This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3629353

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

Fed-StackFPAD: Federated Learning for Face Presentation Attack Detection with Stacking to Tackle Data Heterogeneity

MEHMET FATİH GÜNDOĞAR[1], ÇİĞDEM EROĞLU ERDEM[2], (Member, IEEE), and ÖMER KORÇAK[3], (Member, IEEE)

[1]Department of Computer Engineering, Faculty of Engineering, Marmara University, İstanbul, Türkiye (e-mail: mehmetfatihgundogar@gmail.com)
[2]Department of Electrical and Electronics Engineering, Özyeğin University, İstanbul, Türkiye (email: cigdem.erogluerdem@gmail.com)
[3]Department of Computer Engineering, Faculty of Engineering, Marmara University, İstanbul, Türkiye (e-mail: omer.korcak@marmara.edu.tr)

Corresponding author: Mehmet Fatih Gündoğar (e-mail: mehmetfatihgundogar@gmail.com).

This work was supported by The Scientific and Technological Research Council of Türkiye (TÜBİTAK) under projects EEAG-122E025 and ARDEB 1002 [Grant no.: 124E292]. The numerical calculations reported in this paper were partially performed at TÜBİTAK ULAKBİM, High Performance and Grid Computing Center (TRUBA resources).

ABSTRACT Face presentation attack detection (FPAD) is crucial for ensuring the accuracy and reliability of face recognition and verification systems. However, existing datasets in the literature exhibit limitations in the diversity of presentation attack types they encompass. Deep learning-based solutions that perform well on one dataset may not generalize well to others due to domain shift issues. A potential approach to improve the robustness of deep learning models involves expanding dataset diversity through multi-source data collection. However, this process is hindered by challenges such as extensive data labeling and privacy concerns of centralized data collection. To overcome these challenges, we introduce Fed-StackFPAD, a federated learning (FL) framework for face presentation attack detection that integrates a pretrained self-supervised transformer-based masked autoencoder with stacking-based ensemble learning. By leveraging model aggregation across multiple clients, the proposed framework enhances performance without requiring centralized data sharing. The meta-model in the stacking phase combines complementary strengths of the federated model and data center-specific models in the ensemble. Experimental evaluations using 4 different datasets demonstrate that Fed-StackFPAD outperforms existing state-of-the-art methods, highlighting its effectiveness in addressing domain adaptation challenges through FL, self-supervised masked image modeling using vision transformers, and stacking-based model aggregation. Fed-StackFPAD demonstrates a significant improvement, lowering the average half total error rate (HTER) by 13.7% over the current best FL methods. To ensure reproducibility, we publicly share the fine-tuned models and will make the code available on https://www.github.com/mehmetfatihgundogar/Fed-StackFPAD.

INDEX TERMS Face presentation attack detection, federated learning, deep learning, stacking, self-supervised learning, ensemble learning.

I. INTRODUCTION

FACE recognition and verification processes have diverse industrial applications, including self-service passport controls, secure building entrance systems, and mobile device access. However, such systems are inherently vulnerable to face presentation attacks, which exploit various spoofing techniques. These attacks can be performed using easily accessible mediums such as wearable masks, printed photographs, and video replays obtained from social media platforms [1]. Figure 1 illustrates several common attack instruments alongside the corresponding spoof types. Given the severity of this threat, face presentation attack detection (FPAD) is a critical component of face recognition and verification systems. In most cases, software-based FPAD solutions are preferred due to their ease of implementation and cost-effectiveness.

The literature extensively explores the FPAD problem [7]–[16], primarily utilizing publicly available datasets. These datasets encompass various types of spoofing attacks captured in image and video formats. However, as each dataset is collected under different conditions, significant variations exist in terms of resolution, illumination, subject ethnicity, and

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3629353



(a) Printed photo (b) Warped photo (c) Video replay (d) 3D mask (e) Silicon mask (f) Doll

FIGURE 1. Sample face presentation attacks from multiple FPAD datasets: (a) Printed photo attack from Oulu-Npu [2], (b) Warped photo attack from Casia-Fasd [3], (c) Video replay attack from Replay-Attack [4], (d) 3D mask attack from 3DMAD [5], (e) Silicon mask attack from MLFP [6], (f) Doll attack from WFAS [1].

other factors. Due to the inherently non-IID (non-independent and identically distributed) nature of FPAD datasets, achieving a universal FPAD solution remains a challenge. While increasing data diversity through the collection of additional spoofing attack samples is a theoretically sound approach, it is often impractical in real-world scenarios. Data acquisition for training machine learning or deep learning models is both costly and labor-intensive. Moreover, training such models necessitates centralizing data, which requires substantial computational resources proportional to dataset size. Beyond computational constraints, privacy regulations further complicate efforts to aggregate data in a single repository.

FL emerges as a viable solution in scenarios where data is distributed across multiple locations and stringent privacy safeguards are necessary [17]. FL-based models eliminate the need for centralized data collection by allowing individual data centers to train models locally. Instead of sharing raw data, these centers exchange model weights to collaboratively train a global model. This approach not only preserves data privacy but also distributes computational burdens across multiple data centers. Additionally, aggregating locally trained model weights enhances the performance of the global model. While FL has been widely explored across various domains such as healthcare, mobile sensing, mission-critical systems, biometric authentication, and FL applications on graphs and semi-supervised settings [18]–[24], its application in FPAD remains limited. In this work, each data center acts as a federated client; thus, we use the terms *client* and *data center* interchangeably throughout the paper.

Another critical challenge in FPAD is the scarcity and heterogeneity of labeled data across domains. One promising strategy to mitigate this challenge is to employ self-supervised pretraining. Self-supervised learning enables models to acquire robust and transferable feature representations without relying on extensive labeled data, making it particularly effective in low-data and non-IID settings. In self-supervised learning, models are typically trained on an auxiliary task, known as a pretext task, that does not require human-annotated labels but encourages the extraction of meaningful semantic features from the input data. Recent studies, such as [25], demonstrate that self-supervised pretraining, particularly using Vision Transformer (ViT) [26] models with masked autoencoders (MAE) [27], can effectively address data heterogeneity in FL environments by enabling better local representation learning and improving global model aggregation. In this setting, masked autoencoder serves as the pretext task, where parts of the input image are randomly masked and the model is trained to reconstruct the missing regions, thus forcing it to learn rich and generalizable visual features. Moreover, starting from a pretrained model reduces the need for extensive domain-specific adaptation at each data center, thereby improving computational efficiency and allowing the federated system to achieve strong performance with minimal local optimization. While our study does not involve additional self-supervised training, we leverage a ViT encoder that has been pretrained in a self-supervised manner on the large-scale ImageNet-22K dataset [28]. By initializing from such pretrained weights, we benefit from the strong generalization and transferability properties learned during self-supervised training—without requiring access to any unlabeled data in our setup.

The primary objective of FL is to derive a global model by iterating through multiple FL rounds, leveraging contributions from different data centers. However, in FPAD, the spoofing mediums, image capture devices, illumination, and ethnicity of subjects vary widely, yielding strongly non-IID FPAD dataset distributions. As a consequence, the aggregated global FL model tends to overfit dominant domains and underrepresent minority ones, leading to residual domain-shift errors at inference time. These dominant domains are typically the ones with a higher number of training samples, which causes the global model to bias toward their data distributions. Stacking-based ensemble learning approaches offer a principled remedy for this imbalance [29]. Stacking preserves the specialized decision surfaces learned locally and lets a lightweight meta-classifier fuse their posterior probabilities. Concretely, each model in the stacking ensemble captures complementary cues that are characteristic of its own domain, such as color-print artefacts in one dataset versus moiré patterns or screen flicker in another. The meta-classifier observes these heterogeneous probability vectors and learns to re-weight them on a validation set that mimics the target distribution.

In this study, we propose a novel framework, Fed-StackFPAD, which makes the following key contributions to the literature:

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3629353

- We present, to the best of our knowledge, the first adaptation of a self-supervised pretrained Vision Transformer (ViT-MAE) model trained using a masked autoencoder within an FL framework for face presentation attack detection (FPAD). By leveraging the strong generalization capabilities of the pretrained model, our approach effectively addresses the data heterogeneity challenge commonly encountered in FPAD scenarios.
- To enhance representation learning under non-IID conditions, we utilize the ViT-MAE encoder pretrained on the large-scale ImageNet-22K dataset [28]. Even without additional pretraining on our FPAD datasets, this foundation model significantly improves downstream classification performance, yielding lower HTER across diverse domains. Our results confirm that ViT-MAE offers strong generalization and robustness to data heterogeneity in federated settings.
- We propose a stacking-based ensemble strategy that aggregates predictions from both data center-specific models and the global federated model using a meta-classifier to mitigate domain shift and leverage complementary decision boundaries.
- Experimental results across multiple FPAD datasets demonstrate that the proposed stacking mechanism significantly lowers HTER (on average by 13.7%) compared to current state-of-the-art single data center and FL methods, confirming its effectiveness in non-IID scenarios.

The remainder of this paper is structured as follows: Section II reviews recent centralized FPAD studies, as well as FL-based FPAD and stacking-based approaches in the literature. Section III details the proposed Fed-StackFPAD architecture. Section IV presents the experimental setup and results. Section V provides a comprehensive discussion of our findings. Finally, Section VI concludes the paper with key insights and future research directions.

II. RELATED WORK

In this section, we briefly review prior studies in the literature related to FPAD, FL-based FPAD frameworks, and ensemble learning approaches based on stacking.

A. FPAD FOR FACE RECOGNITION AND VERIFICATION

Traditional machine learning-based approaches for FPAD have often relied on handcrafted features. Early studies explored texture-based descriptors such as local binary patterns (LBP) [30] and histogram of oriented gradients (HoG) [31], typically combined with classical classifiers like linear discriminant analysis (LDA) and support vector machines (SVM).

A comprehensive survey by [32] reviewed various FPAD methodologies, while another by Yu et al. [33] focused specifically on deep learning-based techniques. Recent advances have demonstrated that deep neural networks, when trained with sufficient data, significantly outperform traditional handcrafted methods [14], [34]–[41].

In [9], the authors addressed the challenge of detecting 3D mask-based presentation attacks. They proposed a novel temporal similarity feature, TSrPPG, derived from remote photoplethysmography (rPPG) signals in the time domain. By analyzing the amplitude, gradient, and phase characteristics of rPPG signals from bona fide subjects, the method effectively distinguishes genuine and spoof samples. Unlike prior rPPG-based approaches, this method does not require extended observation windows. However, its performance has not been validated against presentation attacks involving other types of attack instruments beyond 3D masks.

