Transferable face image privacy protection based on federated learning and ensemble models

Jingjing Yang^{1,2,3} · Jiaxing Liu² · Runkai Han² · Jinzhao Wu^{1,4,5}

Received: 2 February 2021 / Accepted: 11 May 2021 / Published online: 2 June 2021
© The Author(s) 2021

Abstract

Face image features represent significant user privacy concerns. Face images cannot be privately transferred under existing privacy protection methods, and data across various social networks are unevenly distributed. This paper proposes a method for face image privacy protection based on federated learning and ensemble models. A federated learning model based on distributed data sets was established by means of federated learning. On the client side, a local facial recognition model was obtained by local face data training and used as the input of PcadvGAN to train PcadvGAN for several rounds. On the server side, a parameter aggregator based on a differential evolutionary algorithm was established as the discriminator of PcadvGAN server, and a client facial recognition model was ensembled simultaneously. The discriminator of the PcadvGAN server experienced mutation, crossover, and interaction with the ensemble model to reveal the optimal global weight of the PcadvGAN model. Finally, the global optimal aggregation parameter matrix of PcadvGAN was obtained by calculation. The server and the client shared the global optimal aggregation parameter matrix, enabling each client to generate private face images with high transferability and practicality. Targeted attack and non-targeted attack experiments demonstrated that the proposed method can generate high-quality, transferable, robust, private face images with only minor perturbations more effectively than other existing methods.

Keywords Face image privacy protection · Federated learning · Ensemble model · Adversarial examples · Transferability

Introduction

Social network usage prevails among Internet users wishing to express themselves, contact friends, and conduct business activities. Unfortunately, the leakage of private information is a relatively common occurrence. For example, in 2018,

the private information of over 50 million Facebook users was leaked and used by an outside company for illegal purposes [1]. Regarding the personal information of Internet users, most social networks lack effective privacy protection methods. Criminals and unauthorized parties can obtain the personal information of Internet users for a variety of purposes. Images containing users' facial information characteristics can be used for facial recognition and identity authentication applications [2], and thus can also be used to obtain a user's private data by means of data mining on social networks [3–5]. There are large numbers of public face images present on social media which do not fall under any download limitations. If they are extracted by criminals for nefarious purposes, users' privacy, security, and property may be at risk [6, 7].

Considering the frequency with which private user information is leaked, actively disturbing the features in a face image may effectively safeguard the user's privacy [8]. Traditional face image privacy protection methods involve directly modifying or deleting sensitive regions of the face image, which can damage the useful information on the

✉ Jinzhao Wu
himrwujzh@aliyun.com

- ¹ Chengdu Institute of Computer Applications, Chinese Academy of Sciences, Chengdu 610041, People's Republic of China
- ² School of Information Science and Engineering, Hebei North University, Zhangjiakou 075000, People's Republic of China
- ³ University of Chinese Academy of Sciences, Beijing 100049, People's Republic of China
- ⁴ Guangxi University, Nanning 530000, People's Republic of China
- ⁵ Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Guangxi University for Nationalities, Nanning 530006, People's Republic of China

image and destroy its usability [9]. The basic principle of the current most common face image privacy protection approach is to slightly disturb the image characteristics and reduce the accuracy of a potential facial recognition system while ensuring that the image can still be used on social media. Existing methods primarily include adversarial sample attacks based on gradient attacks and the generative adversarial network (GAN). Liu et al. [10] and Linardos et al. [11] used Fast Gradient Notation (FGSM) [12] to attack a facial recognition model, added slight perturbation to the gradient of the model to reduce the accuracy of the model, and protected private face images accordingly. The main drawback to FGSM is that the added perturbation is easily filtered by the median, and that its robustness is relatively poor.

There have been other valuable contributions to the literature. Xiao et al. [13] proposed the generation of adversarial samples based on a GAN, where adding tiny perturbations to the image via GAN drives the target recognition model toward misclassification. He et al. [14] proposed substituting U-NET for the GAN generator (PriGAN), where the generated private face image reduces the accuracy of the recognition network in social networks without destroying the usability of face images. Yang et al. [15] proposed a face image privacy protection method (PcadvGAN) based on adversarial-blocking principal components, where slight perturbation is added to the principal components of the blocked face image to protect the facial features in the image from being easily extracted while ensuring its usability.

The structures of the potential facial recognition models used by criminals cannot be predicted. An adversarial example attack on a potential facial recognition model is, essentially, a black-box attack. When perturbing the features in a face image, all of the above adversarial sample attack methods seek to reduce the accuracy of the potential facial recognition model in social networks via single-model adversarial training and transferable adversarial examples. However, an experiment by Yang et al. [15] showed that if the potential facial recognition model in the social network differs significantly from the single-model structure adopted in the adversarial sample attack, the adversarial sample is not transferable enough to resist a black-box attack. In other words, previously established methods have low adversarial transferability leading to an ineffective decrease in the accuracy of the facial recognition model in social networks, and thus, ineffective privacy protection effects.

Liu et al. [16] and Tramèr et al. [17] established a method wherein multiple models of different structures are simultaneously subjected to adversarial training to generate adversarial examples with the same strength, which can effectively improve the transferability of adversarial examples and protect private face images. However, this method requires high-quality face image training data in

large quantities; data from major social networks must be shared and integrated for it to function properly. In consideration of industry competition and complex management procedures, a given social network only allows user data to be stored in its own server database. As a result, it is difficult to integrate face image data from different social networks. Even when face image data are successfully integrated, the transferability and robustness of the face image privacy protection method cannot be guaranteed.

The privacy protection of transferable face images was investigated in this study based on federated learning and ensemble models. In the existing method, the weight of each model parameter in the ensemble model is equal. The method we propose regards the generation process of face privacy image as a multi-objective optimization problem, that is, the generated face privacy image is similar to the original face image; it also must mislead multiple face recognition models with different structures. When multiple models of different structures are trained at the same time, the weight of each model is calculated by the server version of the PcadvGAN. The discriminator of the server version of the PcadvGAN uses the differential evolutionary algorithm to solve the multi-objective optimization problem and calculates the optimal weight of the integrated model.

