Proceedings of the 6th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/86/20241552

Federated learning implementation based on DDS data distribution service

Hao Liu

School of Cyberspace Security, Northwestern Polytechnical University, Xi'an, 710072, China

liuhao4577@mail.nwpu.edu.cn

Abstract. Federated learning allows multiple local private clients to collaborate on training the same model without sharing each client's private data. It can achieve collaborative training between users while protecting data privacy and security. Therefore, it is advocated for use in the fields of Internet of Things, Internet of Vehicles, etc. There are security issues in the transmission process of federated learning models. How to ensure data privacy during local data collection, local model parameter transmission, model aggregation and performance testing while maximizing the use of private data is a question that people have been exploring. This paper proposes a DDS-based FL framework (FL-DDS) to achieve secure transmission of model parameters. The client and server are used as nodes in the DDS security domain, and the DDS data distribution mechanism is used to transmit model parameters. Through the DDS topic-based publish-subscribe mechanism, the global model and local model are transmitted through the DDS topic. At the same time, the DDS authentication component, access control component and encryption component are used to achieve domain-level security and intra-domain security of model transmission. Experiments show that (FL-DDS) can protect the privacy of model parameters without affecting communication performance.

Keywords: Deep learning, Data distribution service, Federated learning, parameter transmission

1. Introduction

Big data and artificial intelligence, as data-driven technologies and important engines for developing new productivity, are widely used in multiple industries such as the Internet of Things, smart manufacturing, medical care, and transportation. Data has in fact become a new type of asset. As data security and user privacy issues become more and more difficult, the potential value of data has become increasingly important. How can we conduct large-scale data analysis and computing without leaking data to the outside world? Federated learning (FL) is a breakthrough technology in distributed machine learning. It allows many clients to jointly train the same model without sharing original local privacy data. It is a new privacy computing strategy that aims to solve data decentralization and data privacy issues.

However, the FL model parameters are vulnerable to external or internal attacks when they are transmitted, making the model transmission process unsafe. Recent studies have shown that although participants in FL only exchange model weight data instead of local private data, the possibility of participants' privacy being compromised still exists. Although the original local privacy data remains on

DOI: 10.54254/2755-2721/86/20241552

the local client, Parameters of the interaction model between the client and the server may inadvertently leak sensitive information. These model updates, if intercepted or analyzed, may be reverse engineered to infer the properties of the underlying data, posing significant privacy risks [1-3].

From the perspective of model transfer, we design the Federated Learning DDS framework (FL-DDS) based on the DDS security component specifically for the current federated learning architecture. We use the authentication, access control, data integrity and confidentiality mechanisms provided by the DDS security component to achieve secure distribution of global model parameters and secure upload of local models in federated learning, ensuring the accuracy of training and the privacy of all participants. Specifically, our contributions are as follows: Implement federated learning through the DDS distributed computing framework. Enable the DDS security mechanism to implement federated learning model parameter transmission. Ensure the security of model transmission without losing accuracy

The rest of this paper includes: Section II explains and summarizes the research background of key technologies such as FL, DDS, and ROS. Section III elaborates on the algorithm design of FL-DDS. Section IV introduces the setup and experimental results of the FL-DDS evaluation experiment. Finally, in Section V, we summarize and evaluate our work on FL-DDS and look forward to the potential of the FL-DDS framework.

2. Background

2.1. Federated Learning

The FedAVG algorithm is a classic algorithm in federated learning [4]. Its training framework is shown in Figure 1.

