Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.041897

REVIEW

Multi-Robot Privacy-Preserving Algorithms Based on Federated Learning: A Review

Jiansheng Peng[1,2,*], Jinsong Guo[1], Fengbo Bao[1], Chengjun Yang[2], Yong Xu[2] and Yong Qin[2]

[1]College of Automation, Guangxi University of Science and Technology, Liuzhou, 545000, China

[2]Department of Artificial Intelligence and Manufacturing, Hechi University, Hechi, 547000, China

*Corresponding Author: Jiansheng Peng. Email: sheng120410@163.com

Received: 05 May 2023 Accepted: 18 October 2023 Published: 26 December 2023

ABSTRACT

The robotics industry has seen rapid development in recent years due to the Corona Virus Disease 2019. With the development of sensors and smart devices, factories and enterprises have accumulated a large amount of data in their daily production, which creates extremely favorable conditions for robots to perform machine learning. However, in recent years, people's awareness of data privacy has been increasing, leading to the inability to circulate data between different enterprises, resulting in the emergence of data silos. The emergence of federated learning provides a feasible solution to this problem, and the combination of federated learning and multi-robot systems can break down data silos and improve the overall performance of robots. However, as scholars have studied more deeply, they found that federated learning has very limited privacy protection. Therefore, how to protect data privacy from infringement remains an important issue. In this paper, we first give a brief introduction to the current development of multi-robot and federated learning; second, we review three aspects of privacy protection methods commonly used, privacy protection methods for multi-robot, and Other Problems Faced by Multi-robot Systems, focusing on method comparisons and challenges; and finally draw conclusions and predict possible future research directions.

KEYWORDS

Federated learning; multi-robot; privacy protection; gradient leakage attacks

1 Introduction

Since the 21st century, with the rapid development of artificial intelligence and the Internet of Things (IoT), many dangerous or tedious tasks in our life have been gradually replaced by robots. For example, the cleaning of facades of high-rise buildings [1]; mine clearance in war [2]; and hazardous jobs such as underground exploration in coal mines [3], train track flaw detection and internal inspection of industrial equipment [4] are replaced in industry. Due to the wide application of the Internet of Things and sensor devices, robots have collected a large amount of relevant data during their work, dramatically impacting industrial production. With the continuous improvement of deep learning algorithms, the accuracy of robots in some recognition tasks has far surpassed that of humans, especially in image recognition. However, the prerequisite for high-precision image recognition is a

large amount of high-quality training data. And in most industries, competition between different companies, privacy and security, and distrust lead to data silos, and there is tremendous resistance to centralizing data from even different departments of the same enterprise, which seriously affects the enterprise development.

The increased awareness of privacy protection and the introduction of national laws related to privacy protection issues [5] have made collaboration between different data owners difficult. In 2016, Google proposed a new distributed machine learning paradigm called Federated Learning (FL) [6], which provides an excellent solution to the above problem. FL enables knowledge sharing among multiple participants without data interaction. It effectively improves learning efficiency and privacy.

The combination of multi-robot and FL breaks down data silos, and multi-robot can get more data to improve performance. But over time researchers have found that existing FL are vulnerable to malicious behavior by local IoT devices. Current approaches to counter these malicious behaviors include data encryption and information fuzzing. Since robots are limited in computing power, communication capacity and memory. So using a form of data encryption to improve the privacy of multiple robots adds extra communication overhead and affects the real-time nature of robot decision-making. On the other hand, the use of information fuzzing leads to degradation of model performance and affects the performance and accuracy of the robots when performing tasks. Therefore the application of FL in multi-robot systems requires a trade-off between privacy and robot performance and the use of a lightweight FL framework. In addition to this, each robot is in a different location and environment, and it is not possible to obtain completely independent and identically distributed (IID) data. Therefore, the effect of non-independently and identically distributed (Non-IID) data on FL performance should also be considered.

The research directions and privacy-preserving methods in FL are presented in the literature [7] and the number of references cited in the article is visualized, which is mainly aimed at industrial applications. In the literature [8], the authors provided, an overview of FL using the PRISMA process, summarizing the privacy issues and solutions faced by FL, and the article focuses on applications in medicine. Privacy-preserving [9], the authors presented a review of FL in the context of IoT in terms of communication efficiency, privacy protection, and summarized the challenges faced by FL in IoT applications. Privacy-preserving [10], the authors summarize the privacy issues faced by FL, including those specific to FL compared to distributed ML, and summarize the costs associated with different privacy-preserving techniques. Privacy-preserving [11], the authors summarized FL applications in smart cities, including privacy and security aspects. However, these articles did not mention the scenarios to which various privacy-preserving approaches in FL can be applied, and did not analyze the application of FL in robotics, and we also summarized the latest research results in 2022.

2 Background

2.1 Multi-Robot Systems

With the rapid development of human technology, especially with the arrival of the new crown epidemic, many industries are beginning to consider the use of multi-robot to replace manual work. Multi-robot has higher efficiency, greater robustness, and lower cost compared to a single robot (multiple lower-performance robots will collaborate better than a single powerful robot). The main workflows of multi-robot systems are: machine learning, task decomposition, coalition formation, task assignment, decision perception, and motion planning. Task decomposition [12,13], task assignment [14,15], and motion planning [16,17] of multi-robot have been studied more extensively. In the learning phase

individual robots use the acquired data for model training and the data quality directly affects the accuracy of decision perception. The perception phase mainly uses pattern recognition techniques, e.g., image recognition, speech recognition, etc.

2.2 Federated Learning

Federated learning, mainly consists of a server and several clients. In federated learning, clients can optimize their own models with the knowledge learned by other clients without uploading local data to the server, enabling collaborative learning that effectively protects data privacy.

2.2.1 Federated Learning Workflow

We use robot to simulate a client in FL, and the federal learning workflow is shown in Fig. 1:

1) **Robot selection:** When multi-robot are used for FL, the first step is to select the robots that will participate in the learning. Preference will be given to robots with good network conditions and sufficient power, The communication algorithm is then selected according to the specific requirements.
2) **Model broadcast:** The robot downloads the current model weights and a training program from the server.
3) **Local training:** Each robot is trained locally to update parameters for the model.
4) **Model aggregation:** After a robot has trained a local model, it uploads the model parameters to the server, and after enough robots produce training results in a training cycle, it aggregates the model parameters and removes the dropped robots.
5) **Model update:** The server updates the model after parameter aggregation, and the robot downloads the new model to start the next training cycle.

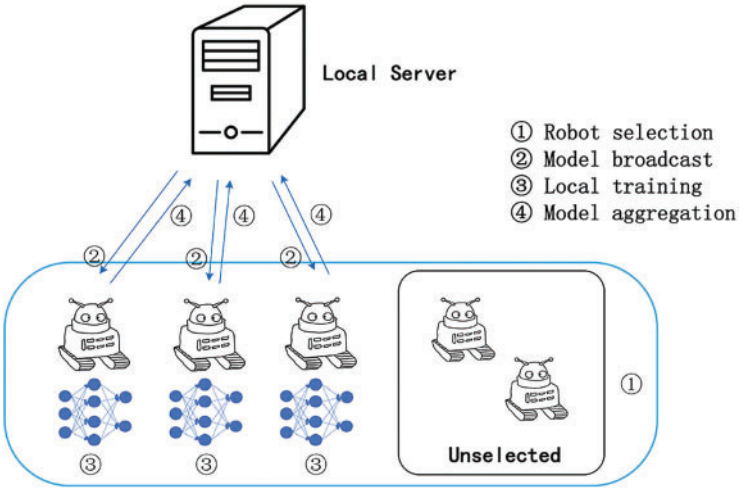

Figure 1: Federated learning (FL) robot workflow

2.2.2 Federated Learning Classification

FL can be classified into Horizontally Federated Learning, Vertically Federated Learning and Federated Transfer Learning [18] according to the different degrees of overlap between the different dimensions of the data, as shown in Fig. 2.

