Multiple Networks Security Strategies for Data Sharing Using MS-FL

Sumaiya Tameem[1], Samreen Sultana[2]

[1]PG Scholar, Department of CSE, Shadan Women's College of Engineering and Technology, Hyderabad, Sumaiyatameem01@gmail.com

[2]Asst Professor, Department of CSE, Shadan Women's College of Engineering and Technology samreencme@gmail.com

ABSTRACT

More and more consumers are using multi-party data for federated machine learning to get the models they want as a result of advancements in data science, artificial intelligence, and data transactions. In order to solve real-world problems, academics have put forth a variety of federated learning frameworks. Nonetheless, contemporary federated learning systems still have three problems that need to be fixed: 1) safeguarding privacy; 2) preventing poisoning; and 3) safeguarding participants' interests. This study suggests MS-FL, a unique federated learning architecture based on several security techniques, as a solution to these problems. Data suppliers don't have to worry about data privacy leaks thanks to the framework's algorithms. It can also protect against malicious nodes' poisoning attacks. Lastly, a blockchain protocol is used to guarantee that the interests of all participants are safeguarded. The theoretical derivation demonstrates this framework's efficacy. The algorithm developed in this paper performs better than previous algorithms, according to experimental results.

1. INTRODUCTION

Legal rules that forbid companies and organizations from revealing sensitive information have been in place in recent years. This leads to the establishment of several data islands and a significant waste of resources because most of the data are closed and isolated. As a result, using data resources responsibly while protecting data privacy has become essential in the current era. Several scholars have proposed the federated learning strategy, which uses data from multiple machine learning partners, to complete machine learning. Fredrikson found that the model could deduce some of the trainers' data information after training, despite the fact that no data privacy was jeopardized throughout the federated learning process. This implies that after training, there is a possibility that the model will reveal private information.

CKKS is a completely homomorphic cryptosystem that is widely used in practice to secure data privacy because it allows (1) addition and multiplication homomorphism of ciphertext and (2) floating-point arithmetic and evaluation of an arbitrary polynomial, which is ideal for privacy-protecting machine learning. To protect the privacy of FL participants, we use CKKS technology in this study. In addition to privacy leaks, the adversary intends to introduce poisoned (biased or misleading) data into a federated learning system during the gradient aggregation phase with the goal of changing the model's parameters to the attacker's advantage. If the aggregation server does not include a defense mechanism and use the average of the aggregate gradients, it has been demonstrated that a single poisoner can take over the entire training process. This project's main focus is on three types of common poisoning attacks: (1) Label-flipping attack, also known as data poisoning. In a data set, each data sample has a category label. On the other hand, malicious nodes can use false labels to train machine learning models. (2) Backdoor attacks, often known as data poisoning. Malicious nodes seek to establish robust associations between the trigger and target label while minimizing their influence on the classification of benign inputs. The trigger is a white block. (3) An arbitrary model assault, often known as model poisoning.

During local model training, malicious nodes could communicate arbitrary wrong gradients to the aggregation server, reducing the final model's accuracy. The increasing value of data has led to a considerable expansion in data trading. In practice, there are situations where a company need specific machine learning models but lacks the required data, and the data owners do not require the corresponding models. For example, even though the schools do not need the model, an educational research organization must train a machine learning model using the academic performance of students from different schools. The major goal of this study is to show how to train a reliable machine learning model using multi-party data while preserving privacy and allowing model requesters and data owners to trade data value. In this study, we create a novel FL framework, MS-FL, to tackle the problem of the aforementioned real-world scenario. It is based on a number of security methods.

OBJECTIVE

The primary objective of this study is to provide and validate MS-FL, a federated learning framework that effectively addresses the problems of privacy protection in multi-party data transactions, poisoning assaults, and participant interest protection. The framework creates a secure and reliable environment for federated learning by utilizing a range of security mechanisms, such as homomorphic encryption and blockchain technology. The study is to show the effectiveness of the framework through theoretical analysis and practical results in order to improve safe federated learning techniques.

PROBLEM STATEMENT

Federated learning allows several companies or devices to work together and build a shared machine learning model without actually sharing personal data. Nevertheless, even when raw data is safeguarded, there are still some serious problems. For example, someone could try to enter false or misleading data into the system to destroy the model. Additionally, if proper safeguards are not in place, private information may still leak during training. It's also difficult to guarantee that everyone is treated equally and that no one cheats. Because of these issues, we need a better federated learning system that can safeguard information, stop hackers, and guarantee that all parties can trust the procedure.

EXISTING SYSTEM

Except for privacy leaks, the adversary intends to input poisoned (biased or misleading) data into a federated learning system during the gradient aggregation phase of the current system in order to change the model's parameters in a way that is advantageous to the attacker. If the aggregation server does not include a defense mechanism and use the average of the aggregate gradients, it has been demonstrated that a single poisoner can take over the entire training process. In a nutshell, homomorphic encryption is a cryptographic method that permits processing of data while it remains encrypted. Unlike ordinary encryption, which necessitates decrypting statistics for every meaningful operation, homomorphic encryption enables computations on encrypted statistics to be performed simultaneously. The results of the same processes performed on the plaintext data are identical to those obtained when similar computations are decoded. In this way, sensitive data can be analyzed, changed, and handled while remaining encrypted, preserving privacy and anonymity. Lower security and data security with analytical efficacy are two of the existing system's drawbacks.

PROPOSED SYSTEM

The major goal of this study is to show how to train a reliable machine-learning model using multi-party data while preserving privacy and allowing model requesters and data owners to trade data value. In this study, we create a novel FL framework, MS-FL, to tackle the problem of the aforementioned real-world scenario. It is based on a number of security methods. An attacker can obtain the data owner's sensitive information, such as training samples and memberships, by identifying shared gradients. To protect each data owner's privacy, we use CKKS technology along with a number of security measures to ensure that their local gradients are kept secret. Furthermore, only the model requestor has access to the finished model following the execution of MS-FL protocols. Among the benefits are the requirement to safeguard private data. High protection and security, multi-party data for trustworthy training.

This paper's primary focus is on how to use multi-party data to train a trustworthy machine learning model while maintaining privacy and enabling the exchange of data value between model requesters and data owners. In order to address the issue of the aforementioned real-world scenario, we develop a unique FL framework, MS-FL, in this study. It is built on various security techniques. MS-FL accomplishes the following benefits:

- **Privacy:** Previous research has demonstrated that by inferring shared gradients, an adversary can retrieve sensitive data belonging to the data owner, like training samples or memberships. We employ several security procedures and CKKS technology to maintain the confidentiality of each data owner's local gradients in order to preserve their privacy. Furthermore, only the model requestor has access to the finished model following the execution of MS-FL protocols.
- **Robustness:** The model training procedure is able to fend off the three poisoning attacks from malevolent nodes mentioned above.
- **Fair Transaction:** The model requestor will receive the model parameters following each training cycle, and they will be required to pay the data owners. This transaction is transparent and atomic thanks to the smart contract employed in MS-FL on the blockchain.
- **Interest protection:** MS-FL ensures that the data owner will not miss out on all potential to profit from the model requestor simply because some of the submitted gradient components differ from the majority, in contrast to certain previous literature.

