Contents lists available at ScienceDirect

Engineering

journal homepage: www.elsevier.com/locate/eng

Review

Secure Federated Evolutionary Optimization—A Survey

Qiqi Liu [a], Yuping Yan [b], Yaochu Jin [a,*], Xilu Wang [c], Peter Ligeti [b], Guo Yu [d], Xueming Yan [e]

[a] *School of Engineering, Westlake University, Hangzhou 310030, China*
[b] *Faculty of Informatics, Department of Computer Algebra, Eötvös Loránd University, Budapest 1053, Hungary*
[c] *Faculty of Technology, Bielefeld University, Bielefeld 33619, Germany*
[d] *Institute of Intelligent Manufacturing, Nanjing Tech University, Nanjing 211816, China*
[e] *School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou 510006, China*

ARTICLE INFO

Article history:
Received 15 April 2023
Revised 19 August 2023
Accepted 12 October 2023

Keywords:
Federated learning
Privacy-preservation
Security
Evolutionary optimization
Data-driven optimization
Bayesian optimization

ABSTRACT

With the development of edge devices and cloud computing, the question of how to accomplish machine learning and optimization tasks in a privacy-preserving and secure way has attracted increased attention over the past decade. As a privacy-preserving distributed machine learning method, federated learning (FL) has become popular in the last few years. However, the data privacy issue also occurs when solving optimization problems, which has received little attention so far. This survey paper is concerned with privacy-preserving optimization, with a focus on privacy-preserving data-driven evolutionary optimization. It aims to provide a roadmap from secure privacy-preserving learning to secure privacy-preserving optimization by summarizing security mechanisms and privacy-preserving approaches that can be employed in machine learning and optimization. We provide a formal definition of security and privacy in learning, followed by a comprehensive review of FL schemes and cryptographic privacy-preserving techniques. Then, we present ideas on the emerging area of privacy-preserving optimization, ranging from privacy-preserving distributed optimization to privacy-preserving evolutionary optimization and privacy-preserving Bayesian optimization (BO). We further provide a thorough security analysis of BO and evolutionary optimization methods from the perspective of inferring attacks and active attacks. On the basis of the above, an in-depth discussion is given to analyze what FL and distributed optimization strategies can be used for the design of federated optimization and what additional requirements are needed for achieving these strategies. Finally, we conclude the survey by outlining open questions and remaining challenges in federated data-driven optimization. We hope this survey can provide insights into the relationship between FL and federated optimization and will promote research interest in secure federated optimization.

1. Introduction

The huge success of deep learning over the past decade can be partly attributed to the availability of big data. In many cases, data are collected from a large number of edge devices and then sent to a central server to be used for training deep learning models. However, this process may raise serious concerns about the security and privacy of the data. To address these concerns, federated learning (FL) [1,2] has been proposed; it involves training a global model by sharing parameters or gradients of the parameters of the local models trained on local raw data, thereby lifting the requirement on sharing sensitive data. It is notable that the training structure and technical settings of FL obey the General Data Protection Regulation (GDPR) [3].

Although no raw data are transmitted to the server or other local devices in the most widely considered horizontal FL [2,4], malicious attackers may still infer the private information of other clients via the transmitted gradients during the learning process [5]. Therefore, malicious attacks—such as poisoning attacks, inference attacks, backdoor attacks, and many others—may pose security threats to FL schemes [6]. As a result, security and other privacy-preserving computing techniques such as encryption and differential privacy (DP) [7] are usually adopted in FL to more strictly protect the privacy and security of private data. Another vulnerability in the server-client-based FL structure is the security of the communication channels. Eavesdroppers could obtain all messages transmitted in the communication channels and thereby

[*] Corresponding author.
E-mail address: jinyaochu@westlake.edu.cn (Y. Jin).

https://doi.org/10.1016/j.eng.2023.10.006

infer sensitive information. Thus, secure learning algorithms must also establish secure communication. Username-password authentication, hypertext transfer protocol, and transport layer security protocols are standard approaches adopted in FL frameworks to build secure communication and a trusted execution environment [8].

