electronics

Communication

FedSpy: A Secure Collaborative Speech Steganalysis Framework Based on Federated Learning

Hui Tian [1,2], Huidong Wang [1], Hanyu Quan [1,2,*], Wojciech Mazurczyk [3] and Chin-Chen Chang [4]

[1] College of Computer Science and Technology, National Huaqiao University, Xiamen 361021, China; htian@hqu.edu.cn (H.T.); whd@stu.hqu.edu.cn (H.W.)
[2] Xiamen Key Laboratory of Data Security and Blockchain Technology, Xiamen 361021, China
[3] Institute of Computer Science, Warsaw University of Technology, 00-665 Warszawa, Poland; wmazurcz@elka.pw.edu.pl
[4] Department of Information and Computer Science, Feng Chia University, Taichung 40724, Taiwan; ccc@o365.fcu.edu.tw
* Correspondence: quanhanyu@hqu.edu.cn

Abstract: Deep learning brings the opportunity to achieve effective speech steganalysis in speech signals. However, the speech samples used to train speech steganalysis models (i.e., steganalyzers) are usually sensitive and distributed among different agencies, making it impractical to train an effective centralized steganalyzer. Therefore, in this paper, we present an effective framework, named FedSpy, using federated learning, which enables multiple agencies to securely and jointly train the speech steganalysis models without sharing their speech samples. FedSpy is a flexible and extensible framework that can work effectively in conjunction with various deep-learning-based speech steganalysis methods. We evaluate the performance of FedSpy by detecting the most widely used Quantization Index Modulation-based speech steganography with three state-of-the-art deep-learning-based steganalysis methods representatively. The results show that FedSpy significantly outperforms the local steganalyzers and achieves good detection accuracy comparable to the centralized steganalyzer.

Keywords: speech steganalysis; speech steganography; federated learning

check for updates

Citation: Tian, H.; Wang, H.; Quan, H.; Mazurczyk, W.; Chang, C.-C. FedSpy: A Secure Collaborative Speech Steganalysis Framework Based on Federated Learning. *Electronics* **2023**, *12*, 2854. https://doi.org/10.3390/electronics12132854

Academic Editor: Andrei Kelarev

Received: 23 May 2023
Revised: 24 June 2023
Accepted: 26 June 2023
Published: 28 June 2023

1. Introduction

Speech steganography embeds secret messages into speech signals to realize covert communications on public channels [1,2], providing a new way for secure information transmission. Compared to speech encryption [3,4], speech steganography can conceal the fact that the secret messages are being sent, thus offering stronger security in some cases. However, it might pose a major threat if used by cybercriminals to transmit stolen data, malware codes, and some other illegal messages. Therefore, speech steganalysis [5,6], whose primary purpose is to detect the existence of hidden messages in speech signals, has been attracting increasing attention in recent years. Particularly, with the development of artificial intelligence, many well-performing speech steganalysis methods based on deep learning [7–18] have been proposed, since deep learning can capture the subtle differences between the steganographic and cover samples.

Although deep-learning-based speech steganalysis has achieved relatively good detection performance in laboratory settings, it still faces some challenges in practical applications. First, deep-learning-based speech steganalysis usually requires a large number of steganographic samples as training data to obtain robust classifiers. For example, the training dataset in [7] includes over one million steganographic speech segments. However, for the offenses of employing speech steganography to transmit unauthorized information, security agencies usually only have a small number of steganographic samples, making it difficult to independently train an effective steganalyzer (i.e., a robust classifier for detecting steganography). Of course, the direct collection of samples from multiple agencies might solve the problem of insufficient samples in a single agency to a certain extent. However, due to the sensitive nature of steganographic samples, even though these agencies are allies in steganalysis, the data privacy concerns and legal constraints (e.g., General Data Protection Regulation) prevent agencies from sharing the limited number of steganographic samples they have, which is a common occurrence in collaborative steganalysis tasks [19]. Thus, the steganographic samples of various security agencies are existing in the form of isolated data silos, which poses a significant obstacle to the deployment of deep-learning-based speech steganalysis. In other words, the current challenge lies in finding an effective approach to deploy deep-learning-based speech steganalysis across multiple security agencies while protecting speech sample privacy. Therefore, the motivation behind this paper is to present a practical solution in the form of collaborative speech steganalysis, which aims to address this challenge comprehensively.

Federated Learning (FL) [20] is an emergent machine learning paradigm, which provides a potential solution to the above issues. In FL, multiple clients (e.g., the security agencies) can train a global model (e.g., a speech steganalysis classifier) collaboratively without sharing training data with each other. Instead, each client trains a local model with its local dataset and uploads the local model to a central server. Then the central server aggregates all local models to update a global model. This process is repeated until the global model is convergent. FL can mitigate the data privacy risk because the raw training data is only kept in an on-premise environment.

Recently, Yang et al. successfully applied FL to image steganalysis and proposed a framework named FedSteg [19]. FedSteg includes a one-round global model update, followed by a local transfer learning at each client end that can decrease the distribution discrepancy between the image data at different ends. However, in this paper, we experimentally show that FedSteg, designed for image steganalysis, is not well suited for speech steganalysis. The reasons are twofold. First, as pointed out in a previous work [7], the distribution discrepancy between different speech data (e.g., different gender, different languages) has little effect on the detection accuracy of speech steganalysis, which indicates that transfer learning is not so significant in speech steganalysis because it cannot further improve detection accuracy. Second, the global model is only updated in FedSteg once, which would also result in low detection accuracy even without the amplification of transfer learning.

