FedSteg: A Federated Transfer Learning Framework for Secure Image Steganalysis

Hongwei Yang, Hui He, *Member, IEEE,* Weizhe Zhang, *Senior Member, IEEE,* and Xiaochun Cao, *Senior Member, IEEE*

***Abstract*—The protection of user private data has long been the focus of AI security. We know that training machine learning models rely on large amounts of user data. However, user data often exists in the form of isolated islands that can not be integrated under many secure and legal constraints. The large-scale application of image steganalysis algorithms in real life is still not satisfactory due to the following challenges. First, it is difficult to aggregate all of the scattered steganographic images to train a robust classifier. Second, even if the images are encrypted, participants do not want irrelevant people to peek into the hidden information, resulting in the disclosure of private data. Finally, it is often impossible for different participants to train their tailored models. In this paper, we introduce a novel framework, referred to as FedSteg, to train a secure, personalized distributed model through federated transfer learning to fulfill secure image steganalysis. Extensive experiments on detecting several state-of-the-art steganographic methods i.e., WOW, S-UNIWARD, and HILL, validate that FedSteg achieves certain improvements compared to traditional non-federated steganalysis approaches. In addition, FedSteg is highly extensible and can be easily employed to various large-scale secure steganographic recognition tasks.**

***Index Terms*—Federated learning, transfer learning, steganalysis, privacy-preserving.**

I. Introduction

IMAGE steganography [1] is the science of hiding information into public digital image without compromising the quality of cover image. With steganography, an image steganizer can securely transmit keys and other private information in an open environment [2]. Due to different embedding strategies, image steganography can be divided into non-adaptive and adaptive steganography. According to the different embedding domain, image steganography can also be categorized into spatial- and JPEG-domain steganography. Spatial domain steganography include WOW [3], S-UNWARD [4], and HILL [5]. Nowadays, Image steganography has been a major research in the field of information security.

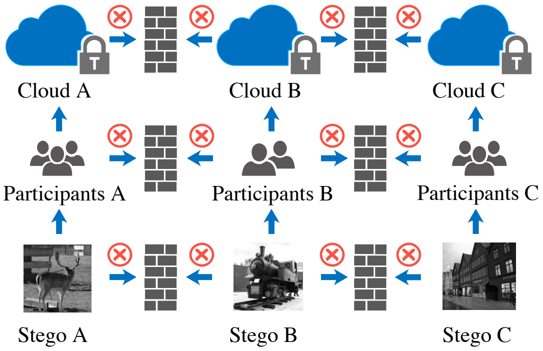


Fig. 1. Three challenges traditional image steganalysis faced: 1) Isolated data are hard to train a robust model; 2) No consideration for personalized model training; 3) Ignorance of the privacy leakage issues when training models.

Image steganalysis, which is a counter technique to image steganography, aims to extract and analyze the steganographic features generated by image steganographic algorithms to detect the existence of hidden information [6]. With the development of information security technology, image steganography and steganalysis promote and develop together. Now, based on the theory of machine learning, the study of steganalysis has made amazing progresses. According to the characteristics of the adaptive steganography algorithms, the corresponding adaptive steganalysis techniques have been produced, such as spatial rich model (SRM) and its several variants, such as tSRM [7], maxSRM [8], and σSRM [9].

Unfortunately, due to the particularity of image steganalysis, there are three critical challenges (Fig. 1) in training a secure and personalized steganalysis model. First, to obtain the information from a stego is to use the secret key, followed by steganalysis methods. In general, only the participants who sent and received the stego knew what the secret information is. Even though images are encrypted, steganographers still do not want unrelated people to peek into the hidden information. However, stegananalysis methods are easy to cause data leakage due to the procedure of training classification model.

Second, we know that for deep learning methods, though they have achieved ideal performance, a sufficient amount of labeled data is still indispensable to obtain robust classifiers [10], [11]. Additionally, in the real world, data often exist in the shape of isolated islands. Although there is a large amount of data in different users, participants, and data

This work was supported in part by the Key Research and Development Program for Guangdong Province (2019B010136001) and the National Key Research and Development Plan under Grant 2017YFB0801801, in part by the National Natural Science Foundation of China (NSFC) under Grant 61672186 and Grant 61872110. Professor Zhang is the corresponding author.

H. Yang is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (email: yanghongwei@hit.edu.cn).

H. He is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (email: hehui@hit.edu.cn).

W. Zhang is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China, and is also with the Pengcheng Laboratory, Shenzhen 518055, China (email: wzzhang@hit.edu.cn).

X. Cao is with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China (email: caoxiaochun@iie.ac.cn).

owners, it is not possible to share them due to privacy and security concerns.

Finally, existing image steganalysis approaches train classifiers based on all raw user data and then use the trained model to identify new steganographic images (stego). This procedure lacks personalization. It is observed that different data owners have different preference for the cover images and the selection of steganographic algorithms. Hence, the traditional image steganalysis algorithms fail to perform personalized training.

In this paper, we proposed a novel FedSteg framework for image steganalysis. As far as we know, there has been no previous work that could tackle these three challenges at the same time. Through federated learning paradigm [12], [13], [14], [15] and additively homomorphic encryption [16], FedSteg collaborates the data from isolated participants to build robust cloud classifiers without compromising the privacy and security of the data owners. After building the cloud model, FedSteg performs transfer learning [17] to realize personalized model learning for each participant. FedSteg can be constantly updated with new stego data. The distributed framework has strong scalability and is suitable for personalized model training that needs privacy and security protection.

We summarize our main contributions as follows:

1. We propose a novel framework FedSteg, the first secure image steganalysis approach at the level of privacy-preserving, which collaborates all the data from different participants to train a general model without leaking any user information to each other, and achieves a personalized classification model through transfer learning.

2. Extensive experiments on large-scale dataset demonstrate that FedSteg decreases the detection error by a certain margin compared to several state-of-the-art non-federated methods, i.e., SRM, maxSRM, and TLU-CNN [18].

3. Finally and importantly, the distributed framework of FedSteg is highly scalable and suitable for personalized model training without leaking the privacy and security of the participants. With the satisfactory performance achieved, it can be easily deployed to other privacy-preserving related tasks.

The rest of the paper is organized as follows, In Section II, we briefly review related works on steganalysis, federated learning, and transfer learning. We describe the proposed FedSteg framework in detail in Section III. The results and analysis of experiments are presented in Section IV. We make a conclusion in Section V.

II. Related Works

In this section, we introduce the related works in the perspective of steganalysis, federated learning, and transfer learning, respectively.

A. *Steganalysis*

With the continuous development of machine learning, the detection error of steganalysis is getting lower and lower, and more and more steganalyzers exploit machine learning methods to train detectors [19], [20].

