Zhao *et al.*
Health Information Science and Systems (2024) 12:49
https://doi.org/10.1007/s13755-024-00306-6

Health Information Science and Systems

RESEARCH

Explainable federated learning scheme for secure healthcare data sharing

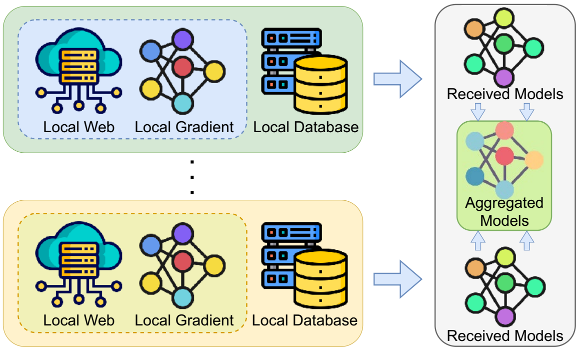
Liutao Zhao[1*], Haoran Xie[2], Lin Zhong[1] and Yujue Wang[3]

Abstract

Artificial intelligence has immense potential for applications in smart healthcare. Nowadays, a large amount of medical data collected by wearable or implantable devices has been accumulated in Body Area Networks. Unlocking the value of this data can better explore the applications of artificial intelligence in the smart healthcare field. To utilize these dispersed data, this paper proposes an innovative Federated Learning scheme, focusing on the challenges of explainability and security in smart healthcare. In the proposed scheme, the federated modeling process and explainability analysis are independent of each other. By introducing post-hoc explanation techniques to analyze the global model, the scheme avoids the performance degradation caused by pursuing explainability while understanding the mechanism of the model. In terms of security, firstly, a fair and efficient client private gradient evaluation method is introduced for explainable evaluation of gradient contributions, quantifying client contributions in federated learning and filtering the impact of low-quality data. Secondly, to address the privacy issues of medical health data collected by wireless Body Area Networks, a multi-server model is proposed to solve the secure aggregation problem in federated learning. Furthermore, by employing homomorphic secret sharing and homomorphic hashing techniques, a non-interactive, verifiable secure aggregation protocol is proposed, ensuring that client data privacy is protected and the correctness of the aggregation results is maintained even in the presence of up to *t* colluding malicious servers. Experimental results demonstrate that the proposed scheme's explainability is consistent with that of centralized training scenarios and shows competitive performance in terms of security and efficiency.

Keywords: Federated learning, Healthcare, Explainability, Security

Graphical abstract



*Correspondence: Zhaolt@bcc.ac.cn
[1] Beijing Academy of Science and Technology, Beijing Computing Center Company Ltd., Beijing, China
Full list of author information is available at the end of the article

Introduction

Smart healthcare [1] refers to the use of modern information technology, including artificial intelligence, to achieve intelligent, personalized, and remote medical services, thereby improving the efficiency and quality of healthcare. In recent years, an increasing number of people have been using lightweight sensors to collect their own data, forming a special data network called Body Area Network (BAN). BAN accumulates a vast amount of medical data [2], however, the special nature of medical data and privacy protection issues pose challenges to analyzing and mining the information in BAN data [3]. Federated Learning (FL), as a machine learning paradigm that emphasizes privacy protection, has been introduced to BANs, enabling multiple data holders to share data for modeling and analysis [4]. By leveraging FL technology, entities such as users and medical institutions can collaboratively explore the potential value of BAN medical data while protecting data privacy, providing strong support for the development of smart healthcare.

Explainability and security are crucial in applying FL to smart healthcare. Explainability requires the model to be clear in function and easy to understand in design and details. In fact, all stakeholders in the smart healthcare field are highly interested in understanding the mechanisms of artificial intelligence. Explainability can enhance the credibility of clinical decisions, help identify and correct biases, build patient trust, and support medical education and training. Currently, two main approaches are used to improve the explainability of artificial intelligence: transparent design models and post-hoc explanation techniques [5]. Transparent design models consider explainability during the model construction phase, making the model inherently transparent and understandable. These models usually have simple structures, making them easy to analyze and explain. However, there is often a trade-off between explainability and model performance, meaning highly explainable models may not perform optimally. Post-hoc explanation techniques mainly include local explainability, feature relevance, simplified explanations, example-based explanations, textual explanations, and visualizations [5], which explain the decision-making process of a model after training through external methods. They are suitable for complex black-box models (such as deep neural networks) and provide insights into model behavior.

Security in federated learning involves multiple aspects. On one hand, the gradients uploaded by clients in federated learning contain a lot of private information, and tampering with aggregated gradients can severely impact the accuracy and security of the model. Therefore, more protection and defense measures are needed for the security and integrity of gradients in federated learning. On the other hand, in traditional single aggregation node scenarios, the failure of a single server node can disrupt the aggregation process. Additionally, during federated training, some participants may intentionally or unintentionally provide low-quality data, affecting the model's performance and reliability, potentially leading to biased, inaccurate predictions, or even rendering the model unusable.

Currently, few federated learning schemes focused on smart healthcare simultaneously address both explainability and security. This is because there is a certain conflict between explainability and security. Techniques such as adding noise and encryption are often used to protect the federated learning process, but these techniques can hinder transparent information flow among federated learning participants, complicating model explanation. On the other hand, some schemes that enhance the explainability or security of federated learning do so at the expense of model performance.

To address the aforementioned issues, this paper proposes a more secure and explainable federated learning scheme tailored for smart healthcare. Our contributions are summarized as follows:

(1) Explainable federated learning model: This paper focuses on the explainability of federated learning by introducing post-hoc explanation techniques in a plug-in manner for global model explainability analysis. This approach avoids affecting the federated learning process and accurately identifies the most important predictive variables in the federated model.
(2) Explainable client contribution: Unlike traditional federated learning methods that accept all private gradients or use threshold controls for aggregation, our scheme analyzes the actual impact of each local gradient on the global gradient to assess its contribution to the global model. By providing explainable evaluations of gradient contributions, our proposed scheme can distinguish clients with low-quality data and intercept their gradient uploads.
(3) Highly robust multi-server system: This paper designs a multi-server verifiable federated learning system that effectively prevents single points of failure associated with single-server scenarios, ensuring high robustness in the aggregation process. This system can complete aggregation tasks without requiring all servers to be online simultaneously, making the overall system more stable and reliable.
(4) Verifiable secure aggregation protocol: Utilizing Shamir's additive homomorphic secret sharing scheme, this paper proposes a verifiable secure aggregation protocol. This protocol, through the

threshold nature of secret sharing, ensures client data privacy even if up to t malicious servers collude.

Compared with the preliminary version of this paper [6], this version further verified the applicability and robustness of the proposed scheme in multiple scenarios. This extended version does not rely on the computationally complex Chameleon hash function, enabling the solution to be completed using regular hash algorithms as anticipated. The solution provided explainability in two dimensions, namely, the contribution value of private gradient values themselves in the global aggregated gradient, and the contribution of each indicator in the gradient to all indicators. This made the model's explainability in the BAN healthcare scenario diverse, helping doctors to learn the main indicator contributions that can be used for transfer learning.

Related works

With the rapid development of artificial intelligence technology, various applications of AI in the medical field have become increasingly common. Examples include using deep learning techniques to recognize handwriting in medical cases or reports, using image segmentation techniques to identify abnormal areas in radiology reports, and predicting the likelihood of future diseases based on comprehensive diagnostic reports and personal information.

(1) Explainable AI: In recent years, the application of explainable AI in smart healthcare has received widespread attention. Many cutting-edge studies aim to improve the transparency and understandability of models, thereby enhancing their credibility and practicality in clinical settings. Che et al. [7] applied model distillation-based explainable methods to the explainability study of medical diagnostic models. They proposed using gradient boosting trees for knowledge distillation to learn explainable models, which not only achieved excellent performance in predicting ventilator-free days for patients with acute lung injury but also provided good explainability for clinicians. Rajpurkar et al. [8] developed a deep learning-based pneumonia detection system (CheXNet) using a large-scale patient chest X-ray dataset. The detection performance of this system even surpassed that of radiologists. By applying the explainable method CAM to explain the decision basis of the detection system and visualize the corresponding explanation results, this system provides clinicians with substantial auxiliary information for analyzing patient medical imaging data and quickly locating patient lesions. Yang et al. [9] built an RNN model with an attention mechanism based on ICU treatment records data to analyze the relationship between medical conditions and ICU mortality, which had often been poorly studied in previous medical practice. Their results indicate that utilizing explainable techniques helps discover potential influencing factors or interactions related to certain outcomes in healthcare, making it possible to learn new diagnostic knowledge from automated medical diagnostic models. Arvaniti et al. [10] showed that, given a well-annotated dataset, a CNN model can successfully achieve automatic Gleason grading of prostate cancer tissue microarrays. Additionally, using explaination methods to provide the grading basis of the automatic grading system can achieve pathologist-level grading results, thereby supporting the simplification of the relatively cumbersome grading tasks.