Thus, in this paper, we propose a novel Secure Collaborative Speech Steganalysis Framework based on federated learning, named FedSpy, whose main contributions can be summarized as follows.

Firstly, FedSpy enables the collaboration of speech steganalysis across isolated clients (agencies). By employing FedSpy, these isolated clients can collaboratively perform steganalysis based on deep learning without compromising the privacy of steganographic samples. This significantly addresses the challenge of insufficient steganographic samples encountered by individual clients during the training of deep-learning-based steganalyzers.

Secondly, within the FedSpy framework, we introduce federated learning into speech steganalysis for the first time. While federated learning has been widely applied in many domains such as finance, healthcare, and the Internet of Things, its potential in speech steganalysis has remained largely unexplored. Our work expands the realm of federated learning, showcasing its efficacy in the domain of speech steganalysis.

Lastly, we implement three state-of-the-art speech steganalysis methods based on deep learning in FedSpy. The experimental results prove that FedSpy can achieve commendable detection accuracy comparable to the centralized steganalysis, surpassing the steganalysis on local data set by a significant margin. Furthermore, these comprehensive experiments convincingly demonstrate the effectiveness and scalability of FedSpy in collaborative speech steganalysis.

2. Related Work

Before describing FedSpy in detail, we first briefly review some representative deep-learning-based speech steganalysis methods proposed in recent years.

Take the detection of Quantization Index Modulation (QIM)-based steganography, one of the most widely used speech steganography methods, as an example. Early studies [5,6] on QIM-oriented speech steganalysis normally extracted hand-crafted and well-designed features from speech and conducted steganalysis using a Support Vector Machine (SVM) classifier. In 2017, Lin et al. [7] first introduced deep learning into QIM-oriented speech steganalysis. They found four codeword correlation patterns in VoIP streams and proposed a Recurrent Neural Network (RNN)-based steganalysis model (named RNN-SM), which achieved higher detection accuracy than those SVM-based methods [5,6]. Then, Yang et al. [8] combined Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) to propose a novel CNN-LSTM model to detect QIM-based steganography with better performance than RNN-SM. In this CNN-LSTM model, the Bi-LSTM network is used for capturing long-term contextual information from speech, while CNN is leveraged to extract local features of each speech frame. In follow-up work, Yang et al. proposed three more effective steganalysis methods based on the teacher–student model [9], the attention mechanism [10], and the multi-head attention mechanism [11], respectively, wherein the model in [11] (named FCEM) has the best performance in terms of both detection accuracy and detection efficiency. Recently, Qiu et al. [12] proposed a novel steganalysis model with distributed representations of codewords based on codeword embedding, Bi-LSTM, and Multi-Layer Perceptron, to further improve the detection accuracy in short-length and low-embedding-rate speech streams, and achieved state-of-the-art performance. (Note that we refer to this model as DRCM in this paper.) Meanwhile, Wei et al. [13] also used codeword embedding and Bi-LSTM to propose a QIM-oriented speech steganalysis method, which can achieve frame-level speech steganography detection.

In addition to QIM-based speech steganography, several deep-learning based steganalysis methods for detecting other speech steganography methods were also proposed recently. For instance, Tian et al. [14] proposed a speech steganalysis model using feature fusion and LSTM to detect Adaptive-Codebook-based steganography. Qiu et al. [15] designed a novel separable convolution network with a dual-stream pyramid-enhanced strategy to detect Fixed-Codebook-based steganography. For the general detection of multiple steganography methods, Hu et al. [16] proposed a novel deep learning model named Steganalysis Feature Fusion Network. Li et al. [17] presented a general steganalysis method based on codeword embedding, Bi-LSTM, and CNN with an attention mechanism. Tian et al. presented a novel Multi-Encoder Network to achieve efficient detection of multiple steganography methods [18].

In summary, deep-learning-based steganalysis is becoming a new trend in speech steganalysis. However, as we discussed above, the existing deep-learning-based steganalysis methods assume that an adequate number of speech samples (i.e., a training dataset containing sufficient samples) are available. But this assumption does not hold in collaborative steganalysis tasks when the clients only have a small number of samples and cannot share the samples directly. Therefore, we propose to utilize federated learning to address this challenge in this paper.

3. The Description of FedSpy

As shown in Figure 1, the system model of FedSpy consists of a Trusted Authority (TA), n clients, and a central server. TA is only responsible for distributing keys to the clients and the server in the initialization phase, and does not participate in the speech steganalysis tasks. Each client C_i has a local speech dataset D_i, including its steganographic samples and cover samples. The entire dataset (a.k.a, the global dataset) is denoted by $D = \cup_i^n D_i$. All clients collaborate in training a deep-learning-based speech steganalysis model by iteratively uploading their local models (i.e., $\boldsymbol{w}_i^{(t)}$) to the central server for aggregation (i.e., $\boldsymbol{w}^{(t)}$), where t indicates the round of iteration. Before describing the details of FedSpy, we would like to introduce two essential building blocks, i.e., deep-learning-based speech steganalysis and secure local model aggregation.