The emergence of deep learning further improves the detection accuracy of steganalysis by optimizing image features and classifiers. The first influential CNN based steganalyzer is proposed by Qian *et al.* [21], which uses Gaussian activation function instead of ReLU or sigmoid and achieves relatively satisfactory result. The XuNet [22] is the first competitive network architecture to adopt absolute activation layer and TanH activation function in the front layer of the network and to perform batch normalization before nonlinear activation layer. YeNet [18] is a CNN-based approach and achieved significant performance improvement in spatial domain by using the TLU (truncated linear unit) activation function, 30 high-pass filters, and the selection channel. A recently proposed architecture SRNet [23] is also an advanced steganalysis methods which achieved state-of-the-art detection accuracy for both spatial and JPEG domain steganography.

It is worth noting that traditional steganalysis approaches usually build models by using data from all participants. In practice, it is impossible because user data is often kept separately and is not easily shared due to privacy concerns. In addition, secure and personalized model training scheme should be considered for better steganalysis.

B. *Federated Learning*

Federated learning [14], [24], [25] is a distributed machine learning approach which enables model training on a large corpus of decentralized data. Federated learning enables participants to collaboratively learn a shared prediction model while keeping all the training data locally, decoupling the ability to do machine learning from the need to store the data in the cloud. The main idea of federated learning is to manipulate user data without compromising the privacy of user data. From the edge computing perspective, federated learning can be seen as an operating system for edge computing [26], [27] because it provides a learning protocol for coordination and security [14], [28], [29], [30], [31].

In general, federated learning can be categorized into three classes [14]: (1) sample-based federated learning (horizontal federated learning), where the datasets of participants share the same feature space but different samples; (2) feature-based federated learning (vertical federated learning), where the datasets of users share the same sample ID space but different feature spaces; (3) federated transfer learning, where datasets of data owners differ not only in samples but also in feature spaces.

With the increasing importance of privacy-preserving, some countries, such as European Union, have enforced general data protection regulation (GDPR) [32] to protect the privacy of data owners. Since then, a lot of research on privacy-preserving machine learning emerged [24], [33], [34]. Due to the advantages of federated learning in solving the data islanding, it provides solutions to problems involving data privacy-preserving. FedSteg belongs to the category of federated transfer learning. To the best of our knowledge, it is the first model to be applied to steganalysis.

C. *Transfer Learning*

Transfer learning [17] has been successfully applied to numerous real-world applications, which aims to leverage on

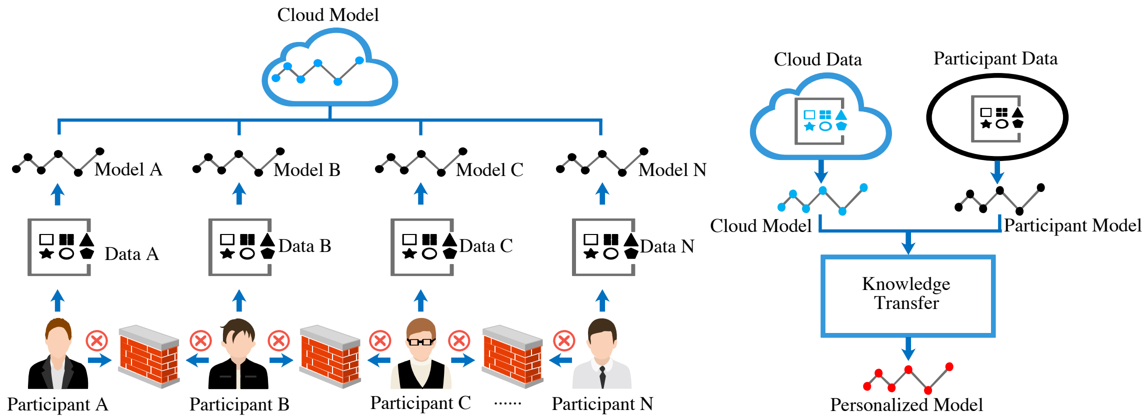


Fig. 2. The main idea of FedSteg. Participants A, B, ..., and N on the left side denote the steganographic images owner, the fire walls denote participants can not share any information with each other, and the process on the right side denotes the personalized model training procedure.

the knowledge collected from a number of different domains to improve the learning performance in new domains, even when these domains may have different distributions.

The main idea of transfer learning is to minimize the mismatch between the source and target domains. For this reason, existing approaches can be divided into two categories: (1) instance reweighting [35], [36], [37], which reuses the instances in the source domain to better match the distribution of the target domain; (2) feature transformation, which performs subspace learning to study the geometrical structure of the subspaces [38], [39], [40] or align the distributions to minimize the marginal or conditional distribution divergence between domains [41]. Nowadays, a large number of deep transfer learning algorithms [42], [43] emerged. In addition, there are some steganalysis studies based on transfer learning [44], [45], [46].

FedSteg exploits the idea of deep transfer learning to achieve a tailored model. As far as we know, it is the first approach to train a security image steganalysis model without accessing the raw data of participants.

III. Federated Learning with FedSteg

In this section, we will introduce the FedSteg framework for secure image steganalysis in detail.

A. Problem Definition

Denote by $\{U_i\}_{i=1}^n$ and $\{D_i\}_{i=1}^n$ the n different users (participants or companies) and the data (stego and cover images) owned by each user. Existing non-federated machine learning methods train a model with all the user data $\bigcup_{i=1}^n D_i$. Our problem is designed to collaborate all the user data to train a federated model in which any user U_i does not leak their data D_i to each other. Let P_{En} and P_{Ef} represent the detection error of the existing non-federated and federated learning methods, respectively. The purpose of FedSteg is to ensure that federated learning is comparable or superior to existing non-federated machine learning approaches in detection error, which can be expressed as:

$$P_{En} - P_{Ef} \geq \epsilon, \tag{1}$$

where ϵ is any small positive real number.

B. Main Idea of FedSteg Framework

The objective of FedSteg is to achieve improved performance for image steganalysis through federated transfer learning without compromising user privacy. We consider a system with three participants and one server as depicted in Fig. 2, which can be extended to more general cases. The framework mainly includes the following four parts. (1) The cloud model are trained with the cover and stego images on the cloud-side. (2) The cloud model is then distributed to all users so that each user can train their local model with local data. (3) The trained user model can then be uploaded to the cloud-side to help train a new cloud model. Note that this step does not share any private data or information but the encrypted model parameters. (4) Each user can perform personalized training by integrating the new cloud model with their previous model and data. Since there exists distribution discrepancy between cloud and user data, transfer learning is performed to make the model more suitable for the user (the right side of Fig. 2). It should be noted that all the parameter sharing procedures may involve the disclosure of user data. Instead, we use additively homomorphic encryption to prevent user privacy from being compromised.