(2) Federated learning: With the development of federated learning, meeting various demands in medical scenarios using federated learning technology has also become more common. Khan et al. [11] conducted a comparative analysis using CNN, AlexNet, ResNet50, and VGG16 models and employed FL to predict pneumonia. The VGG16 model achieved the highest accuracy of 91% in pneumonia prediction. Lee et al. [12] proposed a thyroid prediction model based on ultrasound images using FL and models like ResNet 50 and VGG19. The training set contained 8,457 images, and the validation set included 1691 internal and 100 external images. Results showed that the accuracy of the centralized model was slightly higher than that of the FL model, but FL provided better data privacy protection. The authors suggested that model performance could be further improved through data augmentation.

To better apply federated learning to the field of smart healthcare, some studies focus on enhancing the explainability or security of federated learning.

Raza et al. [13] designed a novel end-to-end framework for ECG-based healthcare using explainable artificial intelligence and deep convolutional neural networks in a federated environment, addressing challenges such as data availability and privacy issues. The proposed framework effectively classifies various arrhythmias. Abid et al. [14] applied the concept of explainability to artificial intelligence and federated machine learning algorithms to enhance the efficiency and security of healthcare systems. They proposed an efficient electronic healthcare framework and detailed model, implementing standardized data-sharing protocols, developing a collaborative framework for federated learning, and prioritizing the integration of explainable AI technologies to improve decision transparency. Komalasari et al. [15] enhanced the security, performance, and privacy of healthcare systems by proposing a robust framework for secure, privacy-preserving federated learning using explainable AI in smart healthcare systems, ensuring their resilience and effectiveness in real-world scenarios.

In terms of enhancing security, techniques such as differential privacy and homomorphic encryption are widely applied to improve the security of federated learning. Wang et al. [16] developed a new differentially private stochastic gradient descent algorithm to address non-convex empirical risk minimization problems, which involve minimizing a non-convex loss function over the training set. This algorithm reduces gradient complexity while maintaining strong privacy guarantees and provides utility guarantees comparable to existing methods. Zhang et al. [17] studied the application of differential privacy in network-distributed machine learning and developed two differentially private protection methods: dual-variable perturbation and primal variable perturbation, for the regularized empirical risk minimization problem. Bonawitz et al. [18] developed a secure federated learning framework based on traditional federated learning algorithms using secret sharing algorithms. This framework enables participants to verify the correctness of aggregation results, ensuring that the central server cannot return incorrect global gradient values and guaranteeing the secure update of participant models. Zhang et al. [19] designed a batch encryption framework by employing new local gradient encoding techniques. On this basis, they designed a new floating-point to long integer conversion algorithm to achieve efficiency improvements while maintaining functionality. Madi et al. [20] addressed the verifiability of federated learning aggregation algorithms by proposing a secure, privacy-preserving, and verifiable framework using homomorphic encryption and verifiable computation. Xie et al. [21] designed a verifiable federated learning aggregation scheme without bilinear operations to reduce computational overhead. This scheme employs homomorphic hash algorithms, access control technologies, and a three-party key agreement protocol to ensure the security and privacy of private gradients and global gradients.

Explainable AI enhances healthcare professionals' trust in AI diagnostic results by providing transparency in model decision-making. Federated learning improves the model's generalization ability on medical data by implementing distributed learning while protecting data privacy. In addition, the application of security and privacy protection technologies, such as differential privacy and homomorphic encryption, provides additional security guarantees for FL, ensuring the effectiveness and reliability of intelligent medical systems in the real world. Many smart healthcare federated learning schemes struggle to balance explainability and security due to their conflicting nature. While noise addition and encryption are crucial for security, they can obscure the model's workings and hinder explanation. Likewise, efforts to improve a model's transparency or security often come at the expense of its performance, creating a dilemma between clarity and efficacy.

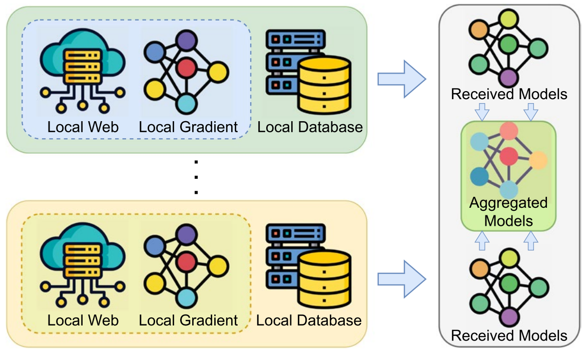


Fig. 1 System model

System model and requirements

In the context of body area networks, the data collected by terminal collection devices belongs to one or more institutions. Each institution, using its own data, cannot effectively construct the target model. Therefore, to utilize the data collected by more devices across multiple institutions and to build high-quality models in medical scenarios, cooperation between multiple institutions is typically required (see Fig. 1).

In the system, m institutions, denoted as S, and n edge computing clients, denoted as C, are involved. To ensure the explainability analysis of the gradient contributions uploaded by clients, prevent clients from conducting gradient attacks on the aggregation process, and ensure that institutions cannot steal or tamper with the federated learning aggregation results through collusion, it is necessary to design a verifiable federated learning aggregation system. This system should be able to achieve fair and efficient client contribution evaluation within the context of body area networks. The system should support analyzing the explainability of the global model using post-hoc explanation techniques. The two types of entities in the system have the following functions:

Client: Participate in the collaborative training of deep learning models through cooperative computation. They use their private datasets, selecting a portion of data for training and testing in each iteration, and upload encrypted local gradients. Through secure multiparty computation, clients collectively obtain aggregation results without exchanging valid information. Clients receive encrypted global gradients returned by the server to update the local model and repeat this process until an accurate neural network model is jointly trained.

Institution: Possess strong computational and data processing capabilities. They aggregate the data sent by clients, assisting in the federated learning training process. However, they also have the potential to steal private information contained within the gradients, necessitating defenses against such possibilities.

System architecture

Definition 1 Consider the scenario of federated learning with n clients C_i and m servers S_j, the scheme $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{SSGen}, \mathsf{Agg}, \mathsf{ConAna}, \mathsf{Ver}, \mathsf{Exp})$ is defined as follows:

- Public parameter generation $\mathsf{Setup}(1^\lambda) \to pp$: Given the security parameter 1^λ, the public parameter pp is generated by all servers S_j through negotiation,
- Key generation $\mathsf{KeyGen}(1^\lambda) \to (\mathsf{pk}_j, \mathsf{sk}_j)$: Given the security parameter 1^λ, server S_j generates its own public-private key pair $(\mathsf{pk}_j, \mathsf{sk}_j)$ and publicly discloses the public key pk_j,
- Secret sharing generation $\mathsf{SSGen}(pp, \{\mathsf{pk}_j\}_{j\in[m]}, x_i, t) \to (\{ct_{i,j}\}_{j\in[m]}, ch_i)$: Given the secret sharing algorithm threshold t, public parameter pp, and server public key list $\{\mathsf{pk}_j\}_{j\in[m]}$, combined with its private input x_i as input, the client outputs the ciphertext of the secret share accepted by the specified server $\{ct_{i,j}\}_{j\in[m]}$ and additional verification information ch_i,
- Aggregation $\mathsf{Agg}(\mathsf{sk}_j, \{ct_{i,j}\}_{i\in[n]}) \to (\hat{y}_j, \hat{r}_j)$: Server S_j based on its private key sk_j and the ciphertext sent to itself $\{ct_{i,j}\}_{i\in[n]}$ as input, outputs the aggregated share $\hat{y}_j$ and additional verification auxiliary value $\hat{r}_j$,
- Contribution analysis $\mathsf{ConAna}(\{[\![r_i]\!]_j\}_{i\in[n]}^{j\in[m]}, t) \to \{\mathcal{C}_i\}_{i\in[n]}$: Given the shares $\{[\![r_i]\!]_j\}_{i\in[n]}^{j\in[m]}$ and threshold t, outputs client contribution $\{\mathcal{C}_i\}_{i\in[n]}$,
- Verification $\mathsf{Ver}(pp, t, \{\hat{y}_j\}_{j\in T}, \{\hat{r}_j\}_{j\in T}, \{ch_i\}_{i\in[n]}) \to (\{y, \perp\}, y^*)$: Given public parameter pp, secret sharing threshold t, $\{\hat{y}_j\}_{j\in T}$, $\{\hat{r}_j\}_{j\in T}$ in server subset T, and collected verification information from client $\{ch_i\}_{i\in[n]}$, outputs a correct unweighted aggregation result $y = \sum_{i=1}^{n} x_i$ or $\perp$, and weighted aggregation result y^*,
- Explainability analysis $\mathsf{Exp}(\{D_i\}_{i\in[n]}, w^*) \to (sc_i*)$: Given the datasets of each client $\{D_i\}_{i\in[n]}$ and the global model obtained from federated modeling w^*, outputs the local explainability analysis result (sc_i*).