Adversarial learning techniques have been widely employed through generative adversarial networks (GANs) for domain adaptation in the FPAD problem. Methods such as ADDA [42], USDAN [43], DupGAN [44], and ADA [45] leverage adversarial objectives to align feature distributions under unsupervised or semi-supervised settings. UDA [46] enhances this approach with metric learning, adversarial unsupervised domain adaptation, and disentangled representation learning. KDPAD [47] and SDA-FAS++ [48] further extend the paradigm by addressing one-class domain adaptation and source-free domain adaptation scenarios, respectively, improving cross-domain generalization for face presentation attack detection.

Although various machine learning, deep learning, and adversarial learning-based domain adaptation methods have demonstrated promising results in FPAD, most of the strong cross-dataset generalization performance has been achieved by adversarial learning-based approaches. Nevertheless, these methods predominantly rely on centralized training frameworks and do not explicitly consider privacy preservation, which is becoming increasingly critical in real-world deployment scenarios. This observation highlights the need for privacy-preserving and FL strategies to further enhance the generalization and security of face presentation attack detection systems.

B. FL-BASED FPAD METHODS

While deep learning-based FPAD methods benefit from access to large-scale, diverse datasets, practical limitations such as data privacy regulations and transmission costs often prevent centralized data collection. FL offers a promising alternative by allowing model training across distributed data centers without sharing raw data. Each data center keeps its data local and only exchanges model updates with a central server.

Several survey studies have reviewed FL's design paradigms, optimization methods, and application domains [18], [19], [49]–[51]. Among these, the most widely adopted optimization method is Federated Averaging (FedAvg) [52], which updates the global model by averaging locally trained model parameters received from data centers.

Despite the growing interest in FL, relatively few studies have applied it to the FPAD problem. In [17], Shao et al. introduced FedPAD, an FL framework for face presentation attack detection. In this setup, each data center trains a local

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3629353

model independently, and the central server aggregates their model updates to construct a global model. However, FedPAD primarily focuses on federated optimization without addressing severe domain heterogeneity across data centers, which may limit its generalization to unseen attack types.

Building on FedPAD, Shao et al. later proposed a test-time adaptive version [53], which improves generalizability by minimizing the Shannon entropy of model predictions during inference. Although this method enhances domain adaptation at test time, it still relies on affine transformations and batch normalization adjustments, which may not fully capture complex domain shifts encountered in real-world FPAD scenarios.

In [54], a more generalized FL framework for FPAD, named FedGPAD, was proposed. It separates domain-specific and domain-invariant features and combines them during decoding to improve robustness. However, FedGPAD's approach may not fully capture the complexities of domain differences in real-world FPAD data due to potential discrepancies in data distributions between data centers, which can challenge the model's ability to generalize across heterogeneous clients.

Another relevant approach is presented in [55], where FL is combined with multitask learning. Each data center is treated as an individual task, and a central server learns a global model by aggregating these local tasks. However, this method solely relies on handcrafted LBP features rather than deep learning-based features, limiting its scalability and representation capacity for more complex FPAD scenarios.

While FL has emerged as a promising method to address privacy concerns in FPAD, existing FL-based FPAD approaches still face important challenges. Most of these methods primarily emphasize privacy preservation and basic federated optimization but do not comprehensively address the effects of severe data heterogeneity across data centers. Consequently, global models often exhibit suboptimal generalization when deployed across diverse attack types and acquisition conditions. These limitations highlight the need for more robust aggregation and ensemble strategies capable of better handling the non-IID nature of FPAD data—precisely the gap that our proposed Fed-StackFPAD framework aims to address.

C. STACKING-BASED APPROACHES

The application of stacking-based ensemble learning in the centralized FPAD domain remains relatively limited. In [56], the authors address photo, video replay, and 3D mask attacks by leveraging both motion-based features—such as eye gaze, head pose, and eye blinking—and texture-based descriptors like histogram of oriented gradients (HoG). Additionally, they incorporate rPPG-based physiological features for liveness detection. These diverse feature representations are processed through a cascade of SVM classifiers, where the prediction probabilities from each SVM are passed to a final SVM classifier for the ultimate decision, forming a stacked ensemble architecture.

In [29], a stacking-based ensemble framework is proposed to improve generalization performance in FPAD tasks. The authors apply a series of data augmentation techniques to synthesize multiple spatiotemporal variants of the original images and train separate base models on each augmented subset. The features extracted from these base models are subsequently used to train a meta-classifier. This approach allows the ensemble to exploit the complementary characteristics of the base models, enhancing the model's ability to learn diverse and discriminative representations across datasets.

Despite the potential of stacking-based ensemble learning to enhance classification performance, its adoption within the FPAD domain remains relatively unexplored. Existing approaches often rely on centralized setups and synthetic data augmentations, which may limit their ability to generalize across real-world presentation attack variations. This highlights the opportunity to integrate stacking mechanisms within FL frameworks to simultaneously achieve privacy preservation and improved robustness against data heterogeneity.

III. METHODOLOGY

In our work, we design the first federated framework for FPAD that combines self-supervised learning as a pretext task for improved feature extraction and stacking-based meta-learning for model aggregation. The proposed framework significantly improves the capability of federated models to overcome the inherent data heterogeneity problem in FPAD. This section presents the problem formulation, details the proposed Fed-StackFPAD framework, and describes the supervised federated fine-tuning and the final ensemble learning process via stacking. To facilitate clarity and consistency in the methodological formulation, a summary of the key notations used throughout this section is provided in Table 1.

A. PROBLEM SETUP

In theory, training a deep learning model on the union of all available data yields better performance compared to using data from a single source. However, due to privacy regulations and bandwidth constraints, centralizing data from multiple centers is often impractical. To address this, we formulate the FPAD problem in a federated setting, aiming to collaboratively train a global model while keeping the data decentralized.

In our FL setup, as in [25], we assume there are N data centers, where each center $k \in \{1, .., N\}$ holds a private dataset D^k, and the full data distribution is expressed as $\mathcal{D} = \bigcup_{k=1}^{N} \mathcal{D}^k$. The global objective function to be minimized during federated training is defined as:

$$\arg\min_{w_G} \mathcal{L}(w_G) = \sum_{k=1}^{N} \frac{|\mathcal{D}^k|}{|\mathcal{D}|} \mathcal{L}_k(w_{G_k}) \tag{1}$$

where $\mathcal{L}_k(w_{G_k})$ denotes the local objective of data center k, and $|\mathcal{D}^k|$, $|\mathcal{D}|$ represent the number of samples in the local and global datasets, respectively.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3629353

TABLE 1. Summary of notations used in the Fed-StackFPAD methodology.

Symbol	Description
E_G^*	ViT encoder pretrained via MAE on ImageNet-22K
D_G^*	ViT decoder pretrained via MAE on ImageNet-22K
E_G	Federated global ViT model encoder (initialized from E_G^*)
L_G	Linear classifier for the federated global ViT model
M_G	Federated global ViT model ($E_G \circ L_G$)
w_G	Weights of the federated global ViT model M_G
E_{G_k}	Federated local ViT model encoder at data center k (initialized from E_G^*)
L_{G_k}	Linear classifier of the federated local ViT model at data center k
M_{G_k}	Federated local ViT model at data center k ($E_{G_k} \circ L_{G_k}$)
w_{G_k}	Weights of the federated local ViT model at data center k
E_k	Encoder of the data center-specific ViT model at data center k (initialized from E_G^*)
L_k	Linear classifier of the data center-specific ViT model at data center k
M_k	Data center-specific ViT model at data center k ($E_k \circ L_k$)
w_k	Weights of the data center-specific ViT model M_k
$\mathcal{F}$	Feature matrix used in stacking ensemble
N	Number of data centers

The local empirical loss over the distribution $\mathcal{D}^k$ is defined as:

$$\mathcal{L}_k(w_{G_k}) = \mathbb{E}_{x\sim\mathcal{D}^k}\left[\ell_k(w_{G_k}; \boldsymbol{x})\right] \tag{2}$$

where ℓ_k is the loss function of data center k, and w_{G_k} denotes the weights of the federated local model obtained by fine-tuning the global model weights w_G on the local dataset $\mathcal{D}^k$. Each image $\boldsymbol{x} \in \mathbb{R}^{H\times W\times C}$ is sampled from the local dataset, where H and W denote image dimensions and C is the number of channels.

In the context of FPAD, data collected across different data centers are inherently heterogeneous. Each dataset may vary significantly in terms of spoofing mediums, video capture devices, environmental conditions, and subject ethnicities, resulting in distinct underlying distributions, i.e., $\mathcal{D}^i \sim P_i(x, y)$ and $\mathcal{D}^j \sim P_j(x, y)$ for data centers $i \neq j$. This non-IID nature poses significant challenges for federated optimization and global model generalization. Therefore, addressing data heterogeneity is a central focus of our federated FPAD framework.

B. PROPOSED FEDERATED FRAMEWORK: FED-STACKFPAD

We propose Fed-StackFPAD, a novel FL framework designed to address the challenges of data heterogeneity and domain shift in face presentation attack detection (FPAD). The overall architecture of the proposed Fed-StackFPAD framework is illustrated in Figure 2. Each data center participates in training using its own private dataset, while the validation and test sets reside on the central federated server. The three main phases of the framework are described in detail in the following subsections and are also summarized in Algorithm 1.

1) Phase 1: Federated Global Model Fine-Tuning

At the beginning of the federated training phase, the central server initializes the global model (M_G) using a ViT encoder [26], pretrained in a self-supervised fashion with MAE on the large-scale ImageNet-22K dataset (E_G^*). This pretraining strategy enables the model to learn rich and transferable visual representations, improving generalization in downstream classification tasks. The global encoder (E_G) is then appended with a simple linear classifier (L_G), forming the complete federated global model, denoted as $M_G = E_G \circ L_G$. Then M_G is distributed to all data centers at the first communication round. Each data center initializes its federated local model as $M_{G_k} \leftarrow M_G$. Image representations are obtained via global average pooling and passed to the classifier, which is trained using cross-entropy loss on the data center's labeled data.

After local training epochs are completed, model updates at round t ($w_{G_k}^t$) are sent to the central server, where a weighted aggregation is performed to update the global model. In subsequent communication rounds, the updated global weights (w_G^{t+1}) are redistributed to data centers, and the training process continues iteratively. This setup allows each data center to personalize a shared high-capacity model while maintaining architectural consistency across the federation.

