Project Report – Privacy-Preserving and Explainable Federated Learning for Robust Digital Forensics

Advanced Research Topics in IT Security (ARTIS)

Omar Abushanab (mail)
Ibrahim Selim (mail)
Malak Abdelaziz (mail)
Peter Schropp (mail)
Matthias Högel (matthias.hoegel@uni-ulm.de)

January 8, 2026

1. Introduction

Introduction of the paper.

2. Related Work

Related work of the paper.

3. Background

3.1. Digital Forensics and the use of Machine Learning

A perpetrator always leaves traces of evidence of their involvement at the crime scene, as described by Dr. Edmond Locard in his exchange principles, which are used in forensic science [DF 3].

The landscape of forensic science has changed due to the rise of electronic devices, which play an increasingly important role in our daily lives and are often connected to the internet and accessible from anywhere. The field has expanded since the early 2000s with digital forensics (DF), which specializes more in the growing number of cybercrimes. However, even crimes that are not classified as cybercrime are becoming increasingly digital in most modern crime scenes. According to the EU, digital evidence is involved in 85% of criminal investigations. This evidence consists of data generated in our daily lives through the use of digital devices, leaving behind a digital footprint. The footprint consists of data generated by wearable devices, emails, cloud service providers, online payments, and other sources [DF 1].

The field of digital forensics can be divided into seven identifiable sub-areas, namely blockchain, networks, mobile, cloud, IoT, file systems & data storage, and multimedia. This project is limited to the sub-area of multimedia, which specializes in image forgery [DF 1].

A major challenge in this field is dealing with large, complex data sets and classifying them. This problem can be addressed by machine learning (ML), whose techniques have expanded and improved in recent years. ML techniques search through the data and look for anomalies and patterns in the investigation process. The largest area for ML in DF is image forensics, accounting for 62.7%. In the field of image forensics, convolutional neural networks (CNNs) are typically used to recognize such complex patterns in the data, which is why this approach was pursued in the project. [DF 3]

3.2. CNNs for face fake detection

3.2.1 Fundamentals

In recent years, there has been intensive research into CNNs in the areas of image classification, facial recognition, and facial expressions, resulting in significant improvements. However, with the emergence of increasingly sophisticated deepfakes, it is becoming more and more difficult to distinguish between real and fake faces [CNN 1]. During the training process, CNNs learn complex patterns and can provide information about whether an input image is fake or real. Among CNN architectures, EfficientNet is one of the most modern models, as it is faster, has fewer parameters and has more capabilities for extracting features than other CNN models. These parameters are also called weights and can be learned by the model during the training process. During the training process, the model is shown many input images of positive (real faces) and negative (fake faces) [5]. The model independently learns complex patterns in the form of weights over several iterations (epochs) in order to distinguish positive from negative examples in the case of a binary classifier, or to assign probabilities to classes in the case of a multiple classifier system. [3][1]. What is special about CNNs are the convolutional layers, which reduce the resolution of the images and extract the spatial local features through weighted convolutions. Such local features can be low-level features such as edges, end points and corners in the first convolutional layers and complex high-level features in the last layers. In the final layers, the high-level features (3D vector) are combined into a fully connected layer (1D) to make a classification decision. In this project, the output consists of two output nodes with probabilities indicating whether the input image is a fake or real face [5].

3.2.2 Transfer Learning

3.2.3 Model Explainability using SHAP

3.3. Federated Learning

3.4. Attack Vectors in FL

4. Methodology

4.1. Experimental Setup

A fake face detection model is developed using federated learning (FL) to distinguish between real and fake faces. First, a centrally trained model without FL is implemented and used as a baseline for performance comparison. Second, a decentralized fake face detection model is trained using FL in a simulated multi-client environment.

4.1.1 Dataset

The well-known 140k real-fake faces dataset was used, which consists of 70,000 real faces and 70,000 fake faces with a image size of 256px [4]. The fake faces were generated using StyleGAN, a generative adversarial network developed by NVIDIA that is capable of producing highly photorealistic synthetic facial images.

4.1.2 Model

The base model chosen is EfficientNet-B0, a convolutional neural network (CNN), which was also used by Khudeyer et al. [2]. The authors trained a fake face detection model using transfer learning. For this purpose, the model was initialized with the pretrained weights of EfficientNetB0 on the ImageNet dataset. A lightweight head was attached to the pre-trained base model,

Epoch	**Learning rate**
$epoch \leq 2$	0.01
$2 < epoch \leq 15$	0.001
$epoch > 15$	0.0001

Table 1: Adjustment of the learning rate during training.

consisting of global average pooling, a 256-dimensional fully connected layer with ReLU activation, batch normalization, and dropout, followed by a 2 dimensional softmax output layer. The output is a 2-dimensional vector with probabilities indicating whether the input image is a fake or real face. The model was optimized with binary cross-entropy loss and the Adam optimizer. All input images for training were resized to 244px.