In order to implement the functionality of explainability technology in the verifiable federated learning scenario in the proposed scheme, this section designs a gradient contribution analysis technique based on gradient integration that will be used in the subsequent text. The explainability analysis in this paper is performed after the completion of the federated modeling. When performing the explainability analysis step, different methods can be used to calculate (sc_i*) to evaluate the impact of variables on prediction results. This paper takes the example of calculating (sc_i*) using integrated gradients, where integrated gradients are a technique used to explain the decisions of deep learning models. It is a method of explainable artificial intelligence that aims at providing transparency and understandability of model decisions.

Definition 2 (Gradient Contribution Calculation) Integrated gradients measure the contribution of each input feature to the final prediction by calculating the gradient of the input features with respect to the model output and integrating along the path from a baseline input (usually a zero input or some average input) to the actual input. The specific steps are as follows:

- Select baseline input: Determine a baseline input, which is usually a zero vector, average input, or other representative input. The baseline input should be a reasonable input for the model, but should not significantly affect the output.
- Linear interpolation path: Generate a path between the baseline input and the actual input. Specifically, multiple intermediate points can be generated through linear interpolation, with these points gradually transitioning from the baseline input to the actual input.
- Gradient computation: For each interpolated point on the path, calculate the gradient of the input features at that point with respect to the model output.
- Integration: Integrate these gradient values along the path to obtain the total contribution of each feature to the model output. The integrated gradient can be represented by the following formula:

$$\mathbf{IntGrad}_i(x) = (x_i - x_i{}') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha$$

where x_i represents the actual input, x_i' represents the baseline input, F represents the model, α represents the coefficient of interpolation and $\frac{\partial F}{\partial x_i}$ represents the gradient of the output of the model with respect to the i-th input feature.

Definition 3 (Correctness) The federated learning system is considered correct if, for any set of client inputs $\{x_i\}_{i=1}^{n}$, the aggregation process Agg yields an output y that accurately reflects the collective contribution of all clients, i.e., $y = \sum_{i=1}^{n} x_i$. This requires that:

- Each client's input x_i is correctly encrypted and shared using the secret sharing scheme SSGen without loss of information.
- The aggregation of shares by servers is performed correctly, using the private key sk_j for decryption and the secret sharing algorithm $\mathcal{SS}.\mathsf{Eval}$ for reconstructing the original inputs.
- The verification process Ver confirms that the aggregated result y matches the expected outcome, ensuring the integrity and accuracy of the federated learning model's output.

System requirements

Definition 4 (Explainability) The server can obtain the contribution value of the user gradients to the global gradient by analyzing and calculating the user gradients. By obtaining the contribution values, it can effectively balance the contributions of all users and reduce the malicious impact of users on the global model.

Definition 5 (Security) Let $\mathcal{A}$ be a probabilistic polynomial-time adversary who can control at most t servers. Without loss of generality, let the controlled servers be $\{S_j\}_{j\in[t]}$. The security experiment $\mathsf{Exp}^{\mathsf{sec}}(\mathcal{A})$ is defined as follows:

1. For each $j \in [t]$, the challenger $\mathcal{C}$ executes $\mathsf{Setup}(1^{\lambda})$ and sends the public parameters pp to the adversary $\mathcal{A}$.
2. The challenger $\mathcal{C}$ acts as an honest server $\{S_j\}_{j\in[t+1,m]}$ to execute $\mathsf{KeyGen}(1^{\lambda})$, generating the corresponding keys $(\mathsf{pk}_j, \mathsf{sk}_j)$, and publicly releasing the public key pk_j. It receives the public keys $\{\mathsf{pk}_j\}_{j\in[m]}$ of the servers controlled by the adversary $\mathcal{A}$.
3. The challenger $\mathcal{C}$ randomly selects $x_i \xleftarrow{\$} \mathbb{F}$ as the input for each client C_i.
4. The challenger $\mathcal{C}$ acts as an honest client C_i for each honest client C_i, where $i \in [n]$, computing $\mathsf{SSGen}(pp, \{\mathsf{pk}_j\}_{j\in[m]}, x_i, t) \rightarrow (\{ct_{i,j}\}_{j\in[m]}, ch_i)$.
5. The challenger $\mathcal{C}$ acts as an honest server $\{S_j\}_{j\in[t+1,m]}$ to interact with the adversary $\mathcal{A}$, and the challenger $\mathcal{C}$ outputs an aggregated result y^*.
6. If $y^* = \sum_{i=1}^{n} x_i$, the experiment outputs 1; otherwise, it outputs 0.

If $\Pr[\mathsf{Exp}^{\mathsf{sec}}(\mathcal{A}) = 1] \leqslant \mathrm{negl}(\lambda)$, the protocol is considered secure.

Definition 6 (Verifiability) Let $\mathcal{A}$ be a probabilistic polynomial-time adversary who can control at most $k\,(\leqslant m)$ servers. Without loss of generality, let the controlled servers be $\{S_j\}_{j\in[k]}$. The verifiability experiment $\mathsf{Exp}^{\mathsf{ver}}(\mathcal{A})$ is considered as follows:

1. The challenger $\mathcal{C}$ acts as an honest server $\{S_j\}_{j\in[k+1,m]}$ to execute $\mathsf{KeyGen}(1^{\lambda})$, generating the corresponding keys $(\mathsf{pk}_j, \mathsf{sk}_j)$, and publicly releasing the public key pk_j. It receives the public keys $\{\mathsf{pk}_j\}_{j\in[k]}$ of the servers controlled by the adversary $\mathcal{A}$.
2. The challenger $\mathcal{C}$ randomly selects $x_i \xleftarrow{\$} \mathbb{F}$ as the input for each client C_i.
3. The challenger $\mathcal{C}$ acts as an honest client C_i for each honest client $C_i, i \in [n]$, computing $\mathsf{SSGen}(pp, \{\mathsf{pk}_j\}_{j\in[m]}, x_i, t) \rightarrow (\{ct_{i,j}\}_{j\in[m]}, ch_i)$.
4. The adversary $\mathcal{A}$ provides a set $N^* \subset [n]$ of clients participating in the aggregation result.
5. For the honest servers $\{S_j\}_{j\in[k+1,m]}$, the challenger $\mathcal{C}$ returns the aggregated shares and additional verification messages $(\hat{y}_j, \hat{r}_j) \leftarrow \mathsf{Agg}(\mathsf{sk}_j, \{ct_{i,j}\}_{i\in[n]})$ to the adversary $\mathcal{A}$.
6. The adversary $\mathcal{A}$ outputs the computed results $\{\hat{y}_j, \hat{r}_j\}_{j\in[k]}$.
7. The adversary $\mathcal{A}$ provides a set $M^* \subset [m]$ of servers available for verification, with $|M^*| \geqslant t+1$.
8. Running the verification algorithm $\mathsf{Ver}(pp, t, \{\hat{y}_j\}_{j\in T}, \{\hat{r}_j\}_{j\in T}, \{ch_i\}_{i\in[n]}) \rightarrow \{y, \perp\}$, if $y^* \neq y = \sum_{i=1}^{n} x_i$, the experiment outputs 1; otherwise, it outputs 0.

If $\Pr[\mathsf{Exp}^{\mathsf{ver}](\mathcal{A})=1} \leqslant \mathrm{negl}(\lambda)$, the protocol is considered verifiable.