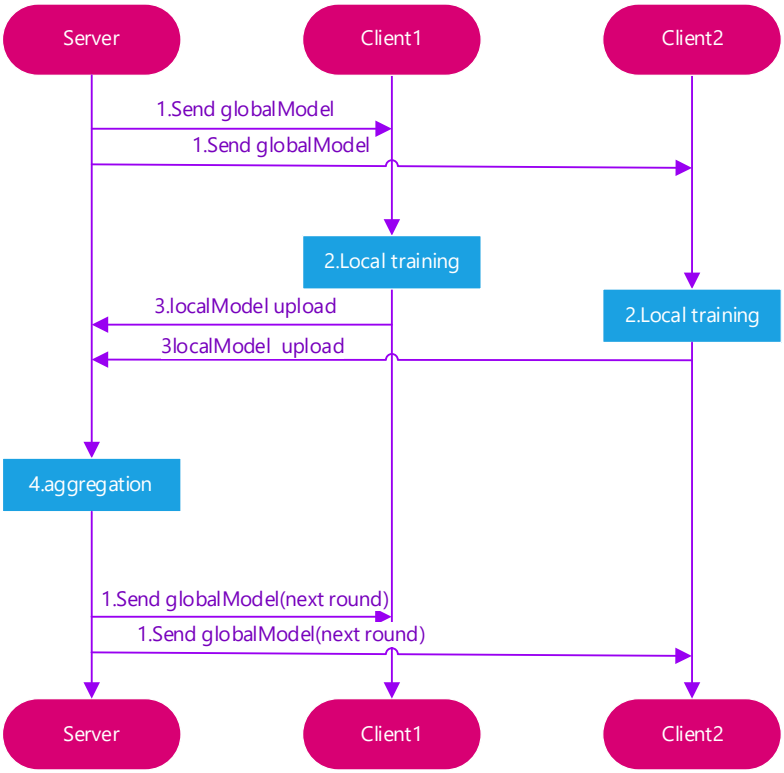

Figure 1. The FedAvg framework

There are four steps to it. The globalModel is initialized and sent to all local clients by the central server during the first stage, known as startup. The next step is local training, where each local client

DOI: 10.54254/2755-2721/86/20241552

independently trains the model using local private data and generates local model updates after a specified local epoch training cycle. During the local model upload stage, the local client then uploads the locally adjusted model parameters to the central server. The global aggregation stage is the last one. During this phase, the global server updates the globalModel parameters and broadcasts them to all clients. The weighted average of all the localModel updates it has received is then calculated. The model is iterated through this process until convergence.

Various techniques have been proposed to address the privacy issue. Differential privacy (DP) is one of the most prominent approaches, which, before to sharing model updates with the central server, adds controlled noise [5]. This ensures that the contribution of any single client's data to the overall model remains indistinguishable, thereby protecting individual privacy. Another method that permits computation on encrypted data is homomorphic encryption that allowing model updates to be securely aggregated without exposing the raw data.

Secure multi-party computing (SMPC), which allows many parties to cooperatively compute functions based on their inputs while maintaining the privacy of those inputs, is another facet of FL privacy security. By dividing the data into shares and distributing them to multiple parties, SMPC ensures that no single party has access to the complete dataset, thereby enhancing privacy.

Additionally, federated learning systems can employ traditional secret sharing (SS) schemes, where a central server aggregates model updates in a way that prevents exposure of any individual client data [6]. This technique, combined with a secure aggregation protocol, ensures that model updates are combined without revealing individual contributions.

In terms of model communication, a unique distributed quantized gradient algorithm was proposed by Jun Sun, Tianyi Chen, and others to solve the federated learning problem of communication efficiency. The adaptive communication of quantized gradients is the feature [7]. To update the model parameters, the global gradient is obtained by summing up all of the local gradients. Quantizing gradients and avoiding quantized gradient communications with less information by reusing prior gradients is the main concept behind conserving communication from workers to servers. Other works focus on the decentralized federated learning framework and propose algorithm called GossipFL, which enables each client to communicate with only one peer with a highly sparse model [8]. Additionally, these works suggest a matrix generation algorithm that can more efficiently use bandwidth resources while maintaining unity.

2.2. Data Distribution Service

The Object Management Group (OMG) proposed a standard for the Data Distribution Service (DDS), making DDS implemented according to this standard interoperable [9]. A Service Plugin Interface (SPI) architecture with particular plugins and APIs is defined by DDS-Security. The five plugins that make up DDS-Security are Data Tagging, Logging, Encryption, Access Control, and Authentication.