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

1. A facial recognition model based on the ensemble client was employed for training. A parameter aggregator based on the differential evolutionary algorithm was used as the discriminator for PcadvGAN+. After multiple rounds of mutation, crossover, and selection, the optimal weight of the ensemble model was calculated. The face image privacy protection model following this parameter fusion shows superior practicability and transferability to those of other methods.
2. The private face image generated with the proposed method shows a certain degree of adversarial transferability for facial recognition models with defensive strategies.
3. Distributed training of the proposed method was performed using federated learning. No face image data were required to be uploaded on the client's social network during the training process. The structure of the client's facial recognition model is unknown to the server throughout to ensure user privacy and data security.
4. If a user is new to the social network or has lost their data under other circumstances, the proposed method can use their data from other social networks to generate a private face image for them.

Related work

PcadvGAN

As mentioned above, features in a given face image create a strong likelihood of private user data leakage. Yang et al. [15] built the PcadvGAN method based on the principal component of adversarial patches. After the face image is divided into blocks, the principal components of the block image are extracted and a small perturbation is added. An image similar to the original face image is generated with the GAN. The tag of the face image generated by the driver generator of the latent recognition network differs from that of the original face image. The loss function of PcadvGAN is calculated as follows:

$$\mathcal{L} = \mathcal{L}_{\text{GAN}} + \chi_1 \mathcal{L}_{\text{adv}} + \chi_2 \mathcal{L}_N + \chi_3 \mathcal{L}_{\text{sim}}, \quad (1)$$

where χ_1, χ_2, χ_3 are the hyperparameters of the loss function. $\mathcal{L}_{\text{GAN}}$ is the GAN loss function. The purpose of $\mathcal{L}_{\text{GAN}}$ is to use the discriminator to drive the face image data generated by the generator toward the same distribution as the original data set. $\mathcal{L}_{\text{adv}}$ is the loss function of the misleading rate of the potential target network, which can be calculated as follows:

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{X'} \ell_f(X', t, t'). \quad (2)$$

The purpose is to make the face image generated by the generator mislead the target face recognition model. The private face image X' is generated by PcadvGAN generator. If the private face image X' can mislead the output of the face recognition model f to be the label t' instead of the original label t , then $\mathcal{L}_{\text{adv}}$ returns a smaller value, otherwise it returns a larger value. $\mathcal{L}_N$ is the loss function that controls the perturbation size and prevents the distortion of the generated face image. $\mathcal{L}_{\text{sim}}$ is calculated as follows:

$$\mathcal{L}_{\text{sim}} = \max \left(\mathbb{E}_{X, X'} \left(\frac{(40 - \text{PSNR}(X, X'))}{40} \right), 0 \right). \quad (3)$$

$\mathcal{L}_{\text{sim}}$ is the loss function for assessing pixel similarity between the generated private face image and the original face image, which can make the generated face privacy image very similar to the original face image. This method is superior to other methods in terms of the quality of the generated face image, the operation speed, and the reduction in accuracy of the target recognition network. Based on PcadvGAN, a PcadvGAN server (PGS) was designed in this study for improved transferability under the framework of federated learning.

Transferability of adversarial examples

Models such as those discussed here are similar at the decision surface of the local space, so the error generated by

the adversarial example can be generalized to other models of different structures [12]. This is the so-called “transferability” of the adversarial example, a term often used in reference to black-box attacks [18]. The concept adversarial example transferability was first proposed by Szegedy et al. [19], where training models with the same structure or different structures on different subsets of the MNIST data set results in target models with high-confidence misclassification. Papernot et al. [20, 21] studied the transferability of adversarial examples between deep neural networks and other models, and launched black-box attacks on Amazon, Google, and MetaMind via the adversarial example transferability of alternative models.

Unlike the method of training alternative models, the method proposed by Cheng et al. [18] centers on a transfer gradient introduced a priori into query-based attacks, where the optimal parameters are derived theoretically to reduce the number of queries required for a successful black-box attack. Carlini et al. [22] proposed a method to generate adversarial examples under different norms which has shown strong transferability experimentally. Moosavi-Dezfooli [23] proposed a universal adversarial perturbation algorithm, where adding the same perturbation to different training data results in misclassification regardless of the specific training data input.

Liu et al. [17] proposed training based on ensemble models for large-scale data sets which achieves adversarial example transfer attacks under multiple models using the transferability of adversarial examples. The training successfully performed black-box attacks on Clarifai.co based on non-targeted adversarial examples and targeted adversarial examples when the specific parameters of the target model and the training set were unknown. The face image data in social networks are classified as large-scale data sets. The proposed method is an improved version of the method proposed by Liu et al. This method improves the transferability of adversarial examples and is suitable for generating private face images while ensuring the feasibility and privacy of face data sharing.

Federated learning

The objective of federated learning is to establish a training model based on distributed data sets under strict privacy protection conditions. This model solves the problem of data islands and enables multiple parties to process big data collaboratively [24–26]. Federated learning eliminates the need for concentrating the facial image data in social networks at a central storage point. Rather, each social network organization trains its own model and ultimately produces a global training model by means of aggregation.

Previous research on federated learning has centered primarily on dealing with statistical challenges [27, 28] and

protecting user privacy [24, 29, 30]. McMahan [31], for example, proposed a master–slave architectural federated learning model where each client trains a deep learning model locally using its own data, and then uploads the parameters of the properly trained deep learning model to the server. The server constructs a global model using the federated average method and shares it with the client, thus well protecting user privacy and data security.

Manually designing deep neural networks requires a great deal of expertise in the field of deep learning. Scholars have begun to prioritize automated machine learning (Auto ML), especially neural architecture searching (NAS). Federated NAS requires maximizing the model performance and minimizing the payload to be transferred between the server and clients [32]. Zhu et al. [33] regarded it as a multi-objective optimization problem and optimized a neural network model structure for federated learning using a multi-objective evolutionary algorithm.

Inspired by the studies discussed above, we sought to improve the master–slave architecture federated learning model and optimized the PcadvGAN using a differential evolutionary algorithm for enhanced parameter aggregation effects. The proposed method allows various social networks to collaborate on a global model for face image privacy protection without sharing their own users' data.