Concrete scheme and analysis

Concrete scheme

Consider the scenario of federated learning with n clients C and m servers S.The i-th client is represented by C_i and the j-th server is represented by S_j. The federated learning process is defined as follows:

1. Public parameter generation algorithm $\mathsf{Setup}(1^{\lambda}) \rightarrow pp$: Each server randomly select $\alpha_j \overset{\$}{\leftarrow} \mathbb{Z}_p$, and calculate $h_j = g_j^{\alpha}$, then follow these steps to calculate:

   - Server S_j randomly selects $r_j \overset{\$}{\leftarrow} \mathbb{Z}_p$, and calculates $a_j = g^{r_j}$,
   - Based on the hash function $\mathcal{H}(\cdot)$, calculate $e = \mathcal{H}(g, h_j, a_j)$,
   - Calculate $z_j = r_j + e_j \cdot \alpha$, output proof (a_j, z_j),
   - After receiving proofs from other servers, server S_j calculates $e_j = \mathcal{H}(g, h_j, a_j)$ according to the above hash function $\mathcal{H}(\cdot)$,
   - Verify whether the equation $g^{z_j} = a_j \cdot h_j^{e_j}$ holds. If it holds, accept the proof and output 1, otherwise reject the proof and output 0,
   - If the verification passes, publishes $(p, \mathbb{G}, g, h = \prod_{j=1}^{m} h_j)$ as the public parameter pp of homomorphic hashing.

2. Key generation algorithm $\mathsf{KeyGen}(1^{\lambda}) \rightarrow (\mathsf{pk}_j, \mathsf{sk}_j)$: Each server S_j runs the key generation algorithm to generate its own public-private key pair $(\mathsf{pk}_j, \mathsf{sk}_j) \leftarrow \mathcal{PKE}.\mathsf{KGen}(1^{\lambda})$, and publishes the public key pk_j,
3. Secret sharing generation algorithm $\mathsf{SSGen}(pp, \{\mathsf{pk}_j\}_{j\in[m]}, x_i, t) \rightarrow (\{ct_{i,j}\}_{j\in[m]}, ch_i)$: The client C_i takes the public parameters pp of the hash, its own private input x_i, the threshold t, and the server's public key set $\{\mathsf{pk}_j\}_{j\in[m]}$ as input and follows the following steps to calculate:

   - Client C_i uses the a priori method for the first few rounds, and then uses the adaptive adjustment algorithm to obtain a sensitivity of $\Delta A = W$, randomly selects noise $\tilde{x}_i$ from the geometric distribution Geom $(\exp(-\varepsilon/W))$, and obtains the secret input value of $x_i \leftarrow x_i + \tilde{x}$,
   - Use the (t, m)-Shamir additive homomorphic secret sharing algorithm to generate the secret sharing share $\{[\![x_i]\!]_j\}_{j\in[m]} \leftarrow \mathcal{SS}.\mathsf{Share}(x_i, t, \{S_j\}_{j\in[m]})$ of the private input x_i for the aggregation server $\{S_j\}_{j\in[m]}$,
   - To calculate the hash value, first randomly select r_i and calculate $ch_i \leftarrow \mathcal{HASH}.\mathsf{Hash}(pp, x_i, r_i)$,
   - Similarly, use the (t, m) -Shamir additive homomorphic secret sharing algorithm to generate the secret share $\{[\![r_i]\!]_j\}_{j\in[m]} \leftarrow \mathcal{SS}.\mathsf{Share}(r_i, t, \{S_j\}_{j\in[m]})$ of the random number r_i for the aggregation server $\{S_j\}_{j\in[m]}$,
   - For $j \in [m]$, use the public key pk_j of the server S_j to generate the corresponding ciphertext $ct_{i,j} \leftarrow \mathcal{PKE}.\mathsf{Enc}(\mathsf{pk}_j, i\|[\![x_i]\!]_j\|[\![r_i]\!]_j)$,
   - The hash value ch_i and the ciphertext $\{ct_{i,j}\}_{j\in[m]}$ are taken as output.

4. Aggregation algorithm $\mathsf{Agg}(\mathsf{sk}_j, \{ct_{i,j}\}_{i\in[n]}) \rightarrow (\hat{y}_j, \hat{r}_j)$: Use the server S_j's own private key sk_j and collect the ciphertext $\{ct_{i,j}\}_{i\in[n]}$ generated by the client $\{C_i\}_{i\in[n]}$, and calculate according to the following steps:

   - Decrypt the ciphertext $\{ct_{i,j}\}_{i\in[n]}$ using its own private key sk_j to obtain $(i\|[\![x_i]\!]_j\|[\![r_i]\!]_j) \leftarrow \mathcal{PKE}.\mathsf{Dec}(\mathsf{sk}_j, \{ct_{i,j}\}_{i\in[n]})$, where $i \in [n]$,
   - Use the secret sharing algorithm $\mathcal{SS}.\mathsf{Eval}$ to calculate the shared aggregate value $\hat{y}_j \leftarrow \mathcal{SS}.\mathsf{Eval}(\{[\![x_i]\!]_j\}_{i\in[n]})$ and aggregate random numbers $\hat{r}_j \leftarrow \mathcal{SS}.\mathsf{Eval}(\{[\![r_i]\!]_j\}_{i\in[n]})$,
   - output $(\hat{y}_j, \hat{r}_j)$.

5. Contribution analysis algorithm $\mathsf{ConAna}(\{[\![r_i]\!]_j\}_{i\in[n],j\in[m]}, t) \rightarrow \{\mathcal{C}_i\}_{i\in[n]}$: Given the share $\{[\![r_i]\!]_j\}_{i\in[n],i\in[m]}$ and the threshold t, the secret sharing algorithm $\mathcal{SS}.\mathsf{Eval}$ is used to calculate the shared aggregate value $x_i \leftarrow \mathcal{SS}.\mathsf{Eval}(\{[\![x_i]\!]_j\}_{j\in[m]})$, and executes the following steps:

   - In order to find the optimal global gradient y^* that can be obtained from all user-provided private gradients, it is necessary to minimize the total distance between all private gradients and the estimated global gradient.

$$\begin{aligned} \min_{y^*, \mathcal{C}} D(y^*, \mathcal{C}) &= \sum_{i\in[n]} g(p_i) \cdot d(y^*, x_i) \\ s.t. \sum_{i\in[n]} p_i &= 1 \end{aligned} \quad , \tag{1}$$

   where $d(\cdot)$ is the distance function, $g(\cdot)$ is the non-negative coefficient function, p_i is the performance of the local private gradient, which is calculated based on the distance,

   - Select the Euclidean distance $d(y^*, x_i) = ||y^* - x_i||$ as the selected distance function, $g(p_i) = 1/p_i$ as the non-negative coefficient function, and further calculate the contribution ratio of client C_i to this gap distance by considering the aggregation weight and the distance between its local model update and the estimated global model update.

$$\ell_i = \frac{g(p_i) \cdot d(y^*, x_i)}{\sum_{i\in[n]} g(p_i) \cdot d(y^*, x_i)} \tag{2}$$

- Given a set of contribution percentages $\{\ell_i\}_{i\in[n]}$, where $\sum_{i\in[n]} \ell_i = 1$, calculate customer contribution $\{\mathcal{C}_i\}_{i\in[n]}$ by solving the following linear equation:

$$\sum_{i\in[n]} \mathcal{C}_i = 1 \qquad \frac{\ell_i}{\ell_k} = \frac{\mathcal{C}_k}{\mathcal{C}_i}, \forall i, k \in [n] \tag{3}$$

6. Verification algorithm $\mathsf{Ver}(pp, t, \{\hat{y}_j\}_{j\in T}, \{\hat{r}_j\}_{j\in T}, \{ch_i\}_{i\in[n]}) \to (\{y, \perp\}, y^*)$: Given public parameters pp,threshold t, collected hash values $\{ch_i\}_{i\in[n]}$, a series of aggregated shared values $\{\hat{y}_j\}_{j\in T}$ and $\{\hat{r}_j\}_{j\in T}$, where $j \in T \subseteq [m]$ and $|T| \geqslant t+1$, the verification process is performed as follows:

- Use the secret reconstruction algorithm to recover $y \leftarrow \mathcal{SS}.\mathsf{Recon}(t, \{\hat{y}_j\}_{j\in T})$ and $r \leftarrow \mathcal{SS}.\mathsf{Recon}(t, \{\hat{r}_j\}_{j\in T})$,
- Use the homomorphism of the hash algorithm to verify whether the equation $\mathcal{HASH}.\mathsf{Hash}(pp, y, r) = \prod_{i\in[n]} ch_i$ holds. If the above equation holds, output y, otherwise output $\perp$,
- According to the user contribution $\{\mathcal{C}_i\}_{i\in[n]}$, the global gradient value is set to $y^* = \sum_{i\in[n]} \mathcal{C}_i \cdot x_i$.

7. Explainability Analysis $\mathsf{Exp}(\{D_i\}_{i\in[n]}, w^*) \to (sc_i*)$: Given the datasets of each client $\{D_i\}_{i\in[n]}$ and the global model w^* obtained by federated modeling, the explainability analysis follows the following steps:

- Each client uses the integrated gradient to solve the local explainability analysis result (sc_i*),
- With the help of the server, the aggregate explainability analysis result $\frac{\sum_i sc_{i*}|D_i|}{|\cup_i D_i|}$ is calculated,
- Each server broadcasts and cross-validates the calculated aggregate explainability analysis results,
- The server synchronizes the aggregate explainability analysis results to the client.