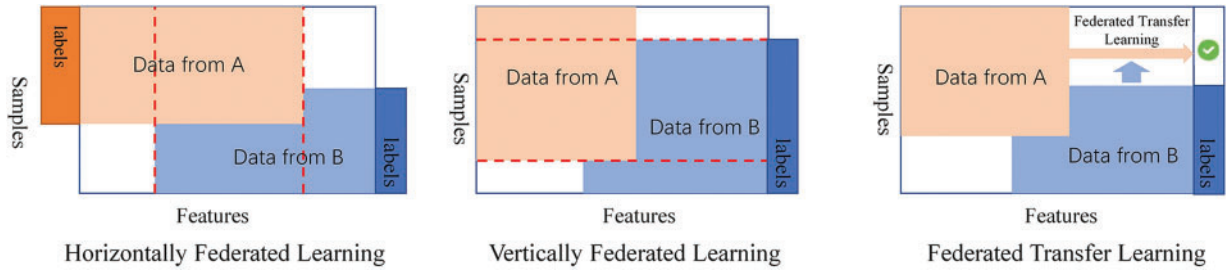

Figure 2: Classification of federated learning (FL)

If there is more feature overlap in the data among the participants, then it is suitable to use Horizontally Federated Learning. Horizontally Federated Learning is often used for joint learning by the same type of companies, which can improve the service quality of Horizontally Federated Learning because similar companies provide similar services. Vertically Federated Learning is mainly used in scenarios where each participant has a high overlap of data samples and a low overlap of features. For example, suppose a bank in a certain region has users' income and expenditure information, and a supermarket has the purchase records of users. In that case, they can collaborate to train a purchase prediction model. Federated Transfer Learning can be used when both feature dimension and sample crossover between data are small. In the case of two firms with low data crossover, for example, Federated Transfer Learning is performed in both firms using a limited common sample set for training a common representation of the feature dimensions in the data of both firms. The model is subsequently applied to the sample prediction of the non-overlapping feature parts of both firms. Federated Transfer Learning is mainly used to cope with the problem of insufficient data and small data volume while ensuring security and privacy for each participant.

3 Common Privacy Protection Methods

3.1 Method Introduction

The main two methods of privacy protection commonly used today are privacy protection through data encryption or fuzzy information. Examples include differential privacy, secure multi-party computing, homomorphic encryption, blockchain, and trusted execution environments.

3.1.1 Differential Privacy

Differential Privacy (DP) is achieved by adding noise to the private data for the purpose of privacy preservation [19,20]. Suppose there is a randomized algorithm M, P denotes probability, S is the set consisting of all outputs, and S_m is any subset of S. For a pair of adjacent data sets D and D', if satisfies:

$$P[M(D) \in S_m] \leq e^{\epsilon} \times P[M(D') \in S_m] + \delta \tag{1}$$

Then the algorithm M provides (ϵ, δ) Differential Privacy. Where the adjacent datasets D and D' differ by only one data. The closer ϵ is to 0, the closer the distributions of D and D' are, the higher the privacy of the algorithm; privacy protection is strongest when $\epsilon = 0$, but the original data will lose availability. δ denotes the probability that the query outputs of adjacent datasets differ by more than a factor e^{ϵ}, and the value is generally small, usually taken as less than the inverse of the dataset size. When $\delta = 0$, Eq. (1) degenerates to $\epsilon-$Differential Privacy.

3.1.2 Secure Multiparty Calculation

Secure Multiparty Computing (SMC) originated from the millionaire's problem proposed by Yao in 1986 [21]. Third parties often require users to upload data to provide services in daily life. The privacy of user data must be guaranteed by a trusted third party, but trusted third parties often do not exist. So an encryption protocol is needed to perform the computation without a trusted third party and without revealing the user's uploaded input. The main current frameworks for SMC are Yao's Obfuscated Circuit, Secret Sharing [22,23], and Unintentional Transmission [24]. The above SMC frameworks have advantages and disadvantages in terms of the number of users, computation and communication efficiency, and the choice must be made according to the actual situation.

3.1.3 Homomorphic Encryption

Data encryption is an important means to protect privacy, but traditional encryption methods do not allow third-party computations to be performed without decryption, so the user's privacy is threatened when the third party is not trusted. Homomorphic encryption (HE) is a special type of encryption [25] that supports third-party computation in an undecrypted form, i.e., the input and output of the third party are in ciphertext form. It is defined as:

$$E(m_1) * E(m_2) = E(m_1 * m_2) \, \forall m_1, m_2 \in M \tag{2}$$

If Eq. (2) is satisfied, it is called a homomorphic operation on the operator "$*$", where m_1, m_2 are the original data and E is the encryption algorithm. Homomorphic encryption can be classified into Partially Homomorphic Encryption (PHE), Somewhat Homomorphic Encryption (SWHE) [26] and Fully Homomorphic Encryption (FHE) [27] according to the type and number of operations supported.

3.1.4 Blockchain

The traditional FL architecture has a server to aggregate model parameters uploaded by clients, and this centralized architecture is not immune to a single point of failure and malicious servers. The combination of blockchain and FL can solve the centralization problem and monitor malicious behavior in model training. The smart contract mechanism unique to blockchain allows models to be aggregated automatically after meeting the conditions without human intervention. The models are stored in the blockchain after aggregation so that even the clients who join later can enjoy the global model. However, some mechanisms of blockchain itself also limit its development on FL, such as limited throughput, encryption and decryption of data, proof of workload, etc., take up a lot of computing resources resulting in lower model training efficiency.

3.1.5 Trusted Execution Environment

A Trusted Execution Environment (TEE) is a standalone execution environment that enables secure storage. The main idea is that a separate secure memory zone can be allocated on an untrusted device for private data protection [28]. TEE has been instantiated in several forms, including Trustzone-M from ARM and Intel's Trusted Execution Environment: the Intel SGX [29].

3.1.6 Group Key

Group Key is a symmetric key that encrypts and decrypts the content of communications between robots in a multi-robot system. Also, Group Key can be used for authentication when new robots join.

It can effectively ensure the data privacy and security of robots and prevent unauthorized robots from entering the system.

3.2 Comparison of Methods

3.2.1 Distributed Learning Privacy Protection

Machine learning has become the primary method for analyzing predictions due to the increase in data volume. Still, a huge amount of data cannot be stored on a single device, and people are reluctant to upload their privacy to a third party. Distributed learning uses data-parallel and model-parallel approaches [30] to improve model training performance and achieve larger data volumes through parallel computing with multiple working nodes. Many researchers have improved privacy for distributed learning. In the literature [31], the authors proposed a selective number of gradients to be uploaded and parameters of the global model to be updated during distributed learning, which protects the privacy of participants' training data while improving model accuracy. However, in this paper, the authors do not consider the Non-IID nature of the data. Due to the enormous communication overhead in peer-to-peer architectures, many scholars have focused on the centralized aggregation of model parameters by servers. The problem of a single point of failure and malicious servers ensues. In the literature [32], the authors proposed a privacy-preserving ADMM distributed machine learning framework that first eliminates the assumption of trusted servers, provides differentiated privacy for users based on the sensitivity of the data and the trustworthiness of the server, and uses added noise to protect privacy, but also has some impact on classification accuracy. In the literature [33], the authors proposed a privacy-preserving framework SPDDL for fog cloud computing. The main idea is to use a threshold signature method to verify the identity of the participants, while the Paillier algorithm is used to encrypt the gradients for secure aggregation. Experiments show that the method is robust to client collusion and withdrawal. In the literature [34], it is proposed to achieve privacy preservation by minimizing the distance correlation between the original data and the segmentation layer activation is minimized, while ensuring the accuracy of the model. In the literature [35], the authors proposed CodedPrivateML, a privacy-preserving distributed training framework that can effectively protect the privacy leakage caused by the collision of multiple clients. It speeds up the training and reduces each worker's computational and communication load, but errors in the quantization process can lead to performance degradation. In the literature [36] a new scheme with anonymous authentication was proposed to address the challenge of privacy-preserving distributed learning.

3.2.2 Federated Learning Privacy Protection

FL has become a popular topic for privacy preservation in distributed learning since Google proposed the FL system. FL privacy and efficiency have been improved compared to distributed learning. However, recent studies have shown that FL systems still have notable privacy leakage potential in the face of malicious attacks. Inference attacks are the leading cause of privacy leakage. When FL trains a model, its data does not leave the local client. Still, a semi-honest or malicious adversary can obtain membership, attribute, and gradient information based on sharing model parameters among clients and uploading model gradients, thus causing privacy leakage. The privacy leakage threat caused by a malicious server or multiple malicious clients colluding can be much larger than that of a single malicious client. Based on this problem, many scholars have conducted in-depth research.

Traditional privacy protection methods still have many drawbacks in practice. For example, it affects the model accuracy, requires huge arithmetic power, etc., and many scholars have improved.