Face image privacy protection based on federated learning and ensemble model training

Problem definition

Assuming that there are N social networks, that is, user data holders $\{S_1, S_2, \dots, S_N\}$, each has a face image data set $\{D_1, D_2, \dots, D_N\}$; there are N sets of facial recognition models in the social network $F = \{f_1, f_2, \dots, f_N\}$. Criminals have L sets of facial recognition models $F' = \{f'_1, f'_2, \dots, f'_L\}$. If $D = D_1 \cup D_2 \cup \dots \cup D_N$. Any face image X , $X \in D$ may be taken, that is, the face image X can be trained and recognized by the facial recognition model sets F and F' . X' is the private face image X after being processed by the privacy protection method. X' and the face image X should be similar visually. For any potential facial recognition model $f_i \in F \cup F'$, however, the difference between face image X and privacy image X' is the preferably more significant—that is, the private face image X' may result in misclassification with high confidence by any potential recognition model f_i , as shown, for example, in Formula (4):

$$\forall X \in D, \exists X', \min ||X - X'||_2$$

$$s.t. \forall f \in F \cup F', f(X) \neq f(X') = T. \quad (4)$$

If the label of T is not specified, Formula (4) is suitable for non-targeted misdirecting attacks, i.e., when the private face image X' to be identified by the potential facial recognition model f_i is not specified. If specify the label of T , Formula (4) is suitable for targeted misdirecting attacks. The private face image X' is misclassified as T by the potential recognition model f_i .

Model structure based on federated learning

A diagram of the model supporting the proposed method is given in Fig. 1. The arrows in Fig. 1 represent the direction of data transmission. The server is an honest and reliable third-party secure computing node in this case, and the client is N social network, the user data holder $\{S_1, S_2, \dots, S_N\}$. All face data are stored on the client. The client does not need to upload face image data at any point during the model training process, but the server and the client do need to interact several times. As described below, the proposed method is operated in eight steps. These eight steps are also marked in Fig. 1.

The flow of the proposed method is shown in Table 1. To allow the server to generate private face images and prevent any leakage of the client's private data, both the server and the client have the same public face datasets (e.g., VGGFACE or self-built datasets). When the clients train PcadvGAN, they add the public face datasets to the training sets and the test sets.

In Step 1, the encrypted user ID alignment technique [33] is applied to manage user group inconsistency in social networks. The user IDs in social networks are aligned to ensure that the label spaces of overlapping data of N clients are consistent while protecting user privacy and not leaking social network data.

In Step 2, the facial recognition model may be defined by the client itself or randomly selected from the server facial recognition model library. The training datasets and model structure of the client facial recognition model are unknown to the server. The proposed method is based on the training of the facial recognition model of the ensemble client. To improve the transferability of the adversarial example of the private face image, the facial recognition model of the client needs a relatively high recognition rate.

In Step 4, to improve the entire model's training efficiency and reduce the number of communications between the server and the client, PcadvGAN is trained several times and the loss function is ensured to be relatively stable before Step 5 can begin.

In Step 6, the PGS generator loads the aggregation parameters $\bar{W}$, selects face images from the public face datasets pre-stored on the server, generates adversarial samples to perturb the original face image, and produces a private face image.

Fig. 1 Model structure of face image privacy protection method based on federated learning and ensemble model training

Table 1 Flow of the proposed method

Flow

1. The encrypted user ID alignment technique aligns user IDs in clients
2. The client uses local face data to train the facial recognition model $f_i \in F$
3. The server issues the initial parameters $\bar{W}$ of PcadvGAN to the client
4. The recognition model f serves as the attack model of the client. PcadvGAN Formula (1) is used as the loss function to train the client's PcadvGAN model $P = \{P_1, P_2, \dots, P_N\}$
5. The server receives the parameters $W = \{W_1, W_2, \dots, W_N\}$ of the model $P = \{P_1, P_2, \dots, P_N\}$ uploaded by N clients
6. The PcadvGAN server (PGS) aggregates the parameters $W = \{W_1, W_2, \dots, W_N\}$ uploaded by the client and generates the aggregated parameters $\bar{W}$. PGS then issues the private face image to the client
7. The client evaluates the private face image and uploads the evaluation results to the server. If the evaluation structure is not ideal, return to Step 6
8. After the global model achieves a satisfactory result, each client receives the global optimal model parameter $\bar{W}$ issued by the server and uses it to update the PcadvGAN client model

end

Steps 6 and 7 involve aggregation of the server-side global model parameters using PGS. The private face image generated by PGS has high transferability after training, which may also lead to misclassification of the facial recognition model set F' by criminals with high confidence.

If User A is a new user of a certain client (social network), the client's facial recognition model cannot recognize them, and its PcadvGAN cannot generate a private face image X' corresponding to the user. However, if User A has a large amount of data in other clients, the proposed method allows this client's PcadvGAN to generate private face images X' for User A after loading the parameters issued by the server.

PcadvGAN sever

In the scenario discussed here, the server performs parameter aggregation and performance evaluation for the global model using PcadvGAN server (PGS). The network architecture of PGS is shown in Fig. 2. The generator structure of PGS is exactly the same as that of PcadvGAN. After the original face image X is blocked and subjected to PCA, the perturbation generated by the generator is added and a private face image X' is generated via PCA

Fig. 2 PcadvGAN server network architecture

inverse transformation and image uniting. Unlike PcadvGAN, the discriminator of PGS is a parameter aggregator based on the differential evolutionary algorithm which does not need to oppose the potential recognition model f . The parameter aggregator optimizes Formula (5) through Algorithm 1 to drive the generator to generate a private face image with high transferability. The model network parameters $\bar{W}$ are reloaded each time before the PGS generator generates a private face image X' . $\bar{W}$ is provided in this case by the parameter aggregator based on the differential evolutionary algorithm.

