SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

Special Focus on Cyber Security in the Era of Artificial Intelligence

July 2022, Vol. 65 170302:1–170302:16
https://doi.org/10.1007/s11432-021-3423-9

Non-IID federated learning via random exchange of local feature maps for textile IIoT secure computing

Bo PENG[1], Mingmin CHI[1,2*] & Chao LIU[1,3*]

[1]*School of Computer Science, Shanghai Key Laboratory of Data Science, Fudan University, Shanghai* 200438*, China;*
[2]*Zhongshan PoolNet Technology Co. Ltd., Zhongshan Fudan Joint Innovation Center, Zhongshan* 528400*, China;*
[3]*Zhengzhou Zhongke Institute of Integrated Circuit and System Application, Zhengzhou* 450001*, China*

Received 1 September 2021/Revised 14 January 2022/Accepted 8 February 2022/Published online 20 June 2022

Abstract With the fast development of artificial intelligence (AI) and industrial Internet of Things (IIoT) technologies, it is challenging to deal with the problems of data privacy protection and secure computing. In recent years, federated learning (FL) is proposed to attack the challenges of learning shared models collaboratively while protecting security based on the data from cross-domain clients. However, data in the real environment is usually not independent and identically distributed (Non-IID) due to the differences in business, working environments, and data acquisition, and thus classic federated methods suffer from significant performance degradation. In this paper, a novel federated framework is proposed for secure textile fiber identification (FedTFI) via cross-domain texture representation based on high-definition fabric images. In addition to sharing the gradient of FedTFI, the local patch of feature maps between cross-domain clients is randomly exchanged to build a richer image texture feature distribution while protecting data security simultaneously for secure computing. Furthermore, a texture embedding layer is designed to provide a joint representation through similarity measure between triplet samples in low-dimensional space. To verify the effectiveness of the proposed framework, two textile image datasets, i.e., one public and the other we collected, are utilized to construct four Non-IID scenarios, including label skew, feature skew, and two combined skew scenarios. The experimental results confirm the effectiveness of our model to obtain better detection accuracies than benchmarks in four Non-IID scenarios by keeping data privacy for secure computing in fabric IIoT.

Keywords federated learning, secure computing, industrial Internet of Things (IIoT), machine vision, texture encoding, image classification, Non-IID

Citation Peng B, Chi M M, Liu C. Non-IID federated learning via random exchange of local feature maps for textile IIoT secure computing. Sci China Inf Sci, 2022, 65(7): 170302, https://doi.org/10.1007/s11432-021-3423-9

1 Introduction

Deep learning (DL) techniques are increasingly deployed in the Internet of Things (IoT), such as vehicle identification, industrial intelligent monitoring, and robotics [1–7]. With the rapid development of the industrial Internet of Things (IIoT), centralized learning artificial intelligence (AI) models can effectively solve the tasks of detection or classification in real applications. However, it brings security risks to the industrial Internet data distributed to different clients.

Federated learning (FL) [8] is an attractive solution for this problem, which allows machine models to learn on distributed datasets while keeping the data localized. In particular, an FL model is firstly trained by local data at each client or enterprise, where it is not necessary to move the data by keeping data privacy. Then, the local model parameters are safely aggregated on a central server by an encryption algorithm, such as homomorphic encryption [9], to generate a global model. In recent years, FL has aroused widespread research interest in academia and industry [10–13] and has been widely used in applications such as financial fraud detection, abnormal text mining, and health care [14–16]. Most current research on FL focuses on improving model prediction accuracy or reducing communication costs. However, due to the difference in business, working environment, or equipment in IIoT applications, the

[*] Corresponding author (email: mmchi@fudan.edu.cn, chaoliu20@fudan.edu.cn)

collected data from different clients usually cannot follow the independent and identically distributed (IID) assumption. Meanwhile, most of the existing methods are based on IID data and thus cannot be directly utilized to effectively share model parameters learned over Non-IID (not independent and identically distributed) data or features in IIoT applications.

To deal with the issues, some methods utilize Data Sharing strategies by sharing part of local raw data or the transformed one. For example, in [17], a sub-dataset is supposed to be globally shared between all the clients to mitigate the Non-IID negative impact; in [18], a self-balancing FL framework is proposed to relieve a global imbalance by adaptive data augmentation and downsampling; in [19], affine transformation is utilized to securely "share" data. However, these methods can only be implemented by sharing raw data, leading to the risk of data privacy leakage. Recently, some studies trying to develop effectively FL algorithms under Non-IID data without sharing local data, such as FedProx [20], SCAFFOLD [11], and FedNova [21]. However, these algorithms are based on a rigid division strategy using simulated data. The performance is still not ideal in real-world distributed data.

Meanwhile, texture information is essential for distinguishing different industrial image processing materials. It is necessary to introduce texture representation learning methods. At present, texture images are successfully discriminated by deep convolutional neural networks (CNNs) [22,23] to identify soil classes, construction materials, etc. [24,25]. Deep-TEN [26] is one of seminal texture models by combining dictionary learning layers and CNN into an end-to-end classification pipeline, achieving satisfactory results on multiple material recognition tasks. In [22], a CNN-based unmixing network is proposed to learn representations given imbalanced samples and small sample size for textile fiber identification. In [27], the CNN feature maps are constructed across different layers as a dynamic process. Such methods have presented a convincing recognition performance on IID data, but when deployed in a federated learning framework under the Non-IID scenario, the performance can be significantly dropped.

To deal with the secure computing of fabric IIoT under the Non-IID scenario, in this paper, we propose a federated textile fiber identification (FedTFI) framework via a cross-domain texture representation. Based on the characteristics of texture images, that is, the repetitive arrangement of pixels pattern, an interpolation module is designed to interpolate the patches of local feature maps into the samples belonging to other domains. Specifically, the cross-domain local patches of feature maps generated by a CNN backbone are firstly randomly sampled and saved in a feature bank. Then a feature interpolation module is proposed, in which a pre-trained feature extractor is defined to obtain a set of feature maps. We randomly sample a feature map with a fixed-scale window to get random and local patches.

To prevent the local interpolation of the feature maps from interfering with the texture representation, we design a texture embedding learning (TEL) module. The fused features are jointly represented by a similarity measurement between samples in a low-dimensional embedding to realize an efficient domain adaptive while ensuring data security. Specifically, the interpolated feature maps are divided into patches and sent to a three-branch transformer encoder by sharing the weights. The representation by transformer encoders is used in two subtasks. Firstly, a texture representation is learned by minimizing the intra-class variation and maximizing the inter-class variation. Secondly, the representation is fed into the texture feature encoding layers as feature descriptors. To avoid the loss of texture information caused by the fixed-scale encoding module, we further design a multi-scale texture encoding (MusTEN) layer with a keyword consistent attention (KCA) mechanism intended to perceive keywords in the multi-scale codebooks effectively.

The main contributions of this paper are as follows.

- We propose a federated learning framework on Non-IID data that can automatically identify fabric materials securely in the textile IIoT scene. In addition to sharing the model's gradient, a random cross-domain interpolation strategy is designed to learn from multiple distributed sources of Non-IID data over the local feature maps securely.

- We propose a TEL module, in which the interpolated and original features are jointly represented in a low-dimensional feature by minimizing the intra-class variation and maximizing the inter-class variation.

- To generate refined representations of different granularities, we propose an MusTEN module, in which the output features are aggregated as a global representation that contains richer texture information. And then, a KCA mechanism is designed to further capture critical semantic information.

