A Blockchain-Enhanced Reversible Watermarking Framework for End-to-End Data Traceability in Federated Learning Systems

3 authors:

Reda Bellafqira
IMT Atlantique

Gouenou Coatrieux
IMT Atlantique, France, Brest

Chloé Berton
IMT Atlantique

A Blockchain-Enhanced Reversible Watermarking Framework for End-to-End Data Traceability in Federated Learning Systems

1st Reda Bellafqira
IMT Atlantique
INSERM UMR 1101 Latim
Brest, France
reda.bellafqira@imt-atlantique.fr

2nd Chloé Berton
IMT Atlantique
INSERM UMR 1101 Latim
Brest, France
chloe.berton@imt-atlantique.fr

3rd Gouenou Coatrieux
IMT Atlantique
INSERM UMR 1101 Latim
Brest, France
gouenou.coatrieux@imt-atlantique.fr

***Abstract*—In federated learning (FL) environments, ensuring data traceability presents significant challenges, particularly when data move between multiple entities such as data centers, edge nodes, and data scientists. This paper presents a novel framework that combines robust reversible watermarking and blockchain technology to achieve end-to-end traceability of medical images in a FL context. Based on the watermark, it becomes possible to interrogate the blockchain about the life cycle of an image to ensure data traceability, authenticity, and integrity. We use a histogram shifting-based reversible watermarking scheme with a new overflow management procedure, integrated with a private blockchain that records all watermarking and verification operations. Experimental results demonstrate the effectiveness of our approach in terms of watermark robustness considering a chest X-ray image dataset. We further show that watermarking does not interfere in the training and inference phase of a VGG-16 classification model for a Covid-19 medical database. A model trained on protected data can be used to classify non-watermarked data as well.**

***Index Terms*—Reversible Robust Watermarking, Blockchain, Histogram shifting, Federated Learning.**

I. Introduction

Federated Learning (FL) has emerged as a paradigm in machine learning, enabling collaborative model training while preserving data privacy through decentralized computation. This approach is particularly valuable in healthcare [1]–[3], where strict regulatory requirements like HIPAA and GDPR traditionally limit data sharing, despite the potential benefits of leveraging vast amounts of sensitive patient data for advancing medical research and improving diagnostic capabilities.

In a typical FL environment [4], multiple entities collaborate in a structured hierarchy: data providers (DPs) maintain their local datasets, a central server (CS) orchestrates the training process, and data scientists (DS) develop models without direct access to the sensitive training data. Each data provider operates through an edge node that processes data locally and interfaces with the central server, which manages communications between data scientists and the federated network of edge nodes. This architecture enables data scientists to train models across distributed datasets while maintaining data provider privacy and regulatory compliance.

Traditional security mechanisms like data watermarking and blockchain technology offer partial solutions to these challenges. Watermarking can embed metadata to record identifiers that can be used to trace the data back to its owner [5], [6], integrity proof [7]–[9], authenticity proof [10], [11]. However, watermarking alone struggles with the complex data flows in FL environments where multiple entities interact without direct data transfers. Meanwhile, blockchain provides immutable, transparent record-keeping [12] but lacks mechanisms for identifying the source of data leaks.

This paper presents a novel framework that synergistically combines robust reversible watermarking with blockchain technology to achieve end-to-end traceability of medical images in FL environments. Our approach integrates a histogram shifting-based reversible watermarking scheme featuring an new overflow management procedure with a private blockchain that records all watermarking and verification operations. The blockchain stores cryptographic hashes of both original and watermarked images for integrity verification, along with encrypted watermarking parameters that enable watermark removal and ownership validation. Each blockchain entry is cryptographically signed by the data provider to authenticate both the watermarking operations and the associated data.

The proposed watermarking algorithm employs histogram shifting on prediction errors, chosen for its computational efficiency and high capacity. To enhance robustness against various attacks, we implement a fixed 256-bit watermark with duplication to utilize the available capacity, employing majority voting for watermark recovery. Experimental results demonstrate our approach's effectiveness in maintaining watermark robustness while preserving image quality on a chest X-ray dataset. Furthermore, we show that the watermarking process does not adversely affect the training or inference capabilities of a VGG-16 classification model for COVID-19 diagnosis, with models trained on protected data maintaining their effectiveness on non-watermarked data.

This work was partly funded by the French government grants managed by the Agence Nationale de la Recherche under the references ANR-22PESN-0006 and the European Union under Grant Agreement 101070222.

The remainder of this paper is organized as follows: Section II presents background information and related work on digital watermarking, blockchain technology, and their combined applications. Section III details our proposed method for end-to-end traceability. Section IV presents experimental results and analysis, and Section V concludes with a discussion of implications and future research directions.