Based on ensemble model training, PGS generates adversarial examples for N client facial recognition models F to improve the transferability of private face images in the facial recognition model set F' used by criminals. The loss function of the PGS discriminator is

$$\mathcal{L}_D = \arg \min_{X'} - \log \left(\left(\sum_{i=1}^N f_i(X') \right) \times 1_T \right), \quad (5)$$

where T is the misclassification label of the adversarial example of targeted misdirecting attacks and 1_T represents the one-hot encoding of label T . $f_i \in F$ is the facial recognition model of the client. The function of $\mathcal{L}_D$ is that the private face image X' misclassifies F with high confidence.

The complete loss function of PGS is

$$\mathcal{L} = \mathcal{L}_D + \chi \mathcal{L}_{sim}, \quad (6)$$

where $\mathcal{L}_{sim}$ is the constraint condition for the private face image X' and the original face image X in PcadvGAN at the pixel level; χ is the hyperparameter. A smaller value of $\mathcal{L}_{sim}$ may lead to higher similarity between the generated private face image X' and the original face image.

Parameter aggregator based on differential evolutionary algorithm

The parameter aggregator based on the differential evolutionary algorithm was used as the discriminator of PGS to optimize Formula (5), so that, F , the facial recognition models of all clients, were misclassified with high confidence. Formula (5) is evidently non-differentiable, and F , the facial recognition models of N clients, are unknown to the server. The differential evolutionary algorithm does not require the objective function to be differentiable or known to find the optimal solution, so it is a workable method to solve the complex multi-model optimization problem in Formula (5) [34, 35].

Table 2 shows the server algorithm flow of the parameter aggregator using the differential evolutionary algorithm optimization technique in Formula (6).

Client function(X): Select the best individual of this round according to the client facial recognition model f . For f in F , return $\mathcal{L}_{adv}(X)$.

Client function run on the client. The facial recognition models have been trained using the client face datasets and run on the client. Each model may have a different structure but both have high recognition accuracy. The face recognition model parameters do not participate in the gradient update during Algorithm 1, but rather, they only return the recognition results to the two functions.

g is the algebra of the weight population, i is the individual label in the weight population, the value range of i is defined according to the training intensity m , K_i is the individual of the weight population, $1 \leq i \leq m$ is the weight matrix of the i th set, j is the gene in the individual K_i , $K_{j,i}$ is the weight of the model parameter matrix W_j , and $1 \leq j \leq N$, that is, N is the total number of genes in the individual; $iter$

Table 2 Server-side algorithm flow of parameter aggregator based on differential evolutionary algorithm

---

Algorithm 1

Input: $W = \{W_1, W_2, L, W_N\}$, PcadvGAN model parameters uploaded by the client.

Output: Aggregated model parameters $\bar{W}$.

//Use (6) to initialize the weighted population matrix $K = \{K_1, K_2, L, K_m\}$ of the model

parameter W . In the i -th population weight matrix, $\sum_{j=1}^N K_{j,i} = 1$.

$$K_{j,i}(0) = K_{j,i}^{\min} + \text{rand}(0,1) \cdot (K_{j,i}^{\max} - K_{j,i}^{\min}) \quad (7)$$

for g in iter **and for** i in m :

// Use (7) to mutate the weighted population matrix.

$$v_i(g+1) = K_{i1}(g) + Q \cdot (K_{i2}(g) - K_{i3}(g)) \quad (8)$$

$$i \neq r_1 \neq r_2 \neq r_3, r_1, r_2, r_3 \in m$$

end for

for i in m **and for** j in N :

//Cross over the genes in K_i individuals.

if $\text{rand}(0,1) < CR$ **then** $u_{j,i}(g+1) = v_{j,i}(g+1)$

else $u_{j,i}(g+1) = K_{j,i}(g)$

end for

for i in m :

// Calculate aggregate parameter matrices $\bar{W}_k$ and $\bar{W}_u$ respectively.

$$\bar{W}_k = K_i \times W \quad \text{and} \quad \bar{W}_u = u_i(g+1) \times W$$

// The generator of PGS loads the parameters $\bar{W}_k$ and $\bar{W}_u$, respectively, and generates

private face images X'_k and X'_u according to the attacked tags.

$X'_k = \text{PGS}(\bar{W}_k)$

$X'_u = \text{PGS}(\bar{W}_u)$

//Obtain the values of L_{adv} loss corresponding to the face images X'_k and X'_u in the client

PcadvGAN.

for p_i in P :

$$\alpha = \sum \text{client function}(X'_k)$$

$$\beta = \sum \text{client function}(X'_u)$$

end for

if $\alpha < \beta$

$K_i(g+1) = K_i$

else

$K_i(g+1) = u_i(g+1)$

end for

// Traverse the weighted population matrix K to obtain the corresponding L_{adv} values.

for i in m :

$$\bar{W}_k^i = K_i \times W$$

$$X'^i_k = \text{PGS}(\bar{W}_k^i)$$

for p_i in P :

$$\alpha_i = \sum \text{client function}(X'^i_k)$$

end for

// Select the optimal solution $\hat{K}$.

$$\alpha_e = \min(\alpha_i)$$

$$\hat{K} = K_e$$

return $\bar{W} = \hat{K} \times W$

end

---

is the number of iterations of the evolutionary algorithm, v is the weight population of the mutation, and Q is the scaling factor during mutation. $Q = \text{rand}(0, 1)$ was defined in the experiment. u is the cross-weighted population, CR is the crossover probability, and $f \in F$ is the client facial recognition model.

The training process of this aggregator is shown in Fig. 3. The most significant difference from the ensemble model training proposed by Liu et al. [17] is that according to the differential evolutionary algorithm, $\hat{K}$, the optimal weight of the attacked target client facial recognition model F , is calculated following mutation, crossover, and selection before the best aggregation parameter matrix $\bar{W}$ of this round of training is derived. After the PGS generator loads the parameter matrix $\bar{W}$ to generate a new private face image X' , the next round of training is initiated repetitively until a satisfactory training result is derived.

Experiment

Experimental environment and data set

The experimental test server hardware environment was an Intel i7-8700 K CPU with 32 GB memory and an NVIDIA GeForce RTX 2080Ti graphics card, plus an Intel i7-9750H CPU client software environment with 16 GB memory and an NVIDIA GeForce RTX 2060 graphics card. The network environment was a 1000 M local area network. The software environment was the Windows10 64bit operating system and the Google Tensorflow framework was used. Codes were written in Python.