The SPI architecture supports multiple authentication schemes, and the default plugin is "DDS: Auth". For authentication and key exchange, this plugin makes use of a public key infrastructure that includes X.509 certificates, RSA or DSA, and Elliptic Curve Diffie-Hellman. Access Control leverages signed XML documents and PKI to define domain protection and participant permissions. The required security plugin "DDS: Crypto" lists the supporting cryptographic techniques, which include 128- and 256-bit AES keys for encryption, RSA, elliptic curve keys, and SHA-256 for digital signatures.

Participants in the domain are identified by means of authentication. Limitations on DDS-related operations are enforced via Access Control. Hashing, encryption, and signature are all managed by encryption. Logging allows security related event monitoring, and data tagging adds labels to data frames to prevent tampering.to the DDS specification, the first three plugins are mandatory. Authentication, Access Control, Encryption, and Logging contribute directly to security, while Data Tagging enhances security indirectly by enabling access control.

DOI: 10.54254/2755-2721/86/20241552

2.3. Robot Operating System 2

Robot Operating System 2, or ROS2, is the next-generation framework for developing robots which is created to get over the drawbacks of its predecessor, ROS1 [10]. It offers real-time performance, cross-platform interoperability with Windows, Linux, and macOS, and strong support for distributed systems. Because ROS2 is built on top of DDS (Data Distribution Service), it guarantees dependable, low-latency data sharing, which is necessary for sophisticated robotic applications.

Better security, compatibility for multi-robot systems, and increased middleware flexibility—which lets developers use the communication libraries of their choice—are some of ROS2's standout features. Its modular architecture makes maintenance and the integration of new features easy. Along with scalability, ROS2 is ideal for a wide range of applications, from small-scale hobby projects to massive industrial robots. More deterministic and real-time capabilities are built into the system, which is essential for applications that need exact timing and synchronization. All things considered, ROS2 is a huge step forward in the robotics industry, offering a more effective and adaptable toolkit for creating complex robotic systems.

3. Federated learning based on DDS middleware

3.1. Problem Statement

Let N be the number of local participants $(P1 - PN)$. Di is a local dataset for Pi . With the aid of a central server, we hope to develop a machine learning model from the dataset without transferring raw data. Training's objective is to resolve:

$$arg\min_{w} \mathcal{L}(w) = \sum_{i=1}^{N} \frac{|D^i|}{|D|} L_i(w) \tag{1}$$

Here $L_i(w)$is the empirical loss of Pi .

3.2. The flow of the FL-DDS algorithm

The security component based on data distribution services (DDS) realizes the security of the Federal average (FedAVG) global model and the security upload of local models, which mainly involves the following steps. First, use DDS's security strategy for global models. On the server side, through the DDS authentication, access control, and data encryption functions to ensure that the global model can only be accepted by the authorized client. After receiving the model, the local client trains a new model locally and uploads the trained model. During the upload process, DDS's security components pass data integrity verification and encryption transmission to ensure the privacy and immutability of local model data during the transmission process. The server receives local models, average federal aggregation, and the next round of global models. This implementation method based on DDS security components ensures the confidentiality, integrity and controllability of weights transmission in the federal learning circuit, and effectively prevent the risk of data leakage and tampering. Particularly separated into the subsequent stages:

1. Global Initialization. In server -side initialization globalModel parameter w^0 , local server initialization local data D^{i}, initialization of all participants' certificates and keys,
2. Client selection. Each process of iteration, the central server chooses a portion $S_t \subseteq \{1,2,\dots,\mathrm{K}\}$ where server is selected from all K clients. where $|S_t| = C \cdot K$ $(0 < C \leq 1)$.
3. the model sends. The server is through DDS topic: GlobalModel sends global model parameters w^t, local client subscribes to DDS topic: GlobalModel receives global model parameters w^t, and initialize local models:
4. Local training. Each selected client $k \in S_t$ receives this round globalModel parameter w^t from the server, and after that uses its local data D_k for local client training. local client k uses its private data D_k to learn rate η for E round random gradient decrease (SGD), update model parameters: $w_k^{t,0} = w^t$. For each local update $i \in \{1,2,\dots,E\}$and each local batch $b \subseteq D_k$: $w_k^{t,i} = w_k^{t,\mathrm{i}-1} -$