In the literature [37], the authors used HE for privacy protection and an improved Paillier algorithm to improve the training speed of the model. But it is not tested for many client cases and contains a large communication overhead. The authors in [38] proposed a privacy-preserving framework, PPFL, which uses layer-by-layer FL training and always updates layers in the TEE of the server and client, effectively suppressing data reconstruction and attribute inference attacks. Still, the assumption in the paper that each participant has a TEE makes the framework highly limited. Blockchain, like FL, is also a promising technology, and the combination of blockchain and FL can also serve to protect privacy.

In the literature [39], the authors combined FL with blockchain technology to solve the problem of a single point of server failure and malicious servers and introduced a DP method in the framework to improve data privacy. Still, it requires a longer convergence time compared to data concentration learning. Withdrawal due to faults in multi-robot execution tasks is inevitable, and faulty devices can impact the robot's communication. The authors in the literature [40] proposed a fault-tolerant privacy-preserving data aggregation scheme that not only helps users protect their privacy from reverse analysis but also tolerates many faulty devices and saves communication and computation time by exploiting the homomorphism of secret sharing. Resistance to reverse analysis attacks from the normal FL scheme of the server. The literature [41] used an adaptive learning rate algorithm to adjust the gradient descent process, improve the model training efficiency, and use DP to provide privacy protection. However, the convergence efficiency of the model leaves something to be desired, and the literature does not provide a quantitative comparison of privacy-preserving performance. The literature [42] used a chained structure to overcome the problem of high client-server communication overhead in FL and uses eliminable DP for privacy preservation. However, the method is vulnerable to collusion attacks between clients. The literature [43] improved on [42] by improving the communication efficiency by grouping clients based on the chained structure and adding noise to each client to protect privacy. Still, the method is based on the premise of trusted servers. A comparison of each method is shown in Table 1.

Table 1: Comparison of privacy protection methods

Literature	Opponent type	Privacy-protection method	Limitations	Applications
[31]	Semi-honest	DP	Single point of failure	Low demand for privacy
[32]	/	DP	Low classification accuracy	Untrusted servers
[33]	/	Paillier	No quantitative assessment of privacy protection performance	Presence of collusion attacks and device dropouts
[35]	Semi-honest	SMC	Performance degradation	Client-specific collusion and reduced load on each client

(Continued)

Table 1 (continued)

Literature	Opponent type	Privacy-protection method	Limitations	Applications
[36]	/	Anonymous authentication	Only hide the data owner, not protect the data	Identity information is more important than data privacy
[37]	/	HE, Paillier	High communication overhead	Reasoning attacks against members
[38]	Semi-honest	TEE	Each participant does not necessarily have TEE	Gradient attack, attribute inference
[39]	/	Blockchain, DP	Slow convergence of the model	Malicious servers, poisoning attacks
[40]	Dishonesty	HE	High communication overhead	Untrusted servers and client exit
[41]	/	DP	Slow convergence of the model	Untrusted servers
[42]	Semi-honest	DP	Vulnerable to collusion attacks	Non-collusion of participants
[43]	Dishonesty	DP	Trusted server required	Against user collusion

3.3 Challenges

Table 2 summarizes the most commonly used privacy protection methods, advantages and disadvantages and application scenarios. But, we can see that these methods have limitations while protecting participants' privacy:

- DP reduces the accuracy of model training due to the presence of noise, so the trade-off between privacy and accuracy is the main problem that DP currently faces [44].
- Secure Multi-party Computation and HE has high computational and communication costs.
- HE increases the risk of privacy breaches when the same key is used.
- Combining Blockchain with FL can solve the single point of failure problem, but the throughput of blockchain is limited.
- TEE is applied under more demanding conditions, and there is no guarantee that every participant will have the hardware to support them.
- Group keys can protect data privacy though. However, key leakage can lead to data decryption or inclusion of malicious participants, and using the same key is more likely to cause key leakage. At the same time, encryption and decryption operations impose an additional computational burden.

Table 2: Commonly used means of privacy protection

Method	Features	Drawbacks	Application scenarios
DP	Add noise to model information to protect privacy	The difficult trade-off between model performance and privacy	Untrustworthy servers
SMC	Computing decentralization	High computing and communication costs	No trusted third-party function calculation problem
HE	Ciphertext syndication	High computing and communication costs	Untrustworthy servers
Paillier algorithm	Prevent malicious clients from syndicating	More rounds of communication between client and server	Prevent untrustworthy multiple client federation
Blockchain	Scalable and FL decentralized	Limited throughput, taking up a lot of computing resources for model training	Decentralized FL
TEE	Prevent malicious server active attacks	There is no guarantee that all participants will have this hardware support	Scenarios requiring device trust

4 Multi-Robot Privacy Protection

4.1 Problem Introduction

Multi-robot have higher efficiency compared to single robots, which means that multi-robot systems are more suitable for scenarios where data is collected for machine learning, and data privacy issues have come into focus in recent years. Therefore, multi-robot machine learning without data transfer is the main research direction at present.

Multi-robot model training encounters a number of problems, for example, the computational power of the robot is difficult to support the model training, and it is difficult to achieve good training results when the robot only uses local data for model training. Therefore, it is necessary to combine local data from other robots to collaborate on model training. The combination of multi-robot systems and cloud computing can solve the problem of insufficient computing power of robots. Cloud robots move complex computing and storage to the cloud for this purpose. Cloud computing helps improve learning efficiency when multi-robot perform machine learning tasks. Cloud robotics also enables knowledge sharing to enhance robot learning and accomplish more complex tasks. On the other hand, the cost of multi-robot systems can be reduced because the robots do not perform computation and data stored locally. However, the communication of cloud robots can be significantly hindered when multi-robot need to make fast decisions in complex environments [45], and the Privacy of data also limits the development of cloud robots [46].

The combination of multi-robot systems and FL can solve the communication and privacy problems of cloud robots. On the one hand, in FL, the robots do not need to communicate frequently with the cloud, but rather the computation is transferred to the edge server closer to the robots, solving the communication latency problem of multi-robot systems. On the other hand, data in FL can be learned collaboratively by multi-robot without leaving the local area, protecting data privacy. Studies have shown that FL improves performance more than data-centralized multi-robot learning [47].

4.2 Comparison of Methods

4.2.1 Multi-Robot Distributed Learning Privacy Protection

There are few studies on privacy protection in multi-robot communication. In the literature [48], the authors grouped the robots according to different regions and assigned different keys to the robots in different regions. They encrypted them using the Paillier algorithm during model upload. The authors also used two groups of encryption with activation and cost functions and experimentally proved that the method is effective against attacked robots and fog nodes. The authors of the literature [49] proposed a re-encrypted multi-robot FL scheme, where each robot uses a different key, the robot uploads the local data to the server after encryption, the server encrypts again to send the data to the fog node, and the fog node and the server train jointly. This method uploads the local data, which is encrypted but still risky in the face of collusion attacks. In the literature [50], the authors combined secure multi-party computation and auction algorithms to achieve task assignments using secure multi-party computation. Experiments show good privacy protection for semi-honest robots but uncertainty in privacy protection when the number of tasks and robots do not match. The traditional dynamic average consensus algorithm is prone to privacy disclosure when subjected to eavesdropping attacks. The literature [51] proposed a privacy-preserving scheme based on state decomposition. The authors decompose an original state into two sub-states and protect data privacy when interacting and communicating with multi-robot by adding perturbations, which can effectively resist external attacks. The authors of the literature [52] proposed a distributed multi-robot training framework based on the augmented Lagrangian function, which finally reaches a consensus on the local model weights. The method can effectively solve the single point of failure problem and is robust to mid-robot dropouts. Still, the method does not support asynchronous updates of multi-robot and is vulnerable to adversarial attacks.

4.2.2 Multi-Robot Federated Learning Privacy Protection

Several scholars have recently combined FL with multi-robot to preserve privacy. The authors in the literature [53] compared different learning approaches for multi-robot. The literature [54] compared centralized learning with FL, and experiments show that FL has advantages in optimizing communication efficiency and privacy preservation and that the combination of FL with multi-robot helps cross-organizational collaboration and breaks down data silos. Withdrawal due to malfunction is inevitable in the multi-robot execution of tasks, and faulty devices can have an impact on robot communication. The authors in the literature [55] proposed a decentralized, FL mechanism that allows connected robots to use different learning algorithms. The authors of the literature [56] proposed a privacy-enhanced FL scheme in industrial robots for combating collusion attacks. The authors in the literature [57] combined multi-robot auction algorithms with FL, where the default server is trusted in the article, and privacy-preserving bidding, auction computation, and model aggregation processes are performed using partial HE.