Two types of experimental data sets were used, the VGGFACE and VGGFACE2 public face data sets. VGGFACE contains more than 2.6 million face images of 2,622 individuals [36]. VGGFACE2 contains more than 3.31 million face images for 8,631 people [37]. Some of the face data of VGGFACE and VGGFACE2 overlap. Before training the client facial recognition model, the overlapping portion of the face data was aligned as label data.

The population generation of the differential evolutionary algorithm was set to $g=40$ and the number of individuals in the population to $m=14$. The scaling factor during mutation was set to $Q=0.5$. The crossover probability CR was operated with the adaptive adjustment strategy. The hyperparameter in the loss function of PGS was set to $\chi=0.1$.

Setting of ensemble model

Face image data sets were randomly extracted for 1,000 individuals appearing in VGGFACE and VGGFACE2. A total of 101,211 images were selected from the training set and test set after clipping the facial area. Three types of facial recognition network models, VGG16 [38], Resnet18 [39], and Resnet50 [40], were trained, respectively. Another 900 individual facial region images were selected from the same set and used to train the facial recognition models VGG19 [41] and Senet50 [42]. One hundred face image test data sets for 900 long-term social network users were randomly extracted to form a face image test data set A_1 . Fifty images were randomly selected from A_1 and 50 face image test data sets of the other 100 individuals, and newer users of social networks from VGG19 and Senet50 models were selected to form a face image test data set A_2 . In the subsequent

Fig. 3 Parameter aggregator training process based on differential evolutionary algorithm

experiment, four facial recognition network models were randomly selected as the training model set. The client facial recognition models F and the remaining facial recognition network model was used as the test model, F' , the "criminal" facial recognition model. A_2 was used when VGG16, Resnet18, and Resnet50 were used as test models and A_1 was used when VGG19 and Senet50 were used as test models. A_1 and A_2 can be considered the same public face datasets for the server and clients.

We have five clients in total. Each client has a facial recognition network model and PcadvGAN. The client's facial recognition model from F and F' serves as the attack model of the PcadvGAN, which trains the client's PcadvGAN model at a learning rate of 10^{-3} . The hyperparameters in the loss function of PcadvGAN models were set to $\chi_1=1, \chi_2=30$, and $\chi_3=0.1$. After every client trains the model of PcadvGAN 100 times, it uploads the model parameters of PcadvGAN to the server.

Table 3 Accuracy rate of facial recognition models

Facial recognition model	Accuracy rate (%)
VGG16	98.43
VGG19	98.92
Resnet18	97.91
Resnet50	99.01
Senet50	98.87

The main objective of the experiment was to test whether there was better transferability on the criminal facial recognition model set when using the client facial recognition model F to train the original face image X and generate the private face image X' . The accuracy rate of each trained facial recognition model is shown in Table 3.

Non-targeted attacks

In the non-targeted attack experiment, a private face image X' was generated based on the face image test data set using the training model set, and then input to both the test model and the training model set. The accuracy of the facial recognition model was tested accordingly.

One of the five facial recognition models was used as the test model and the other models were used as the training set. The proposed method was used to train the model set by means of non-targeted attacks. As a result, the recognition accuracy rate of the generated private face image on the training model set was minimized, and the private face image had high transferability on the test model. The private face image is shown in Fig. 4. The generated private face image suffered from only slight perturbation in quality and was very similar to the original face image visually.

Figure 5 shows changes in the accuracy of each model and the generated private face image $\mathcal{L}_{\text{sim}}$ in the case of non-targeted attacks with VGG16, VGG19, Resnet18, Resnet50, and Senet50 as the test models and the other four models

Fig. 4 Private face image generated by non-targeted attack

Fig. 5 Accuracy of models and generated private face image $\mathcal{L}_{sim}$ trained for non-targeted attacks

as the training set. Abscissas in the figure represent the epochs changes of the PGS and ordinates (a)–(e) represent the recognition accuracy rates of the private face images generated by the test data set as the input to each model. The ordinate (f) represents the changes in average quality

$\mathcal{L}_{sim}$ of the generated private face image during the (a)–(e) training process. A smaller $\mathcal{L}_{sim}$ value indicates higher similarity between the generated and original image. The result $\mathcal{L}_{sim} = 0.25$ indicates an only very slight visual difference between the private image and original image.

In the first case shown in Fig. 5, VGG16 and Resnet18 served as test models and the training model set was attacked by the proposed method. The generated private face images drove the accuracy of the five facial recognition models down to 0%, and the $\mathcal{L}_{\text{sim}}$ values were only 0.2776 and 0.2765, respectively. In the second case, Resnet50 served as the test model. The generated private face images again drove the accuracy of the four facial recognition models in the training model down to 0%. The accuracy of Resnet50 as the test model declined to 5% with an $\mathcal{L}_{\text{sim}}$ value of 0.2799.

In the third case shown in Fig. 5, Senet50 served as the test model. The generated private face images brought the accuracy of the three facial recognition models in the training model set down to 0%. The accuracy of VGG19 in the training model set was reduced to 5% and the accuracy rate of Senet50 as the test model declined to 5% with an $\mathcal{L}_{\text{sim}}$ of 0.2892. When VGG19 served as the test model, the generated private face images made for 0% accuracy of the three facial recognition models in the training model set. The accuracy rate of Resnet50 in the training model set declined to 2%. The accuracy rate of VGG19 as the test model declined to 2% with an $\mathcal{L}_{\text{sim}}$ of 0.2855. The faces generated in all of these experimental cases have high adversarial transferability on the test model, and new users in the social networks with VGG19 and Senet5 did not affect the results.

The experimental process shown in Fig. 5 shows that at the beginning of the operation of the PGS, as the perturbation added to the private face image is large, although the accuracy of the five facial recognition models may be lower, the quality of the private face images can be very low. As the training progresses, PGS would learn to adjust the size of the weight K on the premise of ensuring the quality of the private face image to aggregate the better parameter matrix $\bar{W}$, thereby decreasing the accuracy rates of the training model set and the test model.