2 Related work

2.1 Federated learning on Non-IID data

Federated learning is a distributed machine learning technology in which a shared global model is trained under a central server's coordination from a federation of participating devices. The clients are enabled to collaboratively learn a shared prediction model while keeping all the training data on the device, decoupling the ability to do machine learning from storing the data in the cloud. However, the widely known aggregation strategy FedAvg [28] often suffers when data is Non-IID. Different FL methods have different strategies for Non-IID problems. Some methods consider solving the Non-IID problem through a sample sharing strategy. For example, in [17, 18], part of local raw data are shared between servers; in [19,20], the transformed local data are shared for accelerating the convergence of models while ensuring safety. And in [17], the model alleviates the negative effect of Non-IID by sharing some local data with the server. Although the Data Sharing strategy can significantly enhance the global model performance on Non-IID data, it has obvious shortcomings in downloading parts of the global dataset to each client. It violates the requirement of privacy-preserving learning, which is the fundamental motivation of FL. Some studies develop FL algorithms for Non-IID data by optimizing the gradient transmission and model parameter update process without sharing local data. For example, SCAFFOLD [11] introduces control variates for the server and parties, which are used to estimate the update direction of the server model and the update direction of each client. FedNova [21] normalizes and scales the local updates of each party according to their number of local steps before updating the global model. FedAMP [29] proposes an FL framework that uses a unique adaptive grouping learning mechanism to allow customers with similar data distribution to cooperate more. Meanwhile, it tries to personalize each customer's model to effectively deal with the widespread inconsistency of data distribution. Increase the performance of federated learning. Some methods solve the Non-IID problem by introducing domain generalization technology, such as FedDG [30], which enables each client to exploit multi-source data distributions using the information in frequency space and episodic learning at each local client, and then replace direct raw-data sharing by frequency space features, solving the problem of feature drift under the premise of ensuring privacy. This paper proposes an FL framework for textile image classification, which is different from the existing methods that optimize in the original data stage or the gradient aggregation stage. In our approach, patches of feature maps are interpolated between the clients to obtain a more abundant distribution, and then a fusion feature across domains is learned in the same feature space.

2.2 Texture representation

A definition of texture is given in [31], which states a textured area may have non-uniform or changing spatial distribution characteristics of intensity or color. Texture features are widely used in material quality control, image inpainting scene recognition, and other fields [32,33]. In recent years, different methods have been proposed to find an effective solution for texture recognition. Methods based on BoWs assign each descriptor to the nearest visual word in the codebook typically and record the frequency of the occurrence of each visual word while VLAD [34] aggregates the residual vector with a hard-assignment weights function. With the fast development of deep learning, CNNs are widely used for texture recognition. Some approaches utilize orderless aggregation of CNN based features for texture recognition, such as FV-CNN [35], deep texture encoding network (Deep-TEN) [26] and DEPNet [25]. Some methods have designed optimization methods according to the characteristics of texture image spatial structure. For example, MAPNet [36] utilizes a multi-branch architecture to learn visual texture attributes in a mutually reinforced manner progressively. DSRNet [37] leverages spatial dependency among the captured primitives as structural representation for texture recognition. At the same time, CLASSNet [27] utilizes cross-layer statistics and explicit exploitation of statistical self-similarity for texture representation.

Unlike the current designation of a fixed-size space for texture representation in the encoding layer, we propose an MusTEN that defines multiple codebooks with different clustering centers and fuses the multi-channel encoding features through an attention mechanism.

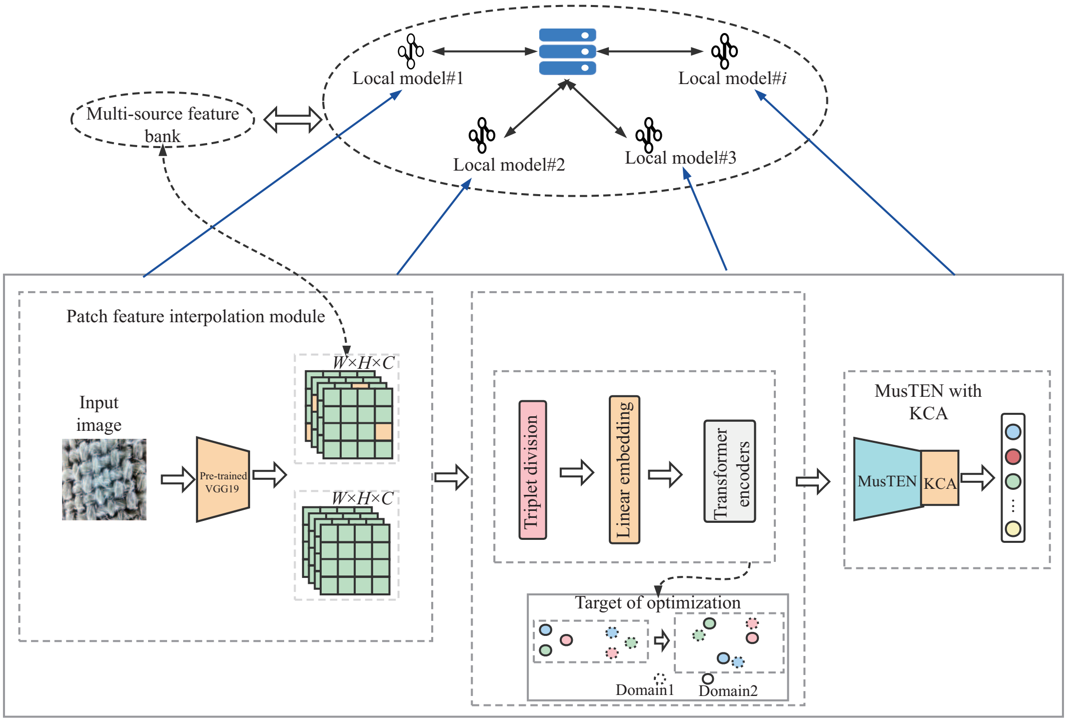

Figure 1 (Color online) Overview of FedTFI. FedTFI is constructed by a PFI module PFI, a TEL module, and an MusTEN module. The deep feature patch is randomly sampled in the PFI module and interpolated crossing domains. Then the interpolated features are jointly represented in the embedding learning module by similarity measurement between samples in low-dimensional space, realizing efficient domain adaptation while ensuring data security. The MusTEN module defines multiple learnable codebooks in different scales to encode the feature descriptors, providing a fusion texture feature with different granularities. A KCA mechanism is embedded in the MusTEN module to further capture keywords in dictionary learning.

3 Proposed Non-IID FL framework

This section introduces the implementation of the patch feature interpolation (PFI) module and then describes the proposed deep TEL module, which is designed to obtain an adaptive representation crossing domains. Finally, we present an MusTEN module with KCA. An overview of FedTFI is shown in Figure 1. We integrate the three proposed modules into the FedAvg [28] framework, realizing the end-to-end collaborative training on Non-IID scenarios.

3.1 PFI module

Different from the existing studies [17, 18, 30] using raw-data or frequency information sharing strategy, we propose a patch feature interpolation module to address the Non-IID problem, which enables the patches of feature maps to transfer between across-domain clients.

Firstly, we design a feature extraction consisting of pre-trained VGG19 layers [38] network together with 1×1 convolutional layers. A group of feature maps is obtained with dimension $W \times H \times C$ by the feature extraction, where W and H are the length and width of the feature map, and C is the number of channels. Then we design an interpolation mechanism to randomly replace the patch in original feature maps with a cross-domain patch from the bank to transmit information between domains.

Specifically, we first construct a semantic feature bank $B = \{B^1, B^2, \ldots, B^K\}$, where each $B^k = \{B_i^k\}_{i=1}^{N_k}$ contains the randomly sampled feature patches with a fixed size in the kth client together with the label, where N_k indicates the number of patch features collected by each client. B is a dynamic queue. When a global epoch starts, we perform feature extraction for each input image, randomly sample a feature map patch in each channel, and save it in B. Next, we interpolate each feature-maps output by feature extraction by randomly accessing one patch-level feature map with the same label. As shown in Figure 1, given a feature vector $x_i^k \in \mathbb{R}^{H \times W \times C}$ from the kth client, for each channel $F_{i,c}^k \in \mathbb{R}^{H \times W}$, we perform interpolation by randomly sampling an item in B_j^n where $n \neq k$ from the feature bank. More formally, for the certain feature map $F_{i,c}^k$ in x_i^k, the process of one-time interpolation with B_j^n can be