FedSteg exploits federated learning as the main computational model. It is responsible for model building and parameter sharing during the entire process. Once the cloud model training is complete, it can be applied directly to the user. Obviously, the samples of the server-side have a different distribution from the data of each user. As a result, the common model lacks personalization. In addition, the user model can not easily be updated continuously due to the privacy security issues. FedSteg uses federated learning paradigm to perform model training and sharing in order to prevent the leakage of user data. The process of model training can be divided into cloud model training and user model training. After the cloud model is trained, it is distributed to the user-side, and each user trains their own local model based on their own data and the cloud model.

In FedSteg, we exploit the CNN based method TLU-CNN to train the cloud and user models. We let $\boldsymbol{\Theta} = \{\theta\}_{i=1}^n$ be the parameters, that is, the weights and biases, of all layers in the pipeline. Let f_C denotes the cloud model to be learned. We train the network by minimizing a loss function as follows

$$\arg\min_{\boldsymbol{\Theta}} L(\boldsymbol{\Theta}|\boldsymbol{X}) = \frac{1}{n}\sum_{i=1}^{n} c(\boldsymbol{y}_i, f_C(\boldsymbol{x}_i)), \tag{2}$$

where $c(\cdot,\cdot)$ represents the classification loss function. Denote by $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{n}$ the samples of the cloud-side.

Once the cloud model is obtained, it is distributed to each user. In order to guarantee the privacy and security of data owners, direct model and parameter sharing among participants are not allowed (Fig. 2). When cloud model share parameters with users, FedSteg exploits additively homomorphic encryption scheme to prevent privacy disclosure. Additively homomorphic encryption scheme has the following property. Any real number u under additively homomorphic encryption scheme can be represented as $\langle u\rangle$. Additively homomorphic encryption is closed to addition operation, that is, for any two real numbers u and v, we have $\langle u\rangle + \langle v\rangle = \langle u+v\rangle$.

Similarly, the loss function for user j can be denoted as

$$\arg\min_{\boldsymbol{\Theta}^j} L(\boldsymbol{\Theta}^j|\boldsymbol{X}^j) = \frac{1}{m^j}\sum_{i=1}^{m^j} c(\boldsymbol{y}_i^j, f_C(\boldsymbol{x}_i^j)), \tag{3}$$

where m^j denotes the number of data in user j.

After acquiring the user model f_j, it is uploaded to the cloud-side for model updating. For model updating, the cloud-side can align the old model with the model from each user. Denote by $f_C^{'}$ the new model of cloud-side. Because the new cloud model $f_C^{'}$ is based on the knowledge of all users, it has better generalization capabilities. We know that training and updating the model is time-consuming, in order to facilitate participants to use the updated cloud model, the cloud-side can enable automatic scheduled update at a specific date and time.

C. Transfer Learning

Federated learning is used to break down data barriers, connect data islanding, and improve data availability. However, Federated learning does not solve the problem of model personalization. Since the model trained on the cloud-side is a general one, it can not learn the personalized information on a particular user. In this study, we train the personalized model by decreasing the domain discrepancy between the cloud and user data.

In FedSteg, we adopt deep transfer learning to train a personalized model for each user. As stated in [47], features in deep networks transition from general (lower layers) to specific (higher layers) along the network, and the transferability of features and classifiers decreases when the cross-domain discrepancy increases. Therefore, once the parameters of the cloud model are obtained, we can perform transfer learning for each user to learn their tailored model.

For both the user and cloud models, we take advantage of the CNN based network architecture TLU-CNN [18] as the training model, and use the idea of transfer learning to achieve personalized steganalyzer. As shown in Fig. 3, the network includes 9 convolutional layers as the feature extraction module, 4 mean pooling layers, and a 2-way fully-connected layer as the classification module. Note that, we replace the fully connected layer FC2 with an alignment layer to adapt the inputs from different domains.

Generally, when CNN is applied to image classification task, the weight parameters (convolution kernels) of the network often initialized with random values. However, for image steganalysis, only a few pixel values of the cover images are slightly different from those of the stego images, so if the weight parameters of network are initialized with random values, the training process often fails to converge [48]. So we exploit the same idea as TCL-CNN to initialize the weights of the first convolutional layer with 30 high-pass filter kernels used in SRM instead of random values [18]. The 7 basic high-pass filters of SRM are as follows

$$1st: \begin{bmatrix} -1 & 1 \end{bmatrix},\, 2nd: \frac{1}{2}\times\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}, \tag{4}$$

$$3rd: \frac{1}{3}\times\begin{bmatrix} -1 & -3 & 3 & 1 \end{bmatrix}, \tag{5}$$

$$EDGE3\times3: \frac{1}{4}\times\begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \end{bmatrix}, \tag{6}$$

$$SQUARE3\times3: \frac{1}{4}\times\begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix}, \tag{7}$$

$$EDGE5\times5: \frac{1}{12}\times\begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \end{bmatrix}, \tag{8}$$

$$SQUARE5\times5: \frac{1}{12}\times\begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{bmatrix}, \tag{9}$$

which can be expanded to 8 filters in order ”1st”, 4 in order ”2nd”, 8 in order ”3rd”, 1 in class ”SQUARE 3×3”, 1 in class ”SQUARE 5×5”, 4 in class ”EDGE 3×3”, and 4 in class ”EDGE5 ×5” for a total of 30 filters. And then all the filters with size less than 5×5 are expanded to 5×5.

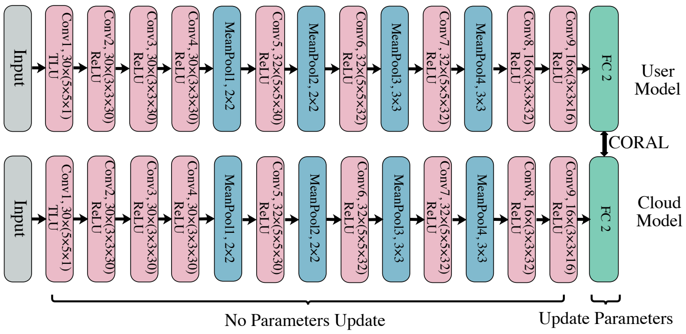


Fig. 3. Knowledge transfer procedure of the proposed FedSteg framework.