DOI: 10.54254/2755-2721/86/20241552

$\eta\nabla\ell\left(w_k^{t,\mathrm{i}-1};\ b\right)$. Among $\ell\left(w_k^{t,\mathrm{i}-1};\ b\right)$ indicates the loss function on the batch b. After the E round local training, the client k gets the updated local client model widgets w_k^{t+1}

5. Local model upload. Each client k publishes its updated local client model widgets w_k^{t+1} to the LocalModel topic, and the server subscribes to the LocalModel topic and stores it.
6. Global aggregation. After the server receives the localModel parameters of all selected client $k \in S_t$, the globalModel parameters are updated according to the number of samples of each clientn_k weighted the average: $w_k^{t+1} = \frac{1}{N}\Sigma_k \in S_t n_k w_k^{t+1}$. Where $N = \Sigma_k \in S_t$,and n_k shows the client's total number of samples.

Through the above steps, while ensuring data privacy, the goals of multiple clients to coordinate the global model.

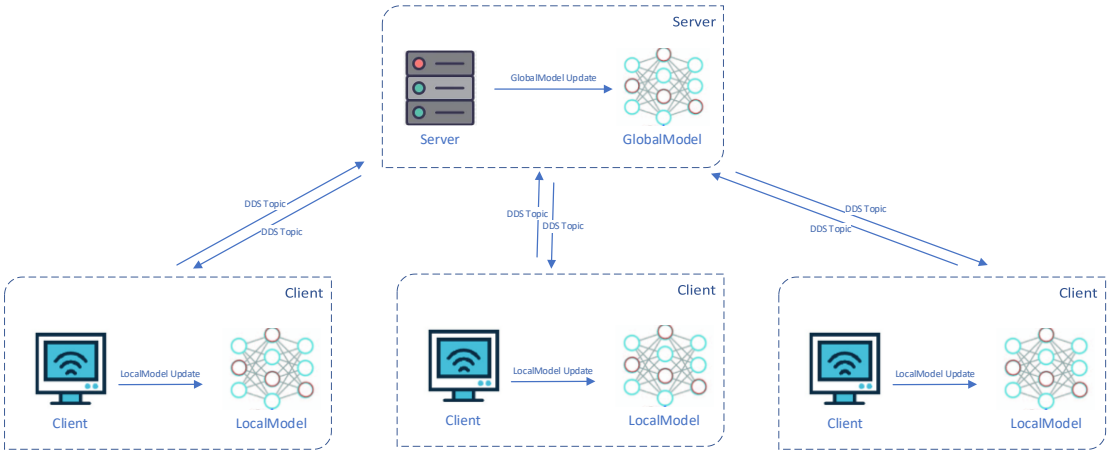

Figure 2. The FL-DDS framework

Figure 2 shows the FL-DDS framework, where DDS TOPIC opens the security function to ensure the security of data transmission. The server -side initialization globalModel widgets are sent to each local client through DDS security TOPIC. Each selected client is trained on local data after receiving the globalModel widgets, and localModel widgets are updated. The updated local client model widgets are uploaded to the central server through the security TOPIC of DDS [11]. After the server receives localModel parameters from multiple clients, the server is weighted average to update the globalModel parameter.

4. Experiment

4.1. Experimental Setup

This paper implement the FL-DDS framework based on the most classic federated learning method FedAvg. This paper conducted experiments on the Cifar10 dataset, which is an image classification dataset commonly used in computer vision research and deep learning model training. It includes 60,000 images, each of which is 32x32 pixels in size and divided into 10 categories. As the basis encoder, we employ a CNN network with two 5x5 convolutional layers.

To implement FL-DDS machine learning training, we employ PyTorch [8]. For local training, using a 0.01 learning rate, the SGD stochastic gradient descent optimizer is used. SGD weight decay is set to 0.00001 and SGD momentum is set at 0.9. There is a batch size of 64. There are ten local epochs in total. There will be one hundred rounds of conversation.

For local data, we randomly and evenly divide the CIFAR-10 dataset into 5 parts to construct independent and identically distributed data partitions. Specifically, the built-in method of PyTorch is used to randomly divide the dataset into 5 non-overlapping new datasets of a given length as local data.