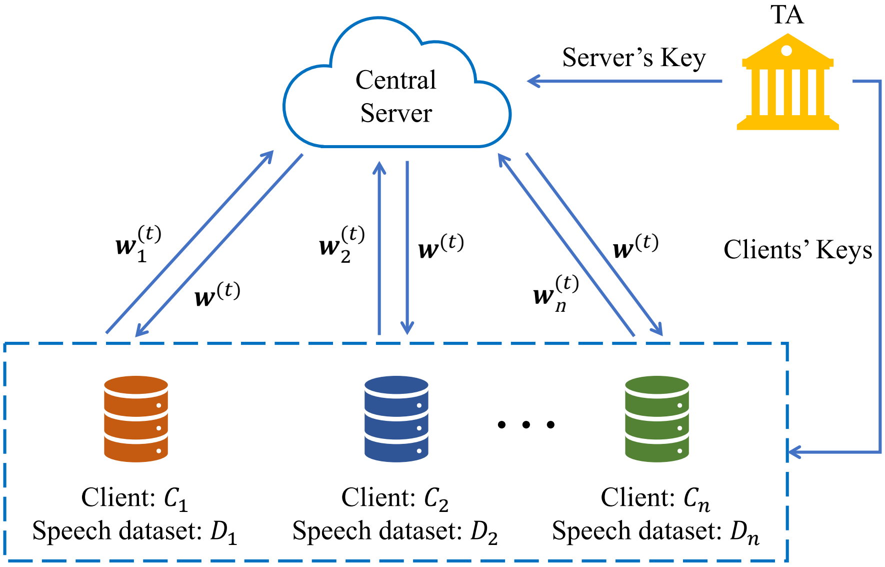


Figure 1. The system model of FedSpy.

3.1. The Building Blocks of FedSpy

Deep-learning-based Speech Steganalysis: As mentioned above, the current trend in speech steganalysis is to introduce deep learning to build effective steganalysis models. Without loss of generality, assume the entire training dataset is $D = \{\langle \boldsymbol{x}_k, y_k \rangle, k = 1, 2, \cdots, K\}$, where K is the number of the speech samples, $\boldsymbol{x}_k$ represents the feature of the k-th speech sample, and the label y_k indicates whether it contains secret information. The goal of the steganalysis model is to minimize the following loss function on the training set:

$$L(D, \boldsymbol{w}) = \frac{1}{K} \sum_{k=1}^{K} c(\boldsymbol{x}_k, y_k, \boldsymbol{w}) \tag{1}$$

where $\boldsymbol{w}$ is the parameter of the steganalysis model, and $c(\cdot)$ represents the classification loss function. The specific forms of $L(\cdot)$ and $c(\cdot)$ depend on the concrete steganalysis model. In FedSpy, since D is distributed over multiple clients, the loss function can be rewritten as:

$$L(D, \boldsymbol{w}) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} L(D_i, \boldsymbol{w}) \tag{2}$$

In general, the most common method of minimizing Equation (1) is using gradient descent (or its variation) [21] as follows:

$$\boldsymbol{w}^{(t+1)} \leftarrow \boldsymbol{w}^{(t)} - \lambda \nabla L(D, \boldsymbol{w}^{(t)}) \tag{3}$$

where $\boldsymbol{w}^{(t)}$ indicates the value of parameters $\boldsymbol{w}$ after the t-th iteration, and λ is the parameter of the learning rate. In FedSpy, we will calculate Equation (3) in a federated way, where $D = \cup_i^n D_i$.

Secure Local Model Aggregation: Recent research results have shown that the local models in federated learning (e.g., the gradients) can reveal the sensitive properties of the clients' data [22,23]. Thus, to protect data privacy, the local models cannot be sent to the server in plaintext. FedSteg [19] leverages the Paillier cryptosystem [24] to address this problem, in which each client encrypts its local model using the server's public key. However, this method allows the server to obtain all local models clearly. In other words, it cannot protect data privacy against an inside attacker at the server end. Instead, we integrate a state-of-the-art secure aggregation protocol [25] (which is also based on the Paillier cryptosystem) into FedSpy, where the server can only obtain the aggregated result of the local models of all clients. Specifically, this secure aggregation protocol consists of four algorithms as follows.

- $(PP, SK_{C_i}, SK_S) \leftarrow GenKey(\kappa)$. TA runs this algorithm to generate the system public parameter PP, the server's secret key SK_S, and each client C_i's secret key SK_{C_i}, $i \in \{1, 2, \cdots, n\}$, where κ is the security parameter.
- $[\![m_i]\!] \leftarrow Enc(m_i, SK_{C_i})$. Each client C_i runs this algorithm to encrypt a private message m_i with its secret key SK_{C_i}. The ciphertext is denoted by $[\![m_i]\!]$.
- $[\![m]\!] \leftarrow Aggre([\![m_1]\!], [\![m_2]\!], \cdots, [\![m_n]\!])$. This algorithm is run by the server. It takes as input n ciphertexts from the n client, and outputs the ciphertext of aggregated results, where $m = m_1 + m_2 + \cdots + m_n$.
- $m \leftarrow Dec([\![m]\!], SK_S)$. Given a ciphertext output by $Aggre(\cdot)$, the server runs this algorithm to decrypt it with its secret key SK_S, and obtains the aggregated results.

Note that in this secure aggregation protocol, the server can only decrypt the ciphertext aggregated from the n ciphertexts of the n clients. None of the clients' private messages can be revealed to the server. The details of this secure aggregation protocol and its security analysis can be found in [25].

3.2. The Details of FedSpy

The goal of FedSpy is to enable multiple clients to train a speech steganalysis model collaboratively. Before the training process, TA first generates the cryptographic keys for the server and the clients, which will be used for the secure aggregation of local models. Specifically, TA runs the algorithm $(PP, SK_{C_i}, SK_S) \leftarrow GenKey(\kappa)$, sends SK_S and SK_{C_i} to the server and the client C_i $(i = 1, 2, \cdots, n)$ through a secure channel, respectively, and publishes the system parameter PP. The secure channel can be established using secure communication protocols, such as the Transport Layer Security (TLS) protocol. Note that TA does not participate in the following training process.