As the convolution and max-pooling layers aim to learn general features of images, we do not update parameters during training. However, the fully connected layers focus on learning task- and user-specific features, so we update the parameters during training. In FedSteg, we only have the cloud-side model and the user data, which is different from existing deep transfer learning methods (both source and target data are accessible), such as [49], [42], [50]. For this reason, we refer to the methods in the papers [51], [52] and regularize the weights. The alignment layer we added before the softmax layer is used to further adapt the second order feature statistics of the source and target domains. Formally, the loss of CORAL (correlation alignment) can be calculated as follows

$$L_{coral} = \frac{1}{4d^2} \left\| \boldsymbol{C}_s - \boldsymbol{C}_t \right\|_F^2 , \tag{10}$$

where d is the dimension of embedding features and $\|\cdot\|_F^2$ represents Frobenius norm. The calculation of covariance matrices $\boldsymbol{C}_s$ and $\boldsymbol{C}_t$ can be found in [53]. Finally, the loss function on the user-side can be denoted as

$$\arg\min_{\boldsymbol{\Theta}^j} L(\boldsymbol{\Theta}^j | \boldsymbol{X}^j) = \frac{1}{m^j} \sum_{i=1}^{m^j} c(\boldsymbol{y}_i^j, f_C(\boldsymbol{x}_i^j)) + \lambda L_{coral}, \tag{11}$$

where $\lambda > 0$ is the tradeoff parameter. The detailed procedure of FedSteg is presented in Algorithm 1.

In addition, in the case of low embedding rate, such as 0.1 bpp or below, only about 2% pixel values of steganographic images using S-UNIWARD algorithm are changed due to the embedding of secret information. At this time, it is difficult to train a CNN with sufficient discriminant ability even if the above high-pass filtering and activation function TLU are used. In this study, we also solve the abovementioned issue by adopting the idea of transfer learning [44]. That is, the CNN model with low embedding payload is not trained from scratch, but is fine-tuned on the trained model with high embedding rate corresponding to the same steganography algorithm. For instance, to train a steganalyzer with payload 0.2 bpp, we first train a model from scratch on dataset with payload 0.3 bpp, then fine-tune the model on dataset with 0.2 bpp. Similarly, the model with 0.1 bpp is obtained based on the model with payload 0.2 bpp and so on.

Algorithm 1 Framework of FedSteg.

```
Input: Data $D_1$, $D_2$, ..., $D_n$, and the tradeoff parameter $\lambda$;
Output: Tailored model for each user $f_j$;
1: Train a cloud model $f_C$ using the cloud-side data according to Eq. (2);
2: Additively homomorphic encrypted cloud model $f_C$ is distributed to each participant;
3: Conduct user model on each user-side according to Eq. (3);
4: All homomorphic encrypted user models are updated to the cloud-side. The cloud-side update its model by aligning with user models and a newer cloud model $f_C^{'}$ achieved;
5: Distribute $f_C^{'}$ to all users, then each user achieve their tailored model $f_j$ by performing transfer learning according to Eq. (11);
6: Repeat the above processes for the recurring new user data on the user-side.
```

IV. Experiments

In this section, extensive experiments are conducted to illustrate the effectiveness of our FedSteg framework. We conduct experiment from the following two aspects. On the one hand, we compare the proposed method with several state-of-the-art steganalyzers in spatial domain, e.g., SRM, maxSRM, and TLU-CNN, in terms of detection error ranging from 0.1 to 0.5 bpp. On the other hand, in terms of the scalability of our method, we first replace the CNN structure used in our model with a deeper lightweight VGG network structure. Secondly, we replace the CORAL loss with MMD loss.

TABLE I
Detection Error (P_E) of Four Steganalysis Methods Trained on Participant A, B, and C, and Tested on BOSS_test for WOW with Payload 0.2 or 0.7 bpp

Participants	Methods	BOSS	BOSS+Bows-2	BOSS+Bows-2+EXP
A (0.2 bpp)	SRM	0.3823	**0.3807**	-
	maxSRM	0.3069	**0.3005**	-
	TLU-CNN	0.4198	0.3585	**0.2794**
	FedSteg	0.3563	0.2736	**0.2192**
B (0.2 bpp)	SRM	0.3215	**0.3198**	-
	maxSRM	0.2387	**0.2204**	-
	TLU-CNN	0.3446	0.2791	**0.1883**
	FedSteg	0.2790	0.1836	**0.1776**
C (0.7 bpp)	SRM	0.2557	**0.2489**	-
	maxSRM	0.2213	**0.2105**	-
	TLU-CNN	0.2051	0.1883	**0.1224**
	FedSteg	0.1887	0.1720	**0.1144**

A. Steganographic Methods

In our experiment, three steganographic approaches WOW, S-UNIWARD, and HILL, are used to evaluate the performance of steganalyzers involved. Note that in our experiment, all embedded steganographic algorithms are implemented using STC (syndrome trellis codes) simulator based on publicly available codes. In order to avoid the problem of poor generalization capability of steganalyzer caused by overfitting, we used the random embedding key to create stego images for training other than fixed embedding key.

B. Data Setup

BOSSbase 1.01 contains 10,000 eight bits grayscale images, which is obtained by seven digital cameras and then processed into size of 512 × 512. Bows-2 [54] is a special dataset for bows-2 contest, which is obtained by downsampling and cropping the natural and grayscale images with the size of 512×512×8 bit. Due to the limitation of GPU memory, the original images of 512×512 pixels are not used as the input of the network in our experiment. Instead, we evaluate the performance of FedSteg on images of 256×256 pixels. In order to construct the dataset required for our experiment and to consider the differences in the cover images of different participants, we generate four image datasets in different ways from the image datasets above, and each dataset corresponds to a participant or the cloud-side described as follows:

Cloud: crop the upper left corner of all the images into the size of 256×256 pixels;

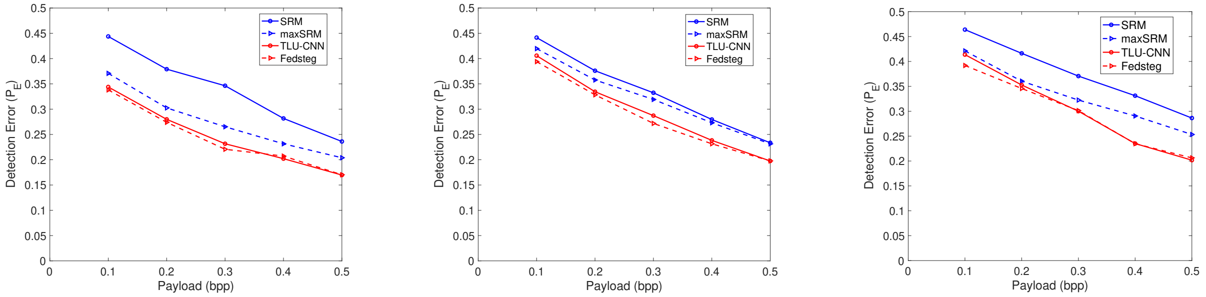
Participant A: crop the central part of all the images into the size of 256×256 pixels;

Participant B: resample all the images into the size of 256×256 pixels;

Participant C: crop the lower right corner of all the images into the size of 256×256 pixels.