The authors of literature [58] proposed decentralized privacy-preserving algorithm. There is no central server in this algorithm, random robots broadcast the learned model, dynamically aggregate into a group based on each robot's reputation, and then start P2P communication. Meanwhile, the model parameters are encrypted by secret sharing. Specifically, each robot in the group obtains a secret slice, and a single secret slice cannot decrypt the model parameters, only a certain number of secret slice can be decrypted. This method effectively solves the single-point-of-failure problem caused by the dependence on the server. However, when multiple robots collude it may have an impact on privacy. Complex FL models cannot be configured on robots due to their limited computational power. Privacy-preserving [59] authors applied a lightweight deep neural network MobileNet on robot. and used the JPQ (Joint Pruning-Quantization) technique to reduce the model storage space and accelerate the inference process. HE is used to encrypt the data during the FL process to protect privacy. Experiments show that the method has good results in robot target recognition. In the literature [60], the authors propose tri-layer FL framework, where model pruning is performed in the first layer, the second layer selects the appropriate client based on the client's current state and history of activity records, and the third layer uses an improved FedAvg to dynamically choose to complete all or part of the work based on the client's resource availability. The authors used 12 mobile robots for the simulation experiments in exchange for faster speed at the cost of losing some accuracy, but with low privacy. The specific privacy-preserving properties are shown in Table 3.

Table 3: Comparison of the effectiveness of Multi-robot privacy protection methods

Literature	Method	Unreliable client	Unreliable server	Participant collusion	External attacks
[48]	HE	√			√
[49]	HE	√	√		
[51]	State decomposition				√
[52]	Decentralization		√		
[56]	HE		√	√	
[57]	HE	√	√		
[58]	Secret sharing	√	√		

4.3 Challenges

Multi-robot systems using traditional centralized learning approaches require uploading local data to a central server, which undoubtedly poses many pitfalls for multi-robot systems. For example, the risk of privacy leakage, the huge communication overhead caused by uploading data, and the lack of personalization of models. In recent years, many scholars have addressed these problems by combining FL with multi-robot systems, but it is clear from the analysis that the application of FL in multi-robot systems still faces a series of problems:

- The study of privacy protection by multi-robot is still in its infancy, and the few articles available leave much to be desired in terms of protecting the privacy of robot data. Robots have limited computing power, and deploying homomorphic encryption algorithms would impose large computational costs on the bot for the bureau, so this may be the reason why more robust privacy-preserving algorithms cannot be deployed on the robot.

- Multi-robot may malfunction and exit the training task while performing the task. Some FL privacy-preserving algorithms require a certain number of robots to operate properly, and robot withdrawal from FL tasks may directly lead to the failure of FL privacy-preserving algorithms. There is less research in dealing with this aspect of unexpected robot withdrawal during operation.
- Studies of FL applications in multi-robot systems are usually specific to robot work scenarios and lack general applicability.
- Multi-robot participation in FL requires a central server to coordinate aggregated model parameter updates, which can also result in privacy leakage of the robot if the server is controlled by an attacker.
- Robots usually have only limited computational power and communication bandwidth, and the application of lightweight FL frameworks in multi-robot systems can reduce computational and communication overheads and improve the efficiency of model updating. However, the model performance will be limited.

5 Other Problems Faced by Multi-Robot Systems

5.1 Problem Introduction

There are some problems with the traditional FL workflow. The server has access to the robot's model updates during model training, intermediate models aggregated by the server in each training round, and the final model. The semi-honest server will follow the communication protocol flow and will not affect the availability and integrity of the model during training but will perform inference attacks on the model parameters as they are received, causing privacy breaches. Malicious servers do not follow the communication protocol and tamper with the sent messages at will to induce other robots to steal more information. Similarly, the presence of malicious robots and the collusion of multiple malicious participants can cause privacy breaches.

FL is vulnerable to attacks by malicious participants, and there are two main types of attacks. One is the attack on model performance, called an "adversarial attack", which is divided into data poisoning, model update poisoning, and model evasion attacks; the other is the "data inference attack", which targets data privacy. The other one is the "data inference attack" for data privacy, which is divided into membership inference attack, attribute inference attack, and gradient attack. As shown in Table 4. To improve the privacy of FL, FL is mostly combined with privacy-preserving algorithms. Due to the specificity of the application scenarios of multi-robot systems, the participation of robots in the improved FL algorithms raises many other issues.

Table 4: Major attacks faced by FL

Malicious behavior	Name	Description
Confrontational attacks	Data poisoning Model update poisoning Model evasion attacks	The main purpose of the malicious device is to affect the FL model performance
Data inference attacks	Member inference attack Attribute Reasoning Attack Gradient Attack	The main purpose of the malicious device is to steal data information from the client

5.2 Federated Learning Privacy Breach Issues

Privacy is the most important property in FL, and in recent years, several scholars have found that privacy protection in FL is not infallible. Inference attacks are the main means of attack against privacy, and there are three main categories of inference attacks: membership inference attacks, attribute inference attacks, and gradient attacks.

Membership inference attacks can be launched when FL participants have access to the model. The original data from the client is obtained by judging whether a certain data sample is present in the training data [61–65]. That is, an attacker can infer an attack by Model parameters for uploading to the server, or by observing the global model in the client. Membership inference attacks can be classified into black-box attacks and white-box attacks in terms of the amount of information obtained about the target model. The black-box attack can only obtain the target model and the output of the model, and this attack mode is the least threatening but the most common; in addition to the input and output of the model, the white-box attack can also obtain information about the structure, parameters, and algorithms of the model, and is highly threatening. Membership inference attacks can be classified into passive membership inference attacks and active membership inference attacks based on whether they actively affect model training.

Attribute inference attacks generally target attributes of training data that are not relevant to the main task in model training [66,67]. Attribute inference attacks can be divided into active attribute inference attacks and passive attribute inference attacks. Active model inference attacks can be used to obtain more information by leading the FL model, but the auxiliary training set desired for attribute inference attacks may sometimes be difficult to obtain.

Gradient attack is an attack method in which attackers reconstruct private data through gradient information. Ongoing research by scholars has shown that openly shared gradient exchanges in FL have great privacy leakage hidden problems, and malicious participants restore users' private information through gradient information with accuracy up to pixel-level original images or matching symbol-level text [68–70]. The attacker reconstructs the client data and the corresponding labels by inferring the gradient information of the training model in each batch.

5.2.1 History of Gradient Attack Development

Information is mainly exchanged in FL by passing gradient messages, and it was once thought that the way of passing gradient messages would not reveal private information. But, the authors of a 2019 paper [68] showed that malicious participants could steal private information such as images and characters through gradient messages. There is a growing awareness of this problem, and in recent years there have been many more studies targeting gradient attacks. We review the history of research on gradient leakage, as shown in Fig. 3.

Deep leakage from gradients (DLG) was first proposed in the literature [68], where a virtual input x' and labeled input y' are randomly initialized, and then these "virtual data" are fed into the model, and "virtual gradients" are obtained. By optimizing the virtual gradient so that it is close to the original gradient, the virtual data and virtual labels can be gradually approximated to the original data. Later, in [71], the authors optimized the DLG and confirm that the label information can be computed analytically from the gradient of the last fully connected layer but it requires the use of a non-negative activation function. The literature [72] confirmed that all networks, are subject to attacks due to gradient leakage and confirms that the reconstruction of the fully connected layer is independent of the network topology. In the literature [73], the authors proposed a recursive procedure to recover data from the gradients of deep neural networks. A novel gradient-based privacy breach

risk level assessment method is also proposed. The authors of the literature [74] proposed a batch tag recovery algorithm that can recover image data in large batches and with high accuracy. In addition, it is demonstrated that single detailed images can be recovered in deep networks. The literature [75] proposed a gradient attack method called GIAS, which is experimentally shown to outperform [72–74] for recovery on image data when given a pre-trained model. The literature also proposed that GIML can be learned by gradient data only when there is no pre-trained model, and the original data recovery is still good For a long time, it was believed that increasing data batches during training could prevent data leakage [71], and in the literature [76], authors proposed a new gradient leakage attack called CAFE, which can recover data at 40 or even larger data batches, and in (Yin et al., 2021) mentioned the limitations of algorithms for recovering data at large batches and developed a CAFE attack for mitigating A defense strategy is developed for mitigating CAFE attacks. The literature [77] summarized the drawbacks of the previously proposed gradient attack methods: poor recovery of large batches of data, low accuracy of label reconstruction when the target batch has a large label repetition rate, and the huge expense of picking hyperparameters. The authors also proposed the E2EGI gradient attack method to achieve sample reconstruction with a batch size of 256, which also has some improvement in algorithm efficiency and duplicate label reconstruction accuracy.

