Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 7759410, 10 pages
https://doi.org/10.1155/2022/7759410

*Research Article*

# Federated Multitask Learning with Manifold Regularization for Face Spoof Attack Detection*

**Yingyue Chen,[1] Liang Chen,[2] Chaoqun Hong,[3] and Xiaodong Wang[3]**

[1]*School of Economic and Management, Xiamen University of Technology, Xiamen, China*
[2]*School of Data and Computer Science, Sun Yat-Sen University, Guangzhou 510006, China*
[3]*School of Computer Science and Information Engineering, Xiamen University of Technology, Xiamen, China*

Correspondence should be addressed to Chaoqun Hong; cqhong@xmut.edu.cn

Received 7 September 2021; Revised 29 November 2021; Accepted 17 March 2022; Published 6 June 2022

Academic Editor: Paolo Spagnolo



Face recognition has been widely used in personal authentication, especially on edge computing devices. However, face recognition systems suffer from face spoof attack. In this paper, a novel method for face spoof attack detection in edge computing scenarios is proposed. It is based on federated learning and improves traditional federated learning with multitask learning and manifold regularization, which is known as federated learning for face spoof attack detection (FedFSAD). In this way, local model learning is completed on edge devices and global model learning only depends on the trained local models without using the original image data. Besides, the performance is improved by imposing hypergraph manifold regularization in the global training of multitask learning. The results of comprehensive experiments show that the detection performance is improved by about 10% and robust against stragglers and network delays, which indicates the effectiveness of FedFSAD.

## 1. Introduction

Personal authentication with face recognition has been widely used currently. However, facial images can be easily captured or faked. With these images, face spoof attack may be conducted. In some interactive applications, such as mobile payment and online banking systems, users are required to perform some predefined actions. In this way, face recognition systems can ensure that a live person is recognized [1]. However, interactions may slow down the authentication process. Besides, requirements are not always satisfied. Therefore, more advanced methods without interactions are needed.

To detect face spoof attack without interactions, researchers make use of different image features, such as motion features [2, 3], texture features [4], and image quality features [5]. Due to the descriptive power, deep features are also used [6, 7]. Since only a single type of features may not sufficient to describe facial images, methods with multiple features are proposed. Some of them make use of additional sensors to obtain different types of features, such as near-infrared illumination [8] and depth information [9]. Atoum et al. used HSV and YCC images instead of RGB images. In this way, features in both color images and depth images can be extracted [10].

To make better use of the above features, learning methods are also critical. With Local Binary Pattern (LBP) features, support vector machines (SVMs) were used to train the detection model [4]. The blinking-based approach using conditional random fields (CRFs) was used to detect face spoof attack [11]. To improve the efficiency for edge face recognition systems, spoof attack score is measured with Hamming distance [12]. Furthermore, some novel methods are proposed to handle multiple features. Deep learning is one of the representative frameworks to train the model with multiple features. For example, both spatial and temporal features in CNN were used for attack detection [7]. Shearlet features, RGB images, and optical flow were used as image quality, pixel colors, and motion cues by Feng et al. [13]. They were also combined in neural networks.

As we know, a large number of face recognition systems run on edge devices due to the growing storage and

computational power. They are used in mobile authentication, security entrance, and so on [14]. These devices can be easily connected by a fast network. Therefore, face spoof attack detection is also needed on these devices. In this scenario, the data come from different sources. Researchers try to make better use of these data. Shao et al. made use of metalearning and the feature space with deep learning to tackle the domain generalization problem [15]. In addition, training images are not directly shared between data owners due to legal and privacy issues, which bring in new challenge in many applications [16–18]. Face spoof attack detection makes use of facial images, and they are also critical personal information. To tackle it, researchers try to store data locally and push more network computation to the edge. Federated learning is a novel framework proposed to train models on devices [19]. Then, these models can be used in classification or regression without touching the training images directly [20]. Shao et al. have done the pioneer work on using federated learning for face spoof attack detection [21]. However, they focus on tackling data centers with significant domain shift effectively but not improving the performance of federated learning framework.

Generally speaking, current methods for face spoof attack detection significantly depend on the quantity of training data. Federated learning can be used to alleviate this issue, but existing models cannot collect and use the distributed data sufficiently and safely. In this paper, a novel method for face spoof attack detection in edge computing scenarios is proposed. It is based on federated learning and improves traditional federated learning with multitask learning and manifold regularization, which is known as federated learning for face spoof attack detection (FedFSAD). The contribution can be summarized as follows:

(1) First, we propose a novel framework for face spoof attack detection in edge computing scenarios. It models the problem of federated learning with the multitask learning idea.

(2) Second, the process of multitask learning is further improved with manifold regularization, in which the inner relationships among different training tasks are explored to learn a unified model.

(3) Third, we propose hypergraph manifold regularization with sparse representation. Multiple vertices are connected by one hyperedge and the connectivities among features are computed by sparse learning.

(4) Finally, with the trained model, face spoof attack is detected in the classification process. Comprehensive experiments are conducted to indicate the effectiveness our method on three commonly-used benchmark datasets for face spoof attack detection.

The remainder of our paper is organized below. In Section 2, we outline the proposed FedFSAD first and then introduce it in detail. After theoretical introduction, in Section 3, we show the improvements of FedFSAD on face spoof attack detection in edge computing scenarios. Finally, in Section 4, we provide some discussion about the novelty and improvements of the proposed method.