2) Phase 2: Data Center-Specific Model Fine-Tuning

After the federated global model fine-tuning phase, each data center independently trains a data center-specific model (M_k) outside the FL loop. The encoder (E_k) is initialized using the weights of the pretrained ViT-MAE encoder (E_G^*). A lightweight linear classifier (L_k) is then appended, forming the complete model as $M_k = E_k \circ L_k$. The model is subsequently optimized using only the local labeled training data available at data center k.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3629353

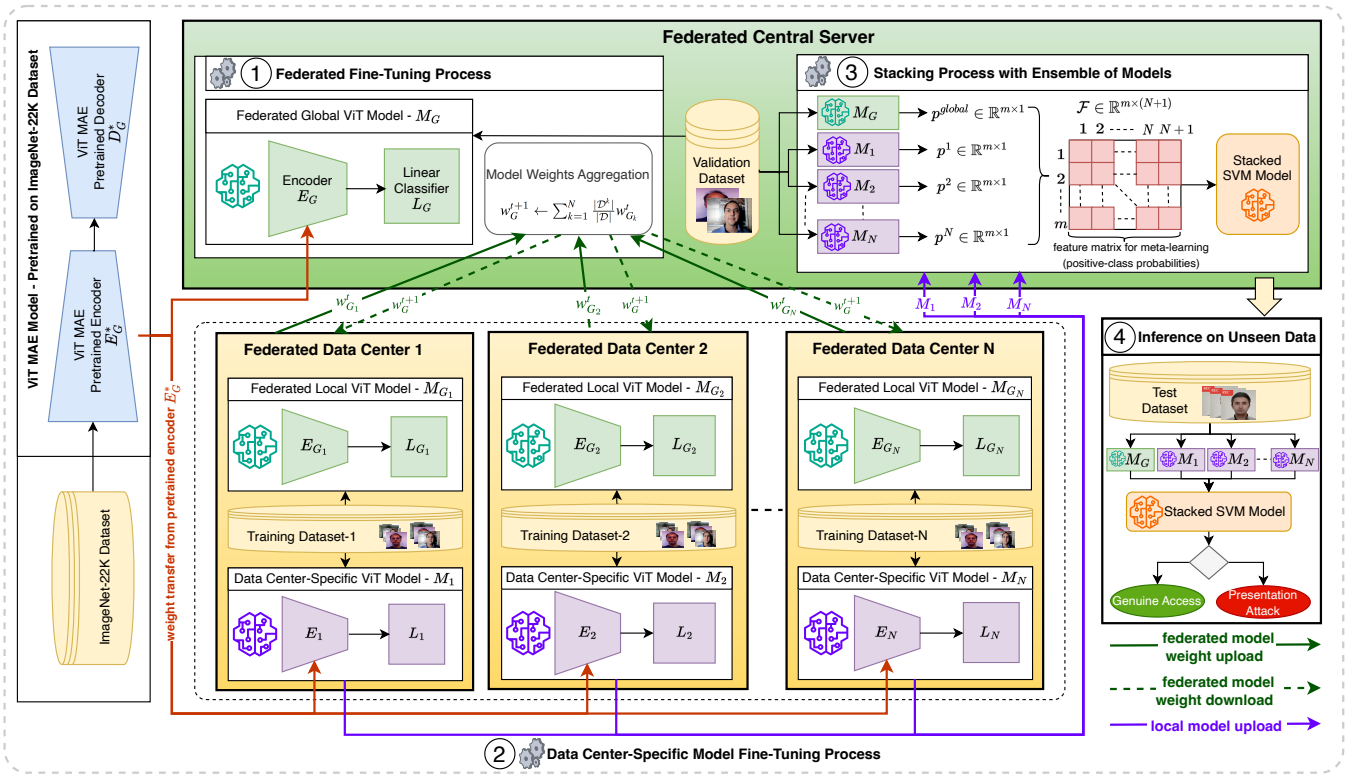


FIGURE 2. **The overall architecture of the proposed Fed-StackFPAD framework. The process consists of four main phases: (1) Federated Fine-Tuning, (2) Data-Center Specific Model Fine-Tuning, (3) Stacking-based Ensemble Learning, (4) Inference on Unseen Data.**
(1) In the first phase, a federated server initializes the global model (M_G) using the ViT encoder weights pretrained via ViT MAE model on the ImageNet-22K dataset (E_G^*). The global weights are distributed to multiple data centers, each of which fine-tunes the model (M_{G_k}) using its own local dataset. After each communication round, data centers send their updated model weights (w_{G_k}) to the server, where they are aggregated to form an updated global model.
(2) In the second phase, each data center independently trains a separate data center-specific model (M_k) initialized from the same pretrained encoder (E_G^*), independent from the FL stage.
(3) In the third phase, both the final global model (M_G) and all data center-specific local models (M_k) are evaluated on a common validation set located on the server. The positive-class (genuine access) probability outputs of each model are concatenated to form a feature matrix, which is then used to train an SVM classifier that learns to combine model predictions.
(4) During inference, the same set of models is applied to unseen test samples to generate prediction probabilities. These outputs are forwarded to the previously trained SVM stacking classifier, which performs the final decision-making step by classifying each input sample as either a presentation attack or a genuine access attempt.

3) Phase 3: Ensemble Learning with Stacking

Upon completion of the federated fine-tuning process (Phase 1), we obtain a global model M_G. Additionally, each data center retains its own locally fine-tuned model M_k, which has been trained on the corresponding local dataset $\mathcal{D}^k$ (Phase 2). For the stacking phase, we perform inference using both the global model (M_G) and all individual data center models (M_k) on a shared validation set $\mathcal{D}_{val}$, which is kept on the central server.

For a validation set containing m samples, each model (M_k) produces a vector of positive-class probabilities: $p^k = [p_1^k, \ldots, p_m^k]^\top \in \mathbb{R}^{m\times 1}$ for each data center $k \in \{1, \ldots, N\}$, and $p^{global} \in \mathbb{R}^{m\times 1}$ for the federated global model (M_G). In this context, the positive class represents *genuine access*, while the negative class corresponds to *presentation attacks*. These probability outputs are concatenated column-wise to form a feature matrix $\mathcal{F} \in \mathbb{R}^{m\times(N+1)}$, where each row corresponds to a validation sample and each column to a model's output.

$$\mathcal{F} = \begin{bmatrix} p_1^1 & p_1^2 & \cdots & p_1^N & p_1^{global} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ p_m^1 & p_m^2 & \cdots & p_m^N & p_m^{global} \end{bmatrix} \in \mathbb{R}^{m\times(N+1)} \quad (3)$$

Finally, an SVM classifier is trained on the feature matrix $\mathcal{F}$ to perform the final classification. This stacking-based meta-learning step integrates predictions from both the federated global model and the data center-specific local models, effectively capturing complementary decision patterns. To optimize the stacking classifier, we perform grid search cross-validation over a range of hyperparameters, exploring various kernel types such as linear and radial basis function (RBF), along with different regularization strengths. The optimal configuration is selected based on validation performance. This model selection process enables the final ensemble to generalize well across diverse validation samples while preserving the discriminative power of individual model outputs.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3629353

Algorithm 1 Proposed Fed-StackFPAD Framework

Input:
E_G^*: ViT encoder pretrained via MAE on ImageNet-22K
$\{\mathcal{D}^1, \ldots, \mathcal{D}^N\}$: Labeled datasets at N data centers
$\mathcal{D}_{val}$: Validation set on server for stacking
T: Number of federated training rounds
η: Learning rate
Output: Final SVM stacking classifier

Phase 1: Federated Global Model Fine-Tuning
Initialize global encoder $E_G \leftarrow E_G^*$
Append E_G with a linear classifier L_G to form global model M_G with weights w_G
Send w_G to all data centers
for each round $t = 1$ to T **do**
  Data Center-Side (in parallel):
  for each data center $k = 1$ to N **do**
    Receive global weights: $w_{G_k}^t \leftarrow w_G$
    for each mini-batch $b \in \mathcal{B}^k$ from $\mathcal{D}^k$ **do**
      $w_{G_k}^t \leftarrow w_{G_k}^t - \eta \nabla_k \ell_k(w_{G_k}^t, b)$
    end for
    Send updated weights $w_{G_k}^t$ to server
  end for
  Server-Side:
  Aggregate weights:
    $w_G^{t+1} \leftarrow \sum_{k=1}^{N} \frac{|\mathcal{D}^k|}{|\mathcal{D}|} w_{G_k}^t$
  Update global weights: $w_G \leftarrow w_G^{t+1}$
end for

Phase 2: Data Center-Specific Model Fine-Tuning (Data Center-Side)
for each data center $k = 1$ to N **do**
  Initialize encoder $E_k \leftarrow E_G^*$
  Append E_k with a linear classifier L_k to form local model M_k with weights w_k
  for each mini-batch $b \in \mathcal{B}^k$ from $\mathcal{D}^k$ **do**
    $w_k \leftarrow w_k - \eta \nabla_k \ell_k(w_k, b)$
  end for
  Send M_k to server
end for

Phase 3: Stacking (Server-Side)
for each data center-specific local model M_k ($k = 1, \ldots, N$) **do**
  Evaluate M_k on $\mathcal{D}_{val}$ to obtain prediction vector p^k
end for
Evaluate M_G on $\mathcal{D}_{val}$ to obtain p^{global}
Construct feature matrix $\mathcal{F} \in \mathbb{R}^{m \times (N+1)}$ using all predictions
Train SVM classifier on $\mathcal{F}$
return Trained SVM stacking model

IV. EXPERIMENTAL STUDY

This section presents the experimental setup and results for evaluating the proposed Fed-StackFPAD framework. We first describe the FPAD datasets used in our experiments, followed by the evaluation metrics and implementation details. We then report results for four experimental settings: single data center baselines, centralized upper bound, federated fine-tuning, and stacking-based ensemble learning. Finally, we compare our approach against state-of-the-art (SOTA) methods in the literature.

A. DATASETS

We evaluate the proposed framework on four widely used video-based face presentation attack detection (FPAD) datasets: Replay-Attack (I), Msu-Mfsd (M), Oulu-Npu (O), and Casia-Fasd (C). A summary of key dataset characteristics, including resolution, attack types, number of subjects, and fold structure, is provided in Table 2.

The **Replay-Attack** dataset [4] consists of 1300 videos captured from 50 subjects using a MacBook camera at a resolution of 320×240 pixels. It includes both printed photo and video replay attacks, recorded under controlled and adverse illumination settings. These conditions simulate office lighting and complex backgrounds, respectively. The dataset is organized into three folds: training, development, and test.