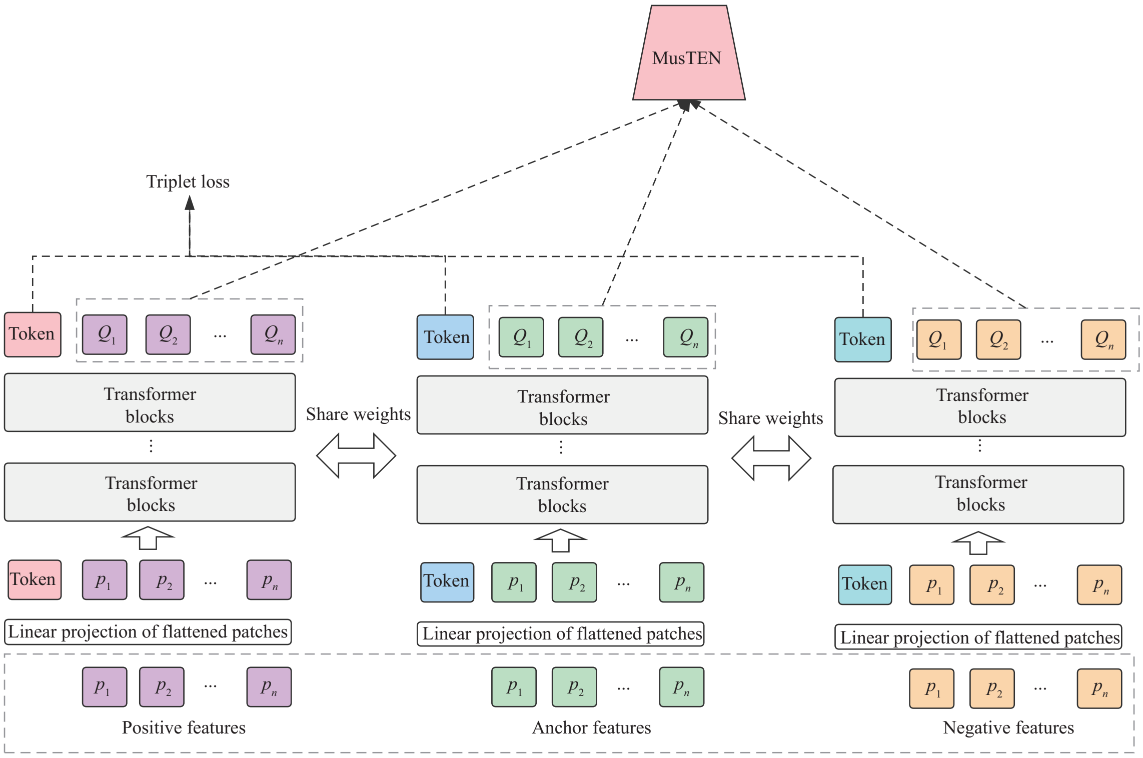

Figure 2 (Color online) The overall structure of the TEL module. TEL comprises three transformer encoders with shared weights and is guided by a triplet loss. We utilize the source features as the anchor, the interpolated feature with the same label as positive features, and different labels as negative features. The fused feature maps are divided into patches of the same size in the encoding process. Then, we utilize a linear projection layer to obtain fixed-dimensional embedding. Next, the embedding of the patches is fed into the transformer encoder, and the output token vector is used for joint representation learning.

represented as

$$F_{i,c}^{k\to n} = F_{i,c}^{k} \odot (1-M) + B_j^n \odot M, \tag{1}$$

$$M = 1_{(h,w\in[-\alpha H:\alpha H,-\alpha W:\alpha W])}, \tag{2}$$

where M is a binary mask, α is a parameter that controls the size of the sampled patch features in B, x denotes the feature-maps output by the feature extractor, c denotes a certain channel in x, and W and H denote the shape of the original feature map. We also define a parameter $p\,(0 \leqslant p \leqslant C)$ to control the number of interpolated feature-maps in each sample.

Such a disturbance provided by PFI allows the original feature to obtain cross-domain information, constructing a richer distribution. In the subsequent representation learning module, we will make the cross-domain feature and the original feature perform joint representation learning in one space to reduce their spatial distance.

3.2 TEL module

In the feature interpolation module, we interpolate the cross-domain local feature map to the local client. Although this approach makes feature distribution diversified, the disturbance may lead the model to fail to converge.

Although the PFI module presents multi-domain distributions to the local client, the gap between cross-domain features may affect the convergence of texture representation. To this end, we design a TEL module. The non-interpolation feature and the interpolation feature are jointly represented in a low-dimensional feature by minimizing the intra-class variation and maximizing the inter-class variation.

From Figure 2 we can see that the TEL module is mainly composed of a three-branch transformer encoder with shared weights. To construct the triple input, we combine the interpolated feature maps output by the PFI module and the original feature maps as a triple input of TEL. We utilize the original feature map as the anchor for each group of input samples. The corresponding interpolated features with the same label as positive samples and those with a different label as negative samples. We first divide the triple feature map into patches, referring to ViT [39], and then we convert the 3D feature map $x_i \in \mathbb{R}^{H\times W\times C}$ into a 2D vector $x_p \in \mathbb{R}^{N\times\left(P^2\cdot C\right)}$, where W and H denote the width and height of the feature map, C denotes the number of channels, and P denotes size of patches. To prevent the disturbance caused by cross-domain information from destroying the original context information, we set the size of p the same as the mask M. We then flatten the patches and map to D dimensions with a 1×1 convolutional embedding layer, resulting in a patch embedding with constant size. Here we utilize the standard transformer architecture as our blocks in Figure 2, which consists of alternating layers of multi-head self-attention (MSA) [40] and multilayer perceptron (MLP) blocks. Layernorm (LN) is applied before every block, while residual connections are deployed after each block. Similar to ViT, we prepend a learnable embedding $x_{\text{emb}} \in \mathbb{R}^{N\times\left(P^2\cdot C\right)}$ to the sequence of embedded patches, which serves as the image representation for the learning of texture embedding. The entire encoding process can be represented as

$$Q_0 = \left[x_{\text{emb}}; x_p^1; x_p^2; \cdots; x_p^N\right] + E_{\text{pos}}, \tag{3}$$

$$Q_l = \text{MSA}(\text{LN}(Q_{l-1})) + Q_{l-1}, \tag{4}$$

$$Q_{l+1} = \text{MLP}(\text{LN}(Q_l)) + Q_l, \tag{5}$$

where $E_{\text{pos}} \in \mathbb{R}^{\left(P^2\cdot C\right)\times D}$ is a position embedding which is added to the patch embedding to retain positional information. Here we use standard learnable 1D position embedding.

Next, we implement cross-domain similarity preserving projections by constraining the distance in feature space between and within classes. We fed the interpolated features into the three-branch network, the weights of the three branches are shared. Then we divide the data into triplets. The positive samples and negative samples all come from interpolated cross-domain features, the features with the same label are defined as positive samples, denoted by x_j^p, and the features with different labels are defined as negative samples, denoted by x_j^n, and anchor feature is denoted by x_j^n. The loss L_t can be expressed as

$$L_t = \frac{1}{N}\sum_{i=1}^{N_2} \max\left(0, m + \left\|F(x_j^a) - F(x_j^p)\right\|\right) - \left\|F(x_j^a) - F(x_j^n)\right\|, \tag{6}$$

where F is the transformer encoder and m is the margin parameter that regularizes the gap between the squared euclidean distance of image pairs.

3.3 An MusTEN with KCA

Traditional deep encoding methods for texture recognition such as Deep-TEN [26] and FV-CNN [41] have a similar structure, including feature extraction, dictionary learning layer, feature encoding layer, and classifier. The texture encoding layer learns a fixed $K\times D$ codebook to encode an orderless representation for texture features, and the hyperparameter K represents the cluster center of learned codebooks. However, the setting of dimension is based on experience. It may not be accurate to express the physical meaning of texture data in a fixed dimension. Unlike previous work, we propose a texture encoding layer containing multiple sub-encoders with multi-scale codebooks. Each sub-encoder refers to a method provided by NetVLAD [42]. The architecture of our proposed model is shown in Figure 3.