## 2. Federated Multitask Learning with Manifold Regularization

*2.1. Outline.* The proposed method can be outlined by Figure 1. The whole framework consists of local model training in edge subsystems and global model training in the server. Local models are trained separately and transferred to the server. Then, the global model is trained using local models and robust to a small fraction of subsystems unpredictably dropping.

*2.2. Notations.* To make the paper clear, we summarize the definitions of notations in Table 1.

*2.3. Federated Learning Framework for Local Model Learning.* The proposed method is based on the routine of federated learning. In the setting of federated learning, local model training is completed on the edge directly. It brings in two advantages. First, plenty of data can be collected in a distributed way. More training data can be used to improve the performance since the quality of images captured by edge devices is not always satisfactory [22, 23]. Second, image data will not be transferred to servers and data privacy can be preserved [24]. Assuming that $X_1, X_2, \ldots, X_m$ is the facial image data captured on $m$ edge devices and they can be represented by image features $F_1, F_2, \ldots, F_m$. To obtain the global model, we need to solve the local subproblem and compute the local models firstly. Then, the parameters of the global model can be updated by incoming local data. Inspired by CoCoA [25] and MOCHA [26], for the $t$-th device and the corresponding image data denoted by $X_t$, the $t$-th $(t \le m)$ subproblem is defined by

$$\begin{aligned}\min_{\Delta\alpha_t} \mathcal{G}_t^{\delta}(\Delta\alpha_t; v_t, \alpha_t) \coloneqq & \sum_{i=1}^{n_t} \mathcal{L}_t^*\left(-\alpha_t^i - \Delta\alpha_t^i\right) \\ & + < w_t(\alpha), X_t\Delta\alpha_t > + \frac{\delta}{2}\left\|X_t\Delta\alpha_t\right\|_{M_t}^2 + c(\alpha).\end{aligned} \tag{1}$$

where $w(\alpha) = \nabla\mathcal{R}^*(X\alpha)$. $w_t(\alpha)$ represents the values of $w(\alpha)$ within the $t$-th task, which indicates the parameters updated by the $t$-th task. $<w_t(\alpha), X_t\Delta\alpha_t>$ is the average of $w_t(\alpha)$ and $X_t\Delta\alpha_t$. $\mathcal{L}_t^*$ is the loss function of the $t$-th task, which demonstrates the differences between the predicted results and the ground truth. $\delta$ is a constant parameter to control the updating speed of the federated model. $c(\alpha) \coloneqq 1/m\mathcal{R}^*(X\alpha)$. $M_t$ is the $t$-th diagonal block of $M$, which is defined as

$$M = (\pi\eta\Omega \otimes I + (1-\eta)I)^{-1}, \tag{2}$$

where $I$ is the identity matrix. Computing $w(\alpha)$ requires $v = X\alpha$ and $v_t$ is the $t$-th block of $v$, which is required to transferred between devices and servers.

With $\alpha$ and $v$, we can define the dual problem as

$$\min_{\alpha}\left\{\mathcal{D}(\alpha) \coloneqq \sum_{t=1}^{m}\sum_{i=1}^{n_t}\mathcal{L}^*\left(-\alpha_t^i\right) + \mathcal{R}^*(X\alpha)\right\}, \tag{3}$$

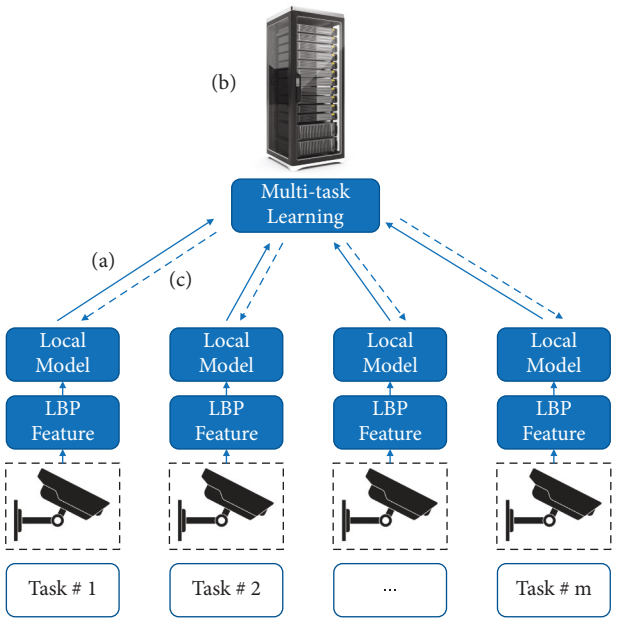



FIGURE 1: The flowchart of the proposed federated learning framework for face spoof attack detection. (a) For each subsystem for face recognition, we set up a task to detect face spoof attack. Facial images are represented by local binary pattern (LBP) features. These features are used to train local models of subsystems. (b) Local models are transferred to the server and multitask learning is applied to train the global model. (c) Finally, the global model is transferred back to subsystems and it can be used for face spoof attack detection on them.

where $\alpha_t^i$ is the dual variable for the data point $(x_t^i, y_t^i)$ and $y$ is the label. Therefore, to solve the above subproblem and compute $\Delta\alpha_t$, we have to find good definitions of $\mathscr{L}$, $\mathscr{R}$, and $M$.