Explainability analysis

Due to the issue of insignificant gradient contributions in marginal value scenarios when traditional methods directly analyze the gradient contributions, the gradient marginal value is shortened and then analyzed step by step until it is reduced to the minimum value of Baseline. Finally, all gradients are summed, and a coefficient interval $\triangle x_i$ is multiplied in the summation to avoid the occurrence of ∞ values when summing the infinitely divided gradients. The segmented gradient values with a length of Baseline are denoted as $x' = \{x_1', x_2', ..., x_n'\}$, and the linear interpolation number is m. The importance of feature x_i is then given by:

$$\begin{aligned}\phi_i^{IG}(f, \boldsymbol{x}, \boldsymbol{x}') &= \sum_{k=0}^{m} \frac{\partial f(\boldsymbol{x}' + \frac{k}{n}(\boldsymbol{x} - \boldsymbol{x}'))}{\partial x_i} \Delta x_i \\ &= \sum_{k=0}^{m} \frac{\partial f(\boldsymbol{x}' + \frac{k}{n}(\boldsymbol{x} - \boldsymbol{x}'))}{\partial x_i} \frac{1}{n}(x_i - x_i')\end{aligned}$$

Then, taking the limit of the gradient importance as $m \to \infty$, the above equation is transformed into integral form:

$$\begin{aligned}\phi_i^{IG}(f, \boldsymbol{x}, \boldsymbol{x}') &= \int_0^1 \frac{\delta f(\boldsymbol{x}' + \alpha(\boldsymbol{x} - \boldsymbol{x}'))}{\delta x_i} d\alpha (x_i - x_i') \\ &= (x_i - x_i') \int_0^1 \frac{\delta f(\boldsymbol{x}' + \alpha(\boldsymbol{x} - \boldsymbol{x}'))}{\delta x_i} d\alpha\end{aligned}$$

The above operation essentially calculates the total contribution of the gradient curve between x and x', that is, $f(x_i) - f(x_i') = \phi_i^{IG}(f, \boldsymbol{x}, \boldsymbol{x}')$, and this equation holds for each dimension of the feature. Therefore, the integral gradient has completeness, i.e.,

$$\phi_i^{IG}(f, \boldsymbol{x}, \boldsymbol{x}') = \int_0^1 \frac{\delta f(\boldsymbol{x}' + \alpha(\boldsymbol{x} - \boldsymbol{x}'))}{\delta x_i} d\alpha (x_i - x_i')$$

The most important thing is that, given $\theta_i(f, \boldsymbol{x}, \boldsymbol{x}') = \int_0^1 \frac{\delta f(\mathbf{x}' + \alpha(\mathbf{x} - \mathbf{x}'))}{\delta x_i} d\alpha$, we have

$$f(\boldsymbol{x}) - f(\boldsymbol{x}') = \phi^{IG}(f, \boldsymbol{x}, \boldsymbol{x}') = \big\langle(\boldsymbol{x} - \boldsymbol{x}'), \boldsymbol{\theta}(f, \boldsymbol{x}, \boldsymbol{x}')\big\rangle$$

Therefore, the relative importance of any sample with respect to the baseline can be linearly expressed by the difference in characteristics $x - x'$ and the result of the integral variational path $\theta(f, x, x')$, which is equivalent to finding a linear model to explain the prediction of the sample x. Thus, the proposed method of integrating gradients can effectively identify the contribution of the integral.

Correctness analysis

Theorem 1 (Correctness) *Under the assumption of the existence of* (t, m)*-Shamir additive homomorphic secret sharing scheme* ($\mathcal{SS}.\mathsf{Share}, \mathcal{SS}.\mathsf{Eval}, \mathcal{SS}.\mathsf{Recon}$) *and the homomorphic hash functions* ($\mathcal{HASH}.\mathsf{Gen}, \mathcal{HASH}.\mathsf{Hash}, \mathcal{HASH}.\mathsf{HashCheck}$), *the secret sharing generation algorithm* SSGen *correctly generates ciphertext shares* $\{ct_{i,j}\}_{j\in[m]}$ *and corresponding verification values for* $\{x_i\}_{i\in[n]}$, *and outputs the final aggregated result* $y = \sum_{i=1}^{n} x_i$ *using correct aggregation shares.*

Proof The correctness of this protocol relies on the correctness of the additive homomorphic secret sharing algorithm, the correctness of public key encryption algorithms, and the homomorphic property of hash functions. When servers S_j and clients $C_i(x_i)$ faithfully execute the protocol, server S_j decrypts to obtain secret shares $\{[\![x_i]\!]_j\}_{i\in[n]}$ and $\{[\![r_i]\!]_j\}_{i\in[n]}$. According to the correctness of the additive homomorphic algorithm, they can reconstruct the secrets $y=\sum_{i\in[n]}x_i$ and $r=\sum_{i\in[n]}r_i$. On the other hand, based on the homomorphic property of the hash function, $\mathcal{HASH}.\mathsf{Hash}(pp,x,r)=\prod_{i\in[n]}ch_i=\mathcal{HASH}.\mathsf{Hash}(pp,x_i,r_i)$, ensuring the final output $y=\sum_{i\in[n]}x_i$. □

Security analysis

Theorem 2 (Security) *Under the assumption of the existence of the discrete logarithm problem, (t, m)-Shamir additive homomorphic secret sharing scheme* $(\mathcal{SS}.\mathsf{Share},\mathcal{SS}.\mathsf{Eval},\mathcal{SS}.\mathsf{Recon})$ *and the homomorphic hash functions* $(\mathcal{HASH}.\mathsf{Gen},\mathcal{HASH}.\mathsf{Hash},\mathcal{HASH}.\mathsf{HashCheck})$, *the above protocol implementation is secure.*

Proof The security proof of the protocol can be achieved using a proof by contradiction. Assume there exists an adversary $\mathcal{A}$ that can break the experiment $\mathsf{Exp}^{\mathsf{sec}}(\mathcal{A})$ with non-negligible probability. Then, we can show that there exists an adversary $R^{\mathcal{A}}$ that can break the discrete logarithm assumption with the same probability.

Given the public parameters $pp=(p,\mathbb{G},g,h)$, the reduction algorithm R receives the challenge $X^*=g^{x^*}$ from the challenger $\mathcal{C}$. Reduction algorithm R first acts as an honest server interacting with the adversary, executing $\mathsf{KeyGen}(1^\lambda)$ to generate the corresponding keys $\{(\mathsf{pk}_j,\mathsf{sk}_j)\}_{j\in[t+1,m]}$ and revealing the public keys $\{\mathsf{pk}_j\}_{j\in[t+1,m]}$ to the adversary. Simultaneously, it receives the public keys $\{\mathsf{pk}_j\}_{j\in[t]}$ from the adversary. Reduction algorithm R randomly chooses $\{x_i\}_{i\in[2,n]}$ as inputs for clients $\{C_i\}_{i\in[2,n]}$ such that $\sum_{i=2}^n x_i=0$.

For $i\in[2,n]$, R acts as an honest client running $\mathsf{SSGen}(pp,\{\mathsf{pk}_j\}_{j\in[m]},x_i,t)\rightarrow(\{ct_{i,j}\}_{j\in[m]},ch_i)$. For $i=1$, reduction algorithm R chooses a random number $\{[\![x_i]\!]_j\}_{j\in[m]}$ as the secret share and randomly selects r_1 as the hash random number, denoted as $ch_1=X^*\cdot h^{r_1}$. Additionally, it generates secret shares for r_1, $\{[\![r_1]\!]_j\}_{j\in[m]}\leftarrow\mathcal{SS}.\mathsf{Share}(r_1,t,\{S_j\}_{j\in[m]})$. It encrypts $ct_{1,j}\leftarrow\mathcal{PKE}.\mathsf{Enc}(\mathsf{pk}_j,1\|[\![x_1]\!]_j\|[\![r_1]\!]_j)$ and publishes $ct_{1,j}$ and ch_i.

Note that the adversary $\mathcal{A}$ controls at most t servers. Based on the security of secret sharing, adversary $\mathcal{A}$ cannot distinguish between the reduction algorithm R randomly selecting secret shares $\{[\![x_i]\!]_j\}_{j\in[m]}$ and the true secret shares of x^*. Furthermore, due to the collision resistance property of the hash function, the adversary cannot efficiently find colliding results that satisfy the conditions. Combining the security of secret sharing, it can be observed that the adversary $\mathcal{A}$ cannot distinguish between the generated distribution and the true distribution.