4.2. Centralized Fake Face Detection Model

The authors of [2] developed a method for fake face detection using CNN, which achieves an accuracy of 99.06%. This approach is used as a benchmark for comparing central fake face detection with fake face detection using FL. This work was reimplemented for verification purposes in order to ensure a meaningful comparison.

The dataset was divided into 100.000 training images, 20.000 test images, and 20.000 validation images. EfficientNetB0 with the hyperparameters of section 4.1.2 is used. Training was performed with a batch size of 32 over 30 epochs with early stopping to reduce training time.

The paper presented a learning rate scheduler that adjusts the learning rate during training based on the epoch, as shown in Table 1. The learning rate scheduler ensures that significant weight adjustments are made early in training. Furthermore, in later iterations, a strong adjustment is prevented by the decreasing learning rate. This leads to faster convergence in early epochs, while weight optimizations can be performed in later itterations.

Due to time constraints, the model was only trained once, as training was very computationally intensive due to the large amount of data.

4.3. Decentralized Fake Face Detection using FL

In this section a decentralized trained fake face detection model is developed. The training process will be explained below. In addition, we will discuss how privacy attacks can be prevented during training.

4.3.1 Research Scenario

The following scenario is fictional.

Several research organizations want to work together to train a fake face detection model. Each individual organization has images of fake faces, but also images of faces that belong to their customers. An ML model should be trained together that can distinguish between real and fake images. To do this, a large data set containing all images would have to be created in order to train the model. However, all organizations are interested in protecting the privacy of their customers and therefore do not want to share the real images. The solution is to train the model using FL. This fake face detection model should be trained with all data from all organizations and should be available to everyone without the need to share data between organizations.

4.3.2 Thread Model

All participants in the fake face detection training process are trusted. This means that all model performance attacks that seek to undermine the convergence of the global model can be excluded. After training, participants send their model weights to a trusted third-party server.

This ensures the secure aggregation of weights. Weights sent by participants to the trusted third party could be captured during transmission. Capturing the weights of individual organizations represents an attack vector for privacy attacks. This attack vector should be reduced by the proposed encryption.

4.3.3 FL Simulation

A simulation environment is developed to simulate a global model with any number of participants. The EfficientNetB0 with the hyperparameters of section 4.1.2 is used as the base model. A batch size of 128 is selected. The training data set was divided among five participants, with each receiving approximately 20,000 images, comprising 10,000 real faces and 10,000 fake faces. On the server, the global model is initialized with the weights from EfficientNetB0, which was pre-trained on the ImageNet dataset. The global model is trained in 10 epochs.
Each epoch consists of the following steps.
First, the weights of the global model are distributed among the 5 participants. Each participant initializes their local model with the global weights. The participants train their local models in 3 epochs. The weights of the local models are encrypted to prevent privacy attacks by intercepting the weights. The server receives the weights and decrypts them. The global model parameters are updated using weighted federated averaging (FedAvg), where each client's contribution is proportional to the number of local samples.

4.3.4 FL attack mitigations

If the local updates of the participants are captured during transmission to the server, privacy attacks on the participants' data can be carried out. To counteract this, the weight updates must be encrypted. The weights are serialized in bytes and encrypted with AES-128. Each participant has a secret key that the server knows as a trusted third party. When a server receives the encrypted data, it can decrypt it for further processing.

4.4. Explainability using SHAP

In order to analyze the centrally trained model and the decentralized trained model, the decisions made by the models should be compared for individual inputs. The aim is to clarify whether the models have developed differently as a result of the different training processes and whether they focus on different details when making decisions. The SHAP method is used for this purpose.

5. Results

This are the results of the paper.

6. Discussion

This is the discussion of the paper.

7. Conclusion

This is the conclusion of the paper.

References

[1] Bilal Hadjadji, Youcef Chibani, and Yasmine Guerbai. "Multiple one-class classifier combination for multi-class classification". In: *2014 22nd International Conference on Pattern Recognition*. IEEE. 2014, pp. 2832–2837.

[2] Raidah Salim Khudeyer and Noor Mohammed Almoosawi. “Fake Image Detection Using Deep Learning”. In: *Informatica* 47.7 (2023).

[3] Ana Carolina Lorena, André CPLF De Carvalho, and João MP Gama. “A review on the combination of binary classifiers in multiclass problems”. In: *Artificial Intelligence Review* 30 (2008), pp. 19–37.

[4] Bojan Tunguz. *140K Real and Fake Faces*. `https://www.kaggle.com/datasets/xhlulu/140k-real-and-fake-faces`. Accessed: 2025-01-05. 2020.

[5] Wei Wang et al. “Development of convolutional neural network and its application in image classification: a survey”. In: *Optical Engineering* 58.4 (2019), pp. 040901–040901. DOI: `10.1117/1.OE.58.4.040901`.