Specifically, we define the feature $Q \in \mathbb{R}^{N\times\left(P^2\cdot C\right)}$ output by transformer encoder as N-dimensional visual descriptors $X = \{x_1, x_2, \ldots, x_N\}$. And a set of multi-scale codebooks $C = \{C_1, C_2, \ldots, C_m\}$ is defined as learnable parameters of MusTEN. Noting that $C_i = \{c_1, c_2, \ldots, c_K\}$ containing D-dimensional keywords, for each descriptor x_i, the residual vector can be expressed as $r_{ij} = x_i - c_K$. Unlike the hard-assignment method that assigns weight for each r_i with a single non-zero vector, the soft-weight assignment is used to assign the descriptor to each codeword through a softmax function. And the codebook of the clustering center is learned with a learnable smoothing factor. The output vector encoded by each codebook E_m is $K_m \times D$ dimensional. The encoding process of a codebook can be represented as

$$e_j = \sum_{i=1}^{N} a_{ij} \cdot r_{ij}, \tag{7}$$

where $a = a_{ij}$ is the assigning weight for residual vector $r = r_{ij}$ and can be given as

$$a_{ij} = \frac{\exp(-s_k||r_{ik}||)}{\sum_{j=1}^{K}\exp(-s_j||r_{ij}||)}, \tag{8}$$

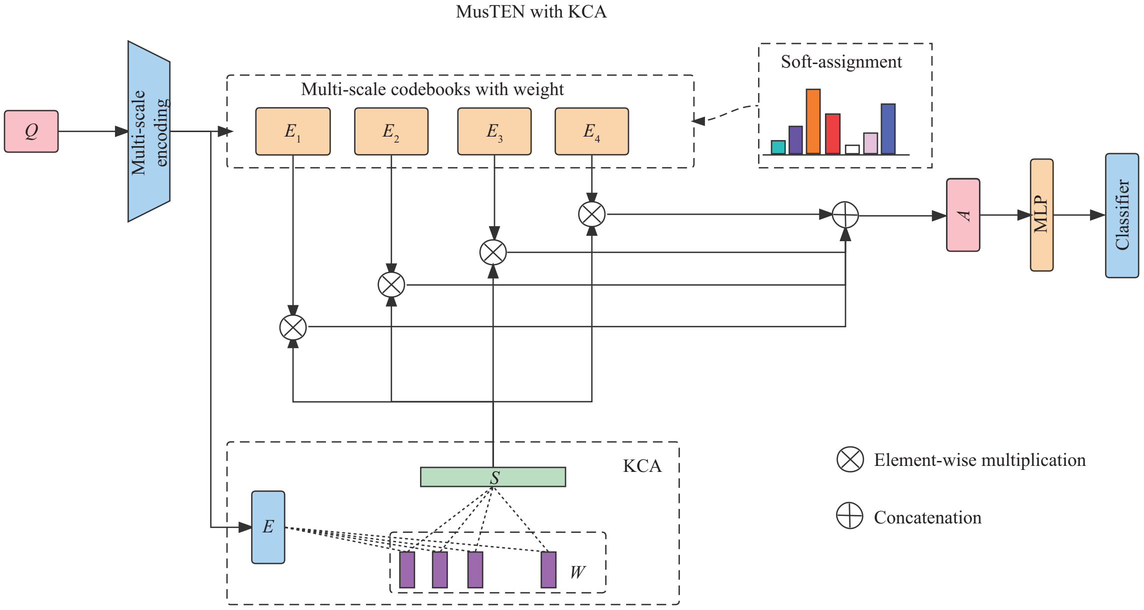

Figure 3 (Color online) The structure of MusTEN with a KCA mechanism. Multi-scale learnable codebooks are defined to encode the feature descriptors, providing representation with varying granularities of texture information. Furthermore, a KCA module is designed to capture the key information in the texture representation to perceive keywords in codebooks.

where $s_1, \ldots, s_k$ are learnable smoothing factors.

We perform the above encoding process using multiple sub-encoders to obtain multi-scale encoding features. The vector with a higher codewords dimension can obtain more texture details, while the vector in the lower dimension can better capture the changing rule of pixels. We concatenate these output vectors directly to form a texture representation with more keywords, which can be represented as

$$E_{\text{out}} = \text{Concat}([E_1, E_2, \ldots, E_M]), \tag{9}$$

where $E_{\text{out}} \in \mathbb{R}^{K\times D}$ is the output of the encoding layer, and K means the sum of the number of sub-encoder keywords. However, the texture representation obtained through the concatenation of multiple encoders increases the keyword dimension, and it is difficult for the model to capture the importance of different keywords. To solve this problem, we propose a keyword consistency attention module embedded after the texture encoding module. The attention weights S for E_{out} can be represented as

$$s = \text{softmax}(\omega^{\text{T}}\tanh(V \cdot E_{\text{out}}^{\text{T}})), \tag{10}$$

where $\omega \in \mathbb{R}^{h\times 1}$ and $V \in \mathbb{R}^{h\times D}$ are trainable parameters for KCA, h is the size of hidden dimension for ω and V, and E_{out} denotes the output of the encoding layer. The embedding A output by MusTEN can be presented as

$$A = s \cdot E_{\text{out}}. \tag{11}$$

Then MLP is utilized to obtain a low-dimensional representation and perform the classification task.

3.4 Federated training

The model is trained with a joint loss, containing triplet loss L_t and a cross-entropy loss L_c. L_t enables the model to have better cross-domain representation performance and obtain more robust feature descriptors. L_c is used to improve the classification ability of the texture coding module for textile images with different components. The total loss can be represented as

$$L_c = \sum_{i=1}^{N} y_i \text{log}(p_i), \tag{12}$$

$$F = L_c + \lambda L_t, \tag{13}$$

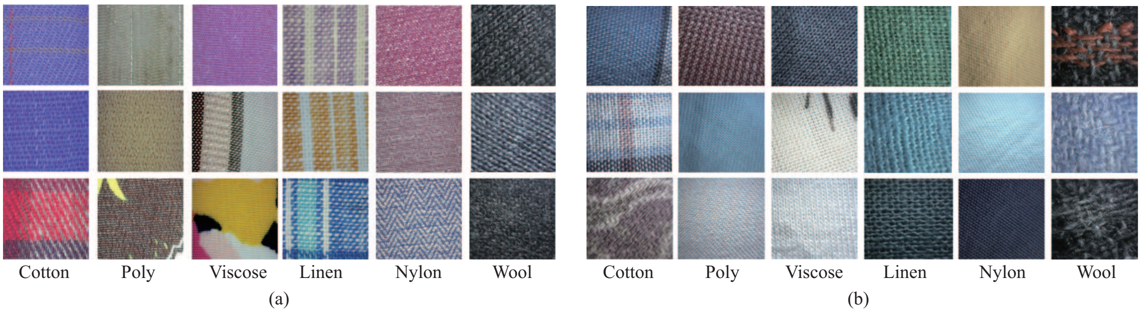

Figure 4 (Color online) Some samples of the high definition textile images. We see that images of the two datasets have obvious differences in view size, magnification, resolution, and shooting angles. (a) Fabric-eyes; (b) fabrics.

where λ is a tradeoff parameter. Assuming there are K clients over which the data is partitioned, $\mathcal{D}_k$ denotes the set of indexes of data points on client k, and the overall loss function can be expressed as

$$f(w) = -\sum_{k=1}^{K} \frac{n_k}{n} F_k(w), \tag{14}$$

where

$$F_k(w) = \frac{1}{n_k} \sum_{r \in \mathcal{D}_k} f_r(w), \tag{15}$$

F_k denotes the local loss of client k, $n_k = |\mathcal{D}_k|$. Referring to FedAvg [28], the local model weights are aggregated at global and in iteration $t+1$ the weights can be given as

$$w_{t+1} = w_t - \eta \sum_k \frac{n_k}{n} g_k, \qquad \text{where} \quad g_k = \nabla F_k(w_t). \tag{16}$$

The iterating of local update can be expressed as

$$w_{t+1}^k = w_t - \eta \nabla F_k(w_t), \tag{17}$$

and the global iteration can be expressed as

$$w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k. \tag{18}$$

The FL framework executes the model training process without compromising user data privacy through the interactive training of multiple participants and central servers.

4 Experiments

4.1 Datasets

To evaluate the performance of the federated learning model in Non-IID scenarios, we construct a feature distribution skew scenario using two textile datasets with differences in the image field of view, angle, and resolution. We have collected 4260 image samples of the fabric-eyes dataset with six widely-used fabric materials, i.e., Cotton, Poly, Viscose, Linen, Nylon, and Wool. The high-definition images of the cloth are acquired with an industrial camera where the focal length and magnification are fixed to ensure that all the acquired textile images are within the same field of view at the same angle. Fabrics [43] is a publicly available database for fine-grained material classification. Our experiment uses the same six materials as the fabric-eyes and consists of a dataset containing 2874 samples. The examples for different materials are shown in Figure 4, and the label distribution is shown in Figure 5.