*2.4. Multitask Learning for Global Model Learning.* The key of CoCoA and MOCHA is the solution to equation (1). In the proposed method, we try to improve it. Therefore, we propose to solve equation (1) using multitask learning. In multitask learning, we set the training process on each edge device as a task. They can be trained separately and then combined to obtain a unified model. This model can be transferred back to the devices and used for face spoof attack detection. There have been several definitions for $\mathscr{L}$ and $\mathscr{R}$ in multitask learning provided by MALSAR [27]. $\mathscr{L}$ can be arbitrary convex loss functions such as the hinge loss and so on. $\mathscr{R}$ can be defined as a clustering loss and computed by a biconvex function:

$$\mathscr{R}(W, \Omega) = \lambda tr\left(W\Omega W^T\right) + (\pi 1 - \lambda)\|W\|_F^2, \tag{4}$$

where the $t$-th column of $W$ indicates the weight of the $t$-th task and $\Omega$ indicates the weights among different tasks. $W$ can be updated according to equation (1). However, updating $W$ with equation (1) requires $M$ and $M$ depends on $\Omega$. Therefore, the key to the proposed method is computing an optimal $\Omega$.

*2.5. Hypergraph Manifold Regularization with Sparse Representation.* Manifold regularization has been widely used to describe the relationships among data [28]. Inspired by it, we make use of manifold regularization to model the weights among different tasks $\Omega$, which is known as the Laplacian matrix modeling the relationship of features in the feature space. In the feature space, the LBP features of images are considered as the vertices and the relationships are connectivities among them. In contrast to the traditional graph, hypergraph allows an edge to connect more than two vertices. Thus, an edge contains a subset of vertices. Hypergraph has been proved to be a better idea to describe the connectivities among related data. Notations used in hypergraph regularization are summarized in Table 2. Based on the patch alignment framework [29], we propose Hypergraph Manifold Regularization with Sparse Representation (HMRSR), in which $\Omega$ can be computed with two steps:

*2.5.1. Part Optimization.* We define one patch to be the vertices connected by one hyperedge. Thus, the patch in the proposed regularization process is defined by

$$\underset{y \in R^{|P|}}{\operatorname{argmin}} \sum_{p,q \subset e} \frac{\sigma(e)}{\varepsilon(e)} \left( \frac{y(p)}{\sqrt{\xi(p)}} - \frac{y(q)}{\sqrt{\xi(q)}} \right)^2. \tag{5}$$

For one patch, we should compute

$$\sum_{p,q \subset e} \frac{\sigma(e)}{\varepsilon(e)} \left( \frac{y(p)}{\sqrt{\xi(p)}} - \frac{y(q)}{\sqrt{\xi(q)}} \right)^2. \tag{6}$$

which means that we randomly choose two vertices in the subset of vertices contained by a hyperedge, $e$, and sum the value of

$$\frac{\sigma(e)}{\epsilon(e)} \left( \frac{y(p)}{\sqrt{\xi(p)}} - \frac{y(p)}{\sqrt{\xi(q)}} \right)^2. \tag{7}$$

Expanding (6) and combining items, we can get the patch optimization for each hyperedge:

$$\frac{1}{2} \sum_{l \subset e} \frac{Y}{D_{vl}^{1/2}} E H_e' D_e^{-1} H_e E' \frac{Y}{D_{vl}^{1/2}}. \tag{8}$$

Matrix $E$ is

$$E = \begin{bmatrix} -e^T \\ I \end{bmatrix}, \tag{9}$$

where $e = [1, \ldots, 1]^T$, $I$ is an $n \times n$ identity matrix.

*2.5.2. Whole Alignment.* In the hypergraph, the weight of a hyperedge is computed by summing the similarity scores of all the pairs of vertices contained in this hyperedge. The similarity score of any pair of vertices is defined as the distance of image features:

$$S(p, q) = \exp\left( \frac{1}{\sigma} \operatorname{Sim}(\operatorname{feat}(p), \operatorname{feat}(q)) \right), \tag{10}$$

Table 1: Definitions of notations.

| Name | Type | Definition |
|---|---|---|
| $\mathscr{L}$ | Function | Loss function in multitask learning |
| $\mathscr{R}$ | Function | Regularization function in multitask learning |
| $\mathscr{L}^*, \mathscr{R}^*$ | Function | Conjugate dual functions of $\mathscr{L}$ and $\mathscr{R}$ |
| $\mathscr{G}$ | Function | Local subproblem in federated learning |
| $tr$ | Function | Trace of the matrix |
| $EXP$ | Function | The exp function |
| $X$ | Matrix | Image data |
| $W$ | Matrix | Weighted matrix of samples within tasks |
| $I$ | Matrix | Identity matrix |
| $F$ | Matrix | Image feature matrix |
| $\Omega$ | Matrix | Weighted matrix among tasks |
| $A$ | Matrix | Affinity matrix among tasks |
| $M$ | Matrix | Symmetric positive definite matrix defined as $M = (\lambda_1 \Omega \otimes I + \lambda_2 I)^{-1}$ |
| $\alpha$ | Vector | The dual variables |
| $w$ | Vector | Primal variables |
| $v$ | Vector | Information transferred among devices |
| $\delta$ | Scalar | Parameter to control the updating speed of the federated model |
| $\lambda$ | Scalar | Parameter to control the global and local weights. |
| $\eta$ | Scalar | Parameter to control the contribution of task relationships |
| $k$ | Scalar | The number of nearest neighbors |
| $m$ | Scalar | The number of local nodes |

