathcal{W}_t^k$. After t-rounds of communication between data centers and the server, the trained global fPAD model parameterized by $\mathcal{W}_t$ can be obtained without compromising the private data of individual *data centers*. Once training is converged, *users* will download the trained model from the server to their devices to carry out fPAD locally.

B. Testing-Time Adaptation

Although federated learning in the training phase can exploit various fPAD information available from multiple data centers to improve the generalization ability of a fPAD model, the fPAD model can not generalize well to some unseen attacks when the domain gap between the training and testing data is large, i.e., significant environmental variations. To address this issue, we further propose the test-time adaptation. We argue that the data presented to the users during testing can provide some hints about their distribution, which can be used to adapt the fPAD model before making the final classification. Specifically, as demonstrated in [28], such hints can be unveiled by Shannon entropy [19] of model predictions which can indicate the distribution shift between seen training data and unseen testing data in an unsupervised way. Therefore, the domain gap between training and testing data can be further reduced by minimizing the Shannon entropy of the fPAD model predictions on the testing data. As illustrated in Fig. 3, given the testing data presented to user $\mathcal{U}$, and the fPAD model $\mathcal{F}_t$ downloaded from the training phase, we calculate the entropy of fPAD model prediction as follows:

$$\mathcal{H}(\mathcal{F}_t(x)) = \sum_{x \sim \mathcal{U}} \mathcal{F}_t(x) \log \mathcal{F}_t(x) + (1 - \mathcal{F}_t(x)) \log(1 - \mathcal{F}_t(x)).$$

To reduce the probability of overfitting during test-time adaptation given limited testing data, we focus on minimizing the above entropy with respect to affine transformation parameters (scale $\mathcal{W}_{t(\gamma)}$ and shift $\mathcal{W}_{t(\beta)}$) of all batch normalization layers in the fPAD model and keep all the other parameters fixed. This process can be carried out as: $\mathcal{W}_{t(\gamma,\beta)} \leftarrow \mathcal{W}_{t(\gamma,\beta)} - \eta \nabla \mathcal{H}(\mathcal{F}_t(x))$. After this test time adaptation, we use the updated fPAD model for the final real/fake classification. Same as federated learning carried out in the training phase, the whole process of the above test-time adaptation also does not need to get access to the private training data from multiple data centers, and thus the generalization ability of fPAD model can be further ameliorated in a privacy preserving manner during testing. Moreover, the above entropy calculation does not require one to design a specific self-supervision task such as image rotation for the supervision of adaptation, which significantly facilitates its compatibility to the task of fPAD.

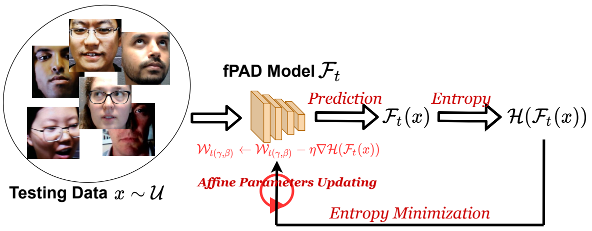
fPAD Model $\mathcal{F}_t$
Prediction
$\mathcal{F}_t(x)$
Entropy
$\mathcal{H}(\mathcal{F}_t(x))$
$\mathcal{W}_{t(\gamma,\beta)} \leftarrow \mathcal{W}_{t(\gamma,\beta)} - \eta\nabla\mathcal{H}(\mathcal{F}_t(x))$
Affine Parameters Updating
Entropy Minimization
Testing Data $x \sim \mathcal{U}$

Fig. 3. Details of test-time adaptation during testing. Note that test-time adaptation updates the parameters of a fPAD model using unlabeled testing data presented to the users without accessing the training data.

On the other hand, we should note that test-time adaptation is sensitive to the quality of initial parameters of a pre-trained model. Federated learning can produce a well pre-trained model in a privacy preserving manner so that a suitable starting point can be provided for the following optimization of testing-time adaptation. This further demonstrates the compatibility of dual-phase privacy preservation for the task of fPAD.