Subsequently, reduction algorithm R acts as an honest server $\{S_j\}_{j\in[t+1,m]}$ interacting with adversary $\mathcal{A}$ until $\mathcal{A}$ outputs the aggregated result y^*. It is observed that

$$y^*=\sum_{i=1}^{n}x_i=x^*+\sum_{i=2}^{n}x_i=x^* \tag{4}$$

Therefore, it is evident that the adversary can break the security of the discrete logarithm $X^*=g^{x^*}$, which contradicts the discrete logarithm assumption. □

Theorem 3 (Verifiability) *Let $\mathcal{A}$ be a probabilistic polynomial-time adversary capable of controlling at most k ($\leqslant m$) servers, without loss of generality, let them be $\{S_j\}_{j\in[k]}$. Based on the collision resistance property of the homomorphic hash function, the above protocol is verifiable.*

Proof We prove the verifiability of the protocol by contradiction. Assume there exists an adversary $\mathcal{A}$ that can break the above experiment with non-negligible probability, then we can show that there exists an adversary $R^{\mathcal{A}}$ that can break the collision resistance property of the hash function with the same probability.

The reduction algorithm R receives the public parameters of the hash function $pp=(p,\mathbb{G},g,h)$ given by the challenger $\mathcal{C}$ and passes these parameters to the adversary $\mathcal{A}$. Subsequently, reduction algorithm R acts as an honest server $j\in[k+1,m]$, performs $\mathsf{KeyGen}(1^\lambda)$ to generate the corresponding keys $(\mathsf{pk}_j,\mathsf{sk}_j)$, and reveals the public key pk_j. Simultaneously, it receives the public keys $\{\mathsf{pk}_j\}_{j\in[k]}$ from servers controlled by the adversary. For each client C_i, R randomly selects x_i as input.

R acts as an honest client C_i. For $i\in[n]$, it computes $\mathsf{SSGen}(pp,\{\mathsf{pk}_j\}_{j\in[m]},x_i,t)\rightarrow(\{ct_{i,j}\}_{j\in[m]},ch_i)$. It receives the set N^* provided by $\mathcal{A}$, which is the set of challenged clients. Acting as an honest server $j\in[k+1,m]$, R sends the computation information of S_j to $\mathcal{A}$, $\mathsf{Agg}(\mathsf{sk}_j,\{ct_{i,j}\}_{i\in[n]})\rightarrow(\hat{y}_j,\hat{r}_j)$, and $\mathcal{A}$ outputs its computation result $\{\hat{y}_j\}_{j\in[k]}$, $\{\hat{r}_j\}_{j\in[k]}$. $\mathcal{A}$ provides a series of verifiable server sets $\{S_l\}_{l\in M^*}$ to $\mathcal{C}$. Using $\{S_l\}_{l\in M^*}$ to run the verification $\mathsf{Ver}(pp,t,\{\hat{y}_j\}_{j\in M^*},\{\hat{r}_j\}_{j\in M^*},\{ch_i\}_{i\in N^*})\rightarrow y^*$,

Table 1 Comparison with related schemes

Schemes	NIVA [22]	Ma et al. [23]	Zhang et al. [24]	Our scheme
Resist collusion attacks	✓	×	×	✓
Resist other $\mathcal{S}$ attacks	×	×	×	✓
Resist curious $\mathcal{C}$ attacks	✓	×	✓	✓
Verifiability	✓	✓	✓	✓
Explainability	×	×	×	✓

it is ensured that with non-negligible probability $\perp \neq y^* \neq \sum_{i \in N^*} x_i$, and running the verification algorithm results in r^* satisfying

$$\mathcal{HASH}.\mathsf{Hash}(pp, \sum_{i \in N^*} x_i, \sum_{i \in N^*} r_i) = \prod_{i \in N^*} ch_i = \mathcal{HASH}.\mathsf{Hash}(pp, y^*, r^*) \tag{5}$$

Therefore, $R^{\mathcal{A}}$ can output with non-negligible probability two distinct pre-images of $\prod_{i \in N^*} ch_i$, namely $(\sum_{i \in N^*} x_i, \sum_{i \in N^*} r_i)$ and (y^*, r^*). This contradicts the collision resistance property of the hash function. □

Analysis

Theoretical analysis

This section compares the proposed scheme with the NIVA scheme [22], Ma et al.'s scheme [23], and Zhang et al.'s scheme [24]. The comparison is conducted from multiple aspects, including functionality, computational overhead, and communication overhead. The number of servers in all these four schemes is set to m, while the number of clients is set to n. The functionality comparison results are shown in Table 1, which indicate that, compared with other schemes, the proposed scheme has higher advantages in security. It can resist attacks from servers and users, as well as collusion attacks to some extent. Moreover, the scheme ensures the explainability of private gradients and the verifiability of global gradients. This demonstrates that the proposed scheme has clear advantages over other schemes.

In Tables 2 and 3, the communication performance and computational efficiency of the client algorithm, server aggregation algorithm, and public verification algorithm are theoretically compared and analyzed. For the schemes by Ma et al. [23] and Zhang et al. [24], where there is no *SSGen* situation, the modules with similar functions in the proposed scheme are analyzed as substitutes for the corresponding overhead. Here, we use $\mathscr{F}$ to represent the transformed large integer of the gradient vector and $\mathscr{G}$ to represent the group element.

As shown in the theoretical analysis of the communication overhead at each stage in Table 2, the proposed scheme is more efficient than all other schemes. The main reason for this is that both users and servers do not need to send too much information, only a small amount of core information is needed for verification to meet the scheme's requirements. Regarding computational efficiency, from Table 3, it is clear that the proposed scheme is also superior to other ones. This is largely due to the high optimization of the aggregation algorithm in the proposed scheme, which does not require excessive auxiliary information to help with verification, thus completing the aggregation task efficiently.

Experimental analysis

In experiments, the proposed scheme runs in a scenario with n clients and m servers. As shown in Table 4, the experiments are conducted on a platform running Ubuntu 22.04, with code executed on hardware featuring an Intel(R) Xeon(R) CPU E5-2603 v2 1.80GHz and 64GB RAM. The experimental code is written in Python 3.10. For cryptographic schemes, we use RSA encryption with a key length of 512 bytes for the public key encryption scheme, a (t, m)-Shamir additive homomorphic secret sharing scheme, and a homomorphic hash function based on the elliptic curve SECP256K1. To verify the effectiveness of the scheme in the field of BANs, the scheme needs to be trained using datasets collected by private medical institutions. In the experiments, the classic lightweight heart disease dataset from IEEE [25] is used for model validation. The dataset contains common indicators, which can also be obtained in BANs, making

Table 2 Communication overhead comparison

Schemes	SSGen	Agg	Ver
NIVA [22]	$(m+1)\mathscr{F} + (2m+3)\mathscr{G}$	$(n+1)\mathscr{F} + (2n+4)\mathscr{G}$	$(\mu+1)\mathscr{F} + (n+3)\mu\mathscr{G}$
Ma et al. [23]	$(2m+1)\mathscr{F} + 4m\mathscr{G}$	$(2n+1)\mathscr{F} + (2n+1)\mathscr{G}$	$2\mathscr{F} + (n+1)\mu\mathscr{G}$
Zhang et al. [24]	$(m+1)\mathscr{F} + (2m+2)\mathscr{G}$	$(2n+1)\mathscr{F} + (n+1)\mathscr{G}$	$2\mathscr{F} + (2n+1)\mu\mathscr{G}$
Our scheme	$(2m+1)\mathscr{F} + 3\mathscr{G}$	$(2n+2)\mathscr{F}$	$(2\mu+1)\mathscr{F} + (n+2)\mathscr{G}$

Table 3 Computational overheads comparison

Scheme	SSGen	Agg	Ver
NIVA [22]	$\mathcal{O}(m^2)$	$\mathcal{O}(n^2+m)$	$\mathcal{O}(m^2+nm)$
Ma et al. [23]	$\mathcal{O}(2m^2+m)$	$\mathcal{O}(n^2)$	$\mathcal{O}(2m^2+n)$
Zhang et al. [24]	$\mathcal{O}(2m^2)$	$\mathcal{O}(2n^2)$	$\mathcal{O}(m^2)$
Our scheme	$\mathcal{O}(m^2)$	$\mathcal{O}(n)$	$\mathcal{O}(m^2+n)$

Table 4 Experimental parameters

Parameters	Value
Run platform	Ubuntu 22.04
Hardware	Intel(R) Xeon(R) CPU E5-2603 v2 1.80GHz and 64GB RAM
Cryptographic scheme	RSA encryption
Key length	512 bytes
Datasets	the classic lightweight heart disease dataset
Learning rate	0.001
Batch size	64
Training epochs	100

this dataset somewhat representative of BAN datasets. The batch size for learning is set to 64, and the training epochs are set to 100. The learning rate is initialized to 0.001 and is dynamically adjusted: if no improvement in validation accuracy is observed within a specified 10 epochs, the learning rate is reduced by a factor of 10%.