Table 2: Definition of notations in the hypergraph regularization.

| Name | Definition |
|---|---|
| $p, q$ | Vertices in the hypergraph |
| $y(p)$ | The label of the vertex $p$ |
| $e$ | Edges in the hypergraph |
| $\sigma(e)$ | The weight of an edge $e$ |
| $\epsilon(e)$ | The degree of an edge, $e$. It illustrates how many vertices are connected by $e$. In traditional graph representation, $\epsilon(e) = 2$. |
| $\xi(m)$ | The degree of a vertex $p$. It is calculated by summing the weighting values of edges connected to this vertex. |
| $D_v$ | The diagonal matrix containing the vertex degrees |
| $D_e$ | The diagonal matrix containing the edge degrees |
| $H$ | In this matrix, $H(p, e) = 1$ if $p \in e$ |
| $\Sigma$ | The diagonal matrix containing the weights of hyperedges |
| $Y$ | The set of labels |
| $V$ | The set of vertices |
| $E$ | The set of edges |

where feat$(p)$ represents the image feature vector of vertex $p$ and Sim is the similarity of feat$(p)$ and feat$(q)$. With the hyperedge weighting matrix, the multiview hypergraph Laplacian can be computed by summing the patch optimization defined in (8) of all the hyperedges:

$$\Omega = \frac{1}{2} \sum_{e \in E} \sum_{v \in e} \frac{Y}{D_v^{1/2}} E H' D_e^{-1} H E' \frac{Y}{D_v^{1/2}}. \tag{11}$$

(11), there are three matrix to be initialized. They are $H$, $D_v$ and $D_e$. $H$ is computed by obtaining the most similar vertices:

$$H(p, q) = \begin{cases} 1, & \text{if } p \text{ is the most similar vertics of } q, \\ 0, & \text{else.} \end{cases} \tag{12}$$

Then, $D_e$ can be computed with $H$. The $l$-th item of $D_e$ can be computed by

$$D_{ell} = \sum_{k=1}^{n} H_{lk}. \tag{13}$$

Finally, $D_v$ can be computed with $S$. The $l$-th item of $D_v$ can be computed by

$$D_{vll} = \sum_{k=1}^{n} S_{lk}. \tag{14}$$

In (10) and (12), we need to define a reasonable measurement for feature similarity. Traditionally, it is computed by feature distances, such as L2 distances and so on. In this paper, we make use of sparse learning. In the result of sparse learning, one vector is represented by the combination of basis vectors and only about 30% coefficients are nonzero. Then, the coefficients can be used to represent the relationship between them [30]. There are several existing

solutions for sparse learning. Among them, we choose the approximation by L1 norm [31]. Then, the $l$-th image feature $F_l$ can be represented as the combination of the whole feature set. In this way, the whole feature set is used as the basis vectors. Then, the coefficient can be computed by

$$P_1(\tau): \min_a \|\beta\| s.t.\ F_l = F\beta. \tag{15}$$

where $\beta$ is the resulting coefficients. To compute (15), we choose the LARS with Lasso modification implemented by SparseLab [32].

*2.6. Implementation Details.* The training process of the proposed FedFSAD is shown in Algorithm 1. With the trained model, the input image can be classified as a real image or a fake image.

## 3. Simulated Evaluations

*3.1. Datasets and Settings.* In the experiments, we use three challenging datasets for face spoof attack detection. The first one is the NUAA Photograph Imposter Database (NUAA). NUAA is collected by generic and commonly-used webcams [33]. It is collected in three sessions. The place and illumination conditions of each session are different. There are 5105 real faces and 7509 fake images from 15 subjects in total.

The second dataset is the Multispectral-Spoof face spoofing database built at Idiap Research institute (MSSPOOF) 2. It contains both color images (VIS) and infrared images (NIR) [34]. Similar to NUAA, images in MSSPOOF are recorded in different light conditions. The number of subjects in the database is 21. There are 70 real faces and 144 fake images for each subject. Examples of the database are shown in Figure 2.

The third dataset is the CASIA Face Antisproofing Database (CASIA-SURF) [35], which is collected by Automation, Chinese Academy of Sciences 3. It contains 29266 training samples, 9608 validating samples, and 57710 testing samples. A color image, a depth image, and an infrared image are provided for each sample.

In the experiments, the performance is measured by the classification accuracy, which is computed by

$$ACC = \frac{\text{Correct count}}{\text{Total count}}. \tag{16}$$

where correct count is the number of correctly classified samples. Classification is completed with a simple SVM regularized by the trained model [26]. For cross validation, we randomly choose 75% samples in training and the rest samples are used in testing. By default, samples in the datasets are equally assigned to tasks. In addition, to simulate the scenario of federated learning, edge subsystems randomly drop. This process is repeated 20 times and the average results are shown. All the facial parts are detected and resized to $200 \times 200$. A laptop with i7-9750H CPU, 16G RAM, and GTX1650 GPU is used. Evaluations are run on MATLAB R2017a.

*3.2. Comparison of Different Manifold Learning Methods.* Manifold regularization has been comprehensively studied. In this part, we demonstrate the effectiveness of the proposed Hypergraph Manifold Regularization with Sparse Representation (HMRSR). We compare it with existing manifold learning methods, such as LDA, DLA, LPP, NPE, LSDA, and ISOMAP [29]. The results are shown in Figure 3. Thanks to hypergraph learning and sparse representation, the proposed HMRSR can capture the connectivities among features and achieve better performance.