While security and privacy protection have attracted an enormous amount of attention in machine learning, not much attention has been paid to privacy and security in optimization—particularly in data-driven optimization [9,10] or surrogate-assisted optimization [11,12], in which an optimizer relies on data to find the optimum of a black-box optimization problem. Most research on evolutionary computation and Bayesian optimization (BO) operates on the fundamental assumption that all resources for an optimization task reside on a single device. However, when several clients aim to collaboratively optimize a task without sharing their sensitive information, the natural question of whether the existing mechanisms are still applicable arises. Furthermore, given the involvement of multiple clients, the security of information transfer between them must be prioritized.

It is worth mentioning that, unlike in machine learning tasks, the data to be protected in optimization tasks include not only the raw data used for optimization but also the parameter settings in the objective function and the global optima. However, common optimization algorithms such as gradient methods, evolutionary algorithms [13,14], BO [10], and alternating direction method of multipliers (ADMM) [15], where the latter is particularly designed for distributed optimization, seldom consider privacy protection.

1.1. Related reviews

Due to the rapid developments in FL, many reviews have been published that provide an overview of recent advances and applications, from various perspectives and with different focuses. Several reviews have attempted to provide a comprehensive overview of the fundamentals and state of the art of FL [6,16–19]. For example, in Ref. [19], FL schemes are introduced and discussed in detail from the perspectives of data distribution, communication architectures, and privacy-preservation mechanisms. By contrast, other reviews [8,16] have focused more on security and privacy-preservation perspectives. Lyu et al. [6] discussed various vulnerabilities and possible attacks of the basic FL scheme, such as poisoning attacks and inference attacks. In addition, Mothukuri et al. [16] recapped defensive and privacy-preserving and -enhancing techniques targeting the unique security threats in FL. Truong et al. [8] discussed the effectiveness of the privacy-preserving schemes adopted in FL from the perspective of GDPR, concluding that some privacy-preserving FL approaches may not meet the GDPR standard. Recently, Cao et al. [20] summarized recent work on FL from the perspective of Bayesian learning, in which the main focus is on FL; however, they discussed very few studies on federated optimization.

A small number of surveys have been published on topics related to privacy-preserving and secure schemes in optimization, a few of which focus on distributed optimization [21–24]. Yang et al. [22] categorized distributed optimization algorithms based on whether an algorithm is in discrete or continuous time. A survey published by Molzahn et al. [24] focused on the applications of distributed optimization algorithms for electric power systems, which can be grouped into online and offline approaches. Still, no privacy-preserving and security technologies are touched upon in these surveys [22–24]. Weeraddana et al. [21] provided a survey of early ideas on secure distributed optimization with or without a central server node, in which privacy-preserving techniques such as cryptography approaches, transformation-based approaches, and decomposition-based methods are implemented. Recently, Li et al. [23] provided a summary of general mutual information-based information-theoretical metrics, relating existing privacy-preserving techniques in distributed optimization to distributed processing. The researchers also discuss passive and eavesdropping adversary models, providing helpful guidelines for the future design of privacy-preserving and secure distributed optimization algorithms.

However, the distributed optimization discussed in the above-mentioned surveys typically assumes that analytical functions are available for formulating the objectives and constraints, and traditional mathematical programming techniques are employed to solve the problems. Regarding privacy-preserving evolutionary algorithms that can handle problems with no explicit expressions, Zhao et al. [25] provided a brief summary of work on privacy-preserving evolutionary computation based on three typical optimization paradigms.

1.2. Motivation

To date, no survey has been published on secure and privacy-preserving data-driven optimization, including BO [10,26] and data-driven evolutionary optimization [9,27], which rely on data to solve complex optimization problems. In fact, privacy-preserving data-driven optimization, such as federated BO or federated data-driven evolutionary optimization, is a new emerging topic, and only sporadic research results have been reported.