The original face image was subtracted concurrently from the private face image generated by the proposed

method. The private face image generated by the PcadvGAN attack single model was used to obtain the perturbation, as shown in Fig. 6a, b. Compared with the perturbation in the private face image generated by PcadvGAN, the perturbation in the private face image generated by the proposed method was significantly more concentrated on the key facial features of the face; the perturbation of key features of the face was also greater. This is why, the private face image generated by the proposed method has stronger adversarial transferability.

The proposed method and four current state-of-the-art methods were used to perform non-targeted attacks on the training model set with the goal of enhancing transferability. The accuracy rate of the generated private face image on the test model was investigated, as shown in Fig. 7. The facial recognition model in each row was used as the test model, and the remaining four facial recognition models were used as the training set. A comparison of the generated private face images $\mathcal{L}_{\text{sim}}$ is shown in Table 4. The methods developed by Carlini et al. [22] and Moosavi-Dezfooli et al. [23] only non-targeted attacked one model in the training set at a given time. The optimal result was selected from the private face images generated by the four models in the test set.

As shown in Fig. 7, the proposed method showed the strongest transferability among all methods tested in this non-targeted attack experiment. It effectively minimized the accuracy rate of the test model and showed the strongest privacy protection effects for face images. Table 4 shows that the quality of the face image generated by the proposed method is higher than that by Liu et al., Cheng et al., Moosavi-Dezfooli et al.'s method, and slightly inferior to the Carlini et al.'s method. Taken together, these results suggest that the proposed method can improve the adversarial transferability of private face images while adding only minor perturbations.

Fig. 6 Distribution of perturbations generated by proposed method and by PcadvGAN

Fig. 7 Accuracy rates of various methods on non-targeted attack training model set

Table 4 Private face images $\mathcal{L}_{sim}$ generated by various methods with non-targeted attack

Model	Proposed method	Liu et al.	Cheng et al.	Carlini et al.	Moosavi-Dezfooli et al.
VGG16	0.2776	0.3446	0.2802	0.2997	0.3138
VGG19	0.2765	0.3368	0.2810	0.2797	0.3207
Resnet18	0.2855	0.3008	0.2870	0.2267	0.2942
Resnet50	0.2799	0.3121	0.2854	0.2362	0.3082
Senet50	0.2892	0.2942	0.2958	0.2143	0.3138

Targeted attacks

In the targeted attack experiment, X' , a private face image with a directional misdirecting label T , was generated based on the face image test data set via the training model set. The generated image was input to both the test model set and the training model set to test whether all facial recognition models in the experiment could be misdirected as the label T . The misdirecting rate of a targeted attack is calculated by the number of face images with a label T divided by the number of images S that can be non-targeted attacked in the face image test data set

$$M_r = \frac{\text{num}(T)}{\text{num}(S)} \quad (9)$$

Figure 8 shows the changes in the directional misdirecting rate and the generated private face image $\mathcal{L}_{sim}$ during a targeted attack on the training model with VGG16, VGG19, Resnet18, Resnet50, and Senet50 as the test models with the

remaining four models as the training set. The abscissa is the epoch changes of the PGS. The ordinates (a)–(e) represent the directional misdirection rate of the private face images generated by the face image test data set as the input for each model. The ordinate (f) represents the changes in the average quality $\mathcal{L}_{sim}$ of the generated private face image during the (a)–(e) training process.

In the first case shown in Fig. 8, the training model set was attacked using the proposed method and the generated private face images brought the misdirection rate of the facial recognition model in the four-training model sets to 100% when VGG19, Resnet18, Resnet50, and Senet50 were used as test models. The misdirection rates of VGG19, Resnet18, Resnet50, and Senet50 as test models were 58%, 87%, 83%, and 81%, respectively. In the second case shown in Fig. 8, when VGG16 was the test model, the generated private face image made the misdirection rate of the four facial recognition models in the training set approximate 100%. Senet50 had the lowest misdirection rate at 91%, and

Fig. 8 Directional misdirection rate of each model and generated private face image L_{sim} during targeted attack training

VGG16 as the test model had a misdirection rate of only 61%. The face images generated in both cases had a high misdirection rate under targeted attacks. However, the quality of the generated face images was slightly lower than that under non-targeted attacks and PGS was required to run more epochs.

The proposed method and other four methods mentioned above were next used to conduct targeted attacks on the training model set under the same experimental settings as the non-targeted attacks discussed above. The misdirecting rates of the generated private face images on the test model are shown in Fig. 9. The facial recognition

Fig. 9 Misdirection rates of testing models following targeted attack on training model set

Table 5 Private face images $\mathcal{L}_{\text{sim}}$ generated by various methods under targeted attacks

Model	Proposed method	Liu et al.	Carlini et al.
VGG16	0.3901	0.4122	0.2983
VGG19	0.3823	0.4406	0.3005
Resnet18	0.3774	0.4475	0.2841
Resnet50	0.3792	0.4426	0.2937
Senet50	0.3845	0.3975	0.3096

model in each row was used as the test model and the remaining four facial recognition models as the training set. Table 5 shows the comparison of the generated private face images $\mathcal{L}_{\text{sim}}$.

As shown in Fig. 9, in the targeted attack experiment, the proposed method maximized the test model misdirection rate compared to other methods. Table 5 shows that the quality of the private face image generated by the proposed method is higher than that of the method proposed by Liu et al., and slightly inferior to that by the Carlini et al. method. The quality of the private face image generated by the proposed method was slightly inferior to the image generated under the non-targeted attack, because the more massive perturbation added by the targeted attack misled the test model to recognize the private face image as a specific label. These experiments altogether showed that the proposed method gives images high transferability without sacrificing quality under targeted attacks.

Robustness experiment

The proposed method and the other four methods were next used to attack the test models with different defense strategies to determine their respective robustness. Private face images were generated using non-targeted attacks. The resulting attack success rates are shown in Table 6.