*3.3. Parameter Sensitivities.* As shown in Table 1, the proposed method depends on 3 parameters. They are $\delta$, $\lambda$, and $k$. The setting of $\delta$ is quite complicated, and we follow MOCHA [26]. Performance is influenced by $\lambda$ and $k$. Reasonable $\lambda$ and $k$ should be set for different datasets. The performance with different $\lambda$ is show in Figure 4. We can figure out that the performance of FedFSAD is achieved when $\lambda = 0.5$ for NUAA, $\lambda = 0.6$ for MSSPOOF, and $\lambda = 0.7$ for CASIA-SURF.

The performance with different $k$ is show in Figure 5. We can figure out that the performance of FedFSAD is achieved when $k = 15$ for NUAA, $k = 20$ for MSSPOOF, and $k = 25$ for MSSPOOF.

*3.4. Comparison with Existing Methods.* First, we emphasize the improvement of the proposed FedFSAD with multi-task learning and manifold regularization. FedFSAD is compared with a fully global model, a fully local model, and previous multitask model MOCHA. Besides, some state-of-the-arts are also included in the numerical comparison. In this experiment, we refer to the following methods:

(1) CoCoA [25]: CoCoA is a communication-efficient framework for distributed learning. It uses train local models in a primal-dual setting. In this way, the amount of transferred information can be reduced.

(2) MOCHA [26]: MOCHA extends CoCoA with multitask learning and is applied to federated learning.

(3) SeetaFace6 [36]: SeetaFace is an open-source project for face applications with computer vision. The latest version, which is named SeetaFace6, provides face spoof attack detection. The model has been trained and we directly use it in testing.

(4) Guided Scale Local Binary Pattern (GS-LBP) [4]: GS-LBP makes use of the edge-preserving property of the guided scale space. Besides, joint quantization is used to encode the spatial locality. Therefore, it can be used as the facial image feature. SVM is used as the classifier.

The result is shown in Table 3. The items with the best performance in each dataset are highlighted in red. Generally speaking, CASIA-SURF is the most difficult. SeetaFace6 and GS-LBP provide stable performance. However, they have not considered the improvements with multiple tasks. Thanks to the application of multi-task learning, MOCHA achieves better performance than the fully global model and the fully local modal. In addition, the proposed FedFSAD is

Figure 2: Sample images in the MSSPOOF dataset are shown. The first row is images taken with in VIS, while the second one is images taken in NIR. The first column is real accesses. The second column is VIS attacks. The third column is NIR attacks. NIR is near-infrared spectra and VIS is visible spectra. They are approaches to capture images.

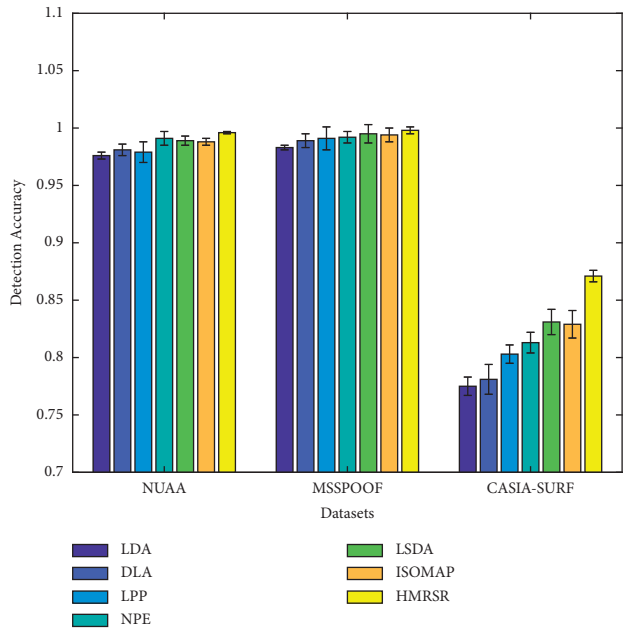



Figure 3: The performance of different manifold learning methods.

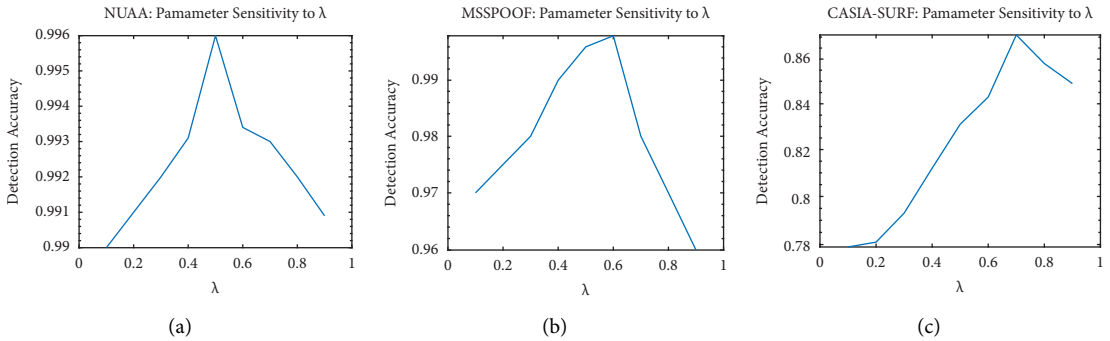
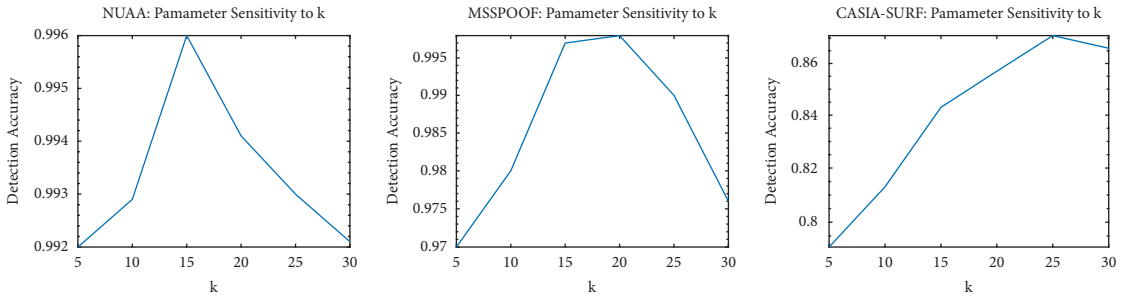