1.3. The scope of this survey

This survey aims to provide a comprehensive overview of the background and recent advances in secure and privacy-preserving data-driven optimization. It introduces various security and privacy-preservation techniques, discusses the relationship between security and privacy preservation, elaborates on similarities and differences in security and privacy-preservation requirements in machine learning and optimization, and outlines future research topics. For a deeper understanding of the research areas in privacy-preserving and secure evolutionary optimization, we first provide a concise overview of distributed optimization and FL, including the challenges encountered in these areas. These three areas share common questions and challenges in the implementation of cryptographic measures or a federated/distributed structure. For example, in distributed optimization, the noise associated with DP diminishes to zero throughout the optimization iterations to ensure convergence. Similarly, in federated or privacy-preserving BO, it is essential to examine the equilibrium between noise levels and algorithmic convergence. We also review privacy-preserving BO in this survey, because many infill sampling criteria developed in BO are widely used in data-driven evolutionary optimization to achieve effective surrogate model management, which is strongly related to secure evolutionary optimization.

We hope that this survey will lay a solid foundation for secure and privacy-preserving evolutionary optimization, raise interest in considering security and privacy preservation in optimization, and promote research in this emerging field.

1.4. Structure

The rest of this survey is organized as follows. Section 2 first introduces the basic definitions and concepts related to this survey, including FL and optimization, security, privacy preservation, and the main privacy-preserving computing techniques. Section 3 presents the fundamentals of FL, including its research history and main research domains in terms of privacy preservation aspects. Section 4 first introduces the difference between learning and optimization, then provides a comprehensive literature review regarding privacy-preserving BO, evolutionary algorithms, distributed optimization, and federated optimization. Section 5 is the main part of the paper, which summarizes the challenges of secure optimization and raises open questions and future directions for researchers. Finally, Section 6 concludes the survey. The main structure of this paper is outlined in Fig. 1.

2. Definitions and terminologies

This section introduces formal definitions of FL and federated optimization, security protection, and privacy preservation. Here, it should be noted that, although security and privacy are strongly related concepts, they have different focuses. We then introduce various privacy-preserving computing techniques, including primitives of DP, homomorphic encryption (HE) [28], multi-party computation (MPC) [29], secret sharing [30], oblivious transfer, and garbled circuits. These schemes are widely used cryptographic building blocks for achieving security and privacy.

2.1. Definitions of FL and optimization

To provide a roadmap from secure FL to secure federated optimization, it is essential to clarify the relationship between machine learning and optimization by introducing the mathematical definition of the objectives in learning and optimization.

2.1.1. Objective in FL

Given a set of training datasets $\mathscr{D} = \mathscr{D}_1, \mathscr{D}_2, \ldots, \mathscr{D}_K$ (in which each $\mathscr{D}_i$ is composed of a pair $(\boldsymbol{z}, l)$, where $\boldsymbol{z}$ and l are the attributes and corresponding label in a learning task, respectively; and K is the number of clients). The objective of FL is to approximate a function $\hat{F}$ from all possible hypotheses to minimize the expected value of loss over dataset $\mathscr{D}$.

$$\hat{F} = \text{argmin}\Theta\left(L(l, F(\boldsymbol{z}))\right) \tag{1}$$

where $L(l, F(\boldsymbol{z}))$ is the loss of $F(\boldsymbol{z})$ to label l, Θ is the expectation, and $F(\boldsymbol{z})$ is the predicted labels of $\boldsymbol{z}$.

2.1.2. Objectives in optimization

An optimization problem is defined as follows.

$$\begin{aligned} &\min \boldsymbol{F} = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x})) \\ &\text{subject to } \boldsymbol{x} = (x_1, x_2, \ldots, x_D), \boldsymbol{x} \in \mathbb{R}^D \end{aligned} \tag{2}$$

where $\boldsymbol{x}$ is a vector of D decision variables in the D-dimensional decision space $\mathbb{R}^D$ in an optimization task, $\mathbb{R}$ denotes one-dimensional decision space, F is the multi-objective objective function composed of M single objective functions, denoted by $f_1, f_2, \ldots, f_M$, and M is the number of objectives. When M = 1, Eq. (2) refers to a single-objective optimization problem; when M > 1, Eq. (2) refers to a multi-objective optimization problem.