The rest of paper is organized as follows. Section II is related work. In Section III, we overview the preliminaries of this paper. Section IV introduces the specific steps and corresponding algorithms to complete federated learning in the application scenario of this paper. Section V and section VI demonstrate security analysis of the system and convergence property of proposed aggregation algorithm in this framework. In section VII, we compare the proposed aggregation algorithm with exiting algorithms in some aspects and give corresponding comparison graphs. Finally, section VIII summarizes this paper.

2. RELATED WORK

Federated learning enables many parties to collaborate on training a model using their data without exchanging raw data. Federated learning provides some privacy protection for users. However, model parameters can still expose private information. Furthermore, because they depend on a trustworthy third party to generate and distribute key pairs to linked participants, existing encrypted federated learning systems are unsuitable for federated learning and vulnerable to security risks. To tackle these issues, we propose a privacy-preserving blockchain-based no trusted third party federated learning system (NttpFL). It is not necessary for a trustworthy third party to distribute keys; instead, the conference key agreement is utilized to negotiate keys between the partners and the federated learning task initiator. We design a double-layer encryption technique to safeguard privacy. Partners cannot access any private data other than their information. The decentralized nature of blockchain complements our system nicely. Additionally, blockchain makes the entire process public and traceable, avoiding the problem of a single node failure. Experimental results show that compared to existing encrypted federated learning, the proposed method significantly reduces communication costs and computing complexity while preserving security and performance. Over the years, there has been a significant increase in interest in machine learning and associated studies. Because of its immense potential, machine learning has achieved remarkable success across a wide range of businesses. However, the essential feature of machine learning is that its algorithms need a large amount of data in order to work. Many Internet of Things (IoT) devices actually have significant amounts of data scattered across them. The processing, connectivity, and sensor capabilities of these devices are constantly improving. With this data, the machine learning technique may be applied successfully **[1].**

We suggest a method for creating a homomorphic encryption scheme that is approximation arithmetic. In addition to allowing the approximate addition and multiplication of encrypted messages, it features a new rescaling procedure to regulate the size of plaintext. By truncating the ciphertext into a smaller modulus, this procedure rounds the plaintext. The main goal is to follow significant figures with a sound that makes a point. This noise, which was initially added to the plaintext for security, is believed to be an error component that occurs during approximations and is removed by rescaling the plaintext. Consequently, an approximate plaintext value with a predetermined degree of precision is generated by our decryption mechanism. We also present a new batching technique for a structure based on RLWE. An isometric ring homomorphism, or complex canonical embedding map, converts a plaintext polynomial, which is an element of a cyclotomic ring with characteristic zero, into a message vector of complex integers. Since this change does not make errors larger, we can preserve the plaintext's correctness after encoding. In contrast to all previous studies, which either demand an increasingly large size of modulus or expensive calculations like bit extraction or bootstrapping, our construction's ciphertext modulus bit size rises linearly with the depth of the circuit being evaluated due to the rescaling process. A key feature of our method is that the accuracy loss during evaluation is constrained by the depth of a circuit and is less than one bit when compared to unencrypted approximation arithmetic, such as floating-point operations. We show that our method can be applied not only to the basic approximation circuits but also to the efficient evaluation of transcendental functions such as the logistic function, exponential function, multiplicative inverse, and discrete Fourier transform. **[2]**

Large clients, such as mobile devices or organizations, can collaborate to create a global model without sharing local data thanks to a novel framework called federated learning (FL). However, as there is no direct access to the customers' data, the global model is vulnerable to attacks by malicious clients utilizing their corrupted data. Many strategies have been proposed to mitigate label flipping attacks, however some of them are unsafe, incur high computational overhead, or even compromise privacy. In this study, we introduce Malicious Clients Detection Federated Learning (MCDFL) to protect against the label flipping attack. It can identify fraudulent clients by recovering a distribution across a latent feature space to assess the quality of each client's data. Considering multiple neural network architectures and attack scenarios, we demonstrate the effectiveness of our proposed method on two benchmark datasets: Fashion-MNIST and CIFAR-10. The results show that our approach can reliably identify harmful clients without incurring excessive costs in a range of scenarios where the ratio of hostile clients falls between 5% and 40%. Federated Learning (FL) has been praised as a promising machine learning technique with regard to data privacy. Machine learning is used extensively in various applications, such as fingerprint liveness detection and face recognition. Because of the increased worry about data privacy, the FL paradigm is used for model training. Specifically, in the absence of a centralized data manager that collects and validates datasets, FL allows the data to be kept locally on clients and provides a central server to interact with the clients and train a global model. Such a paradigm is optimized by locally computed updates for each client. **[3]**

Federated learning (FL) allows data owners to train a common global model without sharing personal data. However, it is vulnerable to Byzantine attackers who can use poisoning assaults to destroy model training. Current defense solutions use the extra datasets to create trusted execution environments or server models to stop threats. Additionally, these strategies can only resist a certain amount of malicious users or types of poisoning attacks. To tackle these challenges, we created a novel federated learning method known as discovery-based federated L earning (TDFL), which can prevent multiple poisoning attacks without requiring additional datasets, even when the proportion of Byzantine users is at least 50%.Specifically, the TDFL considers several scenarios with different levels of malice. We have developed a novel resilient truth discovery aggregation approach that can allocate weights based on user contributions to remove malicious model changes in Honest-majority circumstances (Byzantine <50%). To guarantee global model quality in Byzantine-majority conditions (Byzantine ≥50%), we use a maximum clique-based filter. As far as we are aware, this is the first study to use truth seeking to defend against poisoning attempts. Furthermore, it is the first method to provide significant resilience against a range of attacks carried out by a significant portion of attackers without root datasets. Extensive comparison tests are conducted on diverse datasets under five types of conventional poisoning assaults using five state-of-the-art aggregation rules. The experimental results demonstrate that TDFL is workable and achieves an acceptable Byzantine-robustness level. **[4]**