The training algorithm of FedSpy (Algorithm 1) can be detailed as follows. First, according to the chosen deep-learning-based steganalysis methods, the server generates the initial global model parameter $\boldsymbol{w}^{(0)}$ and broadcasts to all clients (as shown in Line 2). Second, in the subsequent iteration, upon receiving the global model $\boldsymbol{w}^{(t)}$, each client performs the optimization algorithm (e.g., the gradient descent in Algorithm 1) with its individual speech dataset, thereby obtaining its local model $\boldsymbol{w}_i^{(t+1)}$ (as shown in Line 11). Note that we could use another optimizer instead of gradient descent in this step. More specifically, the detailed calculations of this step depend on the concrete steganalysis model (i.e., the deep learning network) integrated into FedSpy. For instance, if we opt for RNN-SM [7] as the underlying model, the parameter $\boldsymbol{w}^{(t)}$ encompasses three sets of weights, namely the Input Weights, the Connection Weights, and the Detection Weights. These weights are locally optimized by each client utilizing the Adam algorithm [26] with cross-entropy loss function. Then the local model $\boldsymbol{w}_i^{(t+1)}$ is encrypted with the client's secret key SK_{C_i} and uploaded to the server (as shown in Lines 12–13). Third, the server securely aggregates all local models to update the global model (as shown in Lines 4–6). The second and third steps are iterated until convergence or reaching a maximum iteration number T. Following the training process, each client can utilize the final global steganalysis model for local detection of steganographic speech samples.

Algorithm 1 The training Algorithm of FedSpy

```
procedure SERVER
    Initialize and broadcast $w^{(0)}$ to all clients
    for each round of iteration $t = 0, 1, \cdots, T-1$ do
        $[\![w^{(t+1)}]\!] \leftarrow Aggre([\![w_1^{(t+1)}]\!], [\![w_2^{(t+1)}]\!], \cdots, [\![w_n^{(t+1)}]\!])$
        $w^{(t+1)} \leftarrow Dec([\![w^{(t+1)}]\!], SK_S)$
        return $w^{(t+1)}$ to all clients
    end for
end procedure
procedure CLIENT $C_i$
    for each round of iterations $t = 0, 1, \cdots, T-1$ do
        $w_i^{(t+1)} = \frac{|D_i|}{|D|}(w^{(t)} - \lambda \nabla L(D_i, w^{(t)}))$
        $[\![w_i^{(t+1)}]\!] \leftarrow Enc(w_i^{(t+1)}, SK_{C_i})$
        return $[\![w_i^{(t+1)}]\!]$ to the server
    end for
end procedure
```

4. Performance Evaluation

In this section, we implement different speech steganalysis methods in FedSpy to show its effectiveness and scalability, as well as to compare it with the FedSteg framework [19], the centralized models, and the local models. Our experiments are implemented using Python on a server with Intel E5-2680 V4 CPU, NVIDIA GeForce RTX 2080 Ti GPU, and 30 GB RAM.

4.1. Basic Steganalysis and Target Steganography Methods

In our experiments, we representatively port three state-of-the-art steganalysis models (i.e., RNN-SM [7], FCEM [11], and DRCM [12]) into FedSpy, although FedSpy can be extended to many other speech steganalysis methods based on deep learning. Like in RNN-SM, FCEM, and DRCM, we take the Complementary Neighbor Vertices (CNV) algorithm [27], a typical approach based on QIM [28], as our benchmark target. Despite this, it is not difficult to see that FedSpy can also be applied to the detection of other speech steganography methods, as long as we adopt the corresponding basic steganalysis method in each client.

4.2. Utilized Dataset

We collected 25,000 speech samples from audio materials for language learning with an 8 kHz sampling rate and 16 bits quantization, including 12,500 English samples and 12,500 Chinese samples. Each sample is one second long and encoded by G.729a speech codec. Furthermore, we produce the corresponding steganographic samples with 10 different embedding rates (i.e., 10%, 20%, $\cdots$, 100%) using the CNV-QIM steganographic method [27], in which the secret messages are simulated as random binary sequences. Therefore, for each embedding rate, we have a total of 50,000 samples, including 25,000 cover and 25,000 steganographic samples. Since the clients usually consist of a few stable entities (e.g., several security agencies), we first fix the number of clients at 4, where each client owns 5000 cover samples and 5000 steganographic samples as training data. The rest of the 5000 cover samples and 5000 steganographic samples are reserved as the testing data. Then, we also investigate the performance of FedSpy with a different number of clients.

4.3. Experimental Results and Performance Evaluation

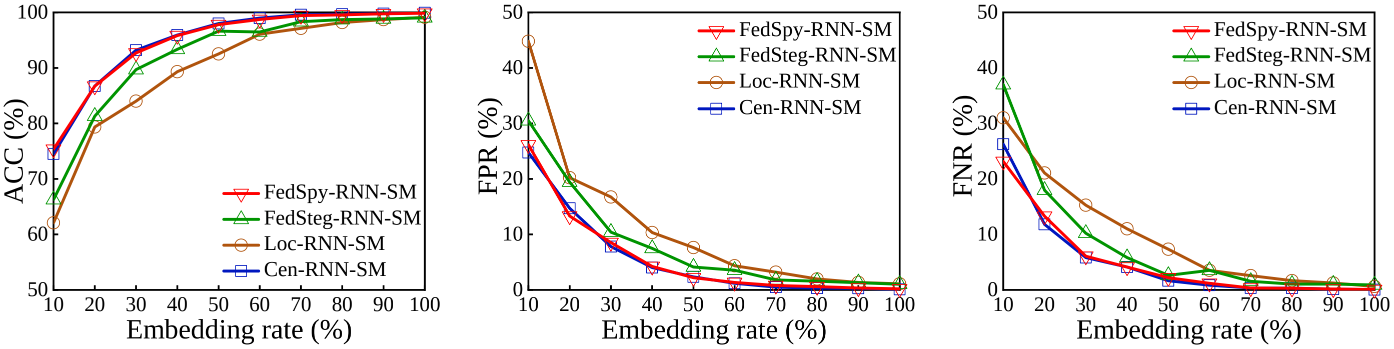
We implement four types of steganalysis models based on RNN-SM [7], FCEM [11], and DRCM [12], respectively, including