The **Msu-Mfsd** dataset [57] contains 280 videos recorded from 35 individuals using two different devices (MacBook Air and Nexus 5) at resolutions of 640×480 and 720×480 pixels. The spoofing attacks include mobile photo and mobile video replays recorded using external cameras and replayed on different screen types. The dataset is split into training and test folds.

The **Oulu-Npu** dataset [2] comprises 4950 videos from 55 subjects. Data were collected using the front-facing cameras of six smartphones with varying resolution settings. Recordings span three sessions with different environmental settings. It includes printed photo and video replay attacks generated with two printers and two display devices. The dataset is divided into training, development, and test folds.

The **Casia-Fasd** dataset [3] includes 600 videos (3 real and 9 spoof videos per subject) from 50 subjects of Asian descent. Recordings were made under three imaging conditions: low quality (long-used USB camera, 640×480), normal quality (new USB camera, 640×480), and high quality (Sony NEX-5, up to 1920×1080). Spoofing attacks consist of warped photo attacks (printed 1920×1080 images deliberately warped), cut photo attacks (eye regions removed to simulate blinking), and video replay attacks (genuine 1280×720 recordings replayed on an iPad screen, downscaled by the display). The dataset is split into training and test folds.

All four datasets are video-based and represent a wide range of spoofing mediums, acquisition conditions, and subject variability. In terms of demographic diversity, Casia-Fasd dataset contains only Asian subjects, whereas the other datasets involve participants from more diverse or mixed ethnic backgrounds. These variations make the selected datasets

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3629353

suitable for evaluating FPAD methods under diverse and cross-domain conditions, and they also serve as widely used benchmarks for fair comparison with prior work.

B. EVALUATION METRICS

We report results using three commonly adopted metrics in the FPAD literature: HTER, equal error rate (EER), and area under the receiver operating characteristic (ROC) curve (AUC), following dataset-specific protocols.

HTER is defined using the EER threshold θ as:

$$HTER(\theta) = \frac{FAR(\theta) + FRR(\theta)}{2} \tag{4}$$

where false acceptance rate (FAR) and false rejection rate (FRR) are computed as:

$$FAR = \frac{\text{\# of false acceptances}}{\text{\# of total attempts}} \tag{5}$$

$$FRR = \frac{\text{\# of false rejections}}{\text{\# of total attempts}} \tag{6}$$

The EER is the threshold point on the ROC curve where FAR equals FRR. HTER values range between 0 and 1, with lower values indicating better performance.

C. DATA PREPARATION

Following a data preparation setup similar to that used in [58], [59], we resize all frames to 224×224 pixels to fit the ViT model input. Additionally, to reduce redundancy and ensure temporal diversity, we sample one frame every five frames from each video. We obtain 18,600 frames from the Idiap Replay-Attack dataset, 6,757 from the Msu-Mfsd dataset, 49,557 from the Oulu-Npu dataset, and 9,126 from the Casia-Fasd dataset for model training.

D. IMPLEMENTATION DETAILS

For the Fed-StackFPAD framework, we adopt the federated training structure and baseline implementation from the public codebase of [25], which originally targets a medical imaging task. We adapt this framework to the FPAD domain by modifying the data handling and model architecture accordingly.

We simulate an FL setup with four data centers, each corresponding to one dataset. The validation and test sets are hosted on the central server. Following practices in similar studies [54], [60], when a dedicated validation set is unavailable, we use the training fold of the target dataset as the validation set and determine the required EER threshold for the calculation of the HTER metric.

Each federated data center initializes its ViT encoder using weights from a ViT-MAE model (MAE-ViT-Base/16) pretrained on the ImageNet-22K dataset [28]. Fine-tuning is performed in a supervised manner using locally labeled data. Each FL round consists of one local epoch, and the FL process runs for 100 rounds.

For data center-specific training, local models are also fine-tuned for 100 epochs using the same pretrained ViT encoder. These local models are later used in the stacking phase.

We perform hyperparameter tuning using the Optuna framework [61] to optimize model performance on the validation set. Specifically, we tune the following parameters: the base learning rate, the standard deviation for linear classifier weight initialization, the batch size, and the number of warmup epochs. The Optuna framework is employed exclusively during the offline training phase and has no impact on the computational cost of the inference phase.

The standard deviation for linear classifier weight initialization controls the spread of the initial weights assigned to the classifier head. Proper initialization helps ensure stable adaptation during fine-tuning on limited FPAD data without significantly perturbing the pretrained encoder. A warmup phase is applied to gradually increase the learning rate at the early stages of training, mitigating the risk of unstable updates. After the warmup period, we apply a cosine learning rate decay, smoothly annealing the learning rate from its base value to a final value, facilitating stable and efficient convergence. This approach supports smooth convergence across training epochs.

The hyperparameter search spaces are defined as follows:

- Base learning rate: log-uniform distribution in $[10^{-5}, 10^{-2}]$
- Standard deviation for linear classifier weight initialization: log-uniform distribution in $[10^{-5}, 10^{-2}]$
- Batch size: categorical choice from {8, 16, 32, 64, 128}
- Warmup epochs: categorical choice from {10, 20}

The best model found by Optuna is used in the stacking ensemble.

For the final SVM classifier, hyperparameter optimization is performed using 5-fold cross-validation and grid search on the validation set $\mathcal{D}_{val}$. The search space is defined as follows:

- Kernel type: categorical choice from {linear, RBF}
- Regularization parameter C: categorical choice from {0.001, 0.01, 0.1, 1, 10, 100, 1000}

The combination yielding the lowest classification error is selected for the final stacked model.

For evaluation, we apply the same frame sampling strategy as in training. For video-level prediction, majority voting is applied over frame-level predictions to classify each video as a genuine access or a presentation attack.

All experiments were conducted using PyTorch on a high-performance computing infrastructure. Training was performed on one NVIDIA Ada RTX 6000 GPU or four NVIDIA Tesla V100 GPUs. The federated training and model evaluation processes were distributed across these devices to efficiently handle the computational demands of ViT-based architectures and Optuna-based hyperparameter optimization.

E. EXPERIMENTAL RESULTS

To further understand the contributions of key components in our framework, we conduct a series of ablation studies, as detailed in the following subsection.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3629353

TABLE 2. Brief overview of the FPAD datasets used in the experiments.

Dataset	Resolution	Format	Attack Types	# of Subjects	# of Genuine Videos	# of Attack Videos	Fold Structure
Replay-Attack [4]	320 × 240	video	printed photo, replay video	50	300	1000	Training, Test Development
Msu-Mfsd [57]	640 × 480 720 × 480	video	mobile photo, mobile video	35	70	210	Training, Test
Casia-Fasd [3]	640 × 480 1920 × 1080	video	warped photo, cut photo, replay video	50	150	450	Training, Test
Oulu-Npu [2]	Six different resolutions	video	printed photo, replay video	55	990	3960	Training, Test Development

TABLE 3. Comparison of single data center, federated (with/without stacking), and centralized models across cross-dataset evaluation scenarios. (Notation: O&C&I indicates that O, C, and I are separate data centers in the federated setup; O+C+I denotes centralized training using all datasets). Best scores for HTER, EER, and AUC are shown in bold. Lower values for HTER, EER and higher values for AUC indicate better performance.

Method	Training Set (Data Center)	Test Set	HTER (%)	EER (%)	AUC (%)	Avg. HTER	Avg. EER	Avg. AUC
Single Data Center (Lowerbound)	O	M	8.83	0.00	97.79	12.72	10.53	93.39
	C	M	5.00	4.44	96.41			
	I	M	5.00	2.22	98.75			
	O	C	12.40	12.77	94.51			
	M	C	13.51	12.77	92.62			
	I	C	17.40	12.22	90.11			
	O	I	8.49	6.33	97.81			
	C	I	23.75	19.33	88.06			
	M	I	3.00	5.33	98.40			
	M	O	8.92	8.98	96.82			
	C	O	18.78	18.24	88.00			
	I	O	27.53	23.70	81.41			
Fed-StackFPAD (w/o stacking)	O&C&I	M	4.58	3.33	**99.35**	8.69	7.29	**97.47**
	O&M&I	C	12.96	10.55	**95.16**			
	O&C&M	I	9.50	9.00	**97.49**			
	I&C&M	O	7.70	6.29	**97.86**			
Fed-StackFPAD (w/ stacking)	O&C&I	M	**4.07**	2.12	95.92	**4.85**	3.43	95.14
	O&M&I	C	**7.91**	4.90	92.08			
	O&C&M	I	2.92	3.43	97.07			
	I&C&M	O	4.50	3.25	95.49			
Centralized (w/o stacking)	O+C+I	M	4.92	4.92	99.04	7.88	6.79	96.85
	O+M+I	C	11.60	7.43	92.70			
	O+C+M	I	8.09	8.90	97.24			
	I+C+M	O	6.89	5.90	98.43			
Centralized (w/ stacking) (Upperbound)	O+C+I	M	7.10	**1.50**	92.89	5.23	**2.38**	94.77
	O+M+I	C	8.07	**3.52**	91.92			
	O+C+M	I	**1.41**	**1.43**	98.58			
	I+C+M	O	**4.32**	**3.08**	95.67			

1) Ablation Studies

In Table 3, we present a comprehensive comparison of (i) our proposed Fed-StackFPAD method, evaluated both with and without stacking, (ii) a centralized upper-bound model evaluated with and without stacking, and (iii) individual single-data center models. All methods are built upon the same ViT-Base/16 architecture to ensure fair comparison. The results are reported under cross-dataset evaluation settings, where one dataset is held out as the test domain while the others are used for training.

The single data center baseline models are trained independently on individual datasets and tested on unseen target datasets. These represent a lower bound due to the limited

TABLE 4. Ablation study in the federated setup (without stacking): comparison of random initialization of ViT encoder weights vs. initialization with the weights of a self-supervised pretrained ViT-MAE encoder on ImageNet-22K. Metric: HTER (%). (Notation: O&C&I→M indicates that O, C, and I are separate data centers in the federated training setup, and M is the held-out test dataset). Lower HTER values indicate better performance.

Target Domain	Random Initialization	Pretrained ViT-MAE Initialization
O&C&I → M	14.73	**4.58**
O&M&I → C	24.97	**12.96**
O&C&M → I	24.62	**9.50**
I&C&M → O	22.73	**7.70**

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3629353

TABLE 5. SOTA comparison under single data center setting (Part 1). Metric: HTER (%). Each entry shows the HTER for a cross-domain setting, where *C*→*I* indicates that the model is trained on dataset *C* and tested on dataset *I*. Lower HTER values indicate better performance. Best result of each column is highlighted in bold.