As shown in Table 6, the private face images generated by Cheng et al., Carlini et al., and Moosavi-Dezfooli et al. have extremely low transferability when the defense strategy is median filtering; there were no such effects observed on the proposed method. When using the Adv. defense strategy, the proposed method outperforms the other four methods on VGG16, VGG19, Resnet18, and Resnet50 network test models. When the test network is Senet50, it is slightly inferior to the Liu et al.'s method and superior to the other method. The proposed method appears to have strong robustness. The perturbation generated by the proposed method is not easy to be defended against by methods such as median filtering, because the perturbation is mainly concentrated on the main features of the face image.

Conclusions

A transferable face image privacy protection method based on federated learning and ensemble models was developed in this study. Comparative experiments demonstrated its superior adversarial transferability, robustness, and practicality in generating private face images. The proposed method is also

Table 6 Accuracy rates of various methods when non-targeted attacking different models with defense strategies

Model	Defense strategy	Proposed method (%)	Liu et al. (%)	Cheng et al. (%)	Carlini et al. (%)	Moosavi-Dezfooli et al. (%)
VGG16	MEDIA FILTER	0	0	83	80	82
	Adv	78	85	97	98	100
VGG19	MEDIA FILTER	6	10	84	84	83
	Adv	90	92	95	100	98
Resnet18	MEDIA FILTER	1	4	86	89	100
	Adv	85	90	98	100	100
Resnet50	MEDIA FILTER	2	3	59	58	85
	Adv	83	79	94	93	98
Senet50	MEDIA FILTER	1	1	74	76	88
	Adv	87	88	96	94	100

capable of generating private face images for users who are brand new to a certain design network under the framework of federated learning.

The experimental results show that private face images generated from more types of facial recognition models in the training model set contribute greater transferability. In the training model set, the facial recognition model with a similar structure to the test model set has a larger weight K than other models. Also in the training model set, the facial recognition model with a dissimilar structure the test model set has a non-zero weight K , indicating that the perturbation it generates contributes to the models in the test set with dissimilar attack structures.

If the parameters of each PcadvGAN are transmitted in plaintext between each client and the server in the federated learning training process, there is still a risk of privacy leakage. The differential privacy protection method may be employed to adjust the model parameters for protection during facial recognition model training at the client [43]. Otherwise, homomorphic encryption [24] can be utilized for transmission between each client and the server. Certain network security strategies [44, 45] can also be adopted to reduce the threat of the fractional-order malware propagation models in social networks. In a Gigabit LAN environment, the proposed method takes 18 min and 38 s for one iteration. In the future, the server-side evolutionary algorithm may be optimized to accelerate the running speed.

Funding Supported by the Three Three Three Talent Project Funding Project in Hebei Province under Grant No. A202001017, the Natural Science Foundation of Hebei Province under grant No.F2021405001, the National Natural Science Foundation of China under Grant No. 61772006, the Science and Technology Major Project of Guangxi under Grant No. AA17204096, the Key Research and Development Project of Guangxi under Grant No. AB17129012, and the Special Fund for Bagui Scholars of Guangxi.

Availability of data and materials The article uses public datasets.

Code availability The article algorithm is implemented based on the Google Tensorflow.

Declarations

Conflicts of interest/competing interests On behalf of all authors, the corresponding author states that there is no conflict of interest.

Consent for publication Written informed consent for publication was obtained from all participants.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Cadwalladr C, Graham-Harrison E (2018) Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach. *The guardian* 17:22
2. Masi I, Wu Y, Hassner T, Natarajan P (2018) Deep face recognition: a survey. In: 2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI), 2018. IEEE, pp 471–478
3. Li H, Zhu H, Du S, Liang X, Shen XS (2016) Privacy Leakage of Location Sharing in Mobile Social Networks: Attacks and Defense. *IEEE Trans Depend Secure Comput* 15:646–660
4. Mollah MB, Azad MAK, Vasilakos A (2017) Security and privacy challenges in mobile cloud computing: Survey and way ahead. *J Netw Comput Appl* 84:38–54