1. FedSpy-RNN-SM (resp. FedSpy-FCEM or FedSpy-DRCM), incorporating RNN-SM (resp. FCEM or DRCM) into FedSpy.
2. FedSteg-RNN-SM (resp. FedSteg-FCEM or FedSteg-DRCM), incorporating RNN-SM (resp. FCEM or DRCM) into FedSteg [12]. In FedSteg, each client has a personalized steganalysis model after transfer learning. Here, we take the average performance of all personalized models as a reference.
3. Loc-RNN-SM (resp. Loc-FCEM or Loc-DRCM), leveraging RNN-SM (resp. FCEM or DRCM) to create a local model for each client with the corresponding local sample set. Here, we take the average performance of all local models as a reference.
4. Cen-RNN-SM (resp. Cen-FCEM or Cen-DRCM), leveraging RNN-SM (resp. FCEM or DRCM) to implement a centralized model with all clients' samples in a centralized manner.

In addition, for RNN-SM, FCEM, and DRCM, we use Adam as the optimizer and cross-entropy as the loss function. The initial learning rate is 0.001, and the batch size is 64. The architectures of RNN-SM, FCEM, and DRCM can refer to [7], [11], and [12], respectively. Since the encryption algorithm in FedSpy can only take integers as input, we expand the model parameters by 1000 times and take the integer part of the parameters for calculations.

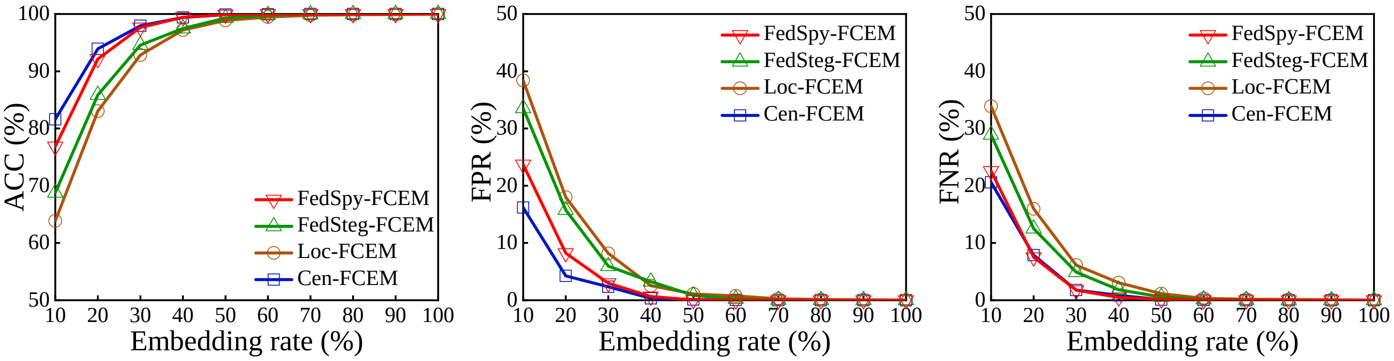
4.3.1. The Analysis on Detection Performance

First, we evaluate the detection performance of FedSpy, including three metrics: Accuracy (ACC), False-Positive Rate (FPR), and False-Negative Rate (FNR). The obtained results are shown in Figures 2–4, from which we can learn the following results.



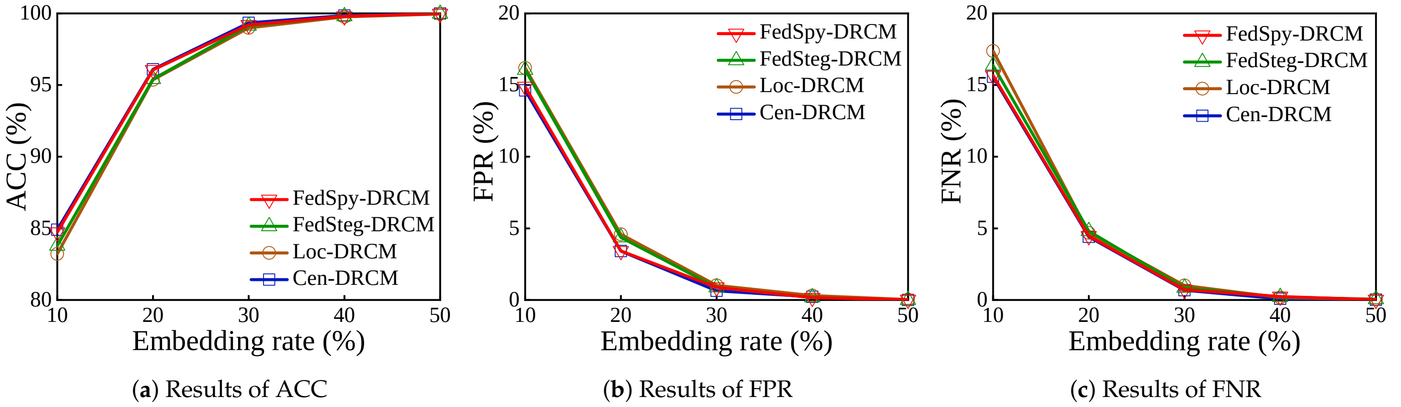
(**a**) Results of ACC (**b**) Results of FPR (**c**) Results of FNR

Figure 2. The experimental results for RNN-SM-based methods.