II. Background & Related Work

A. Digital Watermarking

Digital watermarking [13] is a technique used to embed identifiable and traceable information within digital data. The embedded information, or "watermark," is designed to be imperceptible to human users while still being detectable and verifiable by authorized systems or individuals. Watermarking can assert several security properties, such as ownership [11], [14] and traceability [15], [16], by including the owner's ID and the receiver's ID in the embedded message, respectively. Watermarking is a symmetric process, where the same secret key is used to embed and extract the watermark from the data.

The watermarking properties include imperceptibility, robustness, reversibility, and capacity. Imperceptibility refers to the watermark's hiddenness, ensuring it does not degrade the original content's quality. Robustness is the watermark's ability to resist tampering, removal, or degradation. Reversibility allows watermark extraction without affecting the original data, while capacity is the amount of information that can be embedded without compromising other properties. These properties must be carefully balanced to ensure sufficient protection and traceability while maintaining usability and quality.

Watermarking consists of three steps: watermark generation, watermark embedding, and watermark extraction. Watermark generation involves creating a unique watermark from a message using a hash function and a secret key. Watermark embedding incorporates the watermark into the data using the secret key by modifying the image's pixel values or other properties. Watermark extraction retrieves the embedded watermark using the secret key to verify ownership or trace the origin of a leak. Robust watermarking guarantees copyright protection but does not consider insertion distortion, while reversible watermarking allows lossless watermark extraction and retrieval, maintaining data integrity. However, reversible watermarking schemes are often fragile and unable to resist attacks. Robust reversible watermarking (RRW) combines both benefits, making it suitable for sensitive applications such as medical imaging, military, and remote sensing.

Existing RRW solutions typically combine two watermarking schemes, one robust and one reversible, using various techniques such as embedding watermarks in different image domains [17], pseudorandom code indexing [18], or Pixel Value Ordering (PVO) [19], [20]. However, these schemes often suffer from high distortion, making them unsuitable for applications involving training AI models on watermarked data, such as in Federated Learning environments. In Section III, we propose a histogram shifting modulation based on an overflow management procedure that does not impact AI model accuracy, as demonstrated in Section IV.

B. Blockchain

Blockchain is a decentralized and distributed ledger technology that records transactions in a secure, transparent, and immutable manner [21]–[24]. A blockchain consists of a series of blocks, each containing a set of transactions, with each block linked to the previous one through a cryptographic hash. This structure ensures that once data are recorded in a block, they cannot be altered without changing all subsequent blocks, making blockchain highly resistant to tampering and fraud. The consensus mechanisms employed, such as Proof of Work (PoW) or Proof of Stake (PoS), validate transactions across a network of nodes, ensuring that all participants in the system agree on the current state of the ledger.

Blockchain technology offers several significant advantages in terms of data security. First, it provides an immutable record of all data access and watermarking activities, ensuring a secure and verifiable audit trail. By referencing the blockchain, data ownership can be quickly validated, and data integrity can be verified. Moreover, the decentralized nature of blockchain allows multiple copies of the ledger to be stored by different entities, enhancing security and reliability. Finally, blockchain facilitates transparent data sharing in research collaborations, enabling trustworthy and verifiable exchange of information.

C. Combined Watermarking-Blockchain Solutions

Several existing solutions combine watermarking and blockchain technologies to ensure data traceability. Liu et al. [25] propose a data traceability model for edge nodes, consisting of a blockchain network and an internal network. In their model, data traceability within the internal network is guaranteed using digital watermarking. The blockchain network is composed of master nodes, which are elected by edge nodes based on their computing power. When data moves outside its originating area, it is traced through the blockchain network.

Peng et al. [26] implement a secure digital copyright management system based on a public blockchain. In their system, data providers and data users engage in direct trade, with copyright and transaction data logged in the blockchain. To provide data traceability, the embedded watermarks contain transaction information. Zheng et al. [27] present a copyright protection scheme for videos that combines blockchain and robust reversible watermarking. Their method extracts video keyframes using the image correlation coefficient method. The robust watermarking scheme is based on the Contourlet transform, QR decomposition, and SIFT algorithm, while the reversible watermarking scheme relies on the Arnold Transformation (Cat Map). After identity authentication, the signature of the robust watermark is logged in the blockchain.

However, these methods do not address the impact of watermarking on the performance of AI models. Additionally, they do not provide details on how the blockchain could be used for watermark extraction.

III. Proposed method

A. System architecture & Threat model

As presented in the Introduction, this work is part of the European project PAROMA-MED, which aims to develop a framework to train AI models by data scientists on medical images belonging to different institutions using Federated Learning. When a Data Scientist (DS) requests the Central Server (CS) to start a federated learning session, the Data Providers (DP) send their datasets to their edge nodes to train the AI model on them. In this step, the DPs first watermark their datasets using their ID and the ID of the DS as a message in order to allow ownership verification and traceability. They create a transaction block that contains the hash of the original and the watermarked dataset. Once the Edge Node receives the watermarked dataset and its corresponding transaction info, it verifies if the hashes match to check the integrity of the dataset. Then, it adds the block with its signature to the blockchain. Once the blockchain is updated, the Edge Node shares it with the other edge nodes to provide access to the updated blockchain.