Figure 4: Performance with different values of $\lambda$. (a) NUAA. (b) MSSPOOF. (c) CASIA-SURF.

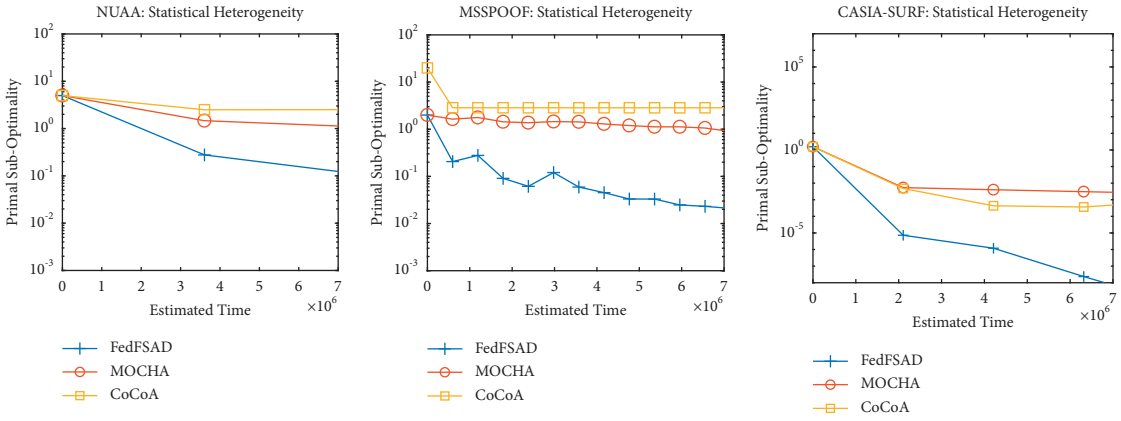



(a) (b) (c)

Figure 5: Performance with different values of $k$. (a) NUAA. (b) MSSPOOF. (c) CASIA-SURF.

**Input:** Image $X$, initialized $\alpha$
**Output:** Multitask face spoof attack detection model.
**Local model learning:**
Extract LBP features of $X$ and get $F$;
Compute feature similarities with (15);
**for all** each task **do**
  Compute $H$ with (12);
  Compute $D_e$ with (13);
  Compute $S$ with (10);
  Compute $D_v$ with (14);
  Compute $\Omega$ with (11);
  Compute $M$;
  Update $\alpha$ and $W$ with (1);
  Solve dual problem with (3);
**end for**
**Global model learning:**
Train the multi-task model with (4);
**return** The multi-task model;

Algorithm 1: Details of training FedFSAD.




(a) (b) (c)

Figure 6: Comparison on statistical heterogeneity. (a) NUAA. (b) MSSPOOF. (c) CASIA-SURF.

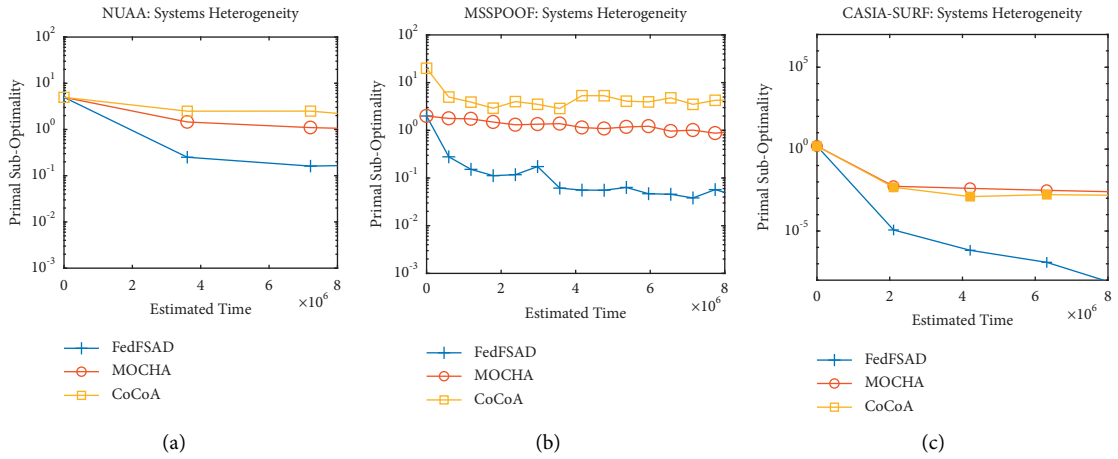



Figure 7: Comparison on system heterogeneity. (a) NUAA. (b) MSSPOOF. (c) CASIA-SURF.