Websites currently function as a mediator in the conventional power data transaction model. However, there are challenging issues with power data trading, such as privacy protection, transaction security, and data reliability. In this study, we propose a novel secure power data trading method (SPDTS). First, the zero-knowledge proof is used to achieve data consistency and availability without revealing the data. SPDTS then makes full use of blockchain technology's immutability and dispersibility to ensure the reliability of data exchanges. To ensure the efficiency of the transaction process, smart contracts are used to handle the processing chores for electricity data. In the interim, a trusted execution environment (TEE) is utilized to guarantee the security of power data. Finally, we propose a differential privacy method to safeguard the privacy information of the power data. Our study indicates that the proposed plan can offer privacy protection, transaction security, and data reliability. We also conduct security analysis and verify the privacy protection feature of the scheme in real-world scenarios. With the advent of cloud computing and big data, the Internet of Things (PIOT) has taken a new turn. Smart terminal devices, such as smart meters, may collect bulk data from the entire power system to allow real-time power grid monitoring. Furthermore, statistical data extracted from power data can be used to correctly diagnose, optimize, and predict the operating status of the power grid, guaranteeing the power system runs securely, dependably, inexpensively, and efficiently. Power data is typically preferred by the trading market. However, in the conventional power data trading approach, sample Web pages are mostly utilized for data exchanges between market participants and grid firms. Market players evaluate and manage the original data by looking at or reporting it over the power trading web **[5].**

A collaborative machine learning architecture known as "federated learning" allows several businesses to train a global model anonymously. Even though it seems promising, privacy and robustness issues occur when an attacker attempts to breach the global model or infer private information from the provided parameters. Although many protocols have been developed to reduce security risks, it is challenging to make federated learning protocols safe from Byzantine attackers while preserving participant privacy. In this research, we propose a differentially private Byzantine-robust federated learning system (DPBFL) with high compute and communication efficiency. The proposed method successfully prevents adversarial attacks by Byzantine parties while guaranteeing differential privacy by utilizing a special aggregation technique in the shuffling model. Theoretically, the learning error of the proposed technique converges to the approximate optimal solution and is dependent on the differential privacy budget and the number of Byzantine participants. The experimental findings on MNIST, Fashion MNIST, and CIFAR10 demonstrate the efficacy and efficiency of the proposed approach. Machine learning excels in a number of data-driven applications, including self-driving cars, picture classification, speech recognition, and recommendation systems. The availability of vast volumes of data is a key component in machine learning's efficacy. However, for most businesses, it might be difficult to collect enough data in a diversity and quantity to produce a good predictive model. Consequently, collaborative learning—which enables learning from several data sources—becomes an increasingly popular solution. However, the training dataset may contain incredibly private information about an individual. Therefore, sharing the training data with all participants or with a central server is undesirable (and often unlawful) from a security and privacy point of view. Thus, putting privacy-preserving collaborative learning into practice has been a challenging research topic in recent years. **[6]**

Multi-domain networking slice orchestration is an essential part of the cloud-native and programmable 5G network. However, existing research solutions are either based on the unreasonable assumption that operators will reveal all private network information, or they are time-consuming secure multi-party computing that is only applicable to certain calculation circumstances. In order to deliver end-to-end network slice orchestration services that are both quick and private, this paper proposes NetChain, a multi-domain network slice orchestration architecture built on blockchain and a trusted execution environment. To provide the strong security, scalability, and information consistency of NetChain, we develop a novel consensus technique called CoNet. Additionally, a bilateral assessment method based on game theory is proposed to guarantee fairness and Quality of Experience by suppressing malicious actions during multi-domain network slice orchestration. Finally, private computing is used to install and evaluate the NetChain prototype on the Microsoft Azure Cloud. The experiment's findings indicate that NetChain functions effectively and is safe when anonymity is preserved. The fifth generation of mobile networks is transforming the communication service experience and enabling new applications in a range of domains, such as industry internet and tactile. A network slice (NS), a concept proposed by 5G, is an autonomous end-to-end (E2E) logical network that operates on a common infrastructure (i.e., computing, storage, and connectivity resources) and may deliver a negotiable service level agreed upon by its providers and clients.

The cornerstones of NS are software defined networking (SDN) and network function virtualization (NFV), which enable E2E provisioning of network resources to meet the service requirements of vertical enterprises. Emerging scenarios indicate that 5G requires agile end-to-end network slice orchestration across several administrative domains. This confronts many challenges, such as inconsistent billing, privacy disclosure, and fairness, because of a lack of trust. To solve the problems, a variety of plans are proposed, ranging from traditional to blockchain-based solutions. **[7]**

Applications that are sensitive to privacy, such as lifestyle prediction, medical diagnosis, and facial recognition, are increasingly using machine-learning (ML) techniques. A model inversion attack, which was recently described in a case study of linear classifiers in personalized medicine by Fredrikson et al., is a misuse of adversarial access to an ML model to gain private genomic data about individuals. Nevertheless, whether model inversion attacks are effective outside of their own setting is unknown. We develop a new type of model inversion attack that exploits confidence levels revealed with predictions. We analyze in detail two of our novel assaults that can be applied to many scenarios: facial recognition using neural networks and lifestyle survey decision trees as used in machine-learning-as-a-service platforms. People who are able to ask models prediction questions in both scenarios are shown their confidence levels. In the alternative case, we show that, with just a person's name and the machine learning model, it is possible to recover recognizable images of their faces. Additionally, we experimentally illustrate attacks that may be used to ascertain whether a lifestyle survey participant admitted to cheating on their partner. Furthermore, by exposing just rounded confidence values and investigating a privacy-aware decision tree training technique that is a simple variant of CART learning, we start the experimental exploration of natural countermeasures. The conclusion is that these MI attacks can be prevented with minimal loss of utility. **[8]**

We study the robustness of distributed stochastic gradient descent (SGD) implementations against Byzantine failures. Distributed machine learning systems have so far mostly ignored the possibility of failures, especially random (i.e., Byzantine) ones. Some of the causes of failures include software bugs, network asynchrony, biases in local datasets, and attackers trying to compromise the system as a whole. Assuming a set of n workers up to f is Byzantine, we inquire as to how robust SGD may be without limiting the dimension or size of the parameter space. Using a linear combination of the vectors proposed by the workers, we first show that no gradient aggregation rule (i.e., current methods) can survive a single Byzantine failure. Next, we incorporate the essential requirements required to guarantee convergence despite Byzantine workers into a resilience feature for the aggregation rule. We assert that Krum, an aggregation rule that meets our resilience condition, is the first provably Byzantine-resilient algorithm for distributed SGD. Krum's experimental evaluations are also presented. **[9]**

Federated learning is the idea that a neural network can be trained collectively on a server. Each user communicates changes to the parameters (gradients) based on local data after obtaining the network's current weights. Because only parameter gradients are communicated and input data remains on the device, this protocol not only efficiently trains neural networks but also provides customers with privacy benefits. But how secure is it to communicate parameter gradients? Previous attacks have provided consumers with a false sense of security because they only operated in simulated situations, even for a single image. However, by using a magnitude-invariant loss and optimization strategies based on adversarial attacks, we demonstrate that it is possible to reliably reconstruct images at high resolution utilizing the knowledge of their parameter gradients. We also demonstrate that this privacy violation can occur in trained deep networks. By examining how architecture and parameters affect the complexity of reconstructing an input image, we show that any input to a fully linked layer can be analytically reconstructed independent of the remaining architecture. Finally, we discuss real-world scenarios and show that even averaging gradients across several iterations or images does not protect user privacy in computer vision federated learning systems. **[10]**

3. METHODOLOGIES

The system model and fundamental assumptions of our suggested framework, MS-FL, are built up in this section. After that, we provide MS-FL with detailed instructions on how to complete model transactions and prevent poisoning attempts. TABLE 1 lists the symbols that occurred below along with their related descriptions for convenience.