Regarding the threat model considered in the federated environment, here are our security hypotheses. First, the edge nodes at the edge of the networks of data providers are considered secure. Then, we assume that there are secure communication channels in the platforms of data providers, ensured by authentication and encryption processes. Each data provider has a pair of encryption keys: a private key used for signatures and a public key used for encryption. Finally, while external users are considered honest but curious, *i.e.,* they respect the instructions for processing provided by the data provider, if they gain access to the data, they may redistribute the data or leak it unintentionally or maliciously.

B. Reversible Watermarking using Histogram Shifting of Prediction Errors

This paper presents a reversible watermarking scheme based on the histogram shifting of prediction errors. The algorithm employs a cross-shaped prediction kernel and includes overflow management to ensure perfect reconstruction. In this section, we present the main three steps of our scheme, which are the generation, embedding, and extraction of the watermark.

1) Watermark generation: The message M to embed can be of variable size and include different metadata, depending on the use case. For example, to ensure ownership, the sender ID can be embedded. To ensure traceability, the receiver ID (data scientist ID) can be added. The message of type string is converted into a watermark coded in 256 bits using the HMAC-SHA256 as a MAC algorithm parametrized with SHA256 as a hash function [28] and a secret key S_k, as presented in the following equation:

$$W = \text{HMAC-SHA256}(M, S_k) \tag{1}$$

In our work, the watermark is encoded in 256 bits and the S_k is coded into 128 bits, which is different for each image in order to avoid having the same watermarked output if an image is watermarked twice. This choice reinforces the security of our scheme.

2) Watermark embedding: Let us first define the notation used throughout this paper. Let I denote the original image of size $M \times N$ encoded in 8 bits, I_w the watermarked image, W the binary watermark sequence, K the prediction kernel, t_{hi} the histogram shifting threshold, s the stride parameter, $P(i,j)$ the predicted value at position (i,j), and $e(i,j)$ the prediction error at position (i,j). For prediction, we employ a cross-shaped kernel K defined as: [0 1/4 0; 0 1/4 0; 0 1/4 0] which computes the mean of the four nearest neighbors.

Algorithm 1 Watermark Embedding

Require: Original image I, watermark bits W, kernel K, stride s, threshold t_{hi}
Ensure: Watermarked image I_w

$I_w := I$ ▷ Create copy of original image
overflow_list := $\emptyset$ ▷ Initialize overflow list
idx_wat := 0 ▷ Watermark bit index
for $y := 0$ **to** $M - k_h$ **step** s **do** ▷ k_h is kernel height
  for $x := 0$ **to** $N - k_w$ **step** s **do** ▷ k_w is kernel width
    region := $I_w[y : y + k_h, x : x + k_w]$
    $P(y + \frac{k_h}{2}, x + \frac{k_w}{2}) := \sum(\text{region} \odot K)$ ▷ Predict center pixel
    center := $I_w[y + \frac{k_h}{2}, x + \frac{k_w}{2}]$
    $e :=$ center $- P(y + \frac{k_h}{2}, x + \frac{k_w}{2})$
    if $e \geq 0$ **then**
      if center $= 255$ **or** center $= 254$ **then**
        overflow_list.append(i) ▷ $i \in \{0, 1\}$
        $I_w[y + \frac{k_h}{2}, x + \frac{k_w}{2}] := I[y + \frac{k_h}{2}, x + \frac{k_w}{2}] + i$
        idx_wat := idx_wat +1
        continue
      end if
      if $e > t_{hi}$ **then**
        $e_w := e + t_{hi} + 1$
      else
        $e_w := 2e + W[\text{idx\_wat} \bmod \text{len}(W)]$
      end if
      $I_w[y + \frac{k_h}{2}, x + \frac{k_w}{2}] := P(y + \frac{k_h}{2}, x + \frac{k_w}{2}) + e_w$
    end if
    idx_wat := idx_wat +1
  end for
end for
for bits **in** overflow_list **do**
  Recalculate prediction and embed overflow bits starting from the last block of the image
end for
  return I_w

The watermark embedding procedure is presented in Alg. 1. The algorithm processes the image in blocks using a sliding window approach with stride s. For each position (y, x), we consider a 3×3 neighborhood centered at $(y + \frac{k_h}{2}, x + \frac{k_w}{2})$, where k_h and k_w are the kernel height and width respectively.