Table 3: Comparison of different models and methods.

| Model | NUAA | MSSPOOF | CASIA-SURF |
|---|---|---|---|
| Global | 0.777 ± 0.031 | 0.739 ± 0.131 | 0.614 ± 0.119 |
| Local | 0.872 ± 0.013 | 0.512 ± 0.129 | 0.533 ± 0.12 |
| MOCHA | 0.993 ± 0.003 | 0.997 ± 0.001 | 0.819 ± 0.006 |
| SeetaFace6 | 0.886 ± 0.007 | 0.793 ± 0.005 | 0.736 ± 0.013 |
| GS-LBP | 0.861 ± 0.005 | 0.776 ± 0.003 | 0.644 ± 0.015 |
| FedFSAD | 0.996 ± 0.001 | 0.998 ± 0.003 | 0.871 ± 0.005 |

Table 4: Cross dataset testing.

| | Train | | |
|---|---|---|---|
| | NUAA | MSSPOOF | CASIA-SURF |
| NUAA | — | 0.633 ± 0.114 | 0.751 ± 0.135 |
| MSSPOOF | 0.798 ± 0.118 | — | 0.867 ± 0.123 |
| CASIA-SURF | 0.736 ± 0.117 | 0.589 ± 0.109 | — |

better than MOCHA due to the usage of manifold regularization.

Second, we show the robustness of the proposed FedFSAD. In the scenario of federated learning, stragglers and network delays are critical. Stragglers appear when it takes too much time to train local models, while network delays appear when it takes too much time to transfer local models to the server. In these experiments, we take CoCoA and MOCHA into comparison and show the results on statistical heterogeneity and system heterogeneity. The results are shown in Figures 6 and 7. As the time elapse, primal suboptimality can be reduced. We can figure out that the proposed FedFSAD is robust and outperforms state-of-the-arts when stragglers and network delays appear. We also conduct the cross datasets testing, which is shown in Table 4. We can figure out that the proposed method is still applicable in cross-set scenario. Besides, if a larger training set, such as CASIA-SURF, is used, the performance is better.

## 4. Conclusion and Discussion

According to the methodology of face spoof attack detection and the improvements of simulated performance, the novelty and contribution of the proposed FedFSAD can be shown.

First, the proposed method tries to tackle the issue of face spoof attack detection on edge devices. Based on the framework of federated learning, we introduce multitask learning. Therefore, we proposed a solution to federated learning with multitask learning. Besides, it is improved by using manifold learning. In this way, the relationships among tasks on edge devices are explored by hypergraph manifold regularization with sparse representation. Therefore, the proposed method is a novel method for face spoof attack detection.

Second, comprehensive simulation has been conducted. According to the results, FedFSAD outperforms exiting methods on accuracy of face spoof attack detection. It proves that a better model is obtained with multitask learning and manifold learning. Besides, we simulate the situations of stragglers and network delays. Although some information is late or missed in these situations, FedFSAD is still better than exiting methods. In this way, robustness is also improved.

In the future, we will focus on the unexpected problems on edge devices, such as hardware failure, network disconnection, and so on. In these situations, both the performance and robustness can be further improved.

## Data Availability

The NUAA dataset is provided for research purposes to a researcher only and not for any commercial use. The data cannot be released and the link cannot be redistributed without the authors' permission. Therefore, we contact them and obtain the data with permission. The MSSPOOF dataset is available at https://www.idiap.ch/dataset/msspoof.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References


[1] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition*, pp. 1867–1874, Columbus, OH, USA, June 2014.

[2] A. Anjos, M. M. Chakka, and S. Marcel, "Motion-based counter-measures to photo attacks in face recognition," *IET Biometrics*, vol. 3, no. 3, pp. 147–158, 2014.

[3] S. Bharadwaj, T. I. Dhamecha, M. Vatsa, and R. Singh, "Computationally efficient face spoofing detection with motion magnification," in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 105–110, IEEE, Portland, OR, USA, June 2013.

[4] F. Peng, L. Qin, and M. Long, "Face presentation attack detection using guided scale texture," *Multimedia Tools and Applications*, vol. 77, no. 7, pp. 8883–8909, 2018.

[5] J. Galbally and S. Marcel, "Face anti-spoofing based on general image quality assessment," in *Proceedings of the 2014 22nd International Conference on Pattern Recognition*, pp. 1173–1178, IEEE, Stockholm, Sweden, Augest 2014.

[6] L. Li, X. Feng, Z. Boulkenafet, Z. Xia, M. Li, and A. Hadid, "An original face anti-spoofing approach using partial convolutional neural network," in *Proceedings of the 2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, January 2017.

[7] Y. Liu, A. Jourabloo, and X. Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 389–398, IEEE, Salt Lake, UT, USA, June 2018.

[8] J. Liu and A. Kumar, "Detecting presentation attacks from 3d face masks under multispectral imaging," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 47–52, IEEE, Salt Lake, UT, USA, June 2018.

[9] Y. Wang, F. Nian, T. Li, Z. Meng, and K. Wang, "Robust face anti-spoofing with depth information," *Journal of Visual Communication and Image Representation*, vol. 49, pp. 332–337, 2017.