TABLE II
COMPARISON RESULTS OF STEGANALYSIS METHODS IN TERMS OF DETECTION ERROR (P_E) FOR THREE STEGANOGRAPHIC ALGORITHMS WITH DIFFERENT PAYLOADS ON PARTICIPANT A

Algorithms	Payload (bpp)	SRM (P_E)	maxSRM (P_E)	TLU-CNN (P_E)	FedSteg (P_E)
WOW	0.1	0.4438	0.3708	0.3439	**0.3381**
	0.2	0.3791	0.3025	0.2799	**0.2740**
	0.3	0.3463	0.2652	0.2316	**0.2207**
	0.4	0.2817	0.2315	**0.2021**	0.2072
	0.5	0.2361	0.2039	**0.1697**	0.1703
S-UNIWARD	0.1	0.4415	0.4197	0.4058	**0.3940**
	0.2	0.3760	0.3577	0.3345	**0.3284**
	0.3	0.3324	0.3192	0.2872	**0.2718**
	0.4	0.2796	0.2731	0.2383	**0.2315**
	0.5	0.2335	0.2314	**0.1975**	0.1997
HILL	0.1	0.4639	0.4212	0.4137	**0.3919**
	0.2	0.4163	0.3604	0.3527	**0.3457**
	0.3	0.3706	0.3224	**0.3001**	0.3013
	0.4	0.3312	0.2905	0.2351	**0.2348**
	0.5	0.2863	0.2533	**0.2017**	0.2063



(a) WOW (b) S-UNIWARD (c) HILL

Fig. 4. Comparison of detection error for four steganalysis methods on detecting three steganographic schemes with different payloads on participant A.

TABLE III
COMPARISON RESULTS OF STEGANALYSIS METHODS IN TERMS OF DETECTION ERROR (P_E) FOR THREE STEGANOGRAPHIC ALGORITHMS WITH DIFFERENT PAYLOADS ON PARTICIPANT B

Algorithms	Payload (bpp)	SRM (P_E)	maxSRM (P_E)	TLU-CNN (P_E)	FedSteg (P_E)
WOW	0.1	0.4045	0.3177	0.3008	**0.2712**
	0.2	0.3285	0.2316	0.1947	**0.1891**
	0.3	0.2613	0.1873	**0.1327**	0.1385
	0.4	0.2096	0.1501	0.1096	**0.1083**
	0.5	0.1789	0.1322	**0.0902**	0.0917
S-UNIWARD	0.1	0.4179	0.3787	0.3315	**0.3307**
	0.2	0.3546	0.2983	0.2528	**0.2337**
	0.3	0.2739	0.2606	**0.1559**	0.1601
	0.4	0.2303	0.2115	**0.1292**	0.1299
	0.5	0.1837	0.1701	0.1096	**0.1019**
HILL	0.1	0.4564	0.3903	0.3541	**0.3428**
	0.2	0.3794	0.3218	0.2757	**0.2678**
	0.3	0.3187	0.2796	**0.2139**	0.2212
	0.4	0.2820	0.2403	**0.1663**	0.1709
	0.5	0.2419	0.2112	**0.1402**	0.1416

For each dataset, we then construct 3 training sets and, 1 validation set, and 1 testing set, respectively:

(1) training set BOSS, which consists of 4,000 images randomly selected from BOSSbase;

(2) training set BOSS+Bows-2, which includes the images in training set BOSS and all the images from Bows-2;

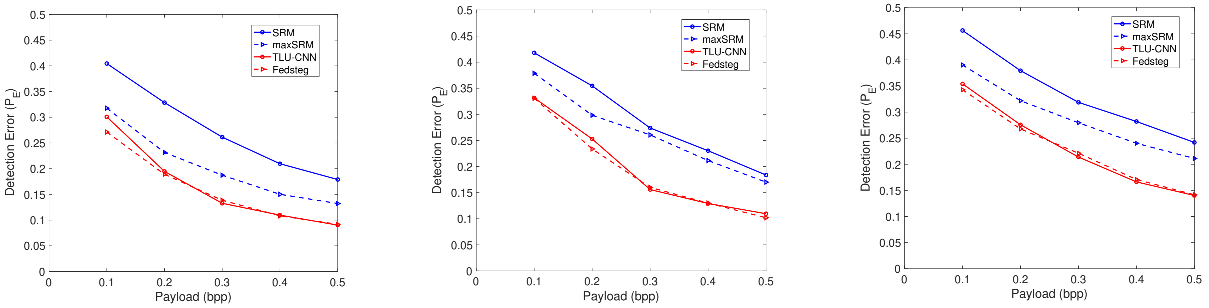
(3) training set BOSS+Bows-2+EXP, which is obtained by performing a random horizontal mirror and rotating at four specific angles (0°, 90°, 180°, and 270°). Therefore, we get 8 expanded images for each image in the BOSS+Bows-2 training set;

(4) validation set BOSS_val, which contains 1,000 images randomly selected from the left 6,000 images in BOSSbase;

(5) testing set BOSS_test, which consists of the remaining 5,000 images in BOSSbase.

In order to avoid overfitting, CNN based steganalysis methods need a lot of training data. Table I summarizes the performance of FedSteg and other three steganalysis approaches trained on participants A, B, and C, and tested on BOSS_test for WOW with payload of 0.2 or 0.7 bpp. The results show that our method also suffers from overfitting when training on dataset BOSS. However, it has a relatively low detection error compared with TLU-CNN. This is mainly because TLU-CNN only conduct training on participant A, while FedSteg is trained on the collaborate BOSS data from participants A, B, and C. Therefore, when the amount of local data is small,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2020.2996612, IEEE Transactions on Network Science and Engineering

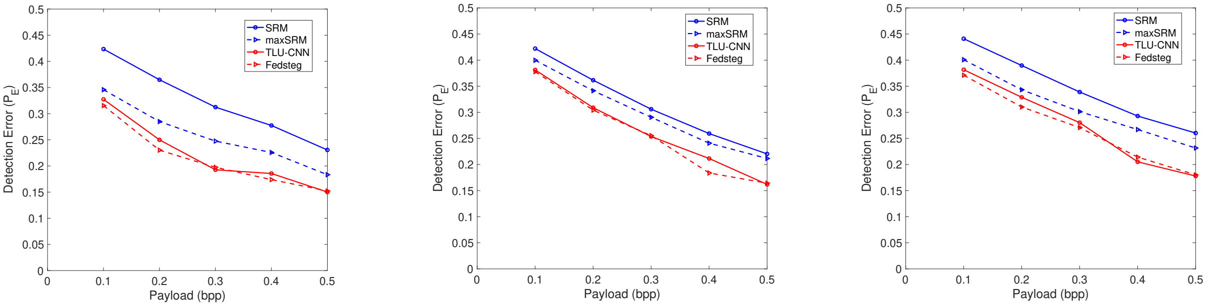


(a) WOW (b) S-UNIWARD (c) HILL

Fig. 5. Comparison of detection error for four steganalysis methods on detecting three steganographic schemes with different payloads on participant B.