The predicted value $P(y+\frac{k_h}{2}, x+\frac{k_w}{2})$ is computed by applying the kernel K to the neighborhood (step 6 in Alg. 1):

$$P(y + \frac{k_h}{2}, x + \frac{k_w}{2}) = \sum(\text{region} \odot K)$$

where $\odot$ denotes element-wise multiplication. The prediction error e is then computed as the difference between the center pixel value and its prediction:

$$e = I[y + \frac{k_h}{2}, x + \frac{k_w}{2}] - P(y + \frac{k_h}{2}, x + \frac{k_w}{2})$$

To handle overflow cases and ensure perfect reconstruction, special consideration is given to pixels with values near the maximum intensity ($2^8 - 1 = 255$ for 8-bit images). For each block:

- If the center pixel value equals 255, we store the value 0 in an overflow vector and leave the pixel unmodified
- If the center pixel value equals 254, we increment the pixel value by 1 and store the value 1 in the overflow vector

For non-overflow cases, the embedding function modifies the prediction error according to the following rule:

$$e_w = \begin{cases} e + t_{hi} + 1 & \text{if } e > t_{hi} \\ 2e + w & \text{if } 0 \leq e \leq t_{hi} \end{cases}$$

where e_w is the modified error, t_{hi} is the histogram shifting threshold, and w is the watermark bit to be embedded. This function creates a gap in the histogram to accommodate the watermark bits while maintaining reversibility. The watermarked pixel value is then updated as:

$$I_w[y + \frac{k_h}{2}, x + \frac{k_w}{2}] = P(y + \frac{k_h}{2}, x + \frac{k_w}{2}) + e_w$$

After processing all blocks in forward order, the algorithm performs a second pass starting from the end of the image to embed the overflow vector. This backward embedding ensures that the overflow information is preserved and can be used during extraction to perfectly reconstruct the original image.

3) Watermark extraction: The extraction process begins by analyzing the watermarked image using the same kernel and stride parameters as during embedding. For each selected position based on the secret key, the algorithm calculates the prediction value $P(i, j)$ using the cross-shaped kernel and determines the prediction error $e_w(i, j)$ as the difference between the actual pixel value and the predicted value. When a positive error is detected, the algorithm checks if the pixel has maximum intensity (255). These positions are tracked as overflow positions for later processing. For each valid position, the original prediction error and watermark bit are extracted using the extraction function which is defined as :

$$(e, w) = \begin{cases} (e_w - t_{hi} - 1, \text{null}) & \text{if } e_w > 2t_{hi} + 1 \\ (\lfloor \frac{e_w - (e_w \bmod 2)}{2} \rfloor, e_w \bmod 2) & \text{otherwise} \end{cases} \tag{2}$$

where e_w is the modified error from the watermarked image, e is the recovered original prediction error, and w is the extracted watermark bit.

Since our watermark consists of 256 bits while the embedding capacity is significantly larger, we utilize this additional capacity to enhance robustness. During embedding, the 256-bit watermark is repeatedly embedded until the available capacity is filled. During extraction, we apply a majority voting scheme on the multiple copies of each embedded bit. For example, if a particular bit position contains more 1s than 0s across all extracted copies, the final recovered bit is determined to be 1. This redundancy-based approach significantly increases the robustness of the watermark against various types of noise and attacks, as errors in individual bit positions can be corrected through the voting process.

After completing the main extraction process, the algorithm handles the overflow cases using the tracked positions to restore the original pixel values. Alg. 2 ensures perfect reconstruction of the original image through two key mechanisms: the bijective mapping in the histogram shifting operation, which guarantees reversibility of the embedding process, and the accurate handling of overflow cases.

4) Combination of watermarking and Blockchain: Another originality of our work is the fact that the watermark embedding and extraction operations are both logged in the blockchain. This allows a life-cycle traceability of the images. The intervention of the blockchain with the watermarking operates as follows: After each watermarking or extraction operation, a new block is appended to the blockchain.

Each block contains the following information: block number, timestamp, hash of the previous block, and the encrypted version of the message, and the secret key (encrypted using the PAROMA-MED public key). Moreover, it contains the hash of the original image and the hash of the watermarked image. Additionally, each block includes its own hash to ensure integrity, along with a digital signature of the block hash, created using the private key of PAROMA-MED. This digital signature verifies the authenticity of the block.

Regarding the Watermark extraction using the blockchain. To access a specific block in the blockchain, a hash of the suspect image is computed, and searched in the blockchain. Thus, to detect whether an image was logged in the blockchain, the following Alg. 3 is implemented. When an image X is suspected to belong to PAROMA-MED, its hash is computed and searched in each block of the blockchain B. If the computed hash exists in the current block, return the message stored in the block. Else, extract the watermark W from the image using the block's secret key S_k^i. If the extracted watermark W equals the watermark in the block, return M_i the message in the block. If there is no match, continue to the next block. At the end, if no matching block is found, then no watermark has been detected, the image X is not part of the blockchain B.