IV. Experiments

To evaluate the performance of the proposed framework, we carry out extensive experiments using five 2D fPAD datasets and two 3D mask fPAD datasets. In this section, we first describe the datasets and the testing protocol used in our experiments. Then we report various experimental results based on multiple fPAD datasets. Discussions and analysis about the results are carried out to provide various insights about FL for fPAD.

A. Experimental Settings

TABLE I
Comparison of seven experimental datasets.

Dataset	Extra light	Complex background	Attack type	Display devices
C	No	Yes	Printed photo Cut photo Replayed video	iPad
I	Yes	Yes	Printed photo Display photo Replayed video	iPhone 3GS iPad
M	No	Yes	Printed photo Replayed video	iPad Air iPhone 5S
O	Yes	No	Printed photo Display photo Replayed video	Dell 1905FP Macbook Retina
S	Yes	Yes	Printed photo Display photo Replayed video	Dell 1905FP iPad Pro iPhone 7 Galaxy S8 Asus MB168B
3	No	No	Thatsmyface 3D mask	Kinect
H	Yes	Yes	Thatsmyface 3D mask REAL-f mask	MV-U3B

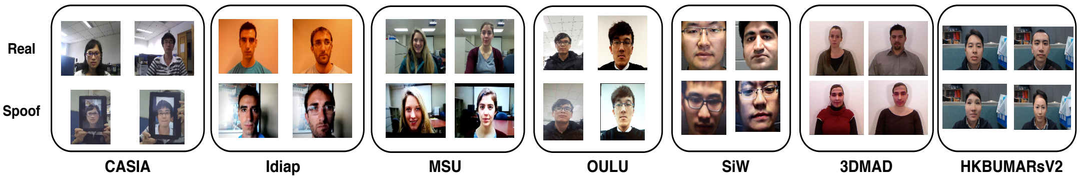

Fig. 4. Sample images corresponding to real and attacked faces from CASIA-MFSD [31], Idiap Replay-Attack [4], MSU-MFSD [29], Oulu-NPU [2], SiW [11], 3DMAD [5], and HKBUMARsV2 [9] datasets.

1) Datasets: We evaluate our method using the following seven publicly available fPAD datasets which contain print, video replay and 3D mask attacks:
1) Oulu-NPU [2] (O for short)
2) CASIA-MFSD [31] (C for short)
3) Idiap Replay-Attack [4] (I for short)
4) MSU-MFSD [29] (M for short)
5) SiW [11] (S for short)
6) 3DMAD [5] (3 for short)
7) HKBUMARsV2 [9] (H for short).
Table I shows the variations in these seven datasets. Some 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, background and so on). Therefore, significant domain shifts exist among these datasets.

2) Protocol: The testing protocol used in the paper is designed to test the generalization ability of fPAD models. Therefore, in each test, performance of a trained model is evaluated against a dataset that it 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 a fPAD model. The trained model is tested 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 Rates (EER) and Area Under Curve (AUC) are used as evaluation metrics in our experiments, which are three most widely-used metrics for the cross-datasets/cross-domain evaluations. Following [12], in the absence of a development set, thresholds required for calculating evaluation metrics are determined based on the data in all *data centers*.

4) Implementation Details: Our deep network is implemented on the platform of PyTorch. We adopt Resnet-18 [6] as the structure of fPAD models $\mathcal{F}^i(i = 1, 2, 3, ..., K)$. In the training phase, Adam optimizer [7] is used for the optimization of federated learning. The learning rate of federated learning is set as 1e-2. The batch size is 64 per data center. Local optimization epoch L is set equal to 3. In the testing phase, Adam optimizer [7] is used for the optimization of test time adaptation. The learning rate of test time adaptation is set as 5e-3.