5. Li H, Chen Q, Zhu H, Ma D, Wen H, Shen XS (2017) Privacy leakage via de-anonymization and aggregation in heterogeneous social networks. *IEEE Trans Depend Secure Comput* 17:350–362. <https://doi.org/10.1109/TDSC.2017.2754249>
6. Wang Y, Kosinski M (2018) Deep neural networks are more accurate than humans at detecting sexual orientation from facial images. *J Pers Soc Psychol* 114:246–257
7. Wu X, Zhang X (2016) Responses to critiques on machine learning of criminality perceptions (Addendum of [arXiv:1611.04135](https://arxiv.org/abs/1611.04135)). *arXiv: Computer Vision and Pattern Recognition*
8. Ilia P, Polakis I, Athanasopoulos E, Maggi F, Ioannidis S (2015) Face/off: preventing privacy leakage from photos in social networks. In: *Proceedings of the 22nd ACM SIGSAC Conference on computer and communications security*, 2015. pp 781–792
9. Yamada T, Gohshi S, Echizen I (2013) Privacy visor: Method for preventing face image detection by using differences in human and device sensitivity. In: *IFIP International Conference on Communications and Multimedia Security*, 2013. Springer, pp 152–161
10. Liu B, Ding M, Zhu T, Xiang Y, Zhou W (2018) Using adversarial noises to protect privacy in deep learning era. In: *Global Communications Conference*, 2018. pp 1–6
11. Linardos P, Little S, McGuinness K (2019) MediaEval 2019: concealed FGSM perturbations for privacy preservation. *arXiv e-prints*: [arXiv:1910.11603](https://arxiv.org/abs/1910.11603)
12. Goodfellow I, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples. *arXiv: Machine Learning*
13. Xiao C, Li B, Zhu J, He W, Liu M, Song D (2018) Generating adversarial examples with adversarial networks. *arXiv: Cryptography and Security*
14. He Y, Zhang C, Zhu X, Ji Y (2019) Generative adversarial network based image privacy protection algorithm. In: *Tenth International Conference on graphics and image processing*, Chengdu, China, 2019. SPIE, p 1106927
15. Yang J, Liu J, Wu J (2020) Facial image privacy protection based on principal components of adversarial segmented image blocks. *IEEE Access* 8:103385–103394
16. Tramèr F, Kurakin A, Papernot N, Goodfellow I, Boneh D, McDaniel P (2017) Ensemble adversarial training: attacks and defenses. *arXiv preprint* [arXiv:170507204](https://arxiv.org/abs/170507204)
17. Liu Y, Chen X, Liu C, Song D (2017) Delving into Transferable Adversarial Examples and Black-box Attacks. In: *5th International Conference on Learning Representations*, Toulon, France, 2017
18. Cheng S, Dong Y, Pang T, Su H, Zhu J Improving black-box adversarial attacks with a transfer-based prior. In: *Advances in Neural Information Processing Systems*, 2019. pp 10934–10944
19. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R (2013) Intriguing properties of neural networks. *arXiv: Computer Vision and Pattern Recognition*
20. Papernot N, McDaniel P, Goodfellow I (2016) Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint* [arXiv:160507277](https://arxiv.org/abs/160507277)
21. Papernot N, McDaniel P, Goodfellow I, Jha S, Celik ZB, Swami A (2017) Practical black-box attacks against machine learning. In: *Proceedings of the 2017 ACM on Asia Conference on computer and communications security*, 2017. pp 506–519
22. Carlini N, Wagner D (2017) Towards evaluating the robustness of neural networks. In: *2017 IEEE Symposium on security and privacy (sp)*, 2017. IEEE, pp 39–57
23. Moosavi-Dezfooli S-M, Fawzi A, Fawzi O, Frossard P (2017) Universal adversarial perturbations. In: *Proceedings of the IEEE Conference on computer vision and pattern recognition*, 2017. pp 1765–1773
24. Yang Q, Liu Y, Chen T, Tong Y (2019) Federated machine learning: Concept and applications. *ACM Trans Intell Syst Technol (TIST)* 10:1–19
25. Konečný J, McMahan HB, Ramage D, Richtárik P (2016) Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint* [arXiv:161002527](https://arxiv.org/abs/161002527)
26. Konečný J, McMahan HB, Yu FX, Richtárik P, Suresh AT, Bacon D (2016) Federated learning: Strategies for improving communication efficiency. *arXiv preprint* [arXiv:161005492](https://arxiv.org/abs/161005492)
27. Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, Ramage D, Segal A, Seth K (2017) Practical secure aggregation for privacy-preserving machine learning. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017. pp 1175–1191
28. Geyer RC, Klein T, Nabi M (2017) Differentially private federated learning: A client level perspective. *arXiv preprint* [arXiv:171207557](https://arxiv.org/abs/171207557)
29. Smith V, Chiang C-K, Sanjabi M, Talwalkar AS (2017) Federated multi-task learning. In: *Advances in Neural Information Processing Systems*, 2017. pp 4424–4434
30. Zhao Y, Li M, Lai L, Suda N, Civin D, Chandra V (2018) Federated learning with non-iid data. *arXiv preprint* [arXiv:180600582](https://arxiv.org/abs/180600582)
31. McMahan B, Moore E, Ramage D, Hampson S, Arcas BA (2017) Communication-efficient learning of deep networks from decentralized data. In: *Artificial Intelligence and Statistics*, 2017. PMLR, pp 1273–1282
32. Zhu H, Zhang H, Jin Y (2021) From federated learning to federated neural architecture search: a survey. *Complex Intell Syst*. <https://doi.org/10.1007/s40747-020-00247-z>
33. Scannapieco M, Figotin I, Bertino E, Elmagarmid AK (2007) Privacy preserving schema and data matching. In: *Proceedings of the 2007 ACM SIGMOD International Conference on Management of data*, 2007. pp 653–664
34. Wu X, Che A (2019) A memetic differential evolution algorithm for energy-efficient parallel machine scheduling. *Omega* 82:155–165
35. Hu Y-L, Chen L (2018) A nonlinear hybrid wind speed forecasting model using LSTM network, hysteretic ELM and Differential Evolution algorithm. *Energy Convers Manage* 173:123–142
36. Wang H, Wang Y, Zhou Z, Ji X, Gong D, Zhou J, Li Z, Liu W (2018) CosFace: large margin cosine loss for deep face recognition. In: *Computer Vision and Pattern Recognition*, 2018. pp 5265–5274
37. Cao Q, Shen L, Xie W, Parkhi OM, Zisserman A (2018) Vggface2: a dataset for recognising faces across pose and age. In: *2018 13th IEEE International Conference on automatic face & gesture recognition (FG 2018)*, 2018. IEEE, pp 67–74
38. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. In: *Computer Vision and Pattern Recognition*, 2014.
39. Zhou Y, Ren F, Nishide S, Kang X (2019) Facial Sentiment Classification Based on Resnet-18 Model. In: *2019 International Conference on electronic engineering and informatics (EEI)*, 8–10 Nov. 2019 2019. pp 463–466. <https://doi.org/10.1109/EEI48997.2019.00106>
40. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *Computer Vision and Pattern Recognition*, 2016. pp 770–778
41. Tan L, Zhang K, Wang K, Zeng X, Peng X, Qiao Y (2014) Group emotion recognition with individual facial emotion CNNs and global image based CNNs. In: *International Conference on multimodal interfaces*, 2017. pp 549–552
42. Hu J, Shen L, Albanie S, Sun G, Wu E (2019) Squeeze-and-excitation networks. *IEEE Trans Pattern Anal Mach Intell* 42:2011–2023. <https://doi.org/10.1109/TPAMI.2019.2913372>
43. Yang J, Wu J, Wang X (2020) Convolutional neural network based on differential privacy in exponential attenuation mode for image classification. *IET Image Proc* 14:3676–3681
44. Xu C, Liao M, Li P (2019) Bifurcation of a Fractional-Order Delayed Malware Propagation Model in Social Networks. *Disc*

- Dynam Nat Soc 2019:7057052. <https://doi.org/10.1155/2019/7057052>
45. Xu C, Liao M, Li P (2019) Bifurcation control for a fractional-order competition model of Internet with delays. Nonlin Dynam 95:3335–3356. <https://doi.org/10.1007/s11071-018-04758-w>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.