IV. Experimental Results and Analysis

A. Dataset and Experimental Setup

The experimental validation of our proposed scheme was conducted using a database of chest X-ray images for COVID-19 positive cases along with Normal and Viral Pneumonia

Algorithm 2 Watermark Extraction

Require: Watermarked image I_w, kernel K, stride s, threshold t_{hi}, secret key K_s
Ensure: Extracted watermark W_{final}, Recovered image I_r

$I_r := I_w$ ▷ Initialize recovered image
$W_{ext} := \emptyset$ ▷ Initialize extracted watermark
overflow_positions $:= \emptyset$
idx_wat $:= 0$
$W_{256} :=$ zero matrix of size 256×2 ▷ For majority voting
for $y := 0$ **to** $M - k_h$ **step** s **do**
  for $x := 0$ **to** $N - k_w$ **step** s **do**
    region $:= I_r[y : y + k_h, x : x + k_w]$
    $P(y + \frac{k_h}{2}, x + \frac{k_w}{2}) := \sum(\text{region} \odot K)$
    center $:= I_r[y + \frac{k_h}{2}, x + \frac{k_w}{2}]$
    $e_w :=$ center $- P(y + \frac{k_h}{2}, x + \frac{k_w}{2})$
    if $e_w < 0$ **then**
      idx_key $:=$ idx_wat $+1$
      continue
    end if
    if center $= 255$ **then**
      overflow_positions.append$(y + \frac{k_h}{2}, x + \frac{k_w}{2})$
      idx_wat $:=$ idx_wat $+1$
      continue
    end if
    $e,$ bit $:=$ extraction_value(e_w, t_{hi}) ▷ Eq: (2)
    if bit $\in \{0, 1\}$ **then**
      W_{ext}.append(bit)
      W_{256}[idx_wat mod 256][0] $:= W_{256}$[idx_wat mod 256][0] $+$ bit
      W_{256}[idx_wat mod 256][1] $:= W_{256}$[idx_wat mod 256][1] $+ 1$
    end if
    $I_r[y + \frac{k_h}{2}, x + \frac{k_w}{2}] := P(y + \frac{k_h}{2}, x + \frac{k_w}{2}) + e$
    idx_wat $:=$ idx_wat $+1$
  end for
end for
if overflow_positions **not** empty **then**
  overflow_bits $:= W_{ext}[-$len(overflow_positions) :]
  for i, $position$ **in** enumerate(overflow_positions) **do**
    $I_r[position] := I_r[position] -$ overflow_bits$[i]$
  end for
end if
$W_{final} :=$ [1 if $W_{256}[i][0]/W_{256}[i][1] > 0.5$ else 0 for i in range(256)] **return** I_r, W_{final}

Algorithm 3 Watermark detection in the blockchain

Require: Input image X, blockchain B with m blocks
Ensure: Detection result

$H :=$ SHA256(X)
for B_i **in** blockchain B, $i \in \{1, \ldots, m\}$ **do**
  if $H \in B_i$ **then**
    return M_i
  end if
  $W :=$ Extract(X, S_k^i)
  if $W = W^i$ **then**
    return M_i
  end if
  if no matching block is found **then**
    return "No watermark detected"
  end if
end for

images, comprising 544893 lung X-ray images [1] of size (299, 299) encoded into 8 bits. We utilized the VGG16[2] architecture as our base model for classification and performed comparative analysis between watermarked and non-watermarked datasets. The experiments were designed to evaluate both the impact of watermarking on model performance and the robustness of the watermarking scheme.

1) Implementation Details: We utilized a pre-trained VGG16 network fine-tuned for COVID-19 classification (Covid19, Normal and Viral Pneumonia), with training parameters set to 30 epochs, batch size of 32, Adam optimizer with learning rate 0.001, and the dataset split into 80% for training, 10% for testing, and 10% for validation.

B. Model Performance Analysis

We trained two versions of the VGG16 model: i) Model trained on original (non-watermarked) dataset and ii) Model trained on watermarked dataset. Both models were evaluated on both watermarked and non-watermarked test sets to ensure comprehensive performance assessment.