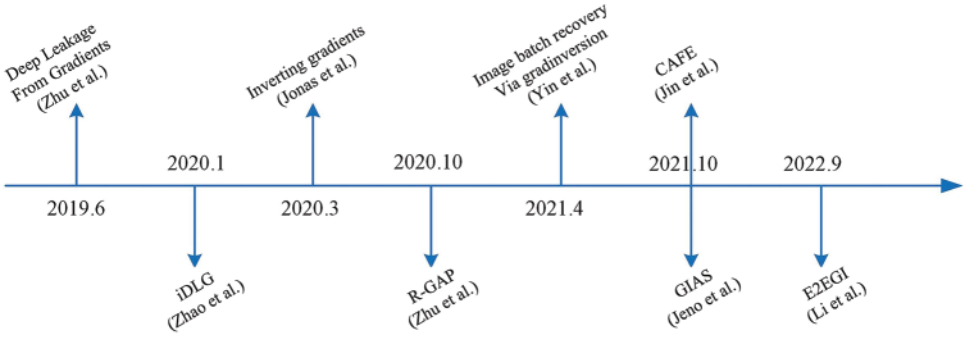

Figure 3: Research history of gradient leakage

As our research progresses we find that increasing the batch size is no longer effective in preventing privacy breaches. The gradient attack algorithm proposed in the literature [78] can obtain individual gradients in arbitrarily large aggregated batches and, crucially, it applies to arbitrary models. The gradient attack algorithm is now able to reconstruct image information with a resolution of 224×224 and steals information with increasing accuracy. So FL needs to be combined with privacy-enhancing algorithms to improve privacy.

5.2.2 Comparison of Methods

The authors in the literature [79] considered a multi-server FL using HE for privacy protection and use the SignSGD idea to decompose the model gradients uploaded by the client and aggregated by multiple server gradients, which can avoid most participants colluding to launch gradient leakage attacks. However, the method uploads too many ciphertexts, the participants have a huge communication burden. Unfavorable real-time multi-robot collaborative sensing and decision-making. The literature [80] grouped participants in FL and used the chaining method to update the model parameters. Each participant's model parameters are protected by the previous participant, reducing the influence of malicious servers on model training. However, it may increase the complexity of multi-robot task management and scheduling.

The literature [37] used an improved HE algorithm to encrypt privacy and uses an enhanced Paillier algorithm to improve computational efficiency, but the method is not resistant to collusion attacks. The literature [70] used DP to encrypt the model gradient, avoids excessive contribution from a single node by compressing the gradient in the model aggregation phase, and uses batch data to add noise in the noise addition phase to reduce the impact on the model accuracy, but still brings inevitable model accuracy degradation. This in turn affects the performance and accuracy of multiple robots when performing tasks. The literature [81] used a decentralized FL that uses DP for encryption when each client communicates with others for model parameters. When the number of robots is large, a large amount of communication bandwidth is used for model parameter sharing and updating. In the literature [82], the authors proposed a privacy-preserving scheme for multi-key HE, where the authors assume that the adversary is honest and curious and encrypts the model parameters using an aggregated public key. This method provides strong privacy protection against the collusion of servers, clients, and multiple participants. But the method has no defense against malicious participants. Many defense mechanisms against malicious participants require the training data to be independently and identically distributed (IID), but this is difficult to achieve in a multi-robot scenario. The authors in the literature [83] extended the classical MPHE approach, where a collective public key and collective key are generated for key generation, the same public key is used for encryption when the model is trained locally, the key is divided into k shares, and the aggregated model is decrypted as long as k users exist when it is decrypted. The literature [84] combined secret sharing with lightweight HE, which is effective in avoiding single-participant or collusion attacks when the participants are semi-honest. The authors use gradient compression to reduce the communication efficiency with a slight loss of convergence rate. The authors in the literature [85] used a HE algorithm to encrypt the model parameters and applied zero-knowledge proofs to determine whether the client is trustworthy. The method was experimentally demonstrated to have excellent resistance to malicious participants. But it will increase the communication burden and algorithmic complexity of the robot. Literature [86] proposed a blockchain-based decentralized FL framework that uses authentication contracts and update contracts to control client authentication and model aggregation while using HE to protect privacy. The server single point of failure problem, data leakage problem are solved. The authors of the literature [87] proposed a decentralized FL framework. Reliable clients are selected by recording the reputation of each client on the blockchain. Each client's contribution is recorded in a nascent block and rewarded. Noise is also added to detect inert clients by using pseudo-noise as a watermark. A comparison of the specific schemes is shown in Table 5.

Table 5: Comparison of gradient leakage defense schemes

Literature	Methods	Decentralization	Server	Client	collusion	Device dropout	Model accuracy
[79]	HE		√	√	√		√
[80]	DP		√	√			√
[37]	HE		√	√		√	√
[41]	DP		√	√	√	√	

(Continued)

Table 5 (continued)

Literature	Methods	Decentralization	Server	Client	collusion	Device dropout	Model accuracy
[42]	DP			√			√
[43]	DP			√	√		√
[81]	DP	√	√	√			
[82]	HE		√	√	√		√
[83]	HE			√		√	√
[84]	HE		√	√	√	√	√
[85]	HE		√	√			√

5.2.3 Challenges

Privacy issues due to gradient leakage have become a key topic of interest for scholars in recent years [88] the integration of FL with privacy-preserving algorithms has become a major means to enhance the privacy of FL, but the fusion of algorithms can have many negative effects on multi-robot systems.

- Multiple robots need to share sensory information with other robots to make quick decisions when collaborating on a task. FL privacy enhancement methods using data encryption add additional communication overhead and affect the real-time nature of the robots' decisions.
- Grouping participants in FL can effectively improve the efficiency of model updating. Grouping the robots participating in FL requires the robots to switch between participating in FL and performing task scheduling. This places extreme demands on multi-robot task management and scheduling.
- Enhancing the privacy of FL using DP leads to a degradation of the model performance. This affects the performance and accuracy when performing tasks with multiple robots.
- Multiple robots participate in decentralized FL, and frequent communication between robots is required to share model parameters. It will increase the communication overhead and latency, which affects the task coordination and decision-making of the multi-robot system.
- In multi-robot systems, due to the differences in location and environment, sensors and task roles of each robot. It leads to the difficulty of obtaining completely independent and identically distributed (IID) data.
- The application of blockchain technology in FL can solve the single-point-of-failure problem, while the immutability of blockchain can record the malicious behavior of robots, but the limited throughput will affect the real-time performance of robots.

5.3 Federated Learning Client Selection Issues

In addition to using privacy-enhancing algorithms, FL can also achieve privacy protection through the selection of clients. On the one hand, the client selection algorithm evaluates the client's reputation, historical participation and security and other indicators to select trusted clients to participate in FL and reduce the risk of privacy leakage. On the other hand, increasing the randomness of client selection can avoid frequent participation of specific clients in FL and reduce the risk of data leakage of individual clients. In addition to this, the FL client selection algorithm can optimize

the efficiency of FL. We divide the client selection purposes into privacy protection and efficiency enhancement. Privacy protection is divided into two client selection methods: fairness selection and privacy assessment. Efficiency enhancement is divided into selection based on client's Self-attribute, data distribution and data similarity. As shown in Table 6.