B. Experimental Results

In this section we demonstrate the practicality and generalization ability of the proposed framework in the real-world scenario. We first compare the performance of the proposed framework with models trained with data from a single data center. As mentioned above, due to the limitation of data privacy that exists in the real-world, 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 the Table II 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 II denoted as **Fused**. Please note that it is impossible for users to carry out feature fusion because a classifier cannot be trained based on the fused features without accessing to any real/spoof data during inference in the users. According to Table II, 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 three for the case considered in this experiment).

FedPAD [25] exploits the federated learning on the task of fPAD. Table II illustrates that the FedPAD improves the performance on most of settings. However, when encountering large domain gap settings such as O&C&M to I, FedPAD still cannot achieve very promising performance. Comparatively, after integrating test-time adaptation with federated learning, on average the proposed framework (**Ours**) is able to significantly improve the performance in all settings and outperform other baselines. Moreover, we plot the ROC curves in Fig 5, which also shows the effectiveness of the proposed framework. These results demonstrate that the proposed method is more effective in facilitating the generalization ability of fPAD with the dual-phase privacy preservation.

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 II. Note that this baseline violates the assumption of preserving data privacy, and therefore is

TABLE II

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	41.61	36.66	67.07
	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			
	C	I	45.99	48.55	51.24			
	M	I	48.50	33.70	66.29			
	M	O	29.80	24.12	84.86			
	C	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			
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			
Ours	O&C&I	M	14.70	16.64	90.57	**23.18**	**23.88**	**83.40**
	O&M&I	C	26.33	29.75	77.77			
	O&C&M	I	28.61	26.04	82.07			
	I&C&M	O	23.09	23.09	83.21			

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 +Test-Time-Adaptation	O	M	28.81	31.35	74.77	35.09	36.43	68.00
	C	M	34.49	35.64	69.46			
	I	M	12.11	16.53	90.23			
	O	C	30.37	30.35	74.69			
	M	C	41.20	42.10	60.69			
	I	C	43.53	42.91	59.21			
	O	I	47.88	46.76	56.71			
	C	I	60.02	65.26	36.34			
	M	I	17.40	17.04	89.65			
	M	O	23.24	23.30	83.65			
	C	O	31.63	31.08	74.34			
	I	O	36.94	37.16	66.60			
Ours	O&C&I	M	14.70	16.64	90.57	**23.18**	**23.88**	**83.40**
	O&M&I	C	26.33	29.75	77.77			
	O&C&M	I	28.61	26.04	82.07			
	I&C&M	O	23.09	23.09	83.21			

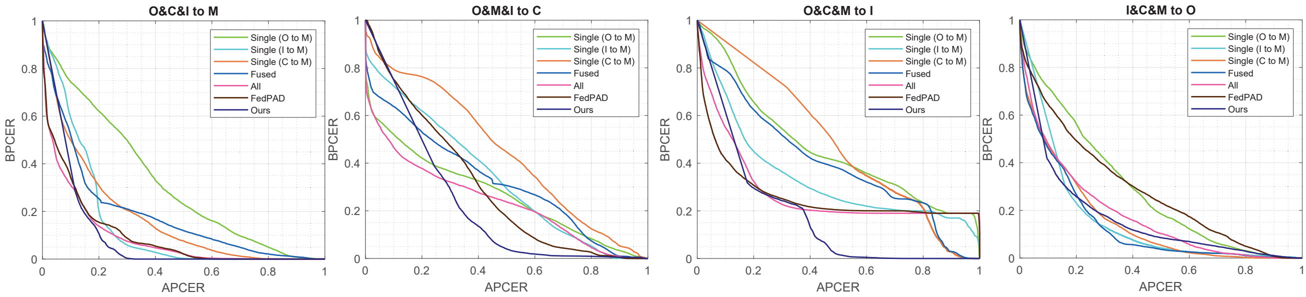

Fig. 5. ROC curves of models trained by data from single data center and various data centers.

not a valid comparison for FL for fPAD. Nevertheless, it indicates the upper bound of performance for the federated learning applied in fPAD. From Table II, it can be seen that the proposed framework is able to perform better than this baseline. This shows the proposed framework is able to obtain a privacy persevering fPAD model without sacrificing fPAD performance.