TABLE IV
COMPARISON RESULTS OF STEGANALYSIS METHODS IN TERMS OF DETECTION ERROR (P_E) FOR THREE STEGANOGRAPHIC ALGORITHMS WITH DIFFERENT PAYLOADS ON PARTICIPANT C

Algorithms	Payload (bpp)	SRM (P_E)	maxSRM (P_E)	TLU-CNN (P_E)	FedSteg (P_E)
WOW	0.1	0.4235	0.3458	0.3273	**0.3154**
	0.2	0.3648	0.2849	0.2499	**0.2302**
	0.3	0.3125	0.2475	**0.1927**	0.1974
	0.4	0.2774	0.2258	0.1856	**0.1739**
	0.5	0.2306	0.1833	**0.1501**	0.1522
S-UNIWARD	0.1	0.4221	0.3997	0.3812	**0.3779**
	0.2	0.3615	0.3414	0.3089	**0.3042**
	0.3	0.3060	0.2905	**0.2536**	0.2553
	0.4	0.2593	0.2409	0.2115	**0.1838**
	0.5	0.2203	0.2116	**0.1619**	0.1642
HILL	0.1	0.4410	0.4008	0.3816	**0.3711**
	0.2	0.3897	0.3432	0.3288	**0.3102**
	0.3	0.3389	0.3014	0.2802	**0.2708**
	0.4	0.2929	0.2670	**0.2051**	0.2142
	0.5	0.2603	0.2316	**0.1779**	0.1802



(a) WOW (b) S-UNIWARD (c) HILL

Fig. 6. Comparison of detection error for four steganalysis methods on detecting three steganographic schemes with different payloads on participant C.

FedSteg can reduce the risk of overfitting to some extent. What is more, the performance is improved when conduct training on dataset BOSS+Bows-2. And when training on dataset BOSS+Bows-2+EXP, the performance is improved by a large margin. Unless otherwise specified, the experiments for our method are trained with resampled images on BOSS+Bows-2+EXP and tested on BOSS_test, and for SRM and maxSRM are trained on BOSS+Bows-2 and tested on BOSS_test.

C. Experimental Details

On both the cloud and the user side, we adopt the CNN based method TLU-CNN [18] for training and prediction. The network is composed of 9 convolutional layers, 4 mean pooling layers, and a 2-way fully connected layer. We also initialize the weights of the first convolutional layer in our experiments with high-pass filter kernels used in SRM instead of random values, but slightly difference with TLU-CNN, we employ the normalized high-pass filter. That is, the central element of these filters is 1 by dividing by the corresponding order, so that the parameters of each filter are distributed in the same order of magnitude. The remaining 8 convolutional layers are initialized using ”Xavier” [55] and the initial biases are set to be 0.1. Meanwhile, the last fully-connected layer is randomly initialized with a Gaussian distribution with a mean value of 0 and a standard deviation of 0.01. Some other hyper-parameters are set as follows: the mini-batch size is 32, which consists of 16 pairs of cover and stego images; the momentum value is 0.5 and the weight decay is 1×10^{-4}.

In the federated learning phase, we adopt additively homomorphic encryption scheme for secure parameter sharing. While, in the transfer learning phase, we only update the parameters of the fully connected layer, not the convolutional and pooling layers.

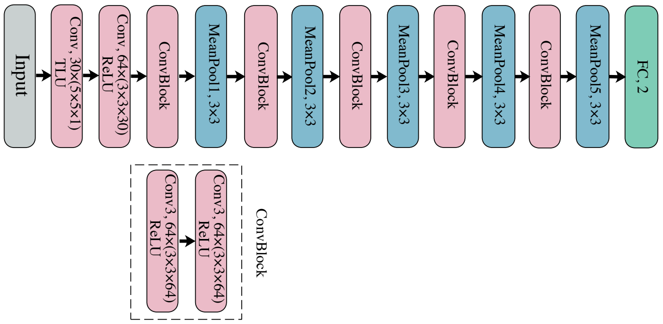


Fig. 7. A lightweight VGG network structure consists of convolution blocks and mean pooling layer connected alternately five times to learn better feature representations, the dotted box below denotes the convolution block with two convolution layers.

D. Experimental Results

In this section, we conduct a large number of experiments and compare the experimental results with three state-of-the-art steganalyzers, i.e. SRM, maxSRM, and TLU-CNN, under a wide variety of payloads ranging from 0.1 to 0.5 bpp. Tables II-IV show the comparison results in terms of detection error for all the tested approaches on participants A, B, and C. Fig. 4 to Fig. 6 show the testing results of the proposed framework and other steganalyzers in terms of the detection error of three steganographic methods, i.e., WOW, S-UNIWARD, and HILL.

Fig. 4 to Fig. 6 show that our proposed framework achieve better detection performance by a certain margin in embedding schemes, payload, and dataset. We can lead the following conclusions: (1) FedSteg can achieve significant improvement gains over SRM and maxSRM irrespective of the embedding methods, payloads, and datasets. Specially, compare to SRM and maxSRM, the total average detection errors of WOW are improved by 9.54% and 3.28%, respectively. (2) The average detection error of FedSteg for participant A is relatively higher, mainly due to the generation scheme of dataset. We know that the central part of images are the most complex regions. Therefore, it is not surprising that the detection error is high.

It is worth noting that, as shown in Tables II-IV, although FedSteg does not perform best under all payloads in terms of three steganographic methods, i.e., WOW, S-UNIWARD, and HILL on all participant side, the detection error is somewhat decreased. Some conclusions can be drawn as follows: (1) When the embedding capacity is low, FedSteg performs better than TLU-CNN, and we speculate that the main reason is that, compared with single domain knowledge training, FedSteg uses data distributed among three users (participants A, B, and C) to jointly train a model. On the one hand, it can prevent overfitting when the amount of data is small; On the other hand, better robust models can be learned due to the diversity of data samples. (2) With the increase of embedding capacity, more pixel values are changed in the image, and the correlation between adjacent pixels is relatively obvious, so the performance difference between FedSteg and TLU-CNN becomes smaller and smaller.

E. Analysis of Scalability

In this section, we discuss the scalability of our framework from the following three aspects.

First, when training FedSteg, we can replace the CNN based TLU-CNN structure with deeper network structures, such as VGG [56]. It has been validated that very deep convolutional networks can learn better representations and achieve better recognition accuracy. So, to prevent overfitting, we choose a lightweight VGG network structure, as shown in Fig. 7, to illustrate that our approach still performs well in the case of deeper network structure and the experimental results are reported in Table V and Fig. 8.

As can be seen from Table V, the detection error of FedSteg with network structure VGG for steganographic method WOW with payloads ranging from 0.1 to 0.5 bpp on participant B are decreased by 0.24% compared to FedSteg, and 0.82% compared to TLU-CNN. This indicates that FedSteg is highly scalable in the selection of network structures and can integrate many neural networks into our framework.