TABLE I
PERFORMANCE COMPARISON OF MODELS ON DIFFERENT TEST SETS

Model / Test Set	Accuracy	Precision	Recall	F1-Score
Original / Original	0.9536	0.9548	0.9536	0.9539
Original / Watermarked	0.9545	0.9556	0.9545	0.9548
Watermarked / Original	**0.9571**	**0.9579**	**0.9571**	**0.9574**
Watermarked / Watermarked	0.9500	0.9509	0.9500	0.9503

In Figs. 1 and 2, we present the performances of the model during training on both the watermarked and original datasets. The dashed and solid lines represent the watermarked and original datasets, respectively. We can observe a slight improvement in the training performance on the watermarked dataset, which could be explained by the fact that watermarking adds noise to the data, potentially helping the model generalize better. Table I presents the results of the accuracy, recall, precision, and F1 score for four different settings. We can notice that the results are very similar for all four settings, with a slight improvement of the watermarked model when tested on the original test set. This can be attributed to the watermark acting as a regularizing noise factor, potentially improving the model's generalization capabilities.These results demonstrate that our watermarking scheme does not significantly impact

[1]https://www.kaggle.com/datasets/dvtiendat/covid-classification-dataset
[2]https://www.kaggle.com/code/vunhuduc/vgg16-final

Fig. 1. Training and Validation Loss vs. Epochs

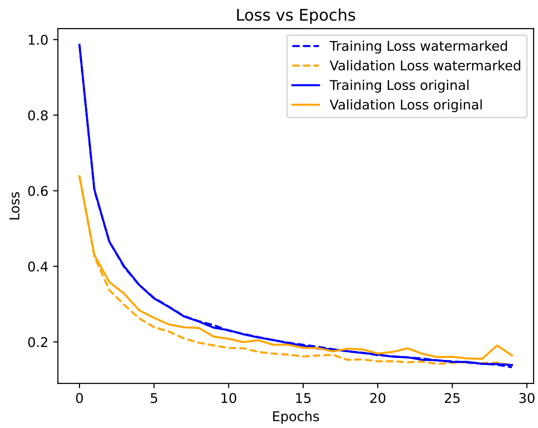
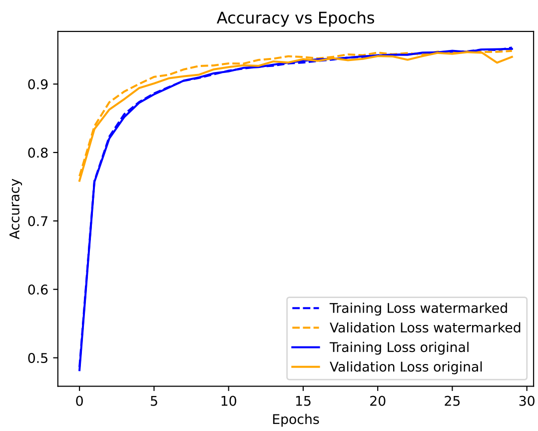


Fig. 2. Training and Validation Accuracy vs. Epochs



the performance of the AI model and may even lead to slight improvements in some cases.

C. Watermark Robustness Analysis

We evaluated the robustness of our watermarking scheme against various image processing attacks. The following attacks were implemented: histogram shifting [29] with a shift value of -10, contrast adjustment with an increase factor of 1.5 and a decrease factor of 0.7 [30], gamma correction with $\gamma = 2.2$ [31], standard and adaptive histogram equalization [32], and Gaussian noise with a mean of 0 and a variance of 1. We evaluate the robustness of the watermark by using the bit error rate (BER), which represents the proportion of bits that are incorrectly decoded during the watermark extraction process. A BER equal to zero indicates a perfect match between the embedded and extracted watermarks, while a BER of 0.5 means that there is no correlation between the watermarks. In our scheme, the watermark is the HMAC of the message. Due to the confusion and diffusion properties of the HMAC, it is very difficult to find two different messages whose HMAC values have a BER smaller than 0.3. Therefore, when the BER is smaller than a threshold of 0.3, it means that the watermarks match with a high probability. This property increases the robustness and trustworthiness of our method. The results in Table II demonstrate near-zero BER across most attacks, indicating strong robustness of our watermarking scheme. Perfect reconstruction was achieved in cases where no modifications were applied to the image, as evidenced by a BER of 0 for the "No Attack (Original)" case.

TABLE II
BIT ERROR RATE (BER) UNDER DIFFERENT ATTACKS

Attack Type	**BER**
Histogram Shift	0
Contrast Increase	0
Contrast Decrease	0
Gamma Correction	0
Histogram Equalization	0
Adaptive Histogram Equalization	0
Gaussian Noise	0.20703125
No Attack (Original)	0

D. Embedding Capacity Analysis

The embedding capacity of our proposed approach is determined by the parameters of the histogram shifting operation, specifically the kernel size and stride. For an input image of dimensions $(M \times N)$, the total capacity in bits can be calculated as output_height $\times$ output_width where:

$$\text{output\_height} = \left\lfloor \frac{M - k_h}{s} \right\rfloor + 1 \tag{3}$$

$$\text{output\_width} = \left\lfloor \frac{N - k_w}{s} \right\rfloor + 1 \tag{4}$$

where $k_h \times k_w$ represents the kernel dimensions and s is the stride. In our implementation, we utilize a 3×3 kernel with a stride of 3 on images of size 299×299, yielding a total capacity of $99 \times 99 = 9,801$ bits. Given that our watermark is encoded into 256 bits, we exploit this high capacity by replicating the watermark sequence to fill the available embedding space. During extraction, we employ a majority voting scheme on the replicated watermark bits to recover the original 256-bit watermark. This redundancy-based strategy significantly enhances the robustness of our watermarking scheme provided by the majority voting mechanism, while taking advantage of the scheme's high embedding capacity. The complete implementation of our approach is available on GitHub[3], enabling reproducibility of all experimental results.