1) Compatibility of Dual Phases: To improve the generalization ability of fPAD models, users can also download models trained with data from a single data center and carry out test-time adaptation. Therefore, a natural choice is to integrate test-time adaptation with models trained with data from a single data center (**Single**). We tabulate the comparison between this baseline (**Single+Test Time Adaptation**)

TABLE IV

COMPARISON OF DIFFERENT NUMBER OF DATA CENTERS

Methods	Data Centers	User	HTER	EER	AUC
FedPAD	O&I	C	49.22	49.10	51.19
	O&M&I		42.27	36.95	70.49
	O&M&I&S		41.74	29.90	78.47
Ours	O&I		43.09	42.57	63.21
	O&M&I		26.33	29.75	77.77
	O&M&I&S		**22.90**	**22.39**	**85.89**

TABLE V

COMPARISON OF DIFFERENT NUMBER OF DATA CENTERS.

Methods	Data Centers	User	HTER	EER	AUC
FedPAD	C&M	S	25.11	20.64	88.08
	I&C&M		29.61	14.61	93.30
	I&C&M&O		12.45	**8.98**	**97.18**
Ours	C&M		15.45	14.06	92.30
	I&C&M		16.45	15.34	92.03
	I&C&M&O		**10.62**	9.63	96.35

and the proposed framework in Table III. Table III shows that the proposed framework integrating test-time adaptation with federated learning performs better than **Single+Test Time Adaptation**. fPAD models trained with data from a single data center (**Single**) will easily overfit to data of corresponding local data center and thus generate a poor fPAD model. Comparatively, by exploiting various fPAD data available from multiple data centers, federated learning can produce a more suitable fPAD model during the training phase with privacy preservation. Since test-time adaptation is very sensitive to the quality of pre-trained model, fPAD model trained by the proposed method is equipped with better initial parameters as a better starting point for the following test-time adaptation. In this way, the corresponding test-time adaptation can achieve improved performance as shown in Table III. This demonstrates the proposed framework is more able to exploit the compatibility between dual phases and thus achieve better fAPD performance.

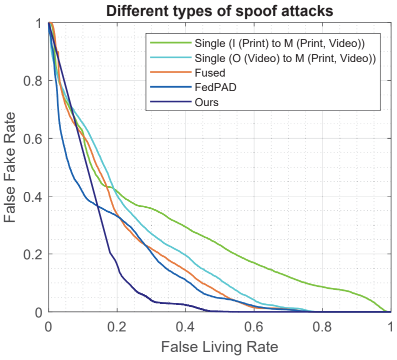

Fig. 6. ROC curves of models trained by different types of 2D spoof attacks.

2) Comparison of different number of data centers: In this section, we investigate the importance of having more data centers during training. Different data centers exploit different characteristics of face presentation attacks. Therefore, we expect aggregating information from more data centers in the proposed framework to produce more robust models with better generalization. In order to verify this point, we increase the number of data centers in the proposed framework and FedPAD method. The comparison is reported in Tables IV and V. The experiments are carried out using five datasets (O, M, I, C, S). In Table IV, we select the dataset C as the data presented to the user and the remaining datasets as the data centers for training the fPAD model with our framework, where we increase the number of data centers from 2 to 4. Another experiment is carried out with a different combination of the same five datasets and the results are shown in Table V. From Tables IV and V, it can be seen that most values of evaluation metrics improve along when the number of data centers increases, and the proposed method achieves the best performance in the first setting and comparable results in the second setting compared to FedPAD. This demonstrates that increasing the number of data centers in the proposed framework can improve the performance.