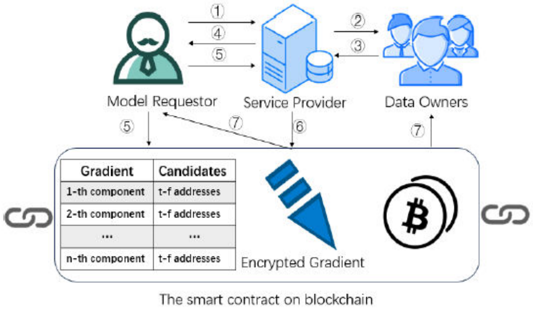
A. System Model

There are three basic entities in our system (Fig. 1):

- **Model Requestor:** model requestors do not have data but need model, they are also the initiator of the machine learning. It is honest but curious.
- **Data Owner:** data owners have data and they are curious about others' privacy. Some of them are malicious, who will poison in the model training.
- **Service Provider (SP):** Service provider is responsible to receive all gradients submitted by data owners and aggregate them. It will add noise in the gradients to protect data owner's privacy. It is also honest but curious.

B. Basic Assumption

- Each data owner has IID (independent and identically distributed) data samples.
- Malicious nodes can initiate three different kinds of attacks: arbitrary model poisoning, backdoor assaults, and label-flipping attacks.
- Any two participants in this system can communicate securely and dependably with one another.



Gradient	Candidates
1-th component	t-f addresses
2-th component	t-f addresses
...	...
n-th component	t-f addresses

FIGURE 1. **SYSTEM MODEL**

MODULES DESCRIPTION:

- **User Interface Design:** Users must enter their username and password in order to connect to the server; only then may they do so. If the user has already left, they can log in directly; if not, they must register their information on the server, including their username, password, email address, city, and country. To maintain the upload and download rate, the database will generate an account for every user. The user ID will be assigned to the name. Usually, logging in allows you to access a certain page. The query will be searched and displayed.
- **Message Sending Process:** This is the project's second module. In this scenario, the user assumes the role of a sender who needs to communicate with another program user. Like sender, message, and recipient, metadata is required for message transmission. This metadata aids in the network's packet data routing so that it can reach its destination. Data users possess data and are interested in the privacy of others. A few of them are malevolent and will contaminate the model during training.
- **Routing:** This is the project's third module. After a message has been successfully outsourced to the network, this module works. Sending data from one node to its next node is the only thing that constitutes routing. Routers are being used in this project to transmit data. The network experiences anonymous routing as a result of these. Routers only know the information about the previous node and the next neighbor node; they are unaware of the sender and recipient metadata.
- **Data Security analysis:** The project's fourth module is this one. Typically, network security is the primary concern since unscrupulous people may be able to access the data that is outsourced. We transform the data that will be outsourced across the network into cipher text in order to prevent this. The symmetrical approach, which uses a single key to both encrypt and decrypt, is not safe. Thus, we are employing anonymous routing based on onions. A matching secret key that aids in decryption will be developed in conjunction with the public key used for encryption. The same procedure will be carried out in various router tiers.
- **Composed Data Analysis:** The project's fifth module is this one. In actuality, it is the project's improving module. If a user wishes to find out what information they shared with a certain user, they must remember everything they shared up until that point. Checking one by one is difficult and time consuming. In order to prevent that, we are utilizing the data analysis idea, which requires the user to enter his name in order to access all of the shared data up to this point. Easily retrieve the shared data in a sequential fashion from the user-specific database.

4. ALGORITHM

SPECIFIC PROCESS:

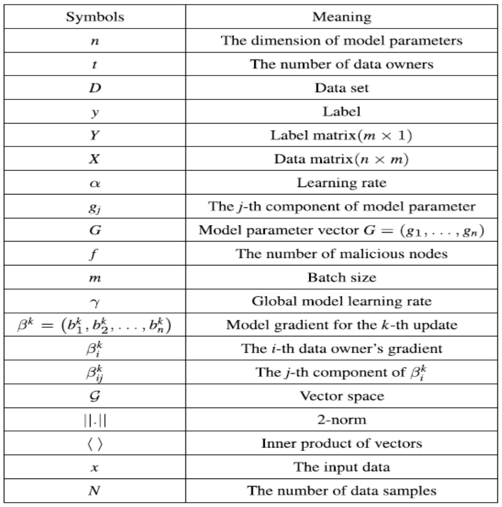
The process of the MS-FL and corresponding algorithm are demonstrated in below.

Step 1: The model requestor uses Algorithm 1 to encrypt the model parameters with the public key and sends them to the service provider (Enc(z,pk) → v denotes that plaintext z is encrypted to ciphertext v by CKKS public key pk).

Step 2: The service provider receives the encrypted model parameters and sends them to data owners.

Step 3: The data owners will deliver the updated model gradients to service provider after executing Algorithm 2.

Step 4: The service provider adds noise into model gradients by Algorithm 3 to protect the privacy of data owners, then send them to model requestor.



Symbols	Meaning
n	The dimension of model parameters
t	The number of data owners
D	Data set
y	Label
Y	Label matrix$(m \times 1)$
X	Data matrix$(n \times m)$
α	Learning rate
g_j	The j-th component of model parameter
G	Model parameter vector $G = (g_1, \ldots, g_n)$
f	The number of malicious nodes
m	Batch size
γ	Global model learning rate
$\beta^k = (b_1^k, b_2^k, \ldots, b_n^k)$	Model gradient for the k-th update
β_i^k	The i-th data owner's gradient
β_{ij}^k	The j-th component of β_i^k
$\mathcal{G}$	Vector space
$\|\|.\|\|$	2-norm
$\langle\,\rangle$	Inner product of vectors
x	The input data
N	The number of data samples

Step 5: Upon receiving encrypted gradients containing noise, the model requestor initiates the process of decryption by utilizing its private key. Once the gradients have been decrypted, the model requestor employs Algorithm 4 (Dec(v, sk)→z denotes that ciphertext v is decrypted to plaintext z by CKKS private key sk) to facilitate the selection of t−f data owners' parameters for aggregation with respect to each gradient component containing noise.

After the selection process is successful, the model requestor creates a table similar to Fig. 3 with the addresses of the data owners that helped aggregate each gradient component. This table allows the model requestor to monitor the overall status of the aggregation process and records the contributions made by each data owner.