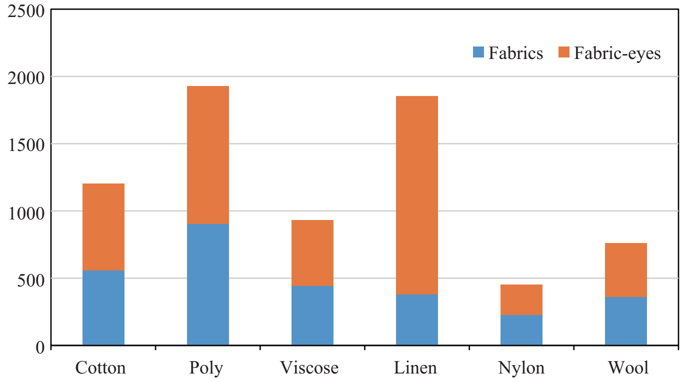

Figure 5 (Color online) Label distribution of the two datasets.

4.2 Experimental setting

4.2.1 *Construction of Non-IID scenarios*

To verify the effectiveness of our framework in different Non-IID scenarios, we divide the clients into M groups, each with N individual clients. We then define four partitioning strategies: label skew, feature skew, and two combined skew scenarios. Specifically, for label distribution skew, similar to [44], we define a variable C, and suppose each group only holds images of C different labels. The different label ids are assigned to each group, and then the images with each label are randomly divided into clients of the corresponding group. In this way, the number of categories held in each group is fixed, with no overlap between the samples. We then use the two datasets with different feature distributions to construct a real-world feature skew scenario. Here we set the number of groups to 2 so that the clients in each group hold data from different sources. For the two combined skew scenarios, we divide the groups by feature offset strategy, and then each client holds specific categories of samples. In our experiments, we set the number of categories C to 1 and 2, respectively.

4.2.2 *Implementation details*

We implement FedAvg and embed the three modules we designed into the network. Table 1 shows the dimensions and specific operations of each module in the framework implementation process. In the PFI module, we firstly utilize pre-trained VGG19 [38] to extract semantic features of raw images. Specifically, we use the first three layers of VGG and a 1×1 convolutional layer to obtain $56 \times 56 \times 128$ feature maps. Then we divide the feature map into 2×2 patches. The random interpolation will be performed on the features of each channel. The entire training process is shown in Algorithm 1.

Algorithm 1 Training procedure for FedTFI algorithm

```
Require: The input image, $I \in W \times H \times 3$;
Ensure: The output labels for classification;
Define mask $M \in \alpha H \times \alpha W$ for PFI;
while $i$<epoch do
    $F$ = Feature_extractor(images);
    Generate the feature bank $B$;
    $F = \mathrm{PFI}(B, F, M)$;
    $x_p, x_n, x_a = \mathrm{Partition}(F)$;
    $Q = \mathrm{TEL}(x_p, x_n, x_a)$;
    $E = \mathrm{MusTEN}(x_p, x_n, x_a)$;
    $A = \mathrm{KCA}(E)$;
    output = Classifier($A$);
    $L_t = \mathrm{TripletLoss}(x_p, x_n, x_a)$;
    $L_c = \mathrm{CrossEntropyLoss}(\mathrm{output}, \mathrm{labels})$;
    $\mathrm{Loss} = L_c + \lambda L_t$;
    $\nabla\theta^K = \mathrm{ComputeGradients}(\mathrm{Loss}, \theta^K)$;
    Send $\nabla\theta^K$ to center server;
    FedAvg($\nabla\theta$);
end while
```

In the TEL module, the dim of the patch embedding layer is 1024. The depth of the transformer

Table 1 The outline of our proposed FedTFI

Layer name	Input size	Output size	Details
Feature_extractor	[3, 224, 224]	[56, 56, 64]	VGG layers and 1 × 1 convolutional layers
Patch divide	[56, 56, 64]	[784, 256]	Randomly sampling the sub-feature maps
Patch embedding	[784, 256]	[784, 1024]	Linear and convolutional layers
Transformer encoder	[784, 1024]	[784 + 1, 1024]	MSA×10
MusTEN encoding	[784, 1024]	[112, 128]	1 × 1 convolution, VLAD based encoders
KCA layer	[112, 128]	[112, 128]	Attention calculation
FC	[112, 128]	Class-num	–

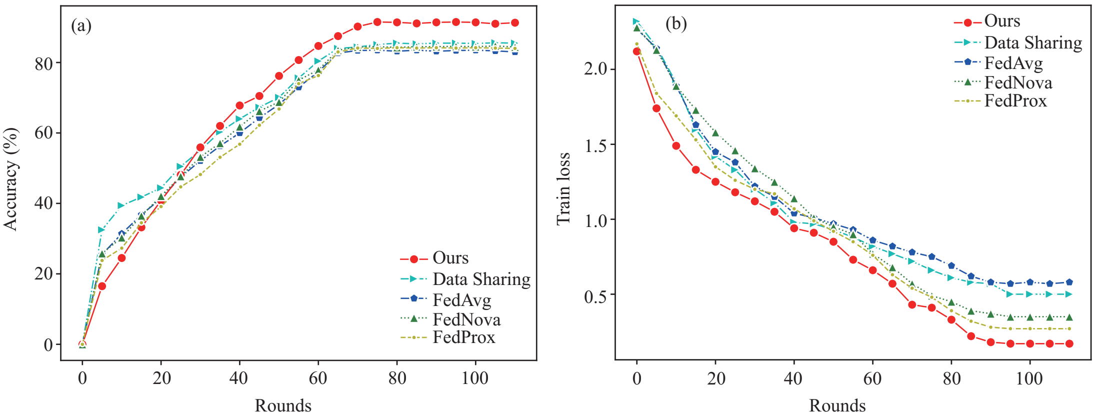

Figure 6 (Color online) Performance evaluation in terms of test accuracy and training loss in the feature skew scenario. Comparison of the five FL methods in terms of (a) test accuracy and (b) training loss.

encoder layer is set as 10, and the number of headers is 8. In MusTEN, we input the 784 × 256 depth feature descriptor obtained from the TEL network. The number of codebooks in MusTEN is set as 4, the number of the keywords in each codebook is defined as 8, 16, 32, and 64, respectively. The dimension of each codebook is 128. Our experiment is deployed on four GPUs of Nvidia GeForce GTX 2080 Ti. The experimental environment is based on PyTorch. We set local-batch size as 128, local epoch as 10, SGD with 1×10^{-3} learning rate, a weight decay of 1×10^{-4} and a momentum of 0.9.

4.3 Comparison among FL models on Non-IID data

To verify the effectiveness of FedTFI, we compare with open-source Non-IID FL benchmarks, including FedProx [20], Data Sharing [17] and FedNova [21]. We also compare our model with the baseline setting FedAvg without considering Non-IID. Meanwhile, we compare our method with state-of-the-art textile & texture feature representation methods, including Deep-TEN [26], DEPNet [25], CU-NET [22] and CLASSNet [27] and deploy them into FL frameworks for textile fiber classification.

Figure 6 shows the test accuracy and training loss of textile fiber identification in the feature skew scenario. We can see that the benchmarks for Non-IID data can improve performance over FedAvg in the case of feature skew. This attributes to their optimized measurements for the feature and labels distribution skew. Moreover, we can see that FedTFI has obvious advantages over other methods in terms of test accuracy and training loss, proving that our model has achieved better convergence. To further prove the convergence of FedTFI, we visualize the F1-score and AUC during the model training process, as shown in Figure 7. FedTFI achieves the best results in terms of all criteria.

We perform textile fiber composition prediction in the constructed real-world feature distribution scenario and visualize the confusion matrix. As shown in Figure 8, compared with the best results achieved in the benchmark method (including Data Sharing, FedNova, and FedProx), our approach has a significant improvement in prediction accuracy. Especially for component pairs that are difficult to predict accurately (cotton and linen, poly and nylon), the heat map shows that the number of our forecast errors has dropped significantly.