V. CONCLUSION AND FUTURE WORK

This paper presents a novel framework for ensuring data traceability throughout the life cycle of medical images in federated learning environments. Our framework integrates robust reversible watermarking with blockchain, employing histogram shifting on prediction errors with an overflow management procedure to guarantee reversibility. The framework's security is achieved through blockchain validation, digital

[3]https://github.com/Bellafqira/HS_Wat_Blockchain

signatures, and cryptography, thereby ensuring data integrity, authenticity, traceability, and ownership verification. Testing on a medical dataset using a VGG16 model has demonstrated that our framework not only preserves model performance but shows slight improvement in accuracy. The method exhibits robust protection against various attacks, leveraging the high capacity of histogram shifting and the error-correction capability of majority voting on redundant watermark bits.

The framework has two main limitations: the high computational cost of blockchain searching operations and vulnerability to geometric transformations such as resizing and rotation. Future work will focus on developing geometric-invariant watermarking techniques. We also plan to implement our solution on established blockchain platforms such as Ethereum, utilizing smart contracts to automate watermark verification and access control, which would enhance the system's scalability and interoperability in real-world healthcare applications.

REFERENCES

[1] A. Lakhan, H. Hamouda, K. H. Abdulkareem, S. Alyahya, and M. A. Mohammed, "Digital healthcare framework for patients with disabilities based on deep federated learning schemes," *Computers in Biology and Medicine*, vol. 169, p. 107845, 2024.

[2] H. Guan, P.-T. Yap, A. Bozoki, and M. Liu, "Federated learning for medical image analysis: A survey," *Pattern Recognition*, p. 110424, 2024.

[3] M. El Azzouzi, R. Bellafqira, G. Coatrieux, M. Cuggia, and G. Bouzille, "Secure extraction of personal information from ehr by federated machine learning," in *Digital Health and Informatics Innovations for Sustainable Health Care Systems*. IOS Press, 2024, pp. 611–615.

[4] K. A. Koutsopoulos, C. Thümmler, A. A. Castillo, A. Abend, S. Covaci, B. Ertl, G. Ledakis, S. Lorin, V. Thouvenot, S. Haddad *et al.*, "Architecture and design choices for federated learning in modern digital healthcare systems," in *Federated Learning for Digital Healthcare Systems*. Elsevier, 2024, pp. 37–58.

[5] Y. Shang, M. Xue, L. Y. Zhang, Y. Zhang, and W. Liu, "Tracking the leaker: An encodable watermarking method for dataset intellectual property protection," in *Proceedings of the ACM Turing Award Celebration Conference-China 2024*, 2024, pp. 114–119.

[6] R. Bellafqira and G. Coatrieux, "Diction: Dynamic robust white box watermarking scheme," *arXiv preprint arXiv:2210.15745*, 2022.

[7] D. Niyitegeka, G. Coatrieux, R. Bellafqira, E. Genin, and J. Franco-Contreras, "Dynamic watermarking-based integrity protection of homomorphically encrypted databases–application to outsourced genetic data," in *Digital Forensics and Watermarking: 17th International Workshop, IWDW 2018, Jeju Island, Korea, October 22-24, 2018, Proceedings 17*. Springer, 2019, pp. 151–166.

[8] R. Bellafqira, M. Al-Ghadi, E. Genin, and G. Coatrieux, "Robust and imperceptible watermarking scheme for gwas data traceability," in *International Workshop on Digital Watermarking*. Springer, 2022, pp. 147–161.

[9] O. Faraj, D. Megías, and J. Garcia-Alfaro, "Zircon: Zero-watermarking-based approach for data integrity and secure provenance in iot networks," *Journal of Information Security and Applications*, vol. 85, p. 103840, 2024.

[10] J. Anderson, S. Lo, and T. Walter, "Authentication security of combinatorial watermarking for gnss signal authentication," *NAVIGATION: Journal of the Institute of Navigation*, vol. 71, no. 3, 2024.

[11] D. Bouslimi, R. Bellafqira, and G. Coatrieux, "Data hiding in homomorphically encrypted medical images for verifying their reliability in both encrypted and spatial domains," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 2496–2499.