Method	C→I	C→M	I→C	I→M	M→C	M→I	Avg.
ADDA [42], [46]	41.8	36.6	49.8	35.1	39.0	35.2	39.6
DRCN [46], [62]	44.4	27.6	48.9	42.0	28.9	36.8	38.1
DupGAN [44], [46]	42.4	33.4	46.5	36.2	27.1	35.4	36.8
ML-Net [46]	43.3	14.0	45.4	35.3	37.8	11.5	31.2
KSA [63]	39.3	15.1	**12.3**	33.3	**9.1**	34.9	24.0
ADA [45]	17.5	9.3	41.6	30.5	17.7	5.1	20.3
UDA [46]	15.6	9.0	34.2	29.0	16.8	3.0	17.9
KDPAD [47]	**3.5**	15.0	31.9	20.8	26.7	**2.9**	16.8
USDAN-Un [43]	16.0	9.2	30.2	25.8	13.3	3.4	16.3
NDA [8]	12.5	5.7	31.2	12.9	21.6	8.0	15.3
SDA-FAS++ [64]	10.0	9.2	18.5	22.9	10.0	3.5	12.4
FedPAD [17]	45.9	27.0	49.2	49.0	39.8	48.5	43.2
Ours	23.7	**5.0**	17.4	**5.0**	13.5	3.0	**11.2**

TABLE 6. SOTA comparison under single data center setting (Part 2). Metric: HTER (%). Each entry shows the HTER for a cross-domain setting, where *O*→*I* indicates that the model is trained on dataset *O* and tested on dataset *I*. Lower HTER values indicate better performance. Best result of each column is highlighted in bold.

Method	O→I	O→M	O→C	I→O	M→O	C→O	Avg.
ADDA [42], [46]	47.4	30.2	41.2	45.4	31.4	36.4	38.6
ML-Net [46]	40.1	32.8	38.7	44.2	31.6	34.3	36.9
UDA [46]	25.4	27.4	19.5	38.5	30.2	28.7	28.2
ADA [45]	26.8	31.5	19.8	29.6	31.2	29.1	28.0
NDA [8]	**7.0**	10.0	15.9	**26.8**	18.8	21.2	16.6
FedPAD [17]	42.2	41.2	31.3	46.9	29.8	33.9	37.5
Ours	8.4	**8.8**	**12.4**	27.5	**8.9**	**18.7**	**14.1**

diversity of training data.

The centralized model aggregates all training data into a single data center and is considered an upper bound under the assumption of full data sharing. In the centralized training setup, where all datasets are collected at a single data center, we apply the same stacking strategy as in Phase 3 of Algorithm 1. In this case, the centralized model plays the role of the global model (M_G) and is complemented by dataset-specific models trained on each individual dataset (analogous to the M_k models in the federated setup). This results in a similar feature matrix $\mathcal{F}$, but with the centralized model's outputs replacing those of the federated global model. Consequently, the same SVM-based meta-classifier is trained on $\mathcal{F}$, enabling effective aggregation of predictions from both centralized and dataset-specific models.

Our proposed Fed-StackFPAD framework achieves an average HTER of 8.69% without stacking, outperforming the single data center baselines (12.72%) and demonstrating the effectiveness of collaborative learning without direct data sharing. Although single data center models may perform competitively in several test scenarios, on the average they tend to suffer from limited generalization due to data scarcity and domain shift. By integrating stacking-based ensemble learning, we further reduce the average HTER to 4.85%, corresponding to a 3.84% absolute decrease in average HTER. This value surpasses the centralized upperbound (5.23%). These improvements are reflected in consistently lower average HTER and EER values across all test domains.

A noteworthy outcome of our study is the clear advantage of using a pretrained self-supervised ViT-MAE encoder over randomly initialized models. Initializing the encoder with weights obtained through self-supervised learning on the large-scale ImageNet-22K dataset yields significantly more informative feature representations for the downstream FPAD task. This benefit is particularly evident in FPAD scenarios characterized by scarce and homogeneous labeled data, which often hinders robust generalization to new and diverse attack domains. To further validate this observation, we conducted an ablation study in the federated fine-tuning setup without stacking, comparing the performance of models initialized with pretrained weights versus random initialization. As shown in Table 4, models initialized with the self-supervised ViT-MAE encoder weights consistently outperformed their randomly initialized counterparts across all target domains, confirming the transferability benefits of self-supervised pretraining for cross-domain FPAD.

F. COMPARISON WITH STATE-OF-THE-ART METHODS

To further validate the effectiveness of our Fed-StackFPAD framework, we compare its performance against a wide range of existing FPAD methods under the single-dataset, centralized, and FL settings. To ensure fairness in the SOTA comparisons, we only include studies that adhere to standardized FPAD evaluation protocols, particularly those employing a

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3629353

TABLE 7. Comparison of centralized models across four cross-dataset evaluation scenarios. (Notation: O+C+I→M indicates that all datasets are centralized for training, and M is used as the test set). Best scores are highlighted in bold.

Method	O+C+I → M		O+M+I → C		O+C+M → I		I+C+M → O	
	HTER (%)	AUC (%)	HTER (%)	AUC (%)	HTER (%)	AUC (%)	HTER (%)	AUC (%)
Binary CNN [38]	29.25	82.87	34.88	71.94	34.47	65.88	29.61	77.54
Color Texture [65]	28.09	78.47	30.58	76.89	40.40	62.78	63.59	32.71
LBP-TOP [66]	36.90	70.80	42.60	61.05	49.45	49.54	53.15	44.09
Auxiliary [67]	22.72	85.88	33.52	73.15	29.14	71.69	30.17	77.61
SSDG-R [68]	7.38	97.17	10.44	95.94	11.71	96.59	15.61	91.54
SSAN [69]	6.67	98.75	10.00	96.67	8.88	96.79	13.72	93.63
PatchNet [70]	7.10	98.46	11.33	94.58	13.40	95.67	11.82	95.07
SDA [71]	15.40	91.80	24.50	84.40	15.60	90.10	23.10	84.30
FedPAD [17]	21.80	90.96	29.46	76.29	30.57	72.21	27.22	82.21
NDA [8]	4.29	99.27	13.22	92.89	5.50	96.90	17.45	89.93
SDA-FAS [48]	5.00	97.96	2.40	99.72	2.62	99.48	5.07	99.01
SDA-FAS++ [48]	3.75	98.92	**1.11**	99.82	**0.25**	**99.99**	3.23	99.65
DG-ES [29]	**1.20**	**99.98**	1.37	**99.99**	5.40	99.97	**2.08**	**99.77**
Ours (w/o stacking)	4.92	99.04	11.60	92.70	8.09	97.24	6.89	98.43
Ours (w/ stacking)	7.10	92.89	8.07	91.92	1.41	98.58	4.32	95.67

TABLE 8. Comparison with current state-of-the-art federated FPAD frameworks. (Notation: O&C&I→M indicates that O, C, and I are separate data centers in the federated training setup, and M is the held-out test dataset). Best scores for HTER, EER, and AUC are shown in bold. Lower values for HTER, EER and higher values for AUC indicate better performance.

Method	O&C&I → M			O&M&I → C			O&C&M → I			I&C&M → O		
	HTER	EER	AUC	HTER	EER	AUC	HTER	EER	AUC	HTER	EER	AUC
FedMA [54], [72]	29.68	25.85	84.68	31.17	30.78	78.81	24.82	26.75	73.05	30.95	25.15	84.28
FedPAD [17]	19.45	17.43	90.24	42.27	36.95	70.49	32.53	26.54	73.58	34.44	34.45	71.74
FedGPAD [54]	12.73	13.36	91.25	28.69	27.55	80.58	10.97	11.11	95.34	21.95	17.91	89.85
Fed-StackFPAD	**4.07**	**2.12**	**95.92**	**7.91**	**4.90**	**92.08**	**2.92**	**3.43**	**97.07**	**4.50**	**3.25**	**95.49**

separate validation set to compute the EER threshold required for HTER calculation.

1) Single-Dataset Comparisons

In Tables 5 and 6, we present the cross-dataset generalization results in single data center scenarios. Our model consistently achieves lower HTER values compared to domain adaptation-based methods such as ADDA [42], DRCN [62], DupGAN [44], and ML-Net [46], as well as deep learning-based approaches like KDPAD [47], SDA-FAS++ [48], and NDA [8]. Notably, our method achieves the best average performance across both benchmark tables, with average HTERs of 11.2% and 14.1% as shown in the last columns of Table 5 and Table 6, respectively.

Here, we use the notation $C \rightarrow M$ to denote training on dataset C and testing on held-out dataset M. In particular, our model yields the lowest HTERs in six representative cross-domain configurations: 5.0% for C→M, 5.0% for I→M, 8.8% for O→M, 12.4% for O→C, 8.9% for M→O, and 18.7% for C→O. This consistent performance gain across six diverse cross-domain setups can be attributed to the transferability of the ViT-based encoder pretrained with masked autoencoders, which facilitates learning domain-invariant features critical for generalization in single-client FPAD scenarios.

2) Centralized Comparisons

In addition to federated and single data center baselines, we evaluate our method under a centralized training setup, where all training data is aggregated into a single repository. Table 7 presents the performance of our model in this configuration compared to existing centralized FPAD methods. Both stacking-based and non-stacking (direct) configurations of our method are included. In this centralized setup, all source datasets are pooled into a single data center, representing an upper-bound scenario compared to federated learning. For example, in the O+C+I→M setting, our stacking-based model leverages predictions from both dataset-specific models and the centralized global model during the ensemble process, whereas the non-stacked model directly trains on the combined data. While our method does not achieve the best scores in the centralized setup, it remains highly competitive across all domains. For instance, in the O+C+I→M scenario, our non-stacked centralized model attains an HTER of 4.92%, which is close to 3.75% achieved by the second best-performing SDA-FAS++ [48] model. However, when examined under the single data center configuration (Table 5), SDA-FAS++ demonstrates higher HTER in certain scenarios—for example, 22.9% in the I→M case—while

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3629353

our method achieves 5.0% in the same setting. Similarly, in the O+C+M→I configuration, our stacking-based model achieves an HTER of 1.41%, which is slightly higher than the 0.25% achieved by the best-performing SDA-FAS++ in the centralized setting. In the single data center configuration (Table 5), SDA-FAS++ demonstrates a higher HTER of 3.5% in the M→I case, while our method achieves 3.0% in the same setting. These observations suggest that although SDA-FAS++ performs well in centralized training, our approach offers more consistent generalization in decentralized and cross-domain conditions, which are critical for real-world FPAD deployments.