[10] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu, "Face anti-spoofing using patch and depth-based cnns," in *Proceedings of the 2017 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 319–328, IEEE, Denver, CO, USA, October 2017.

[11] L. Sun, G. Pan, Z. Wu, and S. Lao, "Blinking-based live face detection using conditional random fields," in *Proceedings of the International Conference on Biometrics*, pp. 252–260, IEEE, Seoul, Korea, August 2007.

[12] H. Jee, S. Jung, and J. Yoo, "Liveness detection for embedded face recognition system. World Academy of Science, Engineering and Technology," *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 2, no. 6, pp. 2142–2145, 2008.

[13] L. Feng, L.-M. Po, Y. Li et al., "Integration of image quality and motion cues for face anti-spoofing: a neural network approach," *Journal of Visual Communication and Image Representation*, vol. 38, pp. 451–460, 2016.

[14] B. B. Gupta and M. Quamara, "An overview of internet of things (iot): architectural aspects, challenges, and protocols," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 21, Article ID e4946, 2020.

[15] R. Shao, X. Lan, and P. C. Yuen, "Regularized fine-grained meta face anti-spoofing," 2019, https://arxiv.org/abs/1911.10771.

[16] A. A. Abd El-Latif, X. Niu, and M. Amin, "A new image cipher in time and frequency domains," *Optics Communications*, vol. 285, no. 21-22, pp. 4241–4251, 2012.

[17] H. Wang, Z. Li, Y. Li, B. B. Gupta, and C. Choi, "Visual saliency guided complex image retrieval," *Pattern Recognition Letters*, vol. 130, pp. 64–72, 2020.

[18] X. Yan, S. Wang, A. A. A. El-Latif, and X. Niu, "Visual secret sharing based on random grids with abilities of and and xor lossless recovery," *Multimedia Tools and Applications*, vol. 74, no. 9, pp. 3231–3252, 2015.

[19] J. Konecny, H. B. Mcmahan, F. X. Yu, A. T. Suresh, D. Bacon, and P. Richtarik, "Federated Learning: Strategies for Improving Communication Efficiency," 2018, http://arXiv.org/abs/Learning.

[20] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: concept and applications," 2019, https://arxiv.org/abs/1902.04885.

[21] R. Shao, P. Perera, P. C. Yuen, and V. M. Patel, "Federated Face Anti-spoofing," 2020, https://arxiv.org/abs/2005.14638.

[22] M. A. Alsmirat, F. Al-Alem, M. Al-Ayyoub, Y. Jararweh, and B. Gupta, "Impact of digital fingerprint image quality on the fingerprint recognition accuracy," *Multimedia Tools and Applications*, vol. 78, no. 3, pp. 3649–3688, 2019.

[23] N. Wang, Q. Li, A. A. Abd El-Latif, T. Zhang, and X. Niu, "Toward accurate localization and high recognition performance for noisy iris images," *Multimedia Tools and Applications*, vol. 71, no. 3, pp. 1411–1430, 2014.

[24] C. Esposito, M. Ficco, and B. B. Gupta, "Blockchain-based authentication and authorization for smart city applications," *Information Processing & Management*, vol. 58, no. 2, Article ID 102468, 2021.

[25] M. Jaggi, V. Smith, M. Takac et al., "Communication-efficient distributed dual coordinate ascent," in *Advances in Neural Information Processing Systems*, vol. 4, pp. 3068–3076, MIT Press, 2014.

[26] V. Smith, C. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," in *Advances in Neural Information Processing Systems*, pp. 4427–4437, MIT Press, Cambridge, MA, USA, 2017.

[27] J. Zhou, J. Chen, and J. Ye, "Malsar: multi-task learning via structural regularization," *Arizona State University*, vol. 21, 2011.

[28] J. Yu, M. Tan, H. Zhang, Y. Rui, and D. Tao, "Hierarchical deep click feature prediction for fine-grained image recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 563–578, 2022.

[29] T. Tianhao Zhang, D. Dacheng Tao, X. Xuelong Li, and J. Jie Yang, "Patch alignment for dimensionality reduction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1299–1313, 2009.

[30] S. Yan and H. Wang, "Semi-supervised learning by sparse representation," in *Proceedings of the 2009 SIAM International Conference on Data Mining*, pp. 792–801, dblp, Carson, NV, USA, April 2009.

[31] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[32] D. Donoho, V. Stodden, and Y. Tsaig, "Sparselab Architecture – Version 2.0," 2007.

[33] X. Tan, Y. Li, J. Liu, and L. Jiang, "Face liveness detection from a single image with sparse low rank bilinear discriminative model," in *Proceedings of the European Conference on Computer Vision: Part VI*, pp. 504–517, Springer, Heraklion, Crete, Greece, September 2010.

[34] I. Chingovska and N. A. A. S. M. Erdogmus, "Face recognition systems under spoofing attacks," in *Face Recognition across the Imaging Spectrum*, T. Bourlai, Ed., Springer, New York, NY, USA, 2015.

[35] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li, "A face antispoofing database with diverse attacks," in *Proceedings of the 2012 5th IAPR International Conference on Biometrics (ICB)*, pp. 26–31, IEEE, New Delhi, India, April 2012.

[36] X. Liu, M. Kan, W. Wu, S. Shan, and X. Chen, "Viplfacenet: an open source deep face recognition sdk," *Frontiers of Computer Science*, vol. 11, no. 2, pp. 208–218, 2017.