Lastly, the model requestor starts the process of creating a smart contract using the previously specified table and a little amount of cryptocurrency. This smart contract acts as a safeguard to guarantee that each individual involved in the aggregation process receives just compensation for their contributions. Following the creation of the smart contract, the model requestor completes the gradient aggregation process by sending the contract's address to the service provider.

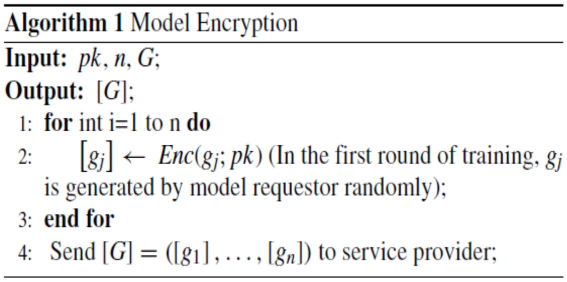
Step 6: The service provider plays a crucial part in enabling the gradient aggregation procedure to be carried out successfully after receiving the smart contract. To achieve this, the service provider starts a stringent set of checks and balances that are intended to guarantee the process's overall accuracy, dependability, and fairness.

A detailed analysis of the quantity of cryptocurrencies present in the smart contract is the first step in this procedure. The service provider does not provide the no-noise encrypted gradient to the smart contract if the quantity of bitcoin is determined to be insufficient to satisfy the amount that the data owners have specified. This phase is essential to guaranteeing that the gradient aggregation process operates in a fair and transparent manner and that the interests of all parties involved are sufficiently safeguarded.

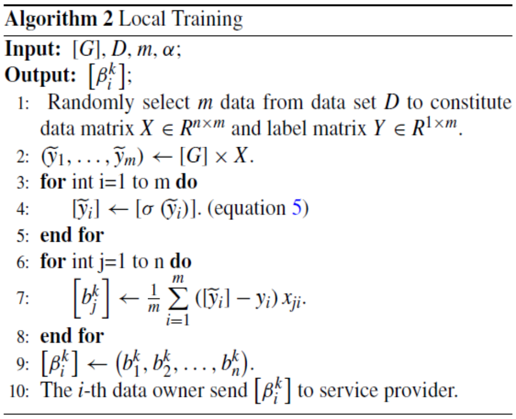
Following the verification of the cryptocurrency included in the smart contract, the service provider proceeds to the next phase of the procedure, which entails choosing noise-free gradient components for aggregation. The model requestor's table, which includes pertinent details about the contributions made by the collaborating data owners, serves as a guide for this selection process.

After determining which gradient components are most suited for aggregation, the service provider starts the aggregation process and uses Algorithm 5 to produce the encrypted aggregate result. When the aggregation process is finished, the service provider uses the smart contract to send the outcome to the model requestor's account.

The final step ensures that everyone participating in the gradient aggregation process is fairly compensated for their contributions by distributing the cryptocurrency contained in the smart contract equally among the corresponding account addresses of the participating data owners on the table. This stage is essential to preserving the data owners' confidence and goodwill as well as their commitment to the gradient aggregation process.



```
Algorithm 1 Model Encryption
Input: $pk, n, G$;
Output: $[G]$;
for int i=1 to n do
    $[g_j] \leftarrow Enc(g_j; pk)$ (In the first round of training, $g_j$ is generated by model requestor randomly);
end for
Send $[G] = ([g_1], \ldots, [g_n])$ to service provider;
```



```
Algorithm 2 Local Training
Input: $[G], D, m, \alpha$;
Output: $[\beta_i^k]$;
Randomly select $m$ data from data set $D$ to constitute data matrix $X \in R^{n\times m}$ and label matrix $Y \in R^{1\times m}$.
$(\widetilde{y}_1, \ldots, \widetilde{y}_m) \leftarrow [G] \times X$.
for int i=1 to m do
    $[\widetilde{y}_i] \leftarrow [\sigma(\widetilde{y}_i)]$. (equation 5)
end for
for int j=1 to n do
    $\left[b_j^k\right] \leftarrow \frac{1}{m}\sum_{i=1}^{m}([\widetilde{y}_i] - y_i)\,x_{ji}$.
end for
$\left[\beta_i^k\right] \leftarrow (b_1^k, b_2^k, \ldots, b_n^k)$.
The $i$-th data owner send $\left[\beta_i^k\right]$ to service provider.
```

```
Algorithm 3 Privacy Protection
Input: $\{[\beta_i^k]\}_{i=1}^t$;
Output: $\left\{\widehat{[\beta_i^k]}\right\}_{i=1}^t$ ($\{[\beta_i^k]\}_{i=1}^t$ with noise);
Generate $n$ pairs of non-zero random numbers $(c_j, d_j)$, $j = 1, 2, \ldots, n$.
for int j=1 to n do
    for int i=1 to t do
        $\widehat{\left[\beta_{ij}^k\right]} \leftarrow c_j \left[\beta_{ij}^k\right] + d_j$. (add noise)
    end for
end for
Send $\left\{\widehat{[\beta_i^k]}\right\}_{i=1}^t$ to model requestor.
```

Step 7: The final step ensures that everyone participating in the gradient aggregation process is fairly compensated for their contributions by distributing the cryptocurrency contained in the smart contract equally among the corresponding account addresses of the participating data owners on the table. This stage is essential to preserving the data owners' confidence and goodwill as well as their commitment to the gradient aggregation process.

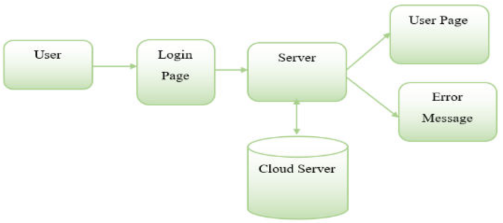
```
Algorithm 4 Aggregation Algorithm
Input: $\left\{\widehat{[\beta_i^k]}\right\}_{i=1}^t$, $sk$;
Output: selected gradient components;
$\left\{\widehat{\beta_i^k}\right\}_{i=1}^t = Dec(\left\{\widehat{[\beta_i^k]}\right\}_{i=1}^t; sk)$.
for int j=1 to n do
    assign $\left\{\widehat{\beta_{ij}^k}\right\}_{i=1}^t$ to the list $\{b_1, b_2, \ldots, b_t\}$ in order.
    for int i=1 to t do
        Pick $t - f - 1$ points that have the smallest distance from $b_i$ in the set $\{b_1, b_2, \ldots, b_t\}$, which constitute $\{b_{1^*}, b_{2^*}, \ldots, b_{(t-f-1)^*}\}$, suppose the distance set between these points and $b_i$ is $\left\{h_{1^*}^i, h_{2^*}^i, \ldots, h_{(t-f-1)^*}^i\right\}$.
        for int j=1 to t-f-1 do
            if $b_{j^*} < b_i$ then
                $h_{j^*}^i \leftarrow -h_{j^*}^i$.
            end if
        end for
        $b_i' \leftarrow \sum_{j=1}^{t-f-1} h_{j^*}^i$.
    end for
    Select the smallest $b_i'$ in the set $\{b_i'\}_{i=1}^t$ and then choose $t - f$ data owners as candidates whose gradient component is or is closest to $b_i$ in the set $\{b_1, b_2, \ldots, b_t\}$.
end for
Take the account addresses of candidates for each component of gradient to establish the smart contract on blockchain.
Put cryptocurrency into smart contract.
```

```
Algorithm 5 Model Transaction
Input: $\{[\beta_i^k]\}_{i=1}^t$;
Output: $[\beta^{k+1}]$;
for int j=1 to n do
    Add $t - f$ values in the set $\left\{\left[\beta_{ij}^k\right]\right\}_{i=1}^t$ corresponding to the accounts in the newly created smart contract to get the result $[\beta^{k+1}]$
end for
Send $[\beta^{k+1}]$ to smart contract.
```