DOI: 10.54254/2755-2721/86/20241552

To develop FL-DDS's model transfer component, we use ROS2. DDS is used by ROS2 for publish-subscribe communications and distributed service discovery amongst its nodes. The most recent development version of ROS is rolling. Additionally, security protections for model transfer are provided by SROS2. For every DDS implementation that supports it, the ROS client library contains a collection of tools and functions called SROS2, which is used to enable DDS security features. We decide to use FastDDS, the default DDS middleware, as the actual DDS implementation out of all the middleware implementations available for ROS2 DDS.

Regarding the studies, they were all carried out on an Ubuntu 20.04 host that included a 32GB RAM, a Micron NAND flash memory (TLC) 512GB disk, and an Intel i7 12700KF processor. Nvidia RTX 3060 was the GPU utilized for processing accelerated by machine learning.

4.2. Accuracy

Figure 3 shows the accuracy of FL-DDS trained on the cifar10 dataset for 100 rounds under 5 local clients.

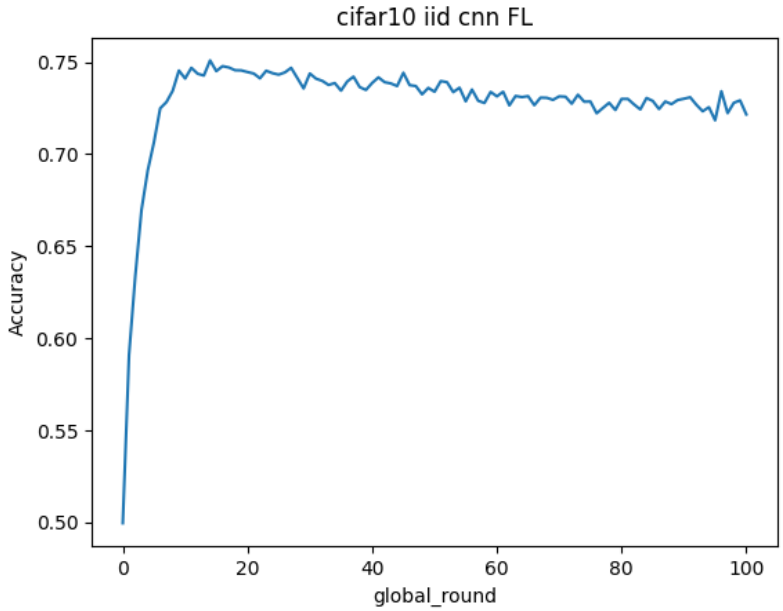

Figure 3. Model Accuracy

In the first few rounds of training, the accuracy increased rapidly, from about 0.50 to more than 0.70. This shows that the model quickly learns effective features in the early stage, and the accuracy is significantly improved. Between 10 and 30 rounds, the accuracy continues to rise slowly and gradually stabilizes. During this period, the model further optimized and adjusted parameters through multiple rounds of local training and global updates, resulting in continuous improvement in performance. After 30 rounds, the accuracy rate stabilizes and remains between 0.70 and 0.75, with small fluctuations. This indicates that the model has reached a good convergence state, and further training will bring limited performance improvement. As can be seen from the Model Accuracy figure, the globalModel reaches a stable convergence state around 30 rounds, indicating that using the FL method to train the CNN model on the CIFAR-10 dataset is effective. The final model accuracy is between 0.70 and 0.75, showing the model's good classification ability on the CIFAR-10 dataset. The experimental results verify the effectiveness and reliability of FL-DDS on distributed data sets. when the data is uniformly and independently disseminated (IID), the model can also achieve higher accuracy faster.

Figure 4 shows the changes in test loss (Test_Loss) of the model under different global rounds (global_round) when using FL-DDS to train the CIFAR-10 data set.