Table 6: Comparison of client selection methods

Literature	Privacy protection		Improve efficiency		
	Fairness	Privacy assessment	Self-attributes	Data distribution	Data similarity
[89]			√		
[90]	√				
[91]				√	
[92]					√
[93]	√		√		
[94]			√		
[95]		√			
[96]					√

The authors of the literature [89] improved the efficiency of model updates by selecting the clients. Firstly, clients that can respond to server requests are screened. Secondly, these clients are predicted to have the arithmetic power, memory, etc., to complete the task, and finally, clients with fewer data are eliminated. The literature [90] proposed the FairFL client selection algorithm, which has two main components TMGCS and SAP. through SAP protocol clients collect state information from each other and then multiple clients collaborate to decide whether to participate in the local update process (TMGCS). The literature [91] proposed to identify, based on weight scatter comparison, the Non-IID degree of the data on the client side. A small portion of IID data is first collected on the server side as an auxiliary dataset, which is compared by the weight difference with the client, and then the client with a higher IID degree of data is selected to participate in the training. However, the auxiliary dataset of this method can adversely affect the model training when it is tampered with by malicious participants. The authors in the literature [92] exploited the correlation between clients to achieve faster model convergence. The client selection was also performed based on the variability of the clients' contributions in model training and the loss values of the models selected for training out of different clients. The authors in the literature [93] proposed a fine-grained client selection scheme. The server ranks the clients by their model training accuracy and training time and selects the top K clients to participate in the next training round. Clients can optimize their local training to improve the probability of being selected next time. The authors of the literature [94] grouped the clients according to their computational power and communication performance and selected pseudo-servers among the clients, which do not participate in model training and only aggregate the model parameters of the members of this group. A CSFedL client selection mechanism is proposed in the literature [95], where the authors suggest that when there are large differences between models, the crossover clients may be malicious participants and this fraction of clients will be discarded when performing global model aggregation. The authors of the literature [96] evaluated the similarity of probability distributions between client data by The clients with high similarity are grouped and then the best-performing client in each group is selected to participate in the training, which greatly reduces the training time of FL.

6 Future Research Directions

Robots have entered all walks of life, providing convenient services and helping us with dangerous jobs. With the development of sensors and deep learning, the efficiency of robots has gradually surpassed that of humans. Applying FL to the multi-robot has broken the data silos between factory enterprises. Therefore, robots have become more efficient and accurate. In FL, each robot trains its model locally and only shares updates of model parameters. However, The updates of model parameters may contain privacy-sensitive information. An attacker can infer and recover the original data by monitoring and analyzing these updates of model parameters, which leads to privacy leakage. To mitigate the risk of leakage, techniques enhancing privacy can be employed in the update and aggregation process of model parameters. However, research shows that many pressing problems still need to be solved and further studied.

(1) Trade-off between privacy and communication cost when using HE algorithms. Multi-robot systems often face dynamic environments and task requirements, and robots need quick adaptation to environmental and task changes. HE algorithms enhance robot privacy and impose significant communication costs that limit its flexibility in dynamic environments.
(2) Most HE algorithms currently applied in multi-robot systems are HE with the same key. This approach simplifies the key management. However, one cracked key of robots will threaten the data privacy of all other robots. In the future, multi-key HE will be the alternative for multi-robot systems with high privacy requirements.
(3) The trade-off between privacy and model accuracy when using the DP algorithm is that the privacy provided by DP comes at the expense of model accuracy. Meanwhile, the data collected by multiple robots operating in the same field but in different environments is inherently diverse, which makes it challenging to achieve good model performance. Preserving privacy by adding cancelable noise in multi-robot systems may be a significant research direction in the future.
(4) Current privacy-preserving schemes for multi-robot systems typically require that a trusted environment be maintained during computation, which means that the privacy of the robot depends on the trustworthiness of the computing device and risks a privacy breach if the trustworthiness of the environment is compromised.
(5) Collusion between multiple malicious robots, resulting in privacy breaches. Collusion among adversarial robots poses a significant threat to bot-local privacy, highlighting the importance of preventing collusion in privacy protection.
(6) Defining the FL client selection algorithm in a multi-robot system can also achieve privacy protection, and making a fair selection for each robot can avoid frequent participation of specific robots in FL and reduce the risk of individual robot data leakage. However, the current research does not have a good balance between the fairness of client selection and FL performance.

Acknowledgement: The authors extend their heartfelt gratitude for the financial support provided by the Guangxi University Excellent Young Teachers Training Program, the Guangxi University Key Laboratory of Artificial Intelligence and Information Processing (Hechi College), and the Guangxi Zhuang Autonomous Region Department of Education.

Funding Statement: The authors are highly thankful to the National Natural Science Foundation of China (No. 62063006), to the Natural Science Foundation of Guangxi Province

(No. 2023GXNSFAA026025), to the Innovation Fund of Chinese Universities Industry-University-Research (ID: 2021RYC06005), to the Research Project for Young and Middle-Aged Teachers in Guangxi Universities (ID: 2020KY15013), and to the Special Research Project of Hechi University (ID: 2021GCC028).

Author Contributions: Jiansheng Peng wrote the manuscript; Jinsong Guo contributed significantly to analysis and manuscript preparation; Fengbo Bao and Chengjun Yang contributed to the conception of the study; Yong Xu and Yong Qin helped perform the analysis with constructive discussions. All authors read and approved the final manuscript.

Availability of Data and Materials: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. Ollero, M. Tognon, A. Suarez, D. Lee and A. Franchi, "Past, present, and future of aerial robotic manipulators," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 626–645, 2021.

[2] F. Crawford, T. Bechtel, G. Pochanin, P. Falorni, K. Asfar *et al.,* "Demining robots: Overview and mission strategy for landmine identification in the field," in *2021 11th Int. Workshop on Advanced Ground Penetrating Radar (IWAGPR)*, Valletta, Malta, pp. 1–4, 2021.

[3] M. G. Li, H. Zhu, C. Q. Tang, S. Z. You and Y. T. Li, "Coal mine rescue robots: Development, applications and lessons learned," in *Int. Conf. on Autonomous Unmanned Systems*, Singapore, pp. 2127–2138, 2022.

[4] P. Oyekola, "Ferromagnetic climbing robot for industrial flaw detection: An implementation of computer vision," in *IEOM Society Int.*, Singapore, pp. 1971–1985, 2021.

[5] Q. B. Li, Z. Y. Wen, Z. M. Wu, S. X. Hu, N. B. Wang *et al.,* "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3347–3366, 2023.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, USA, pp. 1273–1282, 2017.

[7] L. Li, Y. X. Fan, M. Tse and K. Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, pp. 1–15, 2020.

[8] B. Pfitzner, N. Steckhan and B. Arnrich, "Federated learning in a medical context: A systematic literature review," *ACM Transactions on Internet Technology*, vol. 21, no. 2, pp. 1–31, 2021.

[9] C. Briggs, Z. Fan and P. Andras, "A review of privacy-preserving federated learning for the Internet-of-Things," *Federated Learning Systems*, vol. 965, pp. 21–50, 2021.

[10] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha *et al.,* "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021.

[11] R. Al-Huthaifi, T. R. Li, W. Huang, J. Gu and C. S. Li, "Federated learning in smart cities: Privacy and security survey," *Information Sciences*, vol. 632, pp. 833–857, 2023.

[12] E. B. Gil, G. N. Rodrigues, P. Pelliccione and R. Calinescu, "Mission specification and decomposition for multi-robot systems," *Robotics and Autonomous Systems*, vol. 163, pp. 1–18, 2023.

[13] H. F. Zheng and Y. Wang, "Parallel decomposition and concurrent satisfaction for heterogeneous multi-robot task and motion planning under temporal logic specifications," *Discrete Event Dynamic Systems*, vol. 32, no. 2, pp. 195–230, 2022.

[14] M. Otte, M. J. Kuhlman and D. Sofge, "Auctions for multi-robot task allocation in communication limited environments," *Autonomous Robots*, vol. 44, pp. 547–584, 2020.