First, we compare the proposed method with the centralized method, observing the importance of variables under different methods. The importance of variables reflects the sensitivity of the prediction results to the variables. The integrated gradient is used to evaluate the importance of variables under the centralized method as well. The dataset used in the experiments includes eleven variables such as ST slope, sex, etc., with the average value set as the baseline. The importance of variables is arranged in descending order.

Figure 2 shows the comparison of variable importance using the proposed method and the centralized training method. We observe that the explainability analysis results clearly show the importance of each variable, with the importance of ST slope being significantly higher than other variables, warranting special attention. Additionally, we find that the importance of each variable is essentially consistent between the proposed method and the centralized training method. This validates that under the proposed method, although no raw data is transmitted, the model's explainability is not compromised.

Regarding computational overhead and communication overhead, it is assumed that the size of the server set T participating in the verification algorithm in the proposed scheme is μ. In the experiments, to ensure consistency with the NIVA protocol [22], small integer vectors are packed into larger integer vectors for processing, as done in the NIVA protocol [22]. The length of the converted large integer is set to $B = 36$ bytes.

In terms of computational overhead and communication overhead, as shown in Figs. 3 and 4, it can be observed that under the same experimental conditions such that $B = 36$ bytes and the SECP256k1 elliptic curve group element size of 48 bytes, the proposed scheme exhibits significant advantages compared to the NIVA protocol [22], Ma et al.'s scheme [23], and Zhang et al.'s scheme [24]. The figures show that both the computational and communication overheads increase in an approximately linear relationship with the number

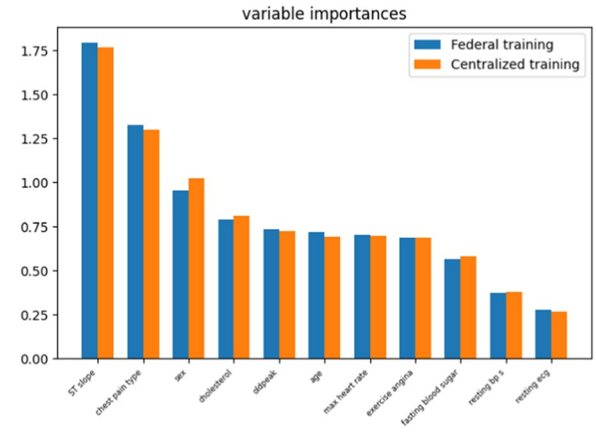


Fig. 2 Explainability analysis results

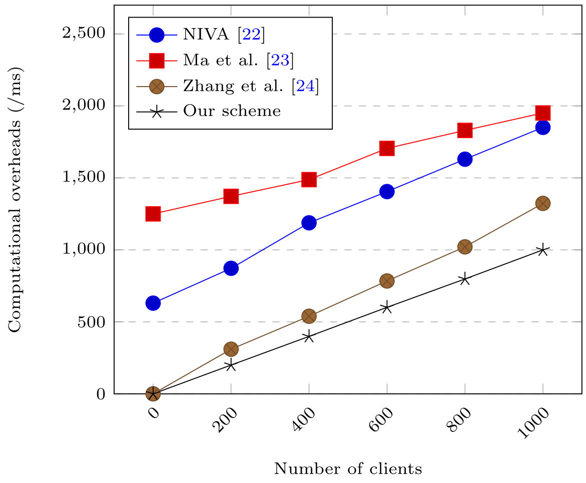


Fig. 3 Comparison of computational overheads

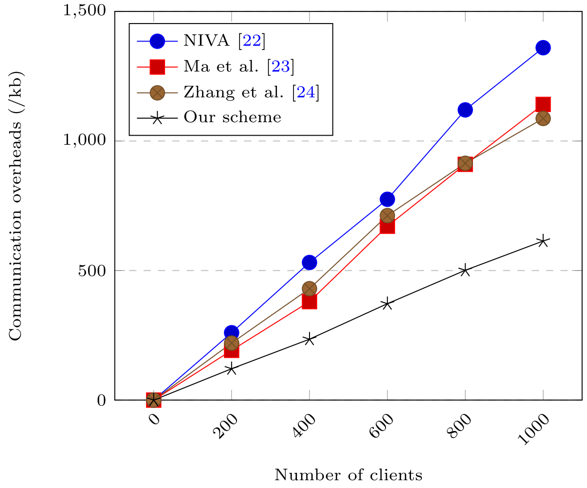


Fig. 4 Comparison of communication overheads

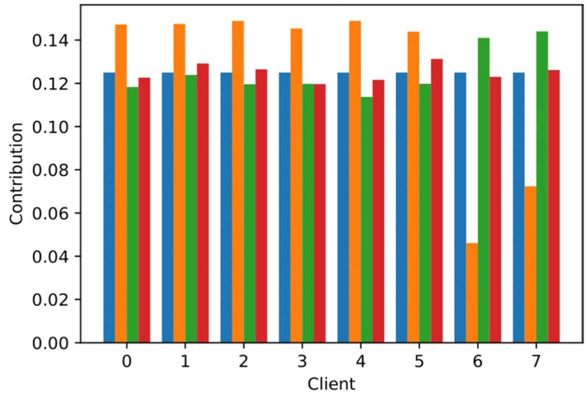


Fig. 5 Uneven label distribution

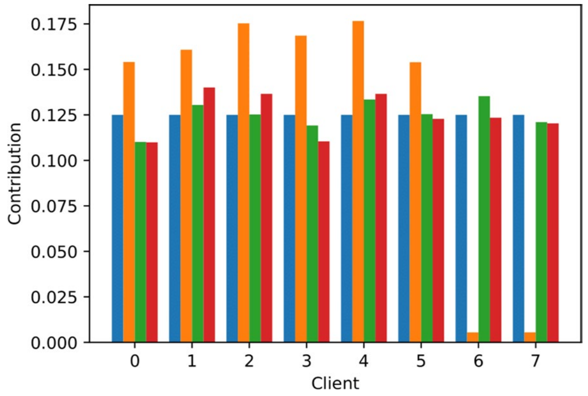


Fig. 6 Gradient malicious attack

of clients, indicating that the experimental results are highly consistent with the theoretical analysis.

To verify the effectiveness of the designed explainability algorithm in identifying anomalous gradients from users, two scenarios were selected to validate the model's performance. In the first scenario, different users hold different quantities of labels under non-independent and identically distributed (non-IID) conditions. There are a total of eight users, with the first six users holding the same number of labels, while the last two users hold the same number of labels but more than the first six users. In the second scenario, two malicious users were selected from a total of eight users in the federated learning experiment. These malicious users have the ability to amplify their private gradients. By analyzing the explainability of the gradients, we test the effectiveness of the proposed scheme. Shapley values, Leave-One-Out (LOO) values, client contribution, and global weighted gradient values are used as comparison values, and the results are presented in Figs. 5 and 6.

From Fig. 5, it can be seen that the aggregation weights are most sensitive to the information content of the last two users, which is greater than that of the first six users. Therefore, the corresponding values are lowered to maintain the optimal direction of the gradients during aggregation. The client contribution is somewhat inferior, as it cannot fully distinguish the gradient changes. From Fig. 6, there is a clear distinction between the attackers and the regular clients, successfully identifying the attackers. By assigning higher values to non-attackers and much lower values to attackers, the scheme effectively mitigates the impact of the attackers. Overall, the experiments demonstrate that the proposed scheme has significant importance in the explainability of gradients, providing high value in practical applications.

The complete training process is simulated alongside scenarios where some users' metrics are missing, to simulate the potential decline in model accuracy due to the loss of part of the user datasets and further verify the model's robustness. The results are shown in Fig. 7.

By comparing the test accuracy results with complete data and the accuracy results with partially missing training metrics on the full category test set, it is verified that federated learning, to some extent, ensures that users can obtain training results with all metrics, even if some training set categories are missing. The proposed scheme not only enhances the accuracy of users within the group using federated learning but also achieves an accuracy level of 84.34% with the heart disease recognition classifier, indicating that the proposed scheme can be deployed in federated learning systems within BANs.