As shown in Table 2, compared with the benchmarks, our FedTFI achieves higher performance in the overall four testing scenarios. This benefits from our patch feature interpolation and texture embedding modules, which present domain adaptive distributions to the local client via cross-domain texture repre-

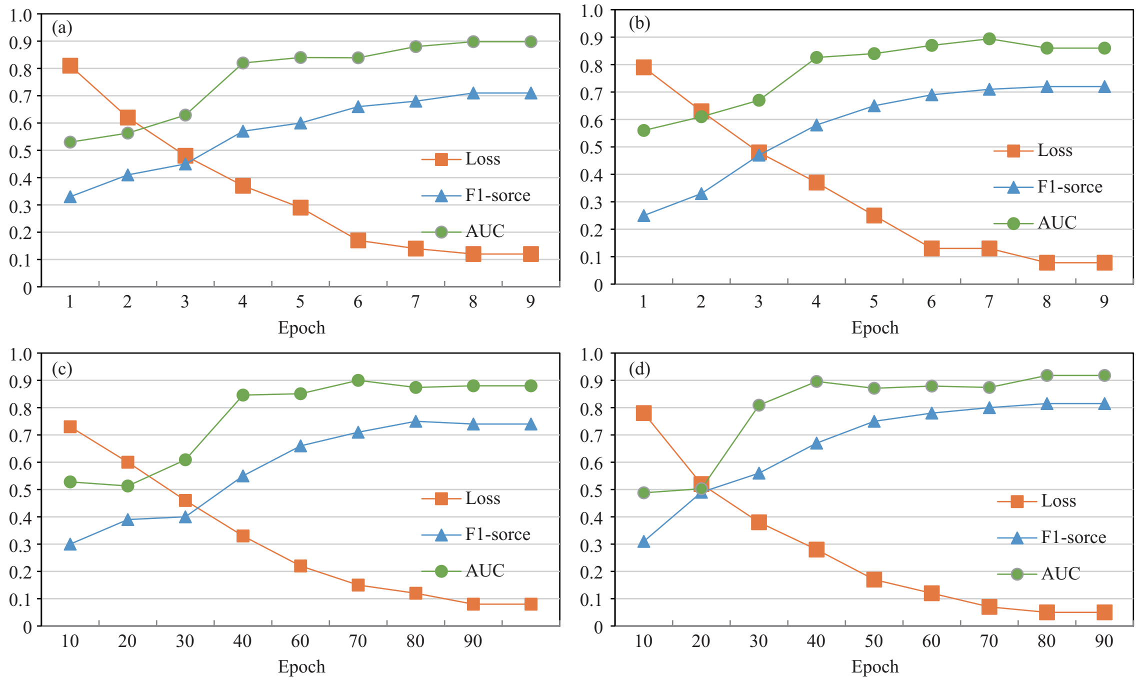

Figure 7 (Color online) Performance evaluation in terms of F1-score, AUC and training loss (scaled-down) in label distribution skew scenario, with respect to training epoch. (a) FedAvg; (b) FedNova; (c) FedProx; (d) ours.

sentation learning. Compared with FedTFI, the local model can only access the individual distribution or fail to integrate different feature distributions in the same feature space for representation. In contrast, FedTFI enables cross-domain features to jointly learn an embedding by similarity measurement between triplet samples in a low-dimension feature space. In addition, the network benefits from MusTEN with KCA, which further enhances the recognition ability of texture features.

Specifically, compared with FedAvg, our FedTFI achieves consistent improvements overall performance increase of 12.5% in label distribution skew ($C = 1$), 8.7% in real-world feature distribution skew, in two combined skew scenarios, 14.8% ($C = 1$) and 10.5% ($C = 2$), respectively.

In feature and label skew scenarios (the first and second columns in Table 2), FedTFI outperforms all the Non-IID optimized benchmarks. The accuracy is significantly improved by 7.6% (67.9% to 75.5% compared with FedProx) and 5.7% (86.7% to 92.4% compared with FedProx). This benefits from optimizing multi-source information fusion at the feature interpolation together with the joint learning of the cross-domain texture feature.

According to the experimental results in the two combined scenarios, our method can still exceed all benchmarks when the labels and feature distribution skew simultaneously. The accuracy is significantly improved by 7.8% (64.9% to 72.7% compared with Data Sharing) and 5.5% (80.8% to 86.3%), respectively.

In addition, different from traditional texture recognition methods, the transformer is utilized in our TEL module. The experimental result shows that when using the transformer as the feature extractor, our FedTFI achieves the best results. Also, we can see that in the Non-IID scenario where the distribution of features and labels are drifting, our model shows a considerable advantage over the CNN-based method. This attributes to the joint learning of PFI and TEL modules, in which the interpolated features are directly fed into the transformer encoder in the form of a patch, which avoids the destruction of the spatial structure of the feature maps. Meanwhile, the TEL module enables the local learning to take full advantage of the multi-source distributions by the joint representation through similarity measure between triplet samples in low-dimensional space.

In addition to the results for classification, we conduct comparative experiments in terms of model size measured by the number of model parameters and efficiency measured by the floating-point operations per second of the model. In Table 3, we see that when compared with other models under the FedAvg framework, the complexity of our model is comparable to others in terms of both criteria.

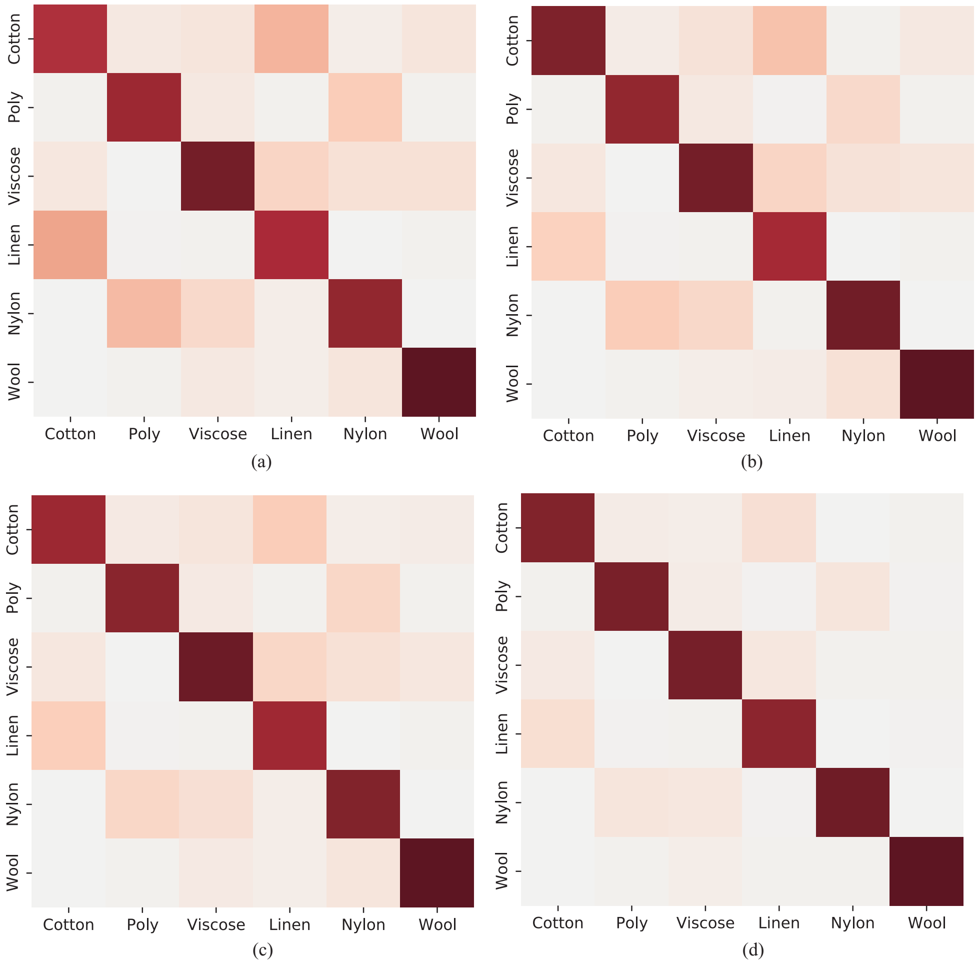

Figure 8 (Color online) Confusion matrices of the four FL algorithms in feature skew scenario, including FedAvg (a), FedNova (b), FedProx (c) and our FedTFI (d). Deep-TEN (ResNet) is used as the texture representation network.