[12] M. G. Lizama, J. Huesa, and B. M. Claudio, "Use of blockchain technology for the exchange and secure transmission of medical images in the cloud: Systematic review with bibliometric analysis," *ASEAN Journal of Science and Engineering*, vol. 4, no. 1, pp. 71–92, 2024.

[13] S. Kumar, B. K. Singh, and M. Yadav, "A recent survey on multimedia and database watermarking," *Multimedia Tools and Applications*, vol. 79, no. 27, pp. 20 149–20 197, 2020.

[14] J. Guo, Y. Li, R. Chen, Y. Wu, C. Liu, and H. Huang, "Zeromark: Towards dataset ownership verification without disclosing watermark," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[15] Y. Zhang, D. Ye, C. Xie, L. Tang, X. Liao, Z. Liu, C. Chen, and J. Deng, "Dual defense: Adversarial, traceable, and invisible robust watermarking against face swapping," *IEEE Transactions on Information Forensics and Security*, 2024.

[16] Z. Ma, G. Jia, B. Qi, and B. Zhou, "Safe-sd: Safe and traceable stable diffusion with text prompt trigger for invisible generative watermarking," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 7113–7122.

[17] X. Wang, X. Li, and Q. Pei, "Independent embedding domain based two-stage robust reversible watermarking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 8, pp. 2406–2417, 2019.

[18] L. Novamizanti, A. B. Suksmono, D. Danudirdjo, and G. Budiman, "Robust reversible watermarking using stationary wavelet transform and multibit spread spectrum in medical images." *International Journal of Intelligent Engineering & Systems*, vol. 15, no. 3, 2022.

[19] F. Peng, X. Li, and B. Yang, "Improved pvo-based reversible data hiding," *Digital Signal Processing*, vol. 25, pp. 255–265, 2014.

[20] L. Novamizanti, A. B. Suksmono, D. Danudirdjo, and G. Budiman, "Robust reversible image watermarking based on independent embedding domain and pixel value ordering," in *2023 IEEE 8th International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*. IEEE, 2023, pp. 1–6.

[21] P. Patel and H. Patel, "Achieving a secure cloud storage mechanism using blockchain technology," *International Journal of Computer Theory and Engineering*, vol. 15, no. 3, pp. 130–142, 2023.

[22] S. S. Kushwaha, S. Joshi, D. Singh, M. Kaur, and H.-N. Lee, "Systematic review of security vulnerabilities in ethereum blockchain smart contract," *IEEE Access*, vol. 10, pp. 6605–6621, 2022.

[23] G. A. Pierro and R. Tonelli, "Can solana be the solution to the blockchain scalability problem?" in *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2022, pp. 1219–1226.

[24] E. Zaghloul, T. Li, M. W. Mutka, and J. Ren, "Bitcoin and blockchain: Security and privacy," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10 288–10 313, 2020.

[25] C. Liu, X. Chen, J. Li, S. Yang, and Y. Sun, "A novel data traceability model based on blockchain and digital watermarking in edge computing," in *Journal of Physics: Conference Series*, vol. 1682, no. 1. IOP Publishing, 2020, p. 012041.

[26] W. Peng, L. Yi, L. Fang, D. XinHua, and C. Ping, "Secure and traceable copyright management system based on blockchain," in *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*. IEEE, 2019, pp. 1243–1247.

[27] J. Zheng, S. Teng, P. Li, W. Ou, D. Zhou, and J. Ye, "A novel video copyright protection scheme based on blockchain and double watermarking," *Security and communication networks*, vol. 2021, no. 1, p. 6493306, 2021.

[28] D. Rachmawati, J. Tarigan, and A. Ginting, "A comparative study of message digest 5 (md5) and sha256 algorithm," in *Journal of Physics: Conference Series*, vol. 978. IOP Publishing, 2018, p. 012116.

[29] K. A. El Drandaly, W. Khedr, I. S. Mohamed, and A. M. Mostafa, "Digital watermarking scheme for securing textual database using histogram shifting model." *Computers, Materials & Continua*, vol. 71, no. 3, 2022.

[30] X.-y. Wang, J. Tian, J.-l. Tian, P.-p. Niu, and H.-y. Yang, "Statistical image watermarking using local rhfms magnitudes and beta exponential distribution," *Journal of Visual Communication and Image Representation*, vol. 77, p. 103123, 2021.

[31] V. Sisaudia and V. P. Vishwakarma, "A secure gray-scale image watermarking technique in fractional dct domain using zig-zag scrambling," *Journal of Information Security and Applications*, vol. 69, p. 103296, 2022.

[32] X. Zhou, Y. Ma, Q. Zhang, M. A. Mohammed, and R. Damaševičius, "A reversible watermarking system for medical color images: balancing capacity, imperceptibility, and robustness," *Electronics*, vol. 10, no. 9, p. 1024, 2021.