[15] B. Park, C. Kang and J. Choi, "Cooperative multi-robot task allocation with teinforcement learning," *Applied Sciences*, vol. 12, no. 1, pp. 272–291, 2021.
[16] C. E. Luis, M. Vukosavljev and A. P. Schoellig, "Online trajectory generation with distributed model predictive control for multi-robot motion planning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 604–611, 2020.
[17] I. Solis, J. Motes, R. Sandström and N. M. Amato, "Representation-optimal multi-robot motion planning using conflict-based search," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4608–4615, 2021.
[18] Q. Yang, Y. Liu, T. J. Chen and Y. X. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.
[19] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, L. Mironov *et al.,* "Deep learning with differential privacy," in *Proc. of the 2016 ACM SIGSAC Conf. on Computer and Communications Security*, Vienna, Austria, pp. 308–318, 2016.
[20] A. El Ouadrhiri and A. Abdelhadi, "Differential privacy for deep and federated learning: A survey," *IEEE Access*, vol. 10, pp. 22359–22380, 2022.
[21] A. C. Yao, "Protocols for secure computations," in *23rd Annu. Symp. on Foundations of Computer Science (SFCS 1982)*, Chicago, USA, pp. 160–164, 1982.
[22] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.
[23] G. R. Blakley, "Safeguarding cryptographic keys," in *Int. Workshop on Managing Requirements Knowledge*, NY, USA, IEEE Computer Society, pp. 313, 1979.
[24] M. O. Rabin, "Transaction protection by beacons," *Journal of Computer and System Sciences*, vol. 27, no. 2, pp. 256–267, 1983.
[25] L. T. Phong, Y. Aono, T. Hayashi, L. Wang and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2018.
[26] A. Acar, H. Aksu, A. S. Uluagac and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–35, 2018.
[27] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. of the Forty-First Annu. ACM Symp. on Theory of Computing*, NY, USA, pp. 169–178, 2009.
[28] F. Mo, H. Haddadi, K. Katevas, E. Marin, D. Perino *et al.,* "PPFL: Privacy-preserving federated learning with trusted execution environments," in *Proc. of the 19th Annu. Int. Conf. on Mobile Systems, Applications, and Services*, NY, USA, pp. 94–108, 2021.
[29] V. Costan and S. Devadas, "Intel SGX explained," *Cryptology ePrint Archive*, 2016. [Online]. Available: https://eprint.iacr.org/2016/086.pdf (accessed on 05/10/2023)
[30] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen *et al.,* "A survey on distributed machine learning," *ACM Computing Surveys (csur)*, vol. 53, no. 2, pp. 1–33, 2020.
[31] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. of the 22nd ACM SIGSAC Conf. on Computer and Communications Security*, Denver, Colorado, USA, pp. 1310–1321, 2015.
[32] X. Wang, H. Ishii, L. Du, P. Cheng and J. Chen, "Privacy-preserving distributed machine learning via local randomization and ADMM perturbation," *IEEE Transactions on Signal Processing*, vol. 68, pp. 4226–4241, 2020.
[33] Y. Li, H. Li, G. Xu, T. Xiang, X. Huang *et al.,* "Toward secure and privacy-preserving distributed deep learning in fog-cloud computing," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11460–11472, 2020.
[34] P. Vepakomma, O. Gupta, A. Dubey and R. Raskar, "Reducing leakage in distributed deep learning for sensitive health data," in *ICLR AI for Social Good Workshop 2019*, Shanghai, China, pp. 1–6, 2019.
[35] J. So, B. Güler and A. S. Avestimehr, "CodedPrivateML: A fast and privacy-preserving framework for distributed machine learning," *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 1, pp. 441–451, 2021.
[36] Y. Jiang, K. Zhang, Y. Qian and L. Zhou, "Anonymous and efficient authentication scheme for privacy-preserving distributed learning," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 2227–2240, 2022.

[37] H. K. Fang and Q. Qian, "Privacy-preserving machine learning with homomorphic encryption and federated learning," *Future Internet*, vol. 13, no. 4, pp. 94–114, 2021.
[38] F. Mo, H. Haddadi, K. Katevas, E. Marin, D. Perino *et al.,* "PPFL: Privacy-preserving federated learning with trusted execution environments," in *Proc. of the 19th Annu. Int. Conf. on Mobile Systems, Applications, and Services*, USA, pp. 94–108, 2021.
[39] Y. H. Qi, M. S. Hossain, J. Nie and X. Li, "Privacy-preserving blockchain-based federated learning for traffic flow prediction," *Future Generation Computer Systems*, vol. 117, pp. 328–337, 2021.
[40] J. Song, W. Wang, T. R. Gadekallu, J. Cao and Y. Liu, "EPPDA: An efficient privacy-preserving data aggregation federated learning scheme," *IEEE Transactions on Network Science and Engineering*, vol. 10, pp. 1–20, 2022.
[41] X. Wu, Y. T. Zhang, M. Y. Shi, P. Li, R. R. Li *et al.,* "An adaptive federated learning scheme with differential privacy-preserving," *Future Generation Computer Systems*, vol. 127, pp. 362–372, 2022.
[42] Y. Li, Y. P. Zhou, A. Jolfaei, D. J. Yu, G. C. Xu *et al.,* "Privacy-preserving federated learning framework based on chained secure multiparty computing," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6178–6186, 2021.
[43] J. R. Cheng, Z. H. Liu, Y. M. Shi, P. Luo and V. S. Sheng, "GrCol-PPFL: User-based group collaborative federated learning privacy protection framework," *Computers, Materials & Continua*, vol. 74, no. 1, pp. 1923–1939, 2023.
[44] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang *et al.,* "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.
[45] N. K. Beigi, B. Partov and S. Farokhi, "Real-time cloud robotics in practical smart city applications," in *2017 IEEE 28th Annual Int. Symp. on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Montreal, Canada, pp. 1–5, 2017.
[46] X. J. Yu, J. P. Queralta, T. Westerlund and J. Heikkonen, "Federated learning in robotic and autonomous systems," *Procedia Computer Science*, vol. 191, pp. 135–142, 2021.
[47] X. J. Yu, J. P. Queralta and T. Westerlund, "Federated learning for vision-based obstacle avoidance in the Internet of Robotic Things," in *2022 Seventh Int. Conf. on Fog and Mobile Edge Computing (FMEC)*, Paris, France, pp. 1–6, 2022.
[48] Y. G. Chen, Y. Ping, Z. L. Zhang, B. C. Wang and S. Y. He, "Privacy-preserving image multi-classification deep learning model in robot system of industrial IoT," *Neural Computing and Applications*, vol. 33, pp. 4677–4694, 2021.
[49] Y. C. Chen, B. C. Wang and Z. L. Zhang, "PDLHR: Privacy-preserving deep learning model with homomorphic re-encryption in robot system," *IEEE Systems Journal*, vol. 16, no. 2, pp. 2032–2043, 2022.
[50] M. Alsayegh, P. Vanegas, A. A. R. Newaz, L. Bobadilla and D. A. Shell, "Privacy-preserving multi-robot task allocation via secure multi-party computation," in *2022 European Control Conf. (ECC)*, London, UK, pp. 1274–1281, 2022.
[51] K. X. Zhang, Z. J. Li , Y. Q. Wang, A. Louati and J. Chen, "Privacy-preserving dynamic average consensus via state decomposition: Case study on multi-robot formation control," *Automatica*, vol. 139, pp. 110182–110191, 2022.
[52] J. Yu, J. A. Vincent and M. Schwager, "DiNNO: Distributed neural network qptimization for multi-robot collaborative learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1896–1903, 2022.
[53] N. Majcherczyk, N. Srishankar and C. Pinciroli, "Flow-FL: Data-driven federated learning for spatio-temporal predictions in multi-robot systems," in *2021 IEEE Int. Conf. on Robotics and Automation (ICRA)*, Xi'an, China, pp. 8836–8842, 2021.
[54] X. J. Yu, J. P. Queralta and T. Westerlund, "Towards lifelong federated learning in autonomous mobile robots with continuous sim-to-real transfer," *Procedia Computer Science*, vol. 210, pp. 86–93, 2022.