(**a**) Results of ACC (**b**) Results of FPR (**c**) Results of FNR

Figure 3. The experimental results for FCEM-based methods.



(**a**) Results of ACC (**b**) Results of FPR (**c**) Results of FNR

Figure 4. The experimental results for DRCM-based methods.

First, for all of RNN-SM, FCEM, and DRCM, FedSpy is superior to FedSteg and the local model in terms of ACC, FPR, and FNR, particularly at the low embedding rates. For instance, when the embedding rate equals 20%, as shown in Figure 3a, FedSpy-FCEM can achieve an ACC greater than 90%, while the ACC of FedSteg-FCEM and the local model is approximately 85%. Even though DRCM has achieved excellent detection performance, FedSpy-DRCM also gains a visible improvement compared to FedSteg-DRCM and the local model when the embedding rate is low (e.g., 10%), as shown in Figure 4.

Second, FedSpy can achieve good detection performance comparable to the centralized model. For example, for the samples with an embedding rate of 20%, the centralized model of RNN-SM can achieve an ACC of 86.74%, while FedSpy-RNN-SM can also achieve an ACC of 86.70%.

Third, the detection performance of FedSpy largely depends on the underlying steganalysis method. For example, as shown in Tables 1–3, for the speech samples with an embedding rate of 20%, FedSpy-DRCM outperforms FedSpy-RNN-SM and FedSpy-FCEM in ACC, FPR, and FNR. They are highly consistent with the comparison results for the centralized models of the underlying steganalysis methods, where DRCM performs best.

Table 1. The statistical results of ACCs for samples of 20% embedding rate.

	FedSpy	**FedSteg**	**Local Model**	**Central Model**
RNN-SM	86.70%	81.31%	79.36%	86.74%
FCEM	92.13%	85.85%	83.03%	93.93%
DRCM	96.06%	95.42%	95.37%	96.10%

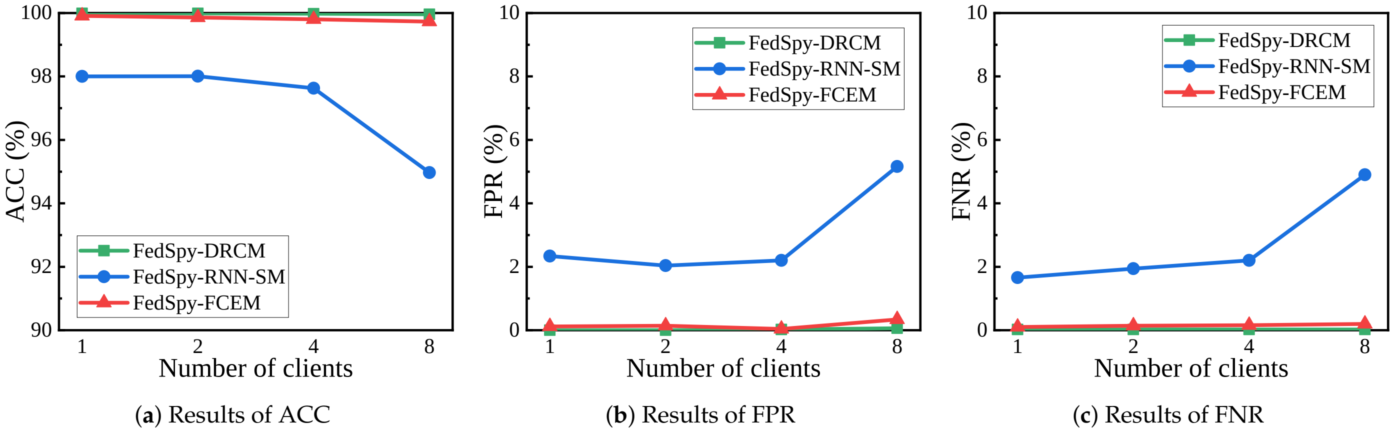
Table 2. The statistical results of FPRs for samples of 20% embedding rate.

	FedSpy	**FedSteg**	**Local Model**	**Central Model**
RNN-SM	13.32%	19.41%	20.21%	14.72%
FCEM	8.24%	15.78%	17.99%	4.26%
DRCM	3.42%	4.36%	4.58%	3.41%

Table 3. The statistical results of FNRs for samples of 20% embedding rate.

	FedSpy	**FedSteg**	**Local Model**	**Central Model**
RNN-SM	13.28%	17.98%	21.07%	11.82%
FCEM	7.50%	12.52%	15, 96%	7.88%
DRCM	4.46%	4.80%	4.68%	4.40%

In the above experiments, the number of clients is initialized to four. In order to further examine the performance of FedSpy, we conduct additional tests to evaluate the impact of the number of clients on ACC, FPR, and FNR in FedSpy, with an embedding rate of 50%. As illustrated in Figure 5a, as the number of clients increases, the ACC of FedSpy-RNN shows a slight decline from 98% to 95%. However, both FedSpy-FCEM and FedSpy-DRCM maintain a consistently high level of ACC, surpassing 99%, with minimal changes. The influence of the number of clients on both FPR and FNR aligns with its impact on ACC, as shown in Figure 5b and Figure 5c, respectively. This is due to the nature of federated learning in FedSpy, where although the number of clients may change and consequently the number of local samples per client may vary, the total number of training samples remains constant. Thus, we can conclude that the impact of the number of clients in FedSpy may depend on the underlying steganalysis model (e.g., RNN-SM). However, the overall performance of FedSpy is independent of the number of clients, as long as the total number of samples remains constant.