In order to prepare for the subsequent iteration, the model requestor uses Algorithm 6 to finish model updating after processing the obtained aggregation results.

5. DATA FLOW DIAGRAM

LEVEL 0



LEVEL 1

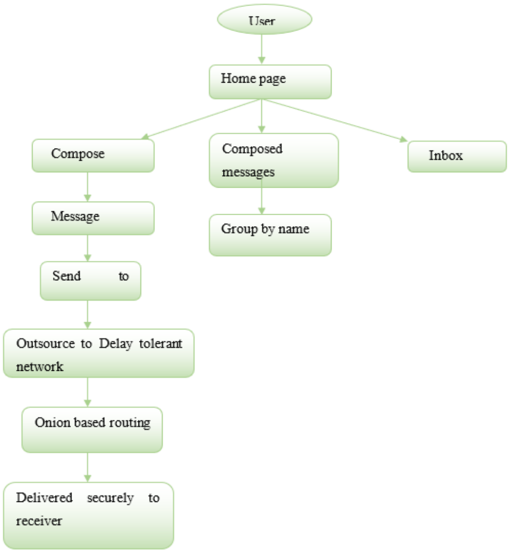


FIGURE 2. **DATA FLOW DIRAGRAM**

6. SYSTEM ARCHITECTURE

This diagram demonstrates how to securely and privately send communications over a network. The user initially registers and login into the system. The website allows the user to view their inbox, compose new messages, and view previously written messages. After being written, a message is sent over a particular kind of network called a delay-tolerant network, which ensures that the message will still be delivered even if the connection is slow or interrupted. To protect its privacy, the communication is sent via a method called onion routing. The connection passes through three routers (Router 1, Router 2, and Router 3) while using this method. Like peeling an onion, an encryption layer is removed as it passes through each router.

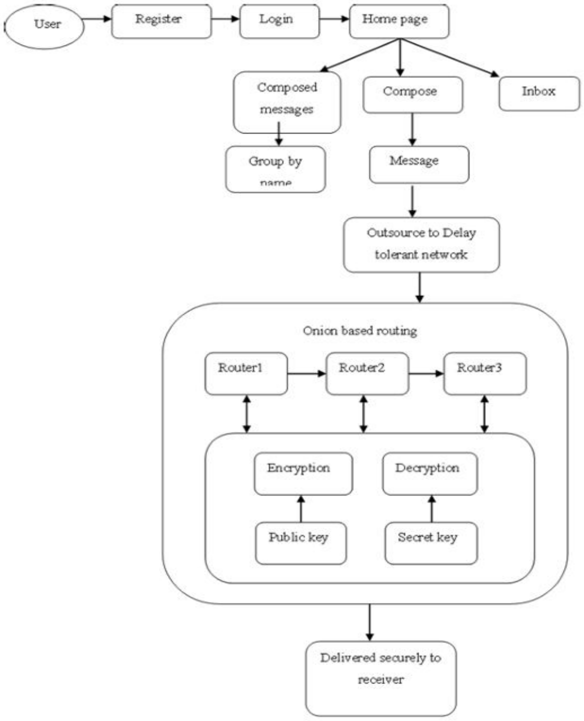


FIGURE 3. **SYSTEM ARCHITECTURE**

The message is encrypted using a public key and decoded with a secret key to ensure that only the intended recipient may see it. Once the secure routing process is finished, the communication is safely sent to the addressee. The complete system helps to ensure private, secure, and reliable communication even in the face of difficult network conditions.

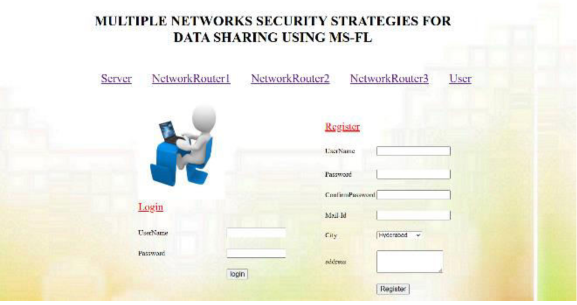
7. RESULTS

The results obtained from the implementation of the proposed system clearly demonstrate its ability to provide secure data sharing across multiple networks using multiple security strategies. The initial results Fig. 4 highlight the login and registration process, where new users were able to create accounts by providing essential details such as username, password, mobile number, city, and address, while existing users could log in using valid identifications. This establishes that the authentication mechanism was successfully implemented, ensuring that only valid users were granted access to the system. By enforcing authentication, the system effectively prevents unauthorized users from entering the communication framework, thereby addressing one of the primary requirements of data security**.** Following authentication, the results confirmed that users were able to compose and transmit secure messages. The message composition interface allowed a sender to enter the message content, specify the intended receiver, and submit the message for delivery. Additional features such as viewing composed messages and checking the inbox were also useful, providing users with control over their communication. This stage validated that the system effectively supports structured communication while simultaneously preparing messages for encrypted transmission across the network.

The results from the transmission phase further confirmed that the system successfully implements multi-layer encryption. Messages were not transmitted in plaintext but were first encrypted using public and secret keys before being forwarded. Each transmitted message displayed information such as the previous node, the encrypted message, the next node, and the key references. Although a decrypt option was provided, it could only be accessed by an authorized recipient keeping the correct secret key. This demonstrates that the system ensures privacy during transmission, as the encrypted form of the message remains unreadable to all unauthorized entities.

As the encrypted messages passed through intermediate routers, the results showed that each router was only capable of forwarding the encrypted content and did not have access to the actual message. The routers displayed routing details and key references, but the original data remained fully hidden. This confirms that even if one or more routers were compromised, the original information could not be exposed. Such an approach reflects the federated learning principle integrated into the system, where no single node in the communication chain has complete control or visibility of sensitive data. By distributing responsibility across multiple nodes, the framework effectively minimizes risks associated with single points of failure.