[55] J. S. Nair, D. D. Kulkarni, A. Joshi and S. Suresh, "On decentralizing federated reinforcement learning in multi-robot scenarios," in *2022 7th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conf. (SEEDA-CECNSM)*, Ioannina, Greece, pp. 1–8, 2022.
[56] M. Hao, H. W. Li, X. Z. Luo, G. W. Xu, H. M. Yang *et al.,* "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6532–6542, 2020.
[57] Q. Chen, L. Yao, X. Wang, Z. L. Jiang, Y. L. Wu *et al.,* "SecMDGM: Federated learning security mechanism based on multi-dimensional auctions," *Sensors*, vol. 22, no. 23, pp. 9434–9455, 2022.
[58] X. K. Zhou, W. Liang, K. I. Wang, Z. Yan, L. T. Yang *et al.,* "Decentralized P2P federated learning for privacy-preserving and resilient mobile robotic systems," *IEEE Wireless Communications*, vol. 30, no. 2, pp. 82–89, 2023.
[59] B. Xue, Y. He, F. Jing, Y. M. Ren, L. L. Jiao *et al.,* "Robot target recognition using deep federated learning," *International Journal of Intelligent Systems*, vol. 36, no. 12, pp. 7754–7769, 2021.
[60] A. Imteaj and M. H. Amini, "FedPARL: Client activity and resource-oriented lightweight federated learning model for resource-constrained heterogeneous IoT environment," *Frontiers in Communications and Networks*, vol. 2, pp. 1–18, 2021.
[61] R. Shokri, M. Stronati, C. Z. Song and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symp. on Security and Privacy (SP)*, San Jose, CA, USA, pp. 3–18, 2017.
[62] Y. H. Gu, Y. B. Bai and S. B. Xu, "CS-MIA: Membership inference attack based on prediction confidence series in federated learning," *Journal of Information Security and Applications*, vol. 67, pp. 1–15, 2022.
[63] A. Suri, P. Kanani, V. J. Marathe and D. W. Peterson, "Subject membership inference attacks in federated learning, arXiv:2206.03317v1, 2022.
[64] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz *et al.,* "ML-Leaks: Model and data independent membership inference attacks and defenses on machine learning models," in *Network and Distributed Systems Security (NDSS) Symp.*, San Diego, USA, pp. 1–15, 2018.
[65] K. Leino and M. Fredrikson, "Stolen memories: Levering model memorization for calibrated white-box membership inference," in *29th USENIX Security Symp. (USENIX Security 20)*, Boston, MA, USA, pp. 1605–1622, 2020.
[66] M. Shen, H. Wang, B. Zhang, L. H. Zhu, K. Xu *et al.,* "Exploiting unintended property leakage in blockchain-assisted federated learning for intelligent edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2265–2275, 2021.
[67] Z. B. Wang, Y. T. Huang, M. K. Song, L. B. Wu, F. Xue *et al.,* "Poisoning-assisted property inference attack against federated learning," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 4, pp. 3328–3340, 2023.
[68] L. G. Zhu, Z. J. Liu and S. Han, "Deep leakage from gradients," in *Advances in Neural Information Processing Systems*, Vancouver, Canada, pp. 1–11, 2019.
[69] H. C. Ren, J. J. Deng and X. H. Xie, "Grnn: Generative regression neural network—A data leakage attack for federated learning," *ACM Transactions on Intelligent Systems and Technology*, vol. 13, no. 4, pp. 1–24, 2022.
[70] H. M. Gong, L. J. Jiang, X. Y. Liu, Y. Q. Wang, L. Wang *et al.,* "Recover user's private training image data by gradient in federated learning," *Sensors*, vol. 22, no. 19, pp. 7157–7177, 2022.
[71] B. Zhao, K. R. Mopuri and H. Bilen, "IDLG: Improved deep leakage from gradients, arXiv:2001.02610, 2020.
[72] J. Geiping, H. Bauermeister, H. Dröge and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" in *Advances in Neural Information Processing Systems*, Vancouver, Canada, pp. 16937–16947, 2020.
[73] J. Y. Zhu and M. Blaschko, "R-GAP: Recursive gradient attack on privacy," in *ICLR 2021*, Seoul, South Kerea, pp. 1–13, 2020.

[74] H. X. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz *et al.,* "See through gradients: Image batch recovery via gradinversion," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, USA, pp. 16337–16346, 2021.
[75] J. Jeon, K. Lee, S. Oh, J. Kim and J. Ok, "Gradient inversion with generative image prior," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29898–29908, 2021.
[76] X. Jin, P. Y. Chen, C. Y. Hsu, C. M. Yu and T. Y. Chen, "CAFE: Catastrophic data leakage in vertical federated learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 994–1006, 2021.
[77] Z. H. Li, L. Wang, G. Y. Chen, Z. Q. Zhang, M. Shafiq *et al.,* "E2EGI: End-to-End gradient inversion in federated learning," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 2, pp. 756–767, 2023.
[78] Y. X. Wen, J. Geiping, L. Fowl, M. Goldblu and T. Goldstein, "Fishing for user data in large-batch federated learning via gradient magnification, arXiv:2202.00580, 2022.
[79] L. Lin, X. Y. Zhang, W. Shen and W. X. Wang, "FastProtector: An efficient federal learning method supporting gradient privacy preservation," *Journal of Electronics and Information*, vol. 45, no. 4, pp. 1–10, 2023.
[80] Z. Z. Zhang, L. B. Wu, D. B. He, Q. Wang, D. Wu *et al.,* "G-VCFL: Grouped verifiable chained privacy-preserving federated learning," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 4219–4231, 2022.
[81] S. Z. Chen, D. X. Yu, Y. F. Zou, J. G. Yu and X. Z. Cheng, "Decentralized wireless federated learning with differential privacy," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6273–6282, 2022.
[82] J. Ma, S. A. Naas, S. Sigg and X. Lyu, "Privacy-preserving federated learning based on multi-key homomorphic encryption," *International Journal of Intelligent Systems*, vol. 37, no. 9, pp. 5880–5901, 2022.
[83] E. Hosseini and A. Khisti, "Secure aggregation in federated learning via multiparty homomorphic encryption," in *2021 IEEE Globecom Workshops (GC Wkshps)*, Madrid, Spain, pp. 1–6, 2021.
[84] C. Fang, Y. B. Guo, Y. J. Hu, B. W. Ma, L. Feng *et al.,* "Privacy-preserving and communication-efficient federated learning in Internet of Things," *Computers & Security*, vol. 103, pp. 1–14, 2021.
[85] X. Ma, Y. Q. Zhou, L. H. Wang and M. X. Miao, "Privacy-preserving byzantine-robust federated learning," *Computer Standards & Interfaces*, vol. 80, pp. 1–12, 2022.
[86] C. S. Feng, B. Liu, K. Q. Yu, S. K. Goudos and S. H. Wan, "Blockchain-empowered decentralized horizontal federated learning for 5G-enabled UAVs," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3582–3592, 2022.
[87] C. Ma, J. Li, L. Shi, M. Ding, T. T. Wang *et al.,* "When federated learning meets blockchain: A new distributed learning paradigm," *IEEE Computational Intelligence Magazine*, vol. 17, no. 3, pp. 26–33, 2022.
[88] J. P. Zhang, H. Zhu, F. W. Wang, J. Q. Zhao, Q. Xu *et al.,* "Security and privacy threats to federated learning: Issues, methods, and challenges," *Security and Communication Networks*, vol. 2022, pp. 1–24, 2022.
[89] S. Abdulrahman, H. Tout, A. Mourad and C. Talhi, "FedMCCS: Multicriteria client selection model for optimal IoT federated learning," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4723–4735, 2021.
[90] D. Y. Zhang, Z. Kou and D. Wang, "FairFL: A fair federated learning approach to reducing demographic bias in privacy-sensitive classification models," in *2020 IEEE Int. Conf. on Big Data (Big Data)*, Atlanta, GA, USA, pp. 1051–1060, 2020.
[91] W. Y. Zhang, X. M. Wang, P. Zhou, W. W. Wu and X. L. Zhang, "Client selection for federated learning with Non-IID data in mobile edge computing," *IEEE Access*, vol. 9, pp. 24462–24474, 2021.
[92] M. X. Tang, X. F. Ning, Y. T. Wang, J. W. Sun, Y. Wang *et al.,* "FedCor: Correlation-based active client selection strategy for heterogeneous federated learning," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, USA, pp. 10102–10111, 2022.
[93] C. N. Li, X. Zeng, M. Zhang and Z. C. Cao, "PyramidFL: A fine-grained client selection framework for efficient federated learning," in *Proc. of the 28th Annu. Int. Conf. on Mobile Computing And Networking*, Australia, pp. 158–171, 2022.
[94] T. Yin, L. X. Li, W. S. Lin, D. H. Ma and Z. Han, "Grouped federated learning: A decentralized learning framework with low latency for heterogeneous devices," in *2022 IEEE Int. Conf. on Communications Workshops (ICC Workshops)*, Seoul, Korea, pp. 55–60, 2022.

[95] X. L. Xu, S. S. Niu, Z. Wang, D. D. Li, H. Yang *et al.,* "Client selection based weighted federated few-shot learning," *Applied Soft Computing*, vol. 128, pp. 1–13, 2022.

[96] J. Wolfrath, N. Sreekumar, D. Kumar, Y. Wang and A. Chandra, "HACCS: Heterogeneity-Aware clustered client selection for accelerated federated learning," in *2022 IEEE Int. Parallel and Distributed Processing Symp. (IPDPS)*, Lyon, France, pp. 985–995, 2022.