(**a**) Results of ACC (**b**) Results of FPR (**c**) Results of FNR

Figure 5. The impact of the number of clients on the ACC, FPR, and FNR in FedSpy

4.3.2. The Analysis on Detection Time

In addition to the preceding analysis on detection performance, we also examine the time complexity of FedSpy. The time complexity of FedSpy, as well as other steganalysis methods based on deep learning, can be divided into two parts, namely the training time and the detection time. Due to the complex architecture of the steganalysis model and the large volume of the training data, the training phase typically demands a substantial amount of time. For instance, the average training duration for a steganalysis model (e.g., RNN-SM, FCEM, and DRCM) spans approximately 20 min in our experiment. Thus, like other research works on speech steganalysis [7,11,12], we only evaluate the detection time of FedSpy, which is crucial in real-time speech steganalysis tasks. Theoretically, since the underlying deep learning network remains unchanged, the detection time of the models trained within FedSpy should align closely with that of the centralized models. Figure 6 shows the comparison results of detection time between the models trained within FedSpy and the centralized models. It is evident that the detection times of both approaches are essentially identical. Owing to the excellent design of the underlying models (i.e., RNN-SM, FCEM, and DRCM), FedSpy-RNN-SM, FedSpy-FCEM, and FedSpy-DRCM are all capable of detecting a speech segment of one second in less than 1 millisecond. Therefore, it can be inferred that FedSpy does not influence the detection time of speech steganalysis, as it is determined by the integrated steganalysis models.

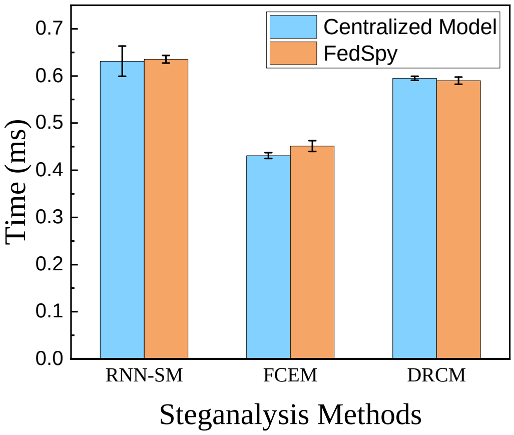


Figure 6. The time of detecting a single sample with one second length and 10% embedding rate.

5. Conclusions and Future Work

In this paper, we presented an effective Secure Collaborative Speech Steganalysis Framework based on federated learning, called FedSpy, which can be organically combined with various deep-learning-based methods. The experimental results show that the steganalysis models in conjunction with FedSpy perform significantly better than each client's local steganalysis model, and achieve good performance comparable to the centralized steganalysis model without privacy protection. To the best of our knowledge, this work is the first exploration of secure collaborative speech steganalysis, and despite achieving remarkable results, there are inevitably some shortcomings as follows. First, this work aims to explore the possibility of applying federated learning to collaborative speech steganalysis with a concern for data privacy. However, there are other security issues to be considered in our future work, such as an insider attack (e.g., the poisoning attack [29]) in the presence of malicious inside participants. Second, in our experiments, all steganographic samples are generated with widely used CNV-QIM algorithm, and are thereby Independent and Identically Distributed (IID). In the future, we would also extend FedSpy to support the detection of multiple speech steganography methods, which is typically a challenging problem of federated learning on non-IID data [30], since steganographic samples generated by different steganographic methods are non-IID. Moreover, we intend to delve into the interpretability, robustness, and generality of the collaborative speech steganalysis models in our future research, which would allow us to gain a better knowledge of how the steganalysis models work.

Author Contributions: Conceptualization, H.T.; methodology, H.T., H.W. and H.Q.; validation, H.W. and H.Q.; formal analysis, H.W. and H.Q.; investigation, H.W. and H.Q.; resources, H.T.; data curation, H.W. and H.Q.; writing—original draft preparation, H.W.; writing—review and editing, H.T., H.Q., W.M. and C.-C.C.; visualization, H.W.; supervision, H.T.; funding acquisition, H.T. and H.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (No. 61972168), the Natural Science Foundation of the Fujian Province, China (No. 2020J05059), and the Scientific Research Funds of Huaqiao University (No. 19BS307).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors would like to thank the editors and reviewers for their valuable comments and helpful suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FL	Federated Learning
QIM	Quantization Index Modulation
SVM	Support Vector Machine
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
TA	Trusted Authority
CNV	Complementary Neighbor Vertices
ACC	Accuracy
FPR	False-Positive Rate
FNR	False-Negative Rate
IID	Independent and Identically Distributed