Finally, the results at the receiver's end confirmed the successful decryption of the transmitted message. The intended receiver, upon applying the correct secret key, was able to retrieve and view the original message content along with the sender's information in the inbox. This validates the achievement of end-to-end confidentiality, as the message remained protected throughout its journey across the network and was accessible only to the authorized receiver.


MULTIPLE NETWORKS SECURITY STRATEGIES FOR DATA SHARING USING MS-FL

Register

UserName

Password

ConfirmPassword

Mail Id

City Hyderabad

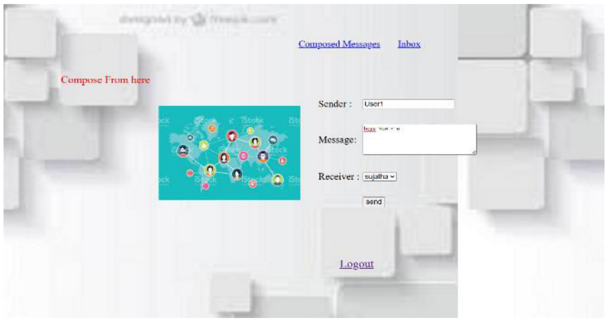
address

Login

UserName

Password

FIGURE 4. **HOME PAGE**

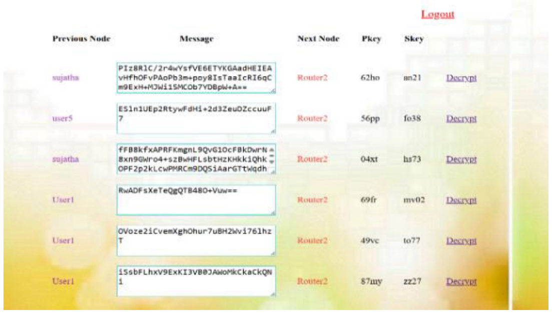


Sender : User1

Message: [illegible]

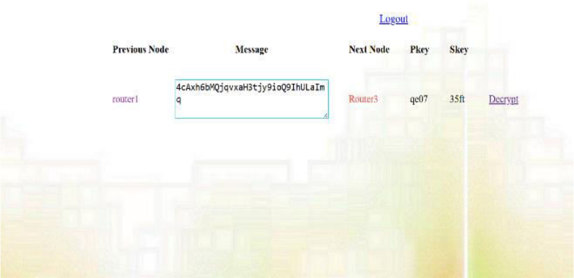
Receiver : sujatha

FIGURE 8. **SENDING MESSAGE FROM ONE USER TO ANOTHER**



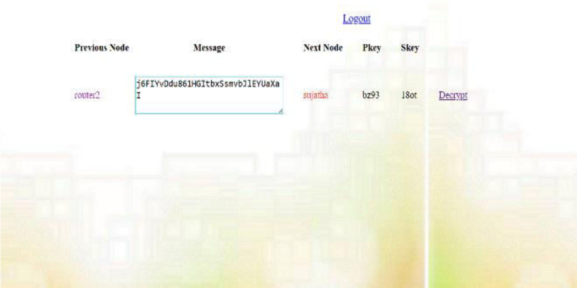
Previous Node	Message	Next Node	Pkey	Skey	
sujatha	PIz8R1C/2r4wYsfVE6ETYKGAadHEIEAvHfhOFvPAoPb3m+poy8IsTaaIcRI6qCm9ExH+MJWi1SMCOb7YDBpW+A==	Router2	62ho	an21	Decrypt
user5	ESln1UEp2RtywFdHi+2d3ZeuDZccuuF7	Router2	56pp	fo38	Decrypt
sujatha	fFB8kfxAPRFKmgnL9QvG1OcFBkDwrN8xn9GWro4+sZBwHFLsbtHzKHkk1QhkOPF2p2kLcwPMRCm9DQSiAarGTtWqdh	Router2	04xt	hs73	Decrypt
User1	RwADFsXeTeQgQTB48O+Vuw==	Router2	69fr	mv02	Decrypt
User1	OVoze2iCvemXghOhur7uBH2Wvi761hzT	Router2	49vc	to77	Decrypt
User1	iSsbFLhxV9ExKI3VB0JAWoMkCkaCkQN1	Router2	87my	zz27	Decrypt

FIGURE 9. **ROUTER1 TRANSACTION DETAILS**



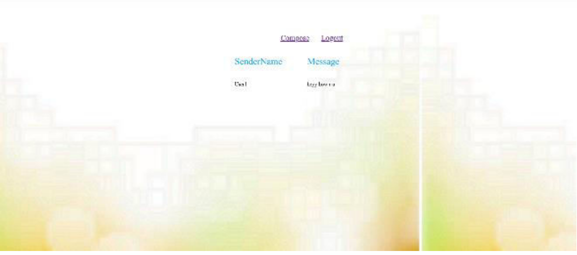
Previous Node	Message	Next Node	Pkey	Skey	
router1	4cAxh6bMQjqvxaH3tjy9ioQ9IhULaImq	Router3	qe07	35ft	Decrypt

FIGURE 10. **ROUTER 2 TRANSACTION DETAILS**



Previous Node	Message	Next Node	Pkey	Skey	
router2	j6FIYvDduB61HGItbxSsmvbJ1EYUaXaI	sujatha	bz93	18ot	Decrypt

FIGURE 11. **ROUTER 3 TRANSACTION DETAILS**



SenderName	Message
User1	[illegible]

FIGURE 12. **RECEIVED MESSAGE TO USER**

8. FUTURE ENHANCEMENT

We intend to improve the accuracy of federated learning models in the future by investigating additional algorithms to protect data privacy and thwart poisoning assaults.

9. CONCLUSION

This study proposes MS-FL, a unique federated learning framework based on various security measures, for data application scenarios where the data owner and the model requestor are not the same. It has been established that this framework protects the privacy of data owners and model requestors. Additionally, it can safeguard FL players' interests, and MS-FL's aggregation technique can fend off common byzantine poisoning assaults. The experimental findings at the conclusion of this research show that the MS-FL aggregation algorithm performs similarly in terms of accuracy and resilience.

10. REFERENCES

[1] S. Bai, G. Yang, G. Liu, H. Dai and C. Rong, "NttpFL: Privacy-preserving oriented no trusted third party fe

derated learning system based on blockchain", IEEE Trans. Netw. Service Manage, vol. 19, no. 4, pp. 3750-3763, Dec. 2022.

[2] D. D, A. K. K and R. M, "Research on homomorphic encryption for arithmetic of approximate numbers", Proc. Int. Conf. Intell. Syst. Commun. IoT Security (ICISCoIS), pp. 409-437, Feb. 2023

[3] Y. Jiang, W. Zhang and Y. Chen, "Data quality detection mechanism against label flipping attacks in federated learning", IEEE Trans. Inf. Forensics Security, vol. 18, pp. 1625-1637, 2023.