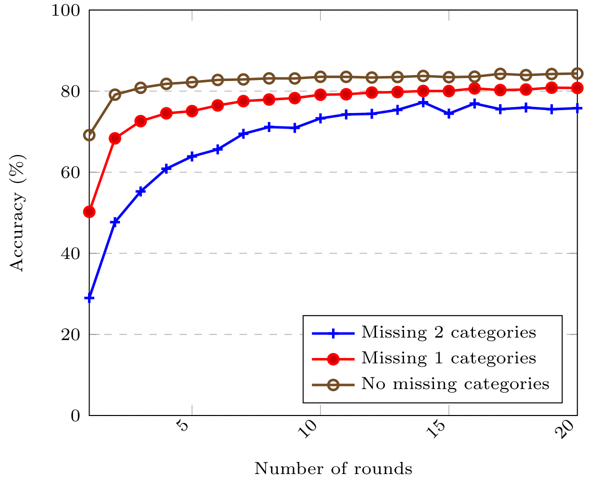


Fig. 7 Comparison under different categories missing conditions

Conclusion

This paper proposed a more secure and explainable federated learning solution tailored for smart healthcare, addressing the explainability and security challenges faced when applying federated learning in the field of smart healthcare. Through post-hoc explaination techniques, our proposed method can analyze the sensitivity of prediction results to input variables, thereby understanding the model's operating mechanism. Experimental validation confirms that the federated environment does not compromise explainability.

In terms of security, this paper addressed common security issues in federated learning under a single-server model. A validated federated learning aggregation system for federated network data within a fair and efficient client contribution assessment system was proposed, which effectively resolves server loss issues and optimizes single-point failure problems. Also, using additive homomorphic secret sharing schemes and homomorphic hash functions, we achieved a verifiable secure aggregation protocol under an explainable gradient model.

Data availability

Not applicable.

Declarations

Conflict of interest

The authors declare that there is no Conflict of interest regarding the publication of this paper.

Author details

[1] Beijing Academy of Science and Technology, Beijing Computing Center Company Ltd., Beijing, China. [2] School of Intelligent Systems Engineering, Sun Yat-sen University, Shenzhen, China. [3] Hangzhou Innovation Institute of Beihang University, Hangzhou, China.

Received: 9 July 2024 Accepted: 17 August 2024
Published online: 13 September 2024

References

1. Muse ED, Barrett PM, Steinhubl SR, Topol EJ. Towards a smart medical home. Lancet. 2017;389(10067):358. https://doi.org/10.1016/S0140-6736(17)30154-X.
2. Fotouhi M, Bayat M, Das AK, Far HAN, Pournaghi SM, Doostari M-A. A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care iot. Comput Netw. 2020;177:107333. https://doi.org/10.1016/j.comnet.2020.107333.
3. Ding Y, Xu H, Zhao M, Liang H, Wang Y. Group authentication and key distribution for sensors in wireless body area network. Int J Distrib Sensor Netw. 2021;17(9):15501477211044338. https://doi.org/10.1177/15501477211044433.
4. Nguyen DC, Pham Q-V, Pathirana PN, Ding M, Seneviratne A, Lin Z, Dobre O, Hwang W-J. Federated learning for smart healthcare: a survey. ACM Computing Surveys. 2022;55(3):1–37. https://doi.org/10.1145/3501296.
5. Hassan Naqvi. An automated system for classification of chronic obstructive pulmonary disease and pneumonia patients using lung sound analysis. Sensors. 2020;6512:22.
6. Zhao L, Xie H, Zhong L, Wang Y Multi-server verifiable aggregation for federated learning in securing industrial iot. In: 2024 IEEE 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD) 2024. IEEE
7. Che Z, Purushotham S, Khemani R, Liu Y Interpretable deep models for icu outcome prediction. In: AMIA Annual Symposium Proceedings, American Medical Informatics Association; 2016. pp. 371–380
8. Rajpurkar P, Irvin J, Zhu K, Yang B, Mehta H, Duan T, Ding D, Bagul A, Langlotz C, Shpanskaya K. Chexnet Radiologist-level pneumonia detection on chest x-rays with deep learning. [Preprint] 2017. Available from https://doi.org/10.48550/arXiv.1711.05225.
9. Yang C, Rangarajan A, Ranka S, Global model interpretation via recursive partitioning. In: 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2018:1563–1570 . https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00256 . IEEE
10. Arvaniti E, Fricker KS, Moret M, Rupp N, Hermanns T, Fankhauser C, Wey N, Wild PJ, Rüschoff JH, Claassen M. Automated gleason grading of prostate cancer tissue microarrays via deep learning. Sci Rep. 2018;8(1):12054. https://doi.org/10.1038/s41598-018-30535-1.
11. Khan SH, Alam MGR, A federated learning approach to pneumonia detection. In: 2021 International Conference on Engineering and Emerging Technologies (ICEET), 2021; 1–6. https://doi.org/10.1109/ICEET53442.2021.9659591 . IEEE
12. Lee H, Chai YJ, Joo H, Lee K, Hwang JY, Kim S-M, Kim K, Nam I-C, Choi JY, Yu HW. Federated learning for thyroid ultrasound image analysis to protect personal information: validation study in a real health care environment. JMIR Med Inform. 2021;9(5):25869. https://doi.org/10.2196/25869.
13. Raza A, Tran KP, Koehl L, Li S. Designing ecg monitoring healthcare system with federated transfer learning and explainable ai. Knowl-Based Syst. 2022;236:107763. https://doi.org/10.1016/j.knosys.2021.107763.
14. Abid R, Rizwan M, Alabdulatif A, Alnajim A, Alamro M, Azrour M. Adaptation of federated explainable artificial intelligence for efficient and secure e-healthcare systems. CMC-Comput Mater Contin. 2024. https://doi.org/10.32604/cmc.2024.046880.
15. Komalasari R Secure and privacy-preserving federated learning with explainable artificial intelligence for smart healthcare systems. In: Federated Learning and Privacy-Preserving in Healthcare AI; 2024, Hershey: IGI Global, pp. 288–313. Chap. 18. https://doi.org/10.4018/979-8-3693-1874-4.ch018
16. Wang L, Jayaraman B, Evans D, Gu Q Efficient privacy-preserving stochastic nonconvex optimization. In: Evans, R.J., Shpitser, I. (eds.) Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence. Proceedings of Machine Learning Research, 2023; PMLR 216; pp. 2203–2213. https://doi.org/10.5555/3625834.3626040

17. Yar M, Dahman AM, Mohammed AW, Vinh HT, Ryan A. Identification of pneumonia disease applying an intelligent computational framework based on deep learning and machine learning techniques. London: Hindawi; 2021.
18. Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, Ramage D, Segal A, Seth K, Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. CCS '17. New York: Association for Computing Machinery; pp. 1175–1191. (2017). https://doi.org/10.1145/3133956.3133982
19. Zhang C, Li S, Xia J, Wang W, Yan F, Liu Y, BatchCrypt: efficient homomorphic encryption for cross-silo federated learning. In: 2020 USENIX Annual Technical Conference (USENIX ATC 20), 2020:493–506. https://doi.org/10.5555/3489146.3489179
20. Madi A, Stan O, Mayoue A, Grivet-Sébert A, Gouy-Pailler C, Sirdey R, A secure federated learning framework using homomorphic encryption and verifiable computing. In: 2021 Reconciling Data Analytics, Automation, Privacy, and Security: A Big Data Challenge (RDAAPS), 2021:1–8 https://doi.org/10.1109/RDAAPS48126.2021.9452005 . IEEE
21. Xie H, Wang Y, Ding Y, Yang C, Zheng H, Qin B. Verifiable federated learning with privacy-preserving data aggregation for consumer electronics. IEEE Trans Consumer Electron. 2024;70(1):2696–707. https://doi.org/10.1109/TCE.2023.3323206.
22. Brunetta C, Tsaloli G, Liang B, Banegas G, Mitrokotsa A, Non-interactive, secure verifiable aggregation for decentralized, privacy-preserving learning. In: Australasian Conference on Information Security and Privacy, New York: Springer; 2021:510–528.
23. Ma X, Zhang F, Chen X, Shen J. Privacy preserving multi-party computation delegation for deep learning in cloud computing. Inform Sci. 2018;459:103–16. https://doi.org/10.1016/j.ins.2018.05.005.
24. Zhang X, Fu A, Wang H, Zhou C, Chen Z, A privacy-preserving and verifiable federated learning scheme. In: ICC 2020-2020 IEEE International Conference on Communications (ICC), 2020:1–6 . https://doi.org/10.1109/ICC40277.2020.9148628 . IEEE
25. Siddhartha M, Heart disease dataset (Comprehensive). https://doi.org/10.21227/dz4t-cm36

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