4.4 Ablation analysis

In this subsection, we verify the effectiveness of different modules in our approach, including the PFI module, the TEL module, and the MusTEN.

We decouple the three critical modules in our experiments and test them in the three scenarios. As shown in Table 4, when the PFI and TEL modules are combined, the results are significantly improved. Specifically, the performance can be improved from 80.5 to 89.8 in feature distribution skew, which verifies the effectiveness of the PFI module and TEL module for feature transfer. On the other hand, in two combined skew scenarios, the performance can be improved from 57.6 to 70.6 and 75.05 to 85.8, respectively. However, when the texture embedding learning module is missing, the model performance has a relatively significant decline, which shows that when only using PFI without TEL module, the interpolation of local features can interfere with the learning process and affect the convergence of the model.

We then visualize the low-dimensional features by t-SNE to evaluate the performance of the TEL module. As shown in Figure 9(a), when adding the TEL module in our approach, compared with Figure 9(b), the results for the low-dimensional feature from different domains have a good fusion effect in the feature space, which further proves the effectiveness of texture representation learning module based on TEL.

Next, we explore the effectiveness of the proposed PFI and TEL modules. Firstly, we remove the MusTEN module from the network and compare our approach with the baselines. As shown in Figure 10(a), compared with other FL algorithms, the accuracy is significantly improved when PFI and TEL modules are deployed. After that, we evaluate the effectiveness of the TEL module. We fix the

Table 2 Four Non-IID federated learning algorithms are used as benchmarks, including FedProx, Data Sharing, FedNova, and FedAvg. Meanwhile, we utilize benchmark texture representation methods combined with the FL frameworks, including Deep-TEN, DEPNet, CU-NET, and CLASSNet. We construct feature distribution skew scenario and label distribution skew scenario. Here, $\#C$ denotes the number of labels held by each client in the label skew scenario. The last two columns of data in the table indicate scenarios where label skew and feature skew exist simultaneously.

FL framework	Method	Backbone	Label skew ($\#C = 1$)	Feature skew	Label & Feature skew ($\#C = 1$)	Label & Feature skew ($\#C = 2$)
Data Sharing	Deep-TEN	ResNet18	64.8	81.7	62.8	77.1
	Deep-TEN	ResNet50	67.1	83.6	63.5	78.8
	DEPNet	ResNet18	64.4	82.5	63.2	76.9
	DEPNet	ResNet50	66.1	85.6	64.9	77.5
	CLASSNet	ResNet18	63.9	81.8	62.7	73.1
	CLASSNet	ResNet50	66.3	85.1	64.7	79.6
	CU-NET	DenseNet	66.2	84.7	62.4	78.7
FedProx	Deep-TEN	ResNet18	64.7	81.2	62	78.8
	Deep-TEN	ResNet50	66.8	83.9	64.5	81.4
	DEPNet	ResNet18	65.4	81.5	61.1	77.6
	DEPNet	ResNet50	67.9	84.1	64.7	80
	CLASSNet	ResNet18	65.9	82.4	62.7	74.9
	CLASSNet	ResNet50	66.7	86.3	64.7	80.5
	CU-NET	DenseNet	67	83.7	62	80.1
FedNova	Deep-TEN	ResNet18	63.5	80.2	61.3	76.8
	Deep-TEN	ResNet50	66.5	83.6	63.7	80.7
	DEPNet	ResNet18	63.9	81.8	61.5	77.4
	DEPNet	ResNet50	66	84.4	63.5	80.3
	CLASSNet	ResNet18	64.2	81.1	60.2	75.4
	CLASSNet	ResNet50	67.4	86.7	65.5	80.8
	CU-NET	DenseNet	66.3	84.2	61.2	79.2
FedAvg	Deep-TEN	ResNet18	58.6	80.5	55.4	73.7
	Deep-TEN	ResNet50	61.1	83.2	56.2	75.5
	DEPNet	ResNet18	58.4	80.7	55.3	73.3
	DEPNet	ResNet50	62.3	83.6	57.6	75.6
	CLASSNet	ResNet18	61.7	79.1	57.1	74.1
	CLASSNet	ResNet50	63	83.7	57.9	75.8
	CU-NET	DenseNet	60.8	83.4	57.2	72.5
FedTFI	Ours	Vision transformer	**75.5**	**92.4**	**72.7**	**86.3**

Table 3 Complexity comparison of different models in terms of number of model parameters (Params) and floating-point operations per second (FLOPs) under FedAvg framework

	Backbone (ResNet50)	Deep-TEN	DEPNet	CLASSNet	Ours
Param (≈, M)	23.57	23.91	25.56	23.7	24.6
FLOPs (≈, G)	4.11	4.12	4.11	4.14	4.51

Table 4 Ablation results to analyze the effect of the three proposed modules, including PFI, TEL module, and MusTEN. We superimpose three modules in turn and observe the experimental results

PFI	TEL	MusTEN	Feature skew	Feature skew ($\#C = 1$)	Feature skew ($\#C = 2$)
–	–	–	80.5	57.6	75.0
✓	–	–	79.9	61.0	70.3
✓	✓	–	89.8	70.6	85.8
✓	✓	✓	92.4	72.7	86.3

PFI module and the TEL module in our framework and replace MusTEN with other texture recognition methods, respectively. As shown in Figure 10(b), MusTEN offers apparent advantages compared with other texture representation methods in the feature skew scenario.

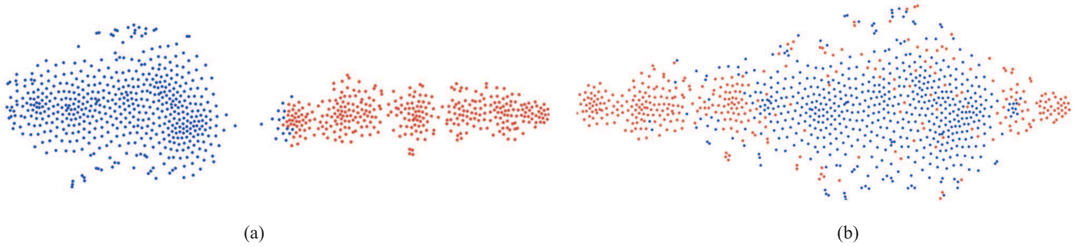

Figure 9 (Color online) t-SNE visualization results. (a) Without TEL; (b) with TEL.

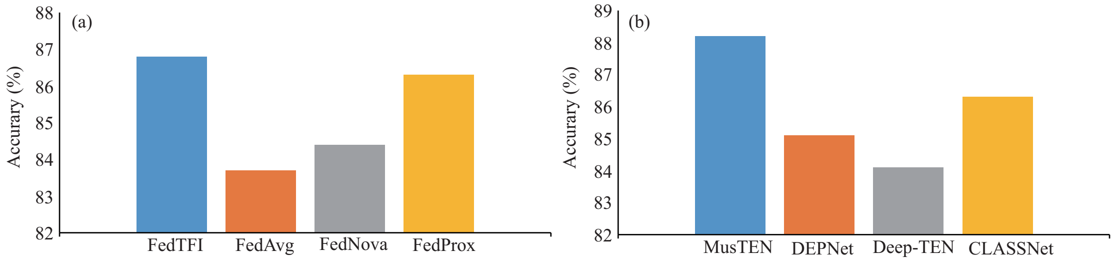

Figure 10 (Color online) Performance comparison in terms of test accuracy. (a) Performance comparison with FL benchmarks for textile fiber identification, including FedTFI (ours), Data Sharing, FedNova, and FedProx; (b) performance comparison with texture representation benchmarks for textile fiber identification, including MusTEN (ours), DEPNet, Deep-TEN and CLASSNet.

5 Conclusion

Data acquired by different sensors in different fabric enterprises cannot follow the IID assumption in the textile industry. To this end, a novel federated framework is proposed to securely identify textile fiber (FedTFI) via cross-domain texture representation based on high-definition fabric images distributed in the IIoT. Here, FedTFI is a Non-IID FL model, constructed by a PFI module, a TEL module, and an MusTEN module. In the PFI module, the deep feature patches are randomly sampled and interpolated among different clients over fabric IIoT. Then, the cross-domain features are jointly represented in the TEL module by minimizing the intra-class variation and maximizing the inter-class variation. Finally, a MusTEN module with a KCA mechanism is designed to define a multi-scale codebook to obtain richer texture features with different granularities.