[4] C. Xu, Y. Jia, L. Zhu, C. Zhang, G. Jin and K. Sharif, "TDFL: Truth discovery based Byzantine robust federated learning", IEEE Trans. Parallel Distrib. Syst., vol. 33, no. 12, pp. 4835-4848, Dec. 2022

[5] Z. Liu, C. Hu, H. Xia, T. Xiang, B. Wang and J. Chen, "SPDTS: A differential privacy based blockchain scheme for secure power data trading", IEEE Trans. Netw. Service Manage, vol. 19, no. 4, pp. 5196-5207, Dec. 2022.

[6] X. Ma, X. Sun, Y. Wu, Z. Liu, X. Chen and C. Dong, "Differentially private Byzantine robust federated learning", IEEE Trans. Parallel Distrib. Syst., vol. 33, no. 12, pp. 3690-3701, Dec. 2022.

[7] G. He, W. Su, S. Gao, N. Liu and S. K. Das, "NetChain: A blockchain-enabled privacy preserving multi-domain network slice orchestration architecture", IEEE Trans. Netw. Service Manage, vol. 19, no. 1, pp. 188-202, Mar. 2022

[8] M. Fredrikson, S. Jha and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures", Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Security, pp. 1322-1333, Oct. 2015.

[9] P. Blanchard, E. Mahdi E. Mhamdi, R. Guerraoui and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent", Proc. Adv. Neural Inf. Process. Syst., vol. 30, pp. 1-11, 2017. 62 [10] J. Geiping, H. Bauermeister, H. Dröge and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?", Proc. Adv. Neural Inf. Process. Syst., vol. 33, pp. 16937-16947, Dec. 2020.

[11] C. Chen, J. Zhou, L. Wang, X. Wu, W. Fang, J. Tan, et al., "When homomorphic encryption marries secret sharing: Secure large-scale sparse logistic regression and applications in risk control", Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining, pp. 2652-2662, Aug. 2021.

[12] M. Nasr, R. Shokri and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning", Proc. IEEE Symp. Security Privacy (SP), pp. 739-753, May 2019.

[13] Y. Jiang, W. Zhang and Y. Chen, "Data quality detection mechanism against label flipping attacks in federated learning", IEEE Trans. Inf. Forensics Security, vol. 18, pp. 1625-1637, 2023.

[14] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin and V. Shmatikov, "How to backdoor federated learning", Proc. Int. Conf. Artif. Intell. Statist, pp. 2938-2948, Jun. 2020.

[15]10. B. Hitaj, G. Ateniese and F. Perez-Cruz, "Deep models under the GAN: Information leakage from collaborative deep learning", Proc. ACM SIGSAC Conf. Comput. Commun. Security, pp. 603-618, Oct. 2017.

[16] Y. Chen, L. Su and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent", Proc. Abstr. ACM Int. Conf. Meas. Model. Comput. Syst., pp. 1-25, Jun. 2018.

[17] D. Yin, Y. Chen, R. Kannan and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates", Proc. Int. Conf. Mach. Learn., pp. 5650-5659, Jul. 2018.

[18] R. Guerraoui and S. Rouault, "The hidden vulnerability of distributed learning in byzantium", Proc. Int. Conf. Mach. Learn., pp. 3521-3530, Jul. 2018.

[19] L. Muñoz-González, K. T. Co and E. Lupu, "Byzantine-robust federated machine learning through adaptive model averaging", arXiv: 1909.05125, 2019.

[20] X. Liu, H. Li, G. Xu, Z. Chen, X. Huang and R. Lu, "Privacy-enhanced federated learning against poisoning adversaries", IEEE Trans. Inf. Forensics Security, vol. 16, pp. 4574-4588, 2021. 63

[21] Y. Song, F. Wei, K. Zhu and Y. Zhu, "Anomaly detection as a service: An outsourced anomaly detection scheme for blockchain in a privacy-preserving manner", IEEE Trans. Netw. Service Manage, vol. 19, no. 4, pp. 3794-3809, Dec. 2022. [22] K. Han, S. Hong, J. H. Cheon and D. Park, "Logistic regression on homomorphic encrypted data at scale", Proc. AAAI Conf. Artif. Intell., vol. 33, pp. 9466-9471, Jul. 2019.

[23] H. Chen, R. Gilad-Bachrach, K. Han, Z. Huang, A. Jalali, K. Laine, et al., "Logistic regression over encrypted data from fully homomorphic encryption", BMC Med. Genomics, vol. 11, no. S4, pp. 3-12, Oct. 2018.

[24] J. H. Cheon, D. Kim, Y. Kim and Y. Song, "Ensemble method for privacy-preserving logistic regression based on homomorphic encryption", IEEE Access, vol. 6, pp. 46938 46948, 2018.

[25] Y. Li, C. Chen, N. Liu, H. Huang, Z. Zheng and Q. Yan, "A blockchain-based decentralized federated learning framework with committee consensus", IEEE Netw., vol. 35, no. 1, pp. 234-241, Jan. 2021. [26] Y. Zhao, J. Zhao, L. Jiang, R. Tan, D. Niyato, Z. Li, et al., "Privacy-preserving blockchain-based federated learning for IoT devices", IEEE Internet Things J., vol. 8, no. 3, pp. 1817-1829, Feb. 2021. [27] W. Zhang, Q. Lu, Q. Yu, Z. Li, Y. Liu, S. K. Lo, et al., "Blockchain-based federated learning for device failure detection in industrial IoT", IEEE Internet Things J., vol. 8, no. 7, pp. 5926-5937, Apr. 2021.

[28] 27. K. Toyoda and A. N. Zhang, "Mechanism design for an incentive-aware blockchain-enabled federated learning platform", Proc. IEEE Int. Conf. Big Data, pp. 395 403, Dec. 2019. [29] A. Hammoud, H. Otrok, A. Mourad and Z. Dziong, "On demand fog federations for horizontal federated learning in IoV", IEEE Trans. Netw. Service Manage, vol. 19, no. 3, pp. 3062-3075, Sep. 2022.

[30] M. Kim, Y. Song, S. Wang, Y. Xia and X. Jiang, "Secure logistic regression based on homomorphic encryption: Design and evaluation", JMIR Med. Informat., vol. 6, no. 2, Apr. 2018. 64 [31] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. Mcmahan, S. Patel, et al., "Practical secure aggregation for privacy-preserving machine learning", Proc. ACM SIGSAC Conf. Comput. Commun. Security, pp. 1175-1191, Oct. 2017.

[32] G. Xu, H. Li, Y. Zhang, S. Xu, J. Ning and R. H. Deng, "Privacy-preserving federated deep learning with irregular users", IEEE Trans. Dependable Secure Comput., vol. 19, no. 2, pp. 1364-1381, Mar. 2022.