3) Generalization ability to various 2D spoof attacks: In reality, due to limited resources, one data center may only be able to collect limited types of 2D attacks. However, various 2D attacks may appear to the users. This section supposes that one data center collects one particular type of 2D attack such as print attack or video-replay attack. As illustrated in Table VI, first, we select real faces and print attacks from dataset I and real faces and video-replay attacks from dataset O to train a fPAD model respectively and evaluate them on dataset M (containing both print attacks and video-replay attacks). In both considered cases as shown in Table VI, the corresponding trained models cannot generalize well to dataset M which contains the additional types of 2D attacks compared to dataset I and O, respectively. 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 VI. FedPAD method obtains a performance gain compared to score fusion. Comparatively, the proposed method further improves the performance, especially with a gain of 12.31% in HTER and 7.47% in EER compared to FedPAD. We also plot the corresponding ROC curve for this comparison in Fig 6, which also demonstrate the superior performance of the proposed method. This demonstrate that test-time adaptation can effectively improve the generalization ability to various 2D attacks when FL model is trained with limited types of fPAD data.

4) Generalization ability to 3D mask attacks : In this section, we investigate the generalization ability of the proposed framework to 3D mask attacks and the comparison is also conducted between the proposed framework and FedPAD method. First, in FedPAD, a fPAD model is trained with data centers exploiting 2D attacks (data from datasets O, C, I and M). This model is tested with 3D mask attacks (data from dataset H). Then, we include one more data center containing 3D mask attacks (dataset 3) into FedPAD and retrain it. Table VII shows that introducing diversity of data centers (by including a 3D mask attack) can improve performance in all evaluation metrics. This demonstrates that increasing data centers with 3D mask attacks within the federated learning framework can improve the generalization ability of fPAD

TABLE VI

EFFECT OF USING DIFFERENT TYPES OF SPOOF ATTACKS

Methods	Data Centers	User	HTER (%)	EER (%)	AUC (%)
Single	I (Print)	M (Print, Video)	38.82	33.63	72.46
	O (Video)	M (Print, Video)	35.76	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
Ours	I (Print) & O (video)	M (Print, Video)	**18.20**	**18.63**	**87.57**

TABLE VII

IMPACT OF ADDING DATA CENTERS WITH DIVERSE ATTACKS

Methods	Data Centers	User	HTER	AUC
FedPAD	O&C&M (2D)	3 (3D)	27.21	76.05
	O&C&M (2D) &H (3D)		34.70	**92.35**
Ours	O&C&M (2D) &H (3D)		**16.97**	90.25

model to the novel 3D mask attacks. We carry out the same experiment based on the proposed framework. In Table VII, it can be seen that the proposed method can significantly improve HTER by 17.73% compared to FedPAD. This means that after adapted with novel 3D mask attack data by test-time adaptation during testing, fPAD model trained with federated learning in the training phase is more able to generalize well to the novel types of 3D mask attacks, which forms a better generalized fPAD framework.

V. CONCLUSION

In this paper, we presented a framework based on the principles of FL and test-time adaptation, targeting the application of fPAD with the objective of obtaining generalized fPAD models while preserving data privacy in both training and testing phases. In the training phase, through communications between *data centers* and the *server*, a global fPAD model is obtained by iteratively aggregating the model updates from various *data centers*. In the testing phase, test-time adaptation is further exploited to minimize the prediction entropy of trained fPAD so that the generalization error on the unseen face presentation attacks can be further reduced. Local private data in the data centers is not accessed during the whole process. Extensive experiments are carried out to demonstrate the effectiveness of the proposed framework. In the future, we will further investigate the generalization improvement among different imbalanced datasets, like FedMix [30] and Fed-Focal Loss [18].

VI. ACKNOWLEDGMENTS

This work is partially supported by Research Grants Council (RGC/HKBU12200820), Hong Kong. Vishal M. Patel was supported by the NSF grant 1801435.

REFERENCES

[1] S. Bengio and J. Mariéthoz. A statistical significance test for person authentication. In *The Speaker and Language Recognition Workshop*, 2004.