DOI: 10.54254/2755-2721/86/20241552

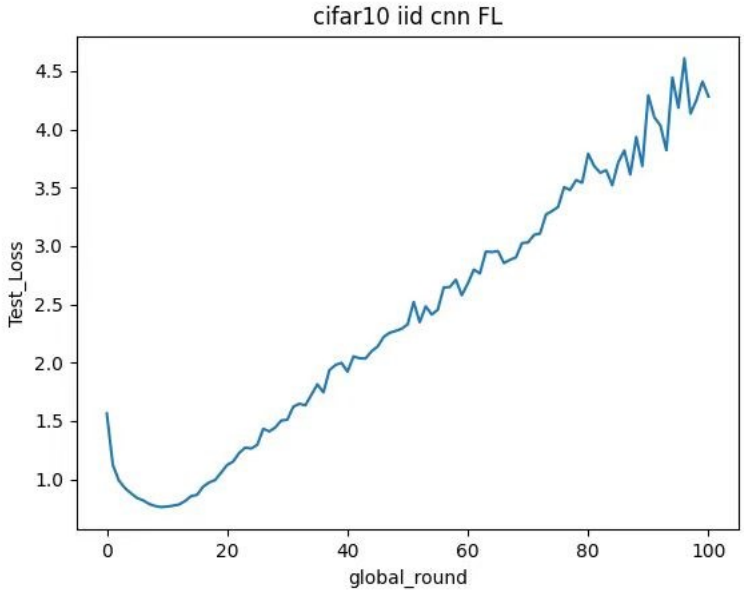

Figure 4. Test Loss

In the first few rounds of training, the test loss drops significantly, from about 1.5 to close to 1.0. This shows that the model can quickly learn effective features in the early stage, thereby significantly reducing the loss. Between rounds 10 and 40, test losses gradually increase. This may be due to the problem of uneven client data distribution or unstable model parameter updates in federated learning. After 40 rounds, the test loss continued to rise, eventually reaching about 4.5. This indicates that the model has problems such as overfitting or too high a learning rate during the later training process, resulting in a decline in model performance.

5. Conclusion

Through experiments on the independent and identically distributed cifar-10 dataset, we evaluated the performance of the FL-DDS algorithm and tested the security overhead of the model transmission. The experimental results show that FL-DDS, as a distributed federated learning framework, ensures the privacy of model transmission through the DDS security component without affecting the communication performance.

In general, the future prospects of FL-DDS are very broad, and future experiments can be improved in many aspects, such is achieving completely decentralized federated learning by utilizing the DDS middleware's decentralized features, and other federated learning frameworks built upon the DDS middleware. FL-DDS may be used to test its viability and usefulness in many domains by applying it to certain situations like smart manufacturing, smart transportation, and smart healthcare.

References

[1] Melis L, Song C, De Cristofaro E, et al. Exploiting unintended feature leakage in collaborative learning. 2019 IEEE symposium on security and privacy (SP). IEEE, 2019: 691-706.

[2] Zhao B, Mopuri K R, Bilen H. idlg: Improved deep leakage from gradients. arXiv preprint arXiv:2001.02610, 2020.

[3] Zhu L, Liu Z, Han S. Deep leakage from gradients. Advances in neural information processing systems, 2019, 32.

[4] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data. Artificial intelligence and statistics. PMLR, 2017: 1273-1282.

DOI: 10.54254/2755-2721/86/20241552

[5] Han L, Fan D, Liu J, et al. Federated learning differential privacy preservation method based on differentiated noise addition. 2023 8th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA). IEEE, 2023: 285-289.
[6] Li Q, Christensen M G. A privacy-preserving asynchronous averaging algorithm based on shamir's secret sharing. 2019 27th European Signal Processing Conference (EUSIPCO). IEEE, 2019: 1-5.
[7] Sun J, Chen T, Giannakis G B, et al. Lazily aggregated quantized gradient innovation for communication-efficient federated learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 44(4): 2031-2044.
[8] Tang Z, Shi S, Li B, et al. Gossipfl: A decentralized federated learning framework with sparsified and adaptive communication. IEEE Transactions on Parallel and Distributed Systems, 2022, 34(3): 909-922.
[9] DDS Security. DDS Security 1.1. 2018. https://www.omg.org/spec/DDS-SECURITY/1.1/PDF.
[10] Open Robotics, C. ROS 2 Documentation — ROS 2 Documentation: Galactic documentation. 2023. https://docs.ros.org/en/galactic/index.html.
[11] Paszke A, Gross S, Massa F, et al. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 2019, 32.