2.2. Security protection

In machine learning or FL, security protection refers to the mechanisms that protect the model from active and passive outsider and insider attacks that may deteriorate the model's performance and availability. To ensure system availability, it protects users' data from being stolen or hacked by malicious parties. There are three pillars in information security; these are considered the primary security attributes and are commonly referred to as "the CIA triad." As a guideline for organizations to evaluate their security capabilities and risks, the CIA triad model includes three core security properties: confidentiality, integrity, and availability [31], which are defined as follows:

- **Confidentiality:** Only persons authorized to access information or systems should be given access to the information or system.
- **Integrity:** Only authorized persons or applications can alter the system or information. Unauthorized modification is not allowed.

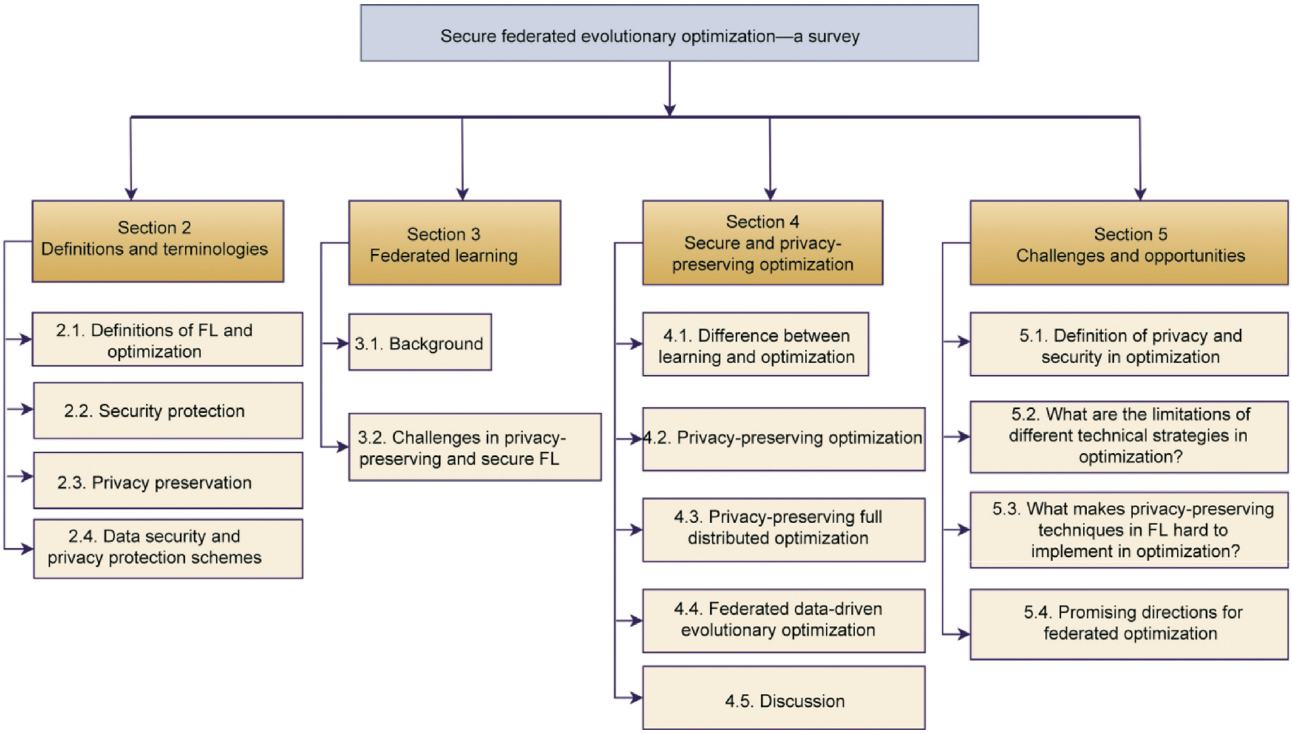
Secure federated evolutionary optimization—a survey
Section 2
Definitions and terminologies
2.1. Definitions of FL and optimization
2.2. Security protection
2.3. Privacy preservation
2.4. Data security and privacy protection schemes
Section 3
Federated learning
3.1. Background
3.2. Challenges in privacy-preserving and secure FL
Section 4
Secure and privacy-preserving optimization
4.1. Difference between learning and optimization
4.2. Privacy-preserving optimization
4.3. Privacy-preserving full distributed optimization
4.4. Federated data-driven evolutionary optimization
4.5. Discussion
Section 5
Challenges and opportunities
5.1. Definition of privacy and security in optimization
5.2. What are the limitations of different technical strategies in optimization?
5.3. What makes privacy-preserving techniques in FL hard to implement in optimization?
5.4. Promising directions for federated optimization