[2] Z. Boulkenafet and et al. Oulu-npu: A mobile face presentation attack database with real-world variations. In *FG*, 2017.

[3] Z. Boulkenafet, J. Komulainen, and A. Hadid. Face spoofing detection using colour texture analysis. In *IEEE Trans. Inf. Forens. Security, 11(8): 1818-1830*, 2016.

[4] I. Chingovska, A. Anjos, and S. Marcel. On the effectiveness of local binary patterns in face anti-spoofing. In *BIOSIG*, 2012.

[5] N. Erdogmus and S. Marcel. Spoofing face recognition with 3D masks. 2014. TIFS.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[7] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *arXiv preprint arXiv:1412.6980*, 2014.

[8] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated learning: Challenges, methods, and future directions. In *arXiv preprint arXiv:1908.07873*, 2019.

[9] S. Liu, X. Lan, and P. C. Yuen. Remote photoplethysmography correspondence feature for 3D mask face presentation attack detection. In *ECCV*, 2018.

[10] S. Liu, P. C. Yuen, S. Zhang, and G. Zhao. 3D mask face anti-spoofing with remote photoplethysmography. In *ECCV*, 2016.

[11] Y. Liu, A. Jourabloo, and X. Liu. Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In *CVPR*, 2018.

[12] Y. Liu, J. Stehouwer, A. Jourabloo, and X. Liu. Deep tree learning for zero-shot face anti-spoofing. In *CVPR*, 2019.

[13] J. Määttä, A. Hadid, and M. Pietikäinen. Face spoofing detection from single images using micro-texture analysis. In *IJCB*, 2011.

[14] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, et al. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2016.

[15] M. Mohri, G. Sivek, and A. T. Suresh. Agnostic federated learning. 2019.

[16] T. F. Pereira and et al. Face liveness detection using dynamic texture. In *EURASIP Journal on Image and Video Processing, (1): 1-15*, 2014.

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

[18] D. Sarkar, A. Narang, and S. Rai. Fed-focal loss for imbalanced data classification in federated learning. *arXiv preprint arXiv:2011.06283*, 2020.

[19] C. E. Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[20] R. Shao, X. Lan, J. Li, and P. C. Yuen. Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In *CVPR*, 2019.

[21] R. Shao, X. Lan, and P. C. Yuen. Deep convolutional dynamic texture learning with adaptive channel-discriminability for 3D mask face anti-spoofing. In *IJCB*, 2017.

[22] R. Shao, X. Lan, and P. C. Yuen. Feature constrained by pixel: Hierarchical adversarial deep domain adaptation. In *ACM MM*, 2018.

[23] R. Shao, X. Lan, and P. C. Yuen. Joint discriminative learning of deep dynamic textures for 3D mask face anti-spoofing. In *IEEE Trans. Inf. Forens. Security, 14(4): 923-938*, 2019.

[24] R. Shao, X. Lan, and P. C. Yuen. Regularized fine-grained meta face anti-spoofing. In *AAAI*, 2020.

[25] R. Shao, P. Perera, P. C. Yuen, and V. M. Patel. Federated face presentation attack detection. *arXiv preprint arXiv:2005.14638*, 2020.

[26] R. Shao, P. Perera, P. C. Yuen, and V. M. Patel. Open-set adversarial defense. In *ECCV*, 2020.

[27] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar. Federated multi-task learning. In *NIPS*, 2017.

[28] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, T. Darrell, U. Berkeley, and A. Research. tent: fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021.

[29] D. Wen, H. Han, and A. K. Jain. Face spoof detection with image distortion analysis. In *IEEE Trans. Inf. Forens. Security, 10(4): 746-761*, 2015.

[30] T. Yoon, S. Shin, S. J. Hwang, and E. Yang. Fedmix: Approximation of mixup under mean augmented federated learning. *arXiv preprint arXiv:2107.00233*, 2021.

[31] Z. Zhang and et al. A face antispoofing database with diverse attacks. In *ICB*, 2012.