3) Federated Comparisons

Table 8 compares the performance of our federated approach against existing federated FPAD frameworks, including FedPAD [17], FedGPAD [54], and FedMA [72]. Fed-StackFPAD with stacking achieves the best overall performance across all four held-out target datasets, demonstrating the benefit of combining both global and data center-specific representations through ensemble learning. Specifically, our method achieves HTERs of 4.07% (O&C&I→M), 7.91% (O&M&I→C), 2.92% (O&C&M→I), and 4.50% (I&C&M→O), outperforming the second-best method (FedGPAD) by absolute margins of 8.66%, 20.78%, 8.05%, and 17.45%, respectively. These results highlight the effectiveness of our stacking-based strategy in addressing domain shift and data heterogeneity problems in federated FPAD settings.

V. DISCUSSION

The experimental results demonstrate that the proposed Fed-StackFPAD framework effectively addresses the limitations of both centralized and single data center FPAD systems. By leveraging FL, the framework allows collaborative model training without requiring raw data exchange across data centers, preserving privacy and reducing communication costs.

While individual single data center models may achieve competitive results in several cross-domain setups (e.g., C→M, I→M, M→I) as shown in Table 3, their performance deteriorates in others (e.g., C→I, I→O) due to limited data diversity. In contrast, the federated approach promotes generalization by aggregating knowledge from heterogeneous sources. Notably, our stacking-based ensemble strategy further enhances the model's ability to capture complementary patterns in data center-specific and global predictions, resulting in improved classification accuracy across multiple domains. This is consistent with the principle of stacked generalization [73], where the outputs of different models are combined via a meta-classifier to leverage their individual strengths. We evaluated several candidate meta-classifiers for the stacking layer (including XGBoost, Random Forest, etc.). Among these, SVM consistently achieved the best average HTER on the validation set. Given its ability to generalize well from structured softmax input vectors and its efficiency in low-dimensional settings, SVM was selected as the most suitable meta-classifier. Although more advanced meta-learning approaches might capture richer interactions between models, we chose a simpler and robust solution that provides strong empirical results and is easier to apply in practice. We also observe that the stacking approach reduces HTER on every held-out dataset (Table 3), confirming its ability to mitigate non-IID effects while retaining the privacy and scalability advantages of FL.

However, we observe a slight reduction in AUC for the ensemble model compared to the non-stacked federated and centralized models. As shown in the ROC curves (Figure 3), this may stem from the fact that the SVM classifier used in the stacking layer does not directly optimize AUC, but instead focuses on minimizing error rate through hard-margin decision boundaries. As discussed in [74], classifiers trained to optimize for accuracy—especially in imbalanced data scenarios—can underperform on ranking-based metrics such as AUC, unless AUC maximization is explicitly incorporated into the learning objective. This suggests that the ensemble model's performance trade-off in AUC may be attributed to the intrinsic limitations of the stacking classifier.

To qualitatively assess the discriminative capability of our proposed Fed-StackFPAD framework, we visualize the learned feature embeddings using t-distributed Stochastic Neighbor Embedding (t-SNE) [75]. Figure 4 presents the two-dimensional projection of feature representations derived from the penultimate layer of the model trained under the O&C&I→M setup, where datasets O, C, and I are used as federated source domains and M is held out as the target. Each class (genuine or attack) contributes 200 randomly sampled frames per domain to the visualization. As shown in Figure 4 (a), the model successfully separates the genuine and attack samples into distinct clusters, reflecting strong intra-class compactness and inter-class separability. Figure 4 (b) further illustrates the domain-wise distribution of the same samples, showing consistent separation regardless of origin of data. These visual patterns demonstrate the robustness of Fed-StackFPAD in learning transferable representations that generalize well across both classes and domains.

Overall, the proposed framework achieves a favorable balance between performance, scalability, and privacy. Its ability to incorporate new data centers without central data sharing, distribute computational load, and adapt to varying local data characteristics makes it a promising candidate for real-world FPAD deployments in distributed environments.

VI. CONCLUSION

In this study, we proposed Fed-StackFPAD, an FL framework tailored for FPAD under decentralized data availability. Our approach combines a ViT encoder pretrained in a self-supervised manner using a masked autoencoder with supervised federated fine-tuning and integrates a stacking-based ensemble layer to fuse predictions from global and data center-specific models. This architecture enables privacy-preserving collaborative learning while enhancing robustness against domain shifts.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3629353

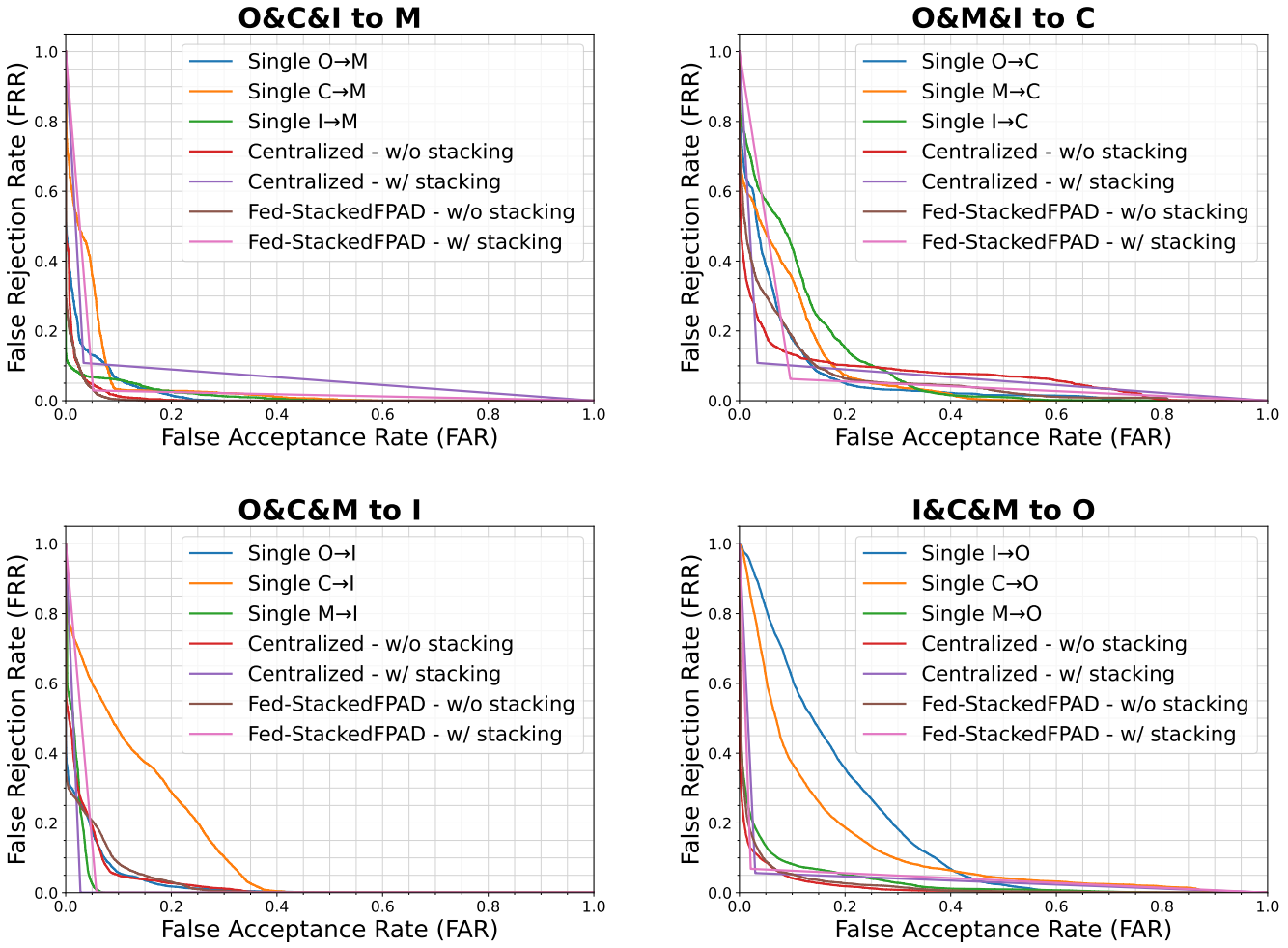
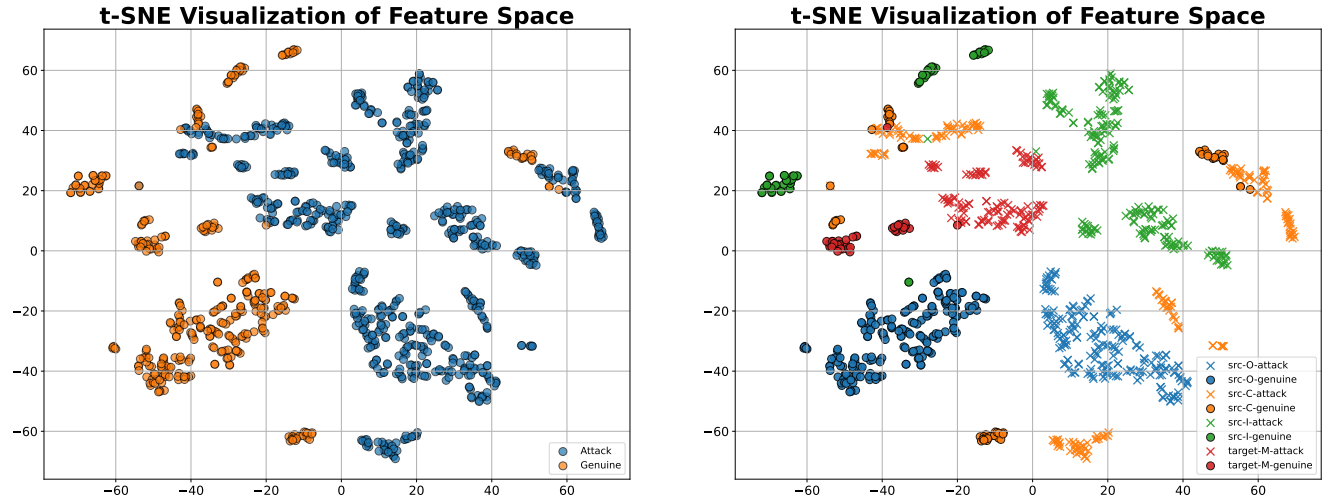


FIGURE 3. **ROC curves of the single data center baselines, centralized model, global federated model, and stacking-based ensemble models for 4 different training and test set combinations.**



(a) **2-class separation:** t-SNE visualization of attack vs. genuine samples using Fed-StackFPAD model embeddings.