To verify the effectiveness of the proposed FedTFI, we build Non-IID scenarios using two textile image datasets (one public available and another collected by our team), including label skew, feature skew, and two combined skew scenarios. Experimental results confirm the effectiveness of our proposed model in four Non-IID methods, obtaining state-of-the-art accuracy on two datasets. In addition, we demonstrate that our approach is more robust than the benchmarks through black-box attack experiments.

Our method allows local clients to access multi-source distribution without revealing data privacy. In the future, the proposed FedTFI will be applied to other industries to solve Non-IID texture image data from IIoT. An encryption algorithm will be applied to parameter sharing in the proposed Non-IID FL framework as a near development. Furthermore, the two datasets used in the papers only contain pure fabric fibers. In real applications, fabrics usually contain the composition of two or over two distinctive fibers, such as 50% cotton and 50% linen. We will apply the proposed Non-IID model to multi-label classification tasks to realize the composition identification of blended fabrics.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant No. 62171139) and Zhongshan Science and Technology Development Project (Grant No. 2020AG016).

References

1 Cheng G, Li R M, Lang C B, et al. Task-wise attention guided part complementary learning for few-shot image classification. Sci China Inf Sci, 2021, 64: 120104

2 Li Y Y, Wang H M, Ding B, et al. RoboCloud: augmenting robotic visions for open environment modeling using Internet knowledge. Sci China Inf Sci, 2018, 61: 050102

3 Liu S C, Zhao H, Du Q, et al. Novel cross-resolution feature-level fusion for joint classification of multispectral and panchromatic remote sensing images. IEEE Trans Geosci Remote Sens, 2021. doi: 10.1109/TGRS.2021.3127710

4 Liu S C, Zheng Y J, Du Q, et al. A novel feature fusion approach for VHR remote sensing image classification. IEEE J Sel Top Appl Earth Observations Remote Sens, 2021, 14: 464–473

5 Shen S Q, Zhang K, Zhou Y, et al. Security in edge-assisted Internet of Things: challenges and solutions. Sci China Inf Sci, 2020, 63: 220302

6 Xie G, Shangguan A Q, Fei R, et al. Motion trajectory prediction based on a CNN-LSTM sequential model. Sci China Inf Sci, 2020, 63: 212207

7 Zheng Y J, Liu S C, Du Q, et al. A novel multitemporal deep fusion network (MDFN) for short-term multitemporal HR images classification. IEEE J Sel Top Appl Earth Observations Remote Sens, 2021, 14: 10691–10704

8 Konečný J, McMahan H B, Ramage D, et al. Federated optimization: distributed machine learning for on-device intelligence. 2016. ArXiv:1610.02527

9 Gentry C. Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, 2009. 169–178

10 Dai Z X, Low B K H, Jaillet P. Federated bayesian optimization via Thompson sampling. In: Proceedings of Advances in Neural Information Processing Systems, 2020. 33

11 Karimireddy S P, Kale S, Mohri M, et al. SCAFFOLD: stochastic controlled averaging for federated learning. In: Proceedings of International Conference of Machine Learning, 2020

12 Pietikäinen M, Hadid A, Zhao G Y, et al. Computer Vision Using Local Binary Patterns. Berlin: Springer, 2011

13 Qi Q, Chen X M, Zhong C J, et al. Physical layer security for massive access in cellular Internet of Things. Sci China Inf Sci, 2020, 63: 121301

14 Brisimi T S, Chen R, Mela T, et al. Federated learning of predictive models from federated electronic health records. Int J Med Inf, 2018, 112: 59–67

15 Wang Y S, Tong Y X, Shi D Y. Federated latent Dirichlet allocation: a local differential privacy based framework. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2020. 34: 6283–6290

16 Yang W S, Zhang Y H, Ye K J, et al. FFD: a federated learning based method for credit card fraud detection. In: Proceedings of International Conference on Big Data, 2019. 18–32

17 Zhao Y, Li M, Lai L Z, et al. Federated learning with Non-IID data. 2018. ArXiv:1806.00582

18 Duan M M, Liu D, Chen X Z, et al. Astraea: self-balancing federated learning for improving classification accuracy of mobile deep learning applications. In: Proceedings of IEEE 37th International Conference on Computer Design (ICCD), 2019. 246–254

19 Reisizadeh A, Farnia F, Pedarsani R, et al. Robust federated learning: the case of affine distribution shifts. In: Proceedings of Advances in Neural Information Processing Systems, 2020

20 Li T, Sahu A K, Zaheer M, et al. Federated optimization in heterogeneous networks. 2018. ArXiv:1812.06127

21 Wang J Y, Liu Q H, Liang H, et al. Tackling the objective inconsistency problem inheterogeneous federated optimization. In: Proceedings of Advances in Neural Information Processing Systems, 2020. 33

22 Feng Z L, Liang W X, Tao D C, et al. CU-Net: component unmixing network for textile fiber identification. Int J Comput Vis, 2019, 127: 1443–1454

23 Kampouris C, Zafeiriou S, Ghosh A, et al. Fine-grained material classification using micro-geometry and reflectance. In: Proceedings of European Conference on Computer Vision, 2016. 778–792

24 Benedykciuk E, Denkowski M, Dmitruk K. Material classification in X-ray images based on multi-scale CNN. Signal Image Video Process, 2021, 15: 1285–1293

25 Xue J, Zhang H, Dana K. Deep texture manifold for ground terrain recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2018. 558–567

26 Zhang H, Xue J, Dana K. Deep TEN: texture encoding network. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2017. 708–717

27 Chen Z L, Li F, Quan Y H, et al. Deep texture recognition via exploiting cross-layer statistical selfsimilarity. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021. 5231–5240

28 McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of Artificial Intelligence and Statistics, 2017. 1273–1282

29 Huang Y T, Chu L Y, Zhou Z R, et al. Personalized cross-silo federated learning on Non-IID data. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2021. 7865–7873

30 Liu Q D, Chen C, Qin J, et al. FedDG: federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021. 1013–1023

31 Mi H B, Xu K L, Feng D W, et al. Collaborative deep learning across multiple data centers. Sci China Inf Sci, 2020, 63: 182102

32 Dong X H, Dong J, Sun G, et al. Learning-based texture synthesis and automatic inpainting using support vector machines. IEEE Trans Ind Electron, 2019, 66: 4777–4787

33 Ferreira M J, Santos C, Monteiro J. Cork parquet quality control vision system based on texture segmentation and fuzzy

grammar. IEEE Trans Ind Electron, 2009, 56: 756–765

34 Jégou H, Douze M, Schmid C, et al. Aggregating local descriptors into a compact image representation. In: Proceedings of 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010. 3304–3311

35 Cimpoi M, Maji S, Vedaldi A. Deep filter banks for texture recognition and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. 3828–3836

36 Zhai W, Cao Y, Zhang J, et al. Deep multiple-attribute-perceived network for real-world texture recognition. In: Proceedings of the IEEE International Conference on Computer Vision, 2019. 3613–3622

37 Zhai W, Cao Y, Zha Z-J, et al. Deep structure-revealed network for texture recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020. 11010–11019

38 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014. ArXiv:1409.1556

39 Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16 × 16 words: transformers for image recognition at scale. 2021. ArXiv:2010.11929

40 Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: Proceedings of Advances in Neural Information Processing Systems, 2017. 5998–6008

41 Hu H, Kang W X, Lu Y T, et al. FV-Net: learning a finger-vein feature representation based on a CNN. In: Proceedings of International Conference on Pattern Recognition, Beijing, 2018. 3489–3494

42 Arandjelovic R, Gronat P, Torii A, et al. NetVLAD: CNN architecture for weakly supervised place recognition. In: Proceedings of Conference on Computer Vision and Pattern Recognition, 2016. 5297–5307

43 Kampouris C, Zafeiriou S, Ghosh A, et al. Fine-grained material classification using micro-geometry and reflectance. In: Proceedings of European Conference on Computer Vision, 2016. 778–792

44 Li Q B, Diao Y Q, Chen Q, et al. Federated learning on Non-IID data silos: an experimental study. 2021. ArXiv:2102.02079