Second, CORAL and MMD (maximum mean discrepancy) [57] losses are two commonly used loss function to minimize the discrepancy between domains in transfer learning, we replace the CORAL loss in our FedSteg framework with MMD loss and also report the result in Table V and Fig. 8. It is not difficult to find that FedSteg using MMD loss is better than TLU-CNN or similar to TLU-CNN in detection error of three steganographic algorithms under payloads ranging from 0.1 to 0.5 bpp on participant B. This demonstrates that our framework is extensible in the method of reducing the loss of difference between the cloud and users.

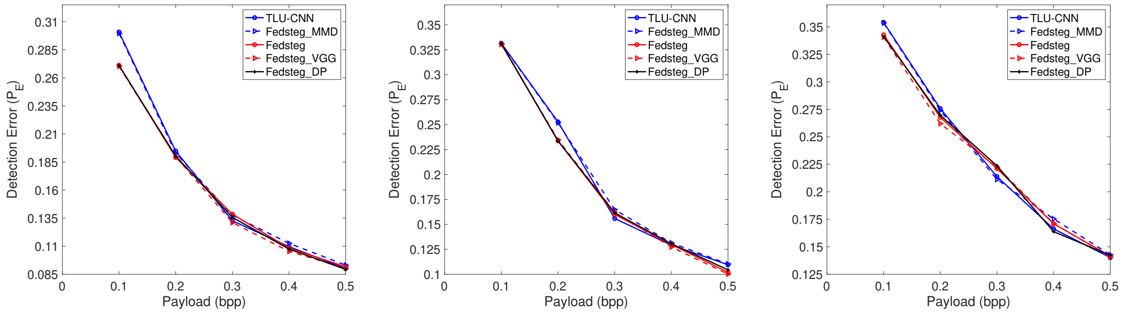
Finally, during federated learning, we can exploit other encryption techniques to achieve secure parameter sharing, such as secure multi-party computation (MPC) [58], Yao's Garbled circuit protocol [59], differential privacy (DP) [60] and other encryption instead of additively homomorphic encryption. We replace the additively homomorphic encryption scheme with DP algorithm and show the result in Table V and Fig. 8. The result shows that FesSteg using DP scheme achieves comparable performance to FedSteg using additively homomorphic encryption scheme. This validates that our method is extensible in terms of the mechanism for secure parameter sharing.

V. Conclusions

In this paper, we propose a novel federated transfer learning framework FedSteg for secure image steganalysis. FedSteg collaborates all the scattered data from different participants to train a general model without leaking the privacy information of the data owners, and performs transfer learning to achieve

TABLE V
COMPARISON RESULTS OF STEGANALYSIS METHODS IN TERMS OF DETECTION ERROR (P_E) FOR THREE STEGANOGRAPHIC ALGORITHMS WITH DIFFERENT PAYLOADS ON PARTICIPANT B

Algorithms	Payload (bpp)	TLU-CNN (P_E)	FedSteg_MMD (P_E)	FedSteg (P_E)	FedSteg_VGG (P_E)	FedSteg_DP (P_E)
WOW	0.1	0.3008	0.2997	0.2712	**0.2703**	0.2709
	0.2	0.1947	0.1929	**0.1891**	0.1898	0.1903
	0.3	0.1327	0.1375	0.1385	**0.1311**	0.1357
	0.4	0.1096	0.1126	0.1083	**0.1054**	0.1074
	0.5	0.0902	0.0931	0.0917	0.0904	**0.0894**
S-UNIWARD	0.1	0.3315	0.3311	0.3307	**0.3304**	0.3312
	0.2	0.2528	0.2519	**0.2337**	0.2349	**0.2337**
	0.3	**0.1559**	0.1648	0.1601	0.1621	0.1615
	0.4	0.1292	0.1315	0.1299	**0.1273**	0.1306
	0.5	0.1096	0.1103	0.1019	**0.1002**	0.1044
HILL	0.1	0.3541	0.3536	0.3428	**0.3407**	0.3413
	0.2	0.2757	0.2740	0.2678	**0.2620**	0.2697
	0.3	0.2139	**0.2113**	0.2212	0.2225	0.2237
	0.4	0.1663	0.1754	0.1709	0.1713	**0.1638**
	0.5	**0.1402**	0.1427	0.1416	0.1408	0.1422



(a) WOW (b) S-UNIWARD (c) HILL

Fig. 8. Comparison of detection error between TLU-CNN and four FedSteg methods on detecting three steganographic schemes with different payloads on participant B.

tailored model for each participant. Experiments on large-scale steganalysis dataset demonstrate the effectiveness and robustness of our method. Furthermore, we also provide a detailed experiment to illustrate the high scalability of FedSteg in terms of network structure, reducing domain discrepancy and encryption scheme for secure parameter sharing, which confirms that the current framework is not limited to any particular learning paradigms or network structures, but rather a general framework for privacy-preserving machine learning.

REFERENCES

[1] Abbas Cheddad, Joan Condell, Kevin Curran, and Paul Mc Kevitt. Digital image steganography: Survey and analysis of current methods. *Signal processing*, 90(3):727–752, 2010.

[2] AP Fabien, Ross J Anderson, and Markus G Kuhn. Information hiding-a survey. *Proceedings of the IEEE*, 87(7):1062–1078, 1999.

[3] Vojtěch Holub and Jessica Fridrich. Designing steganographic distortion using directional filters. In *2012 IEEE International workshop on information forensics and security (WIFS)*, pages 234–239. IEEE, 2012.

[4] Vojtěch Holub and Jessica Fridrich. Digital image steganography using universal distortion. In *Proceedings of the first ACM workshop on Information hiding and multimedia security*, pages 59–68, 2013.

[5] Bin Li, Shunquan Tan, Ming Wang, and Jiwu Huang. Investigation on cost assignment in spatial image steganography. *IEEE Transactions on Information Forensics and Security*, 9(8):1264–1277, 2014.

[6] Huaiqing Wang and Shuozhong Wang. Cyber warfare: steganography vs. steganalysis. *Communications of the ACM*, 47(10):76–82, 2004.

[7] Weixuan Tang, Haodong Li, Weiqi Luo, and Jiwu Huang. Adaptive steganalysis against wow embedding algorithm. In *Proceedings of the 2nd ACM workshop on Information hiding and multimedia security*, pages 91–96, 2014.

[8] Tomas Denemark, Vahid Sedighi, Vojtech Holub, Rémi Cogranne, and Jessica Fridrich. Selection-channel-aware rich model for steganalysis of digital images. In *2014 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 48–53. IEEE, 2014.

[9] Tomáš Denemark, Jessica Fridrich, and Pedro Comesaña-Alfaro. Improving selection-channel-aware steganalysis features. *Electronic Imaging*, 2016(8):1–8, 2016.