(b) **8-class separation:** t-SNE visualization of domain-specific attack/genuine distributions (O, C, I: source; M: target) under the O&C&I→M setup.

FIGURE 4. **t-SNE visualization of feature embeddings from the Fed-StackFPAD model trained under the federated O&C&I→M setup.**

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3629353

Comprehensive evaluations across four public FPAD datasets demonstrate that Fed-StackFPAD consistently outperforms other FL baselines in terms of both HTER and EER. Notably, our method achieves HTER values of 4.07%, 7.91%, 2.92%, and 4.50% across four held-out test domains, while the best-performing alternative federated method (FedGPAD) yields 12.73%, 28.69%, 10.97%, and 21.95% in the same settings. These results correspond to absolute HTER reductions of up to 20.78% and highlight the effectiveness of our stacking-based ensemble mechanism in mitigating domain shift. Furthermore, our framework preserves data privacy and exhibits strong generalization in non-IID cross-domain scenarios.

Future Work. As future work, we plan to extend the ensemble stage by incorporating heterogeneous prediction models, including temporal models that can explicitly capture dynamic visual cues such as eye blinking, facial muscle movement, and subtle motion inconsistencies. Since our current approach processes individual frames, integrating temporal information remains a valuable direction for further improving spoof detection performance. For instance, models utilizing remote photoplethysmography (rPPG), which inherently relies on temporal patterns, will be considered as complementary modalities. In scenarios where datasets support temporal annotations, we also aim to integrate physiological features into the ensemble decision-making process. Additionally, we intend to broaden the evaluation to include more diverse FPAD datasets to further assess the scalability and generalizability of our approach.

Moreover, future work will explore the robustness of the proposed framework against adversarial threats, which are particularly relevant in federated learning scenarios. This includes potential vulnerabilities such as poisoning attacks, backdoor attacks, or adversarial perturbations, both at the model and data level. Investigating defense mechanisms like robust aggregation schemes, anomaly detection, and adversarial training strategies will help strengthen the trustworthiness and real-world applicability of the system.

Limitations. While the proposed Fed-StackFPAD framework demonstrates strong performance in cross-domain FPAD scenarios, it has several practical limitations.

First, the use of Vision Transformer (ViT) architectures pretrained on large-scale datasets such as ImageNet-22K, combined with federated training across multiple clients and an additional stacking ensemble layer, results in considerable computational and memory requirements. These factors may hinder real-time applicability and deployment in edge or mobile devices, especially in latency-sensitive scenarios. However, the proposed approach is primarily designed for server-side inference settings, where computational resources are more readily available and higher model complexity can be tolerated in exchange for improved accuracy. Typical deployment scenarios include biometric verification at border control checkpoints, remote identity verification in financial services, biometric access control in secure facilities, and online proctoring environments. In such contexts, inference is performed on backend servers equipped with GPUs, making the framework more feasible. Nevertheless, future work will explore lightweight ViT variants, model compression, and distillation-based techniques to improve efficiency and facilitate potential deployment on resource-constrained edge devices.

Second, the hyperparameter optimization process conducted using tools such as Optuna requires significant GPU resources and extensive tuning to achieve optimal performance. This process must also be repeated when new data centers with different data distributions are added to the federated system, which can limit the scalability and adaptability of the framework in dynamic real-world settings.

REFERENCES

[1] D. Wang, J. Guo, Q. Shao, H. He, Z. Chen, C. Xiao, A. Liu, S. Escalera, H. J. Escalante, Z. Lei, J. Wan, and J. Deng, "Wild face anti-spoofing challenge 2023: Benchmark and results," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 6 2023, pp. 6380–6391. [Online]. Available: https://ieeexplore.ieee.org/document/10209048/

[2] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng, and A. Hadid, "Oulu-npu: A mobile face presentation attack database with real-world variations," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE, 5 2017, pp. 612–618. [Online]. Available: http://ieeexplore.ieee.org/document/7961798/

[3] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li, "A face antispoofing database with diverse attacks," in *2012 5th IAPR International Conference on Biometrics (ICB)*. IEEE, 3 2012, pp. 26–31. [Online]. Available: http://ieeexplore.ieee.org/document/6199754/

[4] I. Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *2012 BIOSIG - Proceedings of the International Conference of Biometrics Special Interest Group (BIOSIG)*. IEEE, 8 2012. [Online]. Available: https://ieeexplore.ieee.org/document/6313548

[5] N. Erdogmus and S. Marcel, "Spoofing in 2d face recognition with 3d masks and anti-spoofing with kinect," in *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. IEEE, 9 2013, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/6712688/

[6] A. Agarwal, D. Yadav, N. Kohli, R. Singh, M. Vatsa, and A. Noore, "Face presentation attack with latex masks in multispectral videos," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, vol. 2017-July. IEEE, 7 2017, pp. 275–283. [Online]. Available: http://ieeexplore.ieee.org/document/8014774/

[7] M. Huber, M. Fang, F. Boutros, and N. Damer, "Are explainability tools gender biased? a case study on face presentation attack detection," in *2023 31st European Signal Processing Conference (EUSIPCO)*. IEEE, 9 2023, pp. 945–949. [Online]. Available: https://ieeexplore.ieee.org/document/10289865/

[8] W. Wang, P. Liu, H. Zheng, R. Ying, and F. Wen, "Domain generalization for face anti-spoofing via negative data augmentation," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 2333–2344, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10098629/

[9] S.-Q. Liu, X. Lan, and P. C. Yuen, "Learning temporal similarity of remote photoplethysmography for fast 3d mask face presentation attack detection," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 3195–3210, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9852272/

[10] N. Thapa, M. Chaudhari, and K. Roy, "Presentation attack detection: an analysis of spoofing in the wild (SiW) dataset using deep learning models," *Discover Artificial Intelligence*, vol. 3, p. 30, 8 2023. [Online]. Available: https://link.springer.com/10.1007/s44163-023-00077-1

[11] Z. Chen, Y. Lu, X. Deng, J. Meng, S. Zhang, and L. Cao, "Self-paced partial domain-aware learning for face anti-spoofing," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6 2023, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/10095730/

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3629353

[12] C.-L. Chou, "Presentation attack detection based on score level fusion and challenge-response technique," *The Journal of Supercomputing*, vol. 77, pp. 4681–4697, 5 2021. [Online]. Available: https://link.springer.com/10.1007/s11227-020-03461-1

[13] D. Pasmino, C. Aravena, J. E. Tapia, and C. Busch, "Flickr-pad: New face high-resolution presentation attack detection database," in *2023 11th International Workshop on Biometrics and Forensics (IWBF)*. IEEE, 4 2023, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/10157771/

[14] D. Deb and A. K. Jain, "Look locally infer globally: A generalizable face anti-spoofing approach," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1143–1157, 6 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9218954/

[15] Z. Wang, Z. Yu, X. Wang, Y. Qin, J. Li, C. Zhao, X. Liu, and Z. Lei, "Consistency regularization for deep face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1127–1140, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10012352/

[16] J. Xie, Y. Luo, and J. Lu, "A random-patch based defense strategy against physical attacks for face recognition systems," *arXiv preprint arXiv:2304.07822*, 4 2023. [Online]. Available: http://arxiv.org/abs/2304.07822

[17] R. Shao, P. Perera, P. C. Yuen, and V. M. Patel, "Federated face presentation attack detection," *arXiv preprint arXiv:2005.14638*, 5 2020. [Online]. Available: http://arxiv.org/abs/2005.14638

[18] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, pp. 3347–3366, 4 2023.

[19] J. Wen, Z. Zhang, Y. Lan, Z. Cui, J. Cai, and W. Zhang, "A survey on federated learning: challenges and applications," *International Journal of Machine Learning and Cybernetics*, vol. 14, pp. 513–535, 2 2023.

[20] W. Guo, F. Zhuang, X. Zhang, Y. Tong, and J. Dong, "A comprehensive survey of federated transfer learning: challenges, methods and applications," *Frontiers of Computer Science*, vol. 18, no. 6, p. 186356, 2024.

[21] H. Guan, P.-T. Yap, A. Bozoki, and M. Liu, "Federated learning for medical image analysis: A survey," *Pattern recognition*, vol. 151, p. 110424, 2024.

[22] K. Malempati, P. R. Kotigari, S. T. Kandukuri, and M. Eirinaki, "Federated learning on recommender systems," in *2024 IEEE International Conference on Big Data (BigData)*, 2024, pp. 7085–7094.

[23] Y. J. Wong, M.-L. Tham, B.-H. Kwan, E. M. A. Gnanamuthu, and Y. Owada, "An optimized multi-task learning model for disaster classification and victim detection in federated learning environments," *IEEE Access*, vol. 10, pp. 115 930–115 944, 2022.

[24] B. Wang, A. Li, M. Pang, H. Li, and Y. Chen, "Graphfl: A federated learning framework for semi-supervised node classification on graphs," in *2022 IEEE International Conference on Data Mining (ICDM)*, 2022, pp. 498–507.

[25] R. Yan, L. Qu, Q. Wei, S.-C. Huang, L. Shen, D. L. Rubin, L. Xing, and Y. Zhou, "Label-efficient self-supervised federated learning for tackling data heterogeneity in medical imaging," *IEEE Transactions on Medical Imaging*, vol. 42, pp. 1932–1943, 7 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10004993/

[26] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 10 2020. [Online]. Available: http://arxiv.org/abs/2010.11929

[27] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[28] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "Imagenet-21k pretraining for the masses," *arXiv preprint arXiv:2104.10972*, 2021.

[29] U. Muhammad, J. Laaksonen, D. Romaissa Beddiar, and M. Oussalah, "Domain generalization via ensemble stacking for face presentation attack detection," *International Journal of Computer Vision*, vol. 132, no. 12, pp. 5759–5782, 2024.

[30] T. de Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel, "Can face anti-spoofing countermeasures work in a real world scenario?" in *2013 International Conference on Biometrics (ICB)*, 2013, pp. 1–8.