Fig. 1. The structure of this paper.

- **Availability:** The information assets are available to authorized parties when needed.

In information security, another fundamental requirement is to define an adversary model, which formalizes the capability of the participating parties. The participant behavior can be divided into three settings: honest, semi-honest, and malicious, which are defined as follows:

- **Honest:** Honest parties follow the protocol strictly and preserve the data from being leaked.
- **Semi-honest:** A semi-honest participant follows the protocol honestly but may attempt to learn as much as possible from legitimately received messages. This is also known as an honest-but-curious party.
- **Malicious:** A malicious participant can deviate from the protocol in any possible way in order to learn information regarding other parties' input.

2.3. Privacy preservation

In Ref. [18], privacy protection or preservation is defined to be the non-public exposure of sensitive personal information. By "privacy protection" in FL, we refer to the protection of sensitive user information, such as users' local data, gradients, and trained models. To ensure privacy, it is necessary to protect user data from being revealed by third parties without the consent or knowledge of the data owner. Given the importance of data privacy, legal restrictions such as the GDPR have been established to prevent the misuse of sensitive user data.

Most privacy-preservation schemes can be divided into non-cryptographic and cryptographic techniques. A taxonomy of privacy preservation techniques is provided in Fig. 2. In non-cryptographic techniques, researchers use different schemes to protect data privacy. Among these solutions, the most popular schemes are FL, hardware-based trusted execution environment, the non-perturbative masking approach, mainly anonymization, and perturbative masking. Anonymization schemes are expanded into *k*-anonymity [32], *l*-diversity [33], and *t*-closeness [34]. The common method of perturbative masking involves adding noise to sensitive information, where the most classic approach is DP. Cryptographic techniques function powerfully in data protection. In modern cryptography, the techniques can be categorized into normal encryption (e.g., symmetric encryption), asymmetric encryption, identity-based encryption, attribute-based encryption, searchable encryption, HE, MPC, secret sharing, and oblivious transfer, among others. Some of these approaches have been implemented in privacy-preserving FL or optimization. The details of some algorithms can be found in Section 2.4.

2.4. Data security and privacy protection schemes

The basic FL framework is usually insufficient for the strict protection of security and privacy. Therefore, different cryptographic primitives have been proposed and implemented in FL. It should be noted that these cryptographic primitives can also be implemented in privacy-preserving evolutionary/Bayesian optimization and federated optimization to enhance privacy protection and security. To understand these schemes, we introduce in greater detail the most popular cryptographic techniques used in FL, including DP, HE, and secure MPC.

2.4.1. Differential privacy

DP was first proposed by Dwork [7] in 2008. As a perturbation-based privacy-protection method, the core idea of DP is to add noise to ensure statistical privacy for individual information. According to where the noise is added, DP in FL is categorized into global and local DP. Local DP inserts noise into the raw data directly. The server receives the masked data and is not necessarily trusted. However, there is a significant impact on the accuracy, as the addition of noise significantly changes the raw data. By contrast, in global (central) DP, the server aggregates clients' results and masks them before publishing them. In this scenario, the noise compromises a trustworthy server but maintains a certain level of accuracy.