[10] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2010.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[12] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

[13] Kewei Cheng, Tao Fan, Yilun Jin, Yang Liu, Tianjian Chen, and Qiang Yang. Secureboost: A lossless federated learning framework. *arXiv preprint arXiv:1901.08755*, 2019.

[14] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[15] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[16] Ronald L Rivest, Len Adleman, Michael L Dertouzos, et al. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

[17] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[18] Jian Ye, Jiangqun Ni, and Yang Yi. Deep learning hierarchical representations for image steganalysis. *IEEE Transactions on Information Forensics and Security*, 12(11):2545–2557, 2017.

[19] İsmail Avcıbaş, Mehdi Kharrazi, Nasir Memon, and Bülent Sankur.

Image steganalysis with binary similarity measures. *EURASIP Journal on Advances in Signal Processing*, 2005(17):679350, 2005.

[20] Siwei Lyu and Hany Farid. Detecting hidden messages using higher-order statistics and support vector machines. In *International Workshop on information hiding*, pages 340–354. Springer, 2002.

[21] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. Deep learning for steganalysis via convolutional neural networks. In *Media Watermarking, Security, and Forensics 2015*, volume 9409, page 94090J. International Society for Optics and Photonics, 2015.

[22] Guanshuo Xu, Han-Zhou Wu, and Yun Q Shi. Ensemble of cnns for steganalysis: An empirical study. In *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, pages 103–107, 2016.

[23] Mehdi Boroumand, Mo Chen, and Jessica Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5):1181–1193, 2018.

[24] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

[25] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konecny, Stefano Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

[26] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5):637–646, 2016.

[27] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B Letaief. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, 19(4):2322–2358, 2017.

[28] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6):1205–1221, 2019.

[29] Takayuki Nishio and Ryo Yonetani. Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2019.

[30] Xiaofei Wang, Yiwen Han, Chenyang Wang, Qiyang Zhao, Xu Chen, and Min Chen. In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning. *IEEE Network*, 33(5):156–165, 2019.

[31] Kai Yang, Tao Jiang, Yuanming Shi, and Zhi Ding. Federated learning via over-the-air computation. *IEEE Transactions on Wireless Communications*, 19(3):2022–2035, 2020.

[32] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 2017.

[33] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[34] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.

[35] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning*, pages 193–200, 2007.

[36] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[37] Yonghui Xu, Sinno Jialin Pan, Hui Xiong, Qingyao Wu, Ronghua Luo, Huaqing Min, and Hengjie Song. A unified framework for metric transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 29(6):1158–1171, 2017.

[38] Anna Margolis. A literature review of domain adaptation with unlabeled data. *Tec. Report*, pages 1–42, 2011.

[39] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073. IEEE, 2012.

[40] Basura Fernando, Tatiana Tommasi, and Tinne Tuytelaars. Joint cross-domain classification and subspace learning for unsupervised adaptation. *Pattern Recognition Letters*, 65:60–66, 2015.

[41] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer joint matching for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1410–1417, 2014.

[42] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

[43] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[44] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. Learning representations for steganalysis from regularized cnn model with auxiliary tasks. In *Proceedings of the 2015 International Conference on Communications, Signal Processing, and Systems*, pages 629–637. Springer, 2016.

[45] Juan Tian and Yingxiang Li. Convolutional neural networks for steganalysis via transfer learning. *International Journal of Pattern Recognition and Artificial Intelligence*, 33(02):1959006, 2019.

[46] Rabii El Beji, Marwa Saidi, Houcemeddine Hermassi, and Rhouma Rhouma. An improved cnn steganalysis architecture based on “catalyst kernels” and transfer learning. In *International Conference on Digital Economy*, pages 119–128. Springer, 2018.

[47] Wenbo Zhou, Weiming Zhang, and Nenghai Yu. A new rule for cost reassignment in adaptive steganography. *IEEE Transactions on Information Forensics and Security*, 12(11):2654–2667, 2017.

[48] Shunquan Tan and Bin Li. Stacked convolutional auto-encoders for steganalysis of digital images. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, pages 1–4. IEEE, 2014.

[49] Muhammad Ghifary, W Bastiaan Kleijn, and Mengjie Zhang. Domain adaptive neural networks for object recognition. In *Pacific Rim international conference on artificial intelligence*, pages 898–904. Springer, 2014.

[50] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[51] Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. Beyond sharing weights for deep domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 41(4):801–814, 2018.

[52] Yiqiang Chen, Jindong Wang, Chaohui Yu, Wen Gao, and Xin Qin. Fedhealth: A federated transfer learning framework for wearable healthcare. *arXiv preprint arXiv:1907.09173*, 2019.

[53] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[54] Patrick Bas and Teddy Furon. Bows-2. http://bows2.ec-lille.fr, 2007.

[55] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[56] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[57] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2208–2217. JMLR. org, 2017.

[58] Payman Mohassel and Peter Rindal. Aby3: A mixed protocol framework for machine learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 35–52, 2018.

[59] Bita Darvish Rouhani, M Sadegh Riazi, and Farinaz Koushanfar. Deepsecure: Scalable provably-secure deep learning. In *Proceedings of the 55th Annual Design Automation Conference*, pages 1–6, 2018.

[60] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

Hongwei Yang is currently working toward the PhD degree in the School of Computer Science and Technology, Harbin Institute of Technology, China. His research interests include transfer learning, information security, and big data analysis.

Hui He is currently a full professor of network security center in the Department of Computer Science, China. She received the Ph.D. in department of computer science from the Harbin Institute of Technology, China. Her research interests are mainly focused on distributed computing, IoT and big data analysis. She is a member of the IEEE.

Weizhe Zhang is currently a professor in the School of Computer Science and Technology at Harbin Institute of Technology, China. His research interests are primarily in cyberspace security, cloud computing, and parallel computing. He has published more than 100 academic papers in journals, books, and conference proceedings. He is a senior member of the IEEE.

Xiaochun Cao received the B.E. and M.E. degrees in computer science from Beihang University, China, and the Ph.D. degree in computer science from the University of Central Florida, Orlando, USA. After graduation, he spent about three years at ObjectVideo Inc. as a Research Scientist. From 2008 to 2012, he was a Professor with Tianjin University, Tianjin, China. He has been a professor with the Institute of Information Engineering, Chinese Academy of Sciences, since 2012. He is also with the Peng Cheng Laboratory, Cyberspace Security Research Center, China, and the School of Cyber Security, University of Chinese Academy of Sciences, China. He is on the Editorial Boards of the IEEE Transactions on Image Processing, IEEE Transactions on Multimedia and IEEE Transactions on Circuits and Systems for Video Technology. In 2004 and 2010, he was the recipient of the Piero Zamperoni Best Student Paper Award at the International Conference on Pattern Recognition.