[31] J. Yang, Z. Lei, S. Liao, and S. Z. Li, "Face liveness detection with component dependent descriptor," in *2013 International Conference on Biometrics (ICB)*. IEEE, 6 2013, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/6612955/

[32] D. Sharma and A. Selwal, "A survey on face presentation attack detection mechanisms: hitherto and future perspectives," *Multimedia Systems*, vol. 29, pp. 1527–1577, 6 2023. [Online]. Available: https://link.springer.com/10.1007/s00530-023-01070-5

[33] Z. Yu, Y. Qin, X. Li, C. Zhao, Z. Lei, and G. Zhao, "Deep learning for face anti-spoofing: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 1–22, 5 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9925105/

[34] A. George and S. Marcel, "On the effectiveness of vision transformers for zero-shot face anti-spoofing," in *2021 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 8 2021, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/9484333/

[35] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu, "Face anti-spoofing using patch and depth-based cnns," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 10 2017, pp. 319–328. [Online]. Available: http://ieeexplore.ieee.org/document/8272713/

[36] R. Shao, X. Lan, and P. C. Yuen, "Deep convolutional dynamic texture learning with adaptive channel-discriminability for 3d mask face anti-spoofing," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 10 2017, pp. 748–755. [Online]. Available: http://ieeexplore.ieee.org/document/8272765/

[37] Y. Liu and X. Liu, "Spoof trace disentanglement for generic face anti-spoofing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 1–1, 3 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9779478/

[38] J. Yang, Z. Lei, and S. Z. Li, "Learn convolutional neural network for face anti-spoofing," *arXiv preprint arXiv:1408.5601*, 8 2014. [Online]. Available: http://arxiv.org/abs/1408.5601

[39] R. Cai, H. Li, S. Wang, C. Chen, and A. C. Kot, "Drl-fas: A novel framework based on deep reinforcement learning for face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 937–951, 9 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9205636/

[40] R. Shao, X. Lan, J. Li, and P. C. Yuen, "Multi-adversarial discriminative deep domain generalization for face presentation attack detection," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2019-June. IEEE, 6 2019, pp. 10 015–10 023. [Online]. Available: https://ieeexplore.ieee.org/document/8953226/

[41] Z. Yu, C. Zhao, Z. Wang, Y. Qin, Z. Su, X. Li, F. Zhou, and G. Zhao, "Searching central difference convolutional networks for face anti-spoofing," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 6 2020, pp. 5294–5304. [Online]. Available: https://ieeexplore.ieee.org/document/9156660/

[42] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7167–7176.

[43] Y. Jia, J. Zhang, S. Shan, and X. Chen, "Unified unsupervised and semi-supervised domain adaptation network for cross-scenario face anti-spoofing," *Pattern Recognition*, vol. 115, p. 107888, 7 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0031320321000753

[44] L. Hu, M. Kan, S. Shan, and X. Chen, "Duplex generative adversarial network for unsupervised domain adaptation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1498–1507.

[45] G. Wang, H. Han, S. Shan, and X. Chen, "Improving cross-database face presentation attack detection via adversarial domain adaptation," in *2019 International Conference on Biometrics (ICB)*, 2019, pp. 1–8.

[46] G. Wang, H. Han, S. Shan, and X. Chen, "Unsupervised adversarial domain adaptation for cross-domain face presentation attack detection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 56–69, 2021.

[47] Z. Li, R. Cai, H. Li, K.-Y. Lam, Y. Hu, and A. C. Kot, "One-class knowledge distillation for face presentation attack detection," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2137–2150, 2022.

[48] Y. Liu, Y. Chen, W. Dai, M. Gou, C.-T. Huang, and H. Xiong, *Source-Free Domain Adaptation with Contrastive Domain Alignment and Self-supervised Exploration for Face Anti-spoofing*. Springer Nature Switzerland, 2022, pp. 511–528. [Online]. Available: https://link.springer.com/10.1007/978-3-031-19775-8_30

[49] S. Abdulrahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet of Things Journal*, vol. 8, pp. 5476–5497, 4 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9220780/

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3629353

[50] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 3 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0950705121000381

[51] D. H. Mahlool and M. H. Abed, *A Comprehensive Survey on Federated Learning: Concept and Applications*. Springer, 2022, pp. 539–553. [Online]. Available: https://link.springer.com/10.1007/978-981-19-2069-1_37

[52] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282, 2 2016. [Online]. Available: http://arxiv.org/abs/1602.05629

[53] R. Shao, B. Zhang, P. C. Yuen, and V. M. Patel, "Federated test-time adaptive face presentation attack detection with dual-phase privacy preservation," in *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*. IEEE, 12 2021, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/9666952/

[54] R. Shao, P. Perera, P. C. Yuen, and V. M. Patel, "Federated generalized face presentation attack detection," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9780603/

[55] Y. Chen, L. Chen, C. Hong, and X. Wang, "Federated multitask learning with manifold regularization for face spoof attack detection," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–10, 6 2022. [Online]. Available: https://www.hindawi.com/journals/mpe/2022/7759410/

[56] M. F. Gündoğar and Ç. E. Erdem, "Presentation attack detection for face recognition using remote photoplethysmography and cascaded fusion," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 29, pp. 3240–3258, 11 2021. [Online]. Available: https://journals.tubitak.gov.tr/elektrik/vol29/iss7/22

[57] D. Wen, H. Han, and A. K. Jain, "Face spoof detection with image distortion analysis," *IEEE Transactions on Information Forensics and Security*, vol. 10, pp. 746–761, 4 2015. [Online]. Available: http://ieeexplore.ieee.org/document/7031384/

[58] U. Muhammad, Z. Yu, and J. Komulainen, "Self-supervised 2d face presentation attack detection via temporal sequence sampling," *Pattern Recognition Letters*, vol. 156, pp. 15–22, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865522000605

[59] U. Muhammad and M. Oussalah, "Self-supervised face presentation attack detection with dynamic grayscale snippets," in *2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG)*. IEEE, 1 2023, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/10042547/

[60] Y. Liu, J. Stehouwer, A. Jourabloo, and X. Liu, "Deep tree learning for zero-shot face anti-spoofing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[61] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[62] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*. Springer, 2016, pp. 597–613.

[63] H. Li, W. Li, H. Cao, S. Wang, F. Huang, and A. C. Kot, "Unsupervised domain adaptation for face anti-spoofing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1794–1809, 2018.

[64] Y. Liu, Y. Chen, W. Dai, M. Gou, C.-T. Huang, and H. Xiong, "Source-free domain adaptation with domain generalized pretraining for face anti-spoofing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 8, pp. 5430–5448, 2024.

[65] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face spoofing detection using colour texture analysis," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1818–1830, 2016.

[66] T. Pereira, J. Komulainen, A. Anjos, J. De Martino, A. Hadid, M. Pietikäinen, and S. Marcel, "Face liveness detection using dynamic texture," *EURASIP Journal on Image and Video Processing*, vol. 2014, p. 2, 01 2014.

[67] Y. Liu, A. Jourabloo, and X. Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 6 2018, pp. 389–398. [Online]. Available: https://ieeexplore.ieee.org/document/8578146/

[68] Y. Jia, J. Zhang, S. Shan, and X. Chen, "Single-side domain generalization for face anti-spoofing," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8481–8490.

[69] Z. Wang, Z. Wang, Z. Yu, W. Deng, J. Li, T. Gao, and Z. Wang, "Domain generalization via shuffled style assembly for face anti-spoofing," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 4113–4123.

[70] C.-Y. Wang, Y.-D. Lu, S.-T. Yang, and S.-H. Lai, "Patchnet: A simple face anti-spoofing framework via fine-grained patch recognition," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 6 2022, pp. 20 249–20 258. [Online]. Available: https://ieeexplore.ieee.org/document/9879125/

[71] J. Wang, J. Zhang, Y. Bian, Y. Cai, C. Wang, and S. Pu, "Self-domain adaptation for face anti-spoofing," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 4, 2021, pp. 2746–2754.

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

[73] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608005800231

[74] E. LeDell, M. Petersen, and M. J. van der Laan, "Auc-maximizing ensembles through metalearning," *International Journal of Biostatistics*, vol. 12, no. 1, pp. 203–218, 2016.

[75] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008. [Online]. Available: http://jmlr.org/papers/v9/vandermaaten08a.html



MEHMET FATİH GÜNDOĞAR received his B.Sc. degree in Computer Engineering from Ege University, İzmir, Türkiye, in 2006, and his M.Sc. degree in Computer Engineering from Marmara University, İstanbul, Türkiye, in 2020. He is currently pursuing his Ph.D. degree in Computer Engineering at Marmara University, which he started in 2020. His research interests include deep learning, federated learning, and machine learning. He has more than 18 years of professional experience in the private sector, primarily in the telecommunications, finance, and insurance industries. Throughout his career, he has led and contributed to numerous projects involving artificial intelligence, digital transformation, and data analytics.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3629353



ÇİĞDEM EROĞLU ERDEM received her B.S. and M.Sc. degrees in Electrical and Electronics Engineering from Bilkent University, Ankara, Türkiye, in 1995 and 1997, respectively, with high honors. She received her Ph.D. degree in Electrical and Electronics Engineering from Boğaziçi University, İstanbul, Türkiye, in 2002. From September 2000 to June 2001, she was a visiting researcher at the Department of Electrical and Computer Engineering, University of Rochester, NY, USA. Between 2003 and 2004, she was a postdoctoral fellow at the Faculty of Electrical Engineering, Delft University of Technology, the Netherlands, where she was also affiliated with the Video Processing Group at Philips Research Laboratories, Eindhoven. Between 2002 and 2009, she was the Director of Research at Momentum Digital Media Technologies Inc., a technology SME located in İstanbul. Between 2009 and 2016, she was a faculty member at the Department of Electrical and Electronics Engineering, Bahçeşehir University, İstanbul, Türkiye. Between 2016 and 2024, she was a full professor at the Department of Computer Engineering, Marmara University, İstanbul, Türkiye. Since 2024, she has been a full professor at the Department of Electrical and Electronics Engineering, Özyeğin University, İstanbul, Türkiye.

Prof. Erdem's current research interests are in the areas of digital image and video processing, computer vision, and pattern recognition, with applications to biometrics and federated learning. She has served as a referee for numerous technical journals. She has also served as an independent expert and vice chair during project evaluations for the European Commission.



ÖMER KORÇAK received his B.S. (with high honors), M.S. and Ph.D degrees all in Computer Engineering from Bogazici University, in 2002, 2004, and 2009 respectively. Currently, he is an associate professor at Marmara University, Istanbul. In 2011, he was a visiting researcher at Deutsche Telekom Laboratories, Berlin. He is leading the Intelligent Systems Lab in Marmara University. His main research interests include distributed learning, edge intelligence, smart IoT, game theory and incentive mechanism design for networked systems, and security and privacy-preserving mechanisms for intelligent systems.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/