Definition 1. ((ε,δ)-**DP**): A random algorithm $\mathscr{M}$ is (ε,δ)-DP, if for all $\mathscr{S}\subseteq \mathrm{Range}(\mathscr{M})$ and for all, and $\mathscr{D}'$ adjacent datasets:

$$P(\mathscr{M}(\mathscr{D})\in S)\leq \mathrm{e}^{\varepsilon}P(\mathscr{M}(\mathscr{D}')\in S)+\delta \tag{3}$$

where P is the possibility and S represents all possible outcomes of $\mathscr{M}$. The parameter ε is known as the "privacy loss" of information disclosure, while the parameter δ represents the probability of

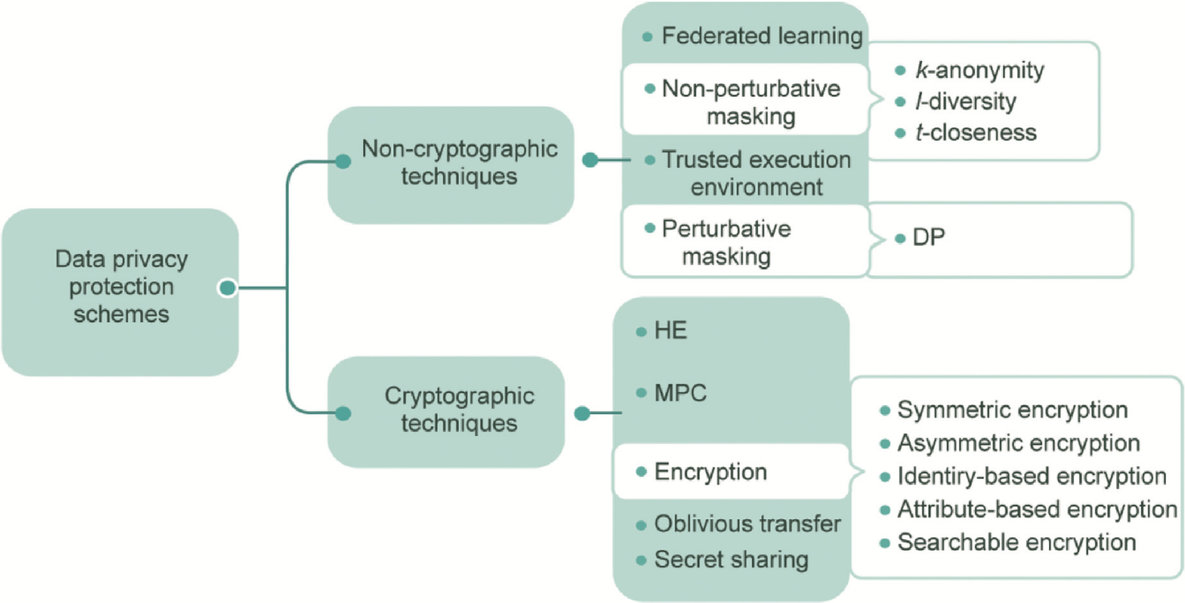

Fig. 2. Data privacy-protection schemes.

information being accidentally leaked. The parameter ε measures the resistance ability of the random algorithm $\mathscr{M}$, where the smaller the parameter ε is, the greater the privacy protection it provides. This is because of the noise generation functions, such as the Laplace noise mechanism. As the noise level is determined by noise $= \mathrm{GS}/\varepsilon$, where GS is the global sensitivity, so the smaller ε is, the bigger the Laplace noise is.

Definition 2. (global sensitivity): Global sensitivity measures the amount of noise to be added to the results. Given a sequence of counting queries $\mathscr{Q}$, global sensitivity measures the maximal change in the result on dataset $\mathscr{D}'$ that removes one record from dataset $\mathscr{D}$. The random algorithm $\mathscr{M}$ satisfies the global sensitivity if the following holds*:*

For $\mathscr{Q}:\mathscr{D}\to\mathscr{R}$, the global sensitivity of $\mathscr{Q}$ is defined as follows*:*

$$\mathrm{GS} = \max_{D,D'} \parallel Q(D) - Q(D') \parallel \tag{4}$$

where $\mathscr{R}$ is real value and Q is the sequence of counting queries.

To generate noise, there are two primary noise generation mechanisms in DP: the Laplace mechanism and the exponential mechanism.

Theorem 1. (the Laplace mechanism): Given an function $\mathscr{Q}:\mathscr{