References

1. Tian, H.; Sun, J.; Chang, C.C.; Huang, Y.; Chen, Y. Detecting bitrate modulation-based covert voice-over-IP communication. *IEEE Commun. Lett.* **2018**, *22*, 1196–1199. [CrossRef]
2. Huang, Y.F.; Tang, S.; Yuan, J. Steganography in inactive frames of VoIP streams encoded by source codec. *IEEE Trans. Inf. Forensics Secur.* **2011**, *6*, 296–306. [CrossRef]
3. Lin, Q.H.; Yin, F.L.; Mei, T.M.; Liang, H. A blind source separation based method for speech encryption. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2006**, *53*, 1320–1328.
4. Xie, S.; Yang, Z.; Fu, Y. Nonnegative matrix factorization applied to nonlinear speech and image cryptosystems. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2008**, *55*, 2356–2367.
5. Li, S.B.; Tao, H.Z.; Huang, Y.F. Detection of quantization index modulation steganography in G. 723.1 bit stream based on quantization index sequence analysis. *J. Zhejiang Univ. Sci. C* **2012**, *13*, 624–634. [CrossRef]
6. Li, S.; Jia, Y.; Kuo, C.C.J. Steganalysis of QIM steganography in low-bit-rate speech signals. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2017**, *25*, 1011–1022. [CrossRef]
7. Lin, Z.; Huang, Y.; Wang, J. RNN-SM: Fast steganalysis of VoIP streams using recurrent neural network. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1854–1868. [CrossRef]
8. Yang, H.; Yang, Z.; Huang, Y. Steganalysis of VoIP streams with CNN-LSTM network. In Proceedings of the ACM Workshop on Information Hiding and Multimedia Security, Paris, France, 3–5 July 2019; pp. 204–209.
9. Yang, H.; Yang, Z.; Bao, Y.; Liu, S.; Huang, Y. Fast steganalysis method for VoIP streams. *IEEE Signal Process. Lett.* **2019**, *27*, 286–290. [CrossRef]
10. Yang, H.; Yang, Z.; Bao, Y.; Huang, Y. Hierarchical representation network for steganalysis of qim steganography in low-bit-rate speech signals. In *International Conference on Information and Communications Security*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 783–798.
11. Yang, H.; Yang, Z.; Bao, Y.; Liu, S.; Huang, Y. Fcem: A novel fast correlation extract model for real time steganalysis of VOIP stream via multi-head attention. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 2822–2826.
12. Qiu, Y.; Tian, H.; Tang, L.; Mazurczyk, W.; Chang, C.C. Steganalysis of adaptive multi-rate speech streams with distributed representations of codewords. *J. Inf. Secur. Appl.* **2022**, *68*, 103250. [CrossRef]
13. Wei, M.; Li, S.; Liu, P.; Huang, Y.; Yan, Q.; Wang, J.; Zhang, C. Frame-level steganalysis of QIM steganography in compressed speech based on multi-dimensional perspective of codeword correlations. *J. Ambient. Intell. Humaniz. Comput.* **2023**, *14*, 8421–8431. [CrossRef]
14. Tian, H.; Qiu, Y.; Mazurczyk, W.; Li, H.; Qian, Z. STFF-SM: Steganalysis Model Based on Spatial and Temporal Feature Fusion for Speech Streams. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2022**, *31*, 277–289. [CrossRef]
15. Qiu, Y.; Tian, H.; Li, H.; Chang, C.C.; Vasilakos, A.V. Separable Convolution Network with Dual-Stream Pyramid Enhanced Strategy for Speech Steganalysis. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 2737–2750. [CrossRef]
16. Hu, Y.; Huang, Y.; Yang, Z.; Huang, Y. Detection of heterogeneous parallel steganography for low bit-rate VoIP speech streams. *Neurocomputing* **2021**, *419*, 70–79. [CrossRef]
17. Li, S.; Wang, J.; Liu, P.; Wei, M.; Yan, Q. Detection of multiple steganography methods in compressed speech based on code element embedding, Bi-LSTM and CNN with attention mechanisms. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2021**, *29*, 1556–1569. [CrossRef]
18. Tian, H.; Wu, J.; Quan, H.; Chang, C. Detecting Multiple Steganography Methods in Speech Streams Using Multi-Encoder Network. *IEEE Signal Process. Lett.* **2022**, *29*, 2462–2466. [CrossRef]
19. Yang, H.; He, H.; Zhang, W.; Cao, X. FedSteg: A federated transfer learning framework for secure image steganalysis. *IEEE Trans. Netw. Sci. Eng.* **2020**, *8*, 1084–1094. [CrossRef]

20. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–19. [CrossRef]
21. Wikipedia Contributors. Gradient Descent—Wikipedia, The Free Encyclopedia. 2023. Available online: https://en.wikipedia.org/wiki/Gradient_descent (accessed on 23 June 2023).
22. Zhu, L.; Liu, Z.; Han, S. Deep leakage from gradients. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 14747–14756.
23. Melis, L.; Song, C.; De Cristofaro, E.; Shmatikov, V. Exploiting unintended feature leakage in collaborative learning. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; pp. 691–706.
24. Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In Proceedings of the Advances in Cryptology—EUROCRYPT'99, Prague, Czech Republic, 2–6 May 1999; pp. 223–238.
25. Wang, F.; Zhu, H.; Lu, R.; Zheng, Y.; Li, H. A privacy-preserving and non-interactive federated learning scheme for regression training with gradient descent. *Inf. Sci.* **2021**, *552*, 183–200. [CrossRef]
26. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
27. Xiao, B.; Huang, Y.; Tang, S. An approach to information hiding in low bit-rate speech stream. In Proceedings of the IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference, New Orleans, LA, USA, 8 December 2008; pp. 1–5.
28. Chen, B.; Wornell, G.W. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. *IEEE Trans. Inf. Theory* **2001**, *47*, 1423–1443. [CrossRef]
29. Wang, Z.; Ma, J.; Wang, X.; Hu, J.; Qin, Z.; Ren, K. Threats to Training: A Survey of Poisoning Attacks and Defenses on Machine Learning Systems. *ACM Comput. Surv.* **2022**, *55*, 1–36. [CrossRef]
30. Zhu, H.; Xu, J.; Liu, S.; Jin, Y. Federated learning on non-IID data: A survey. *Neurocomputing* **2021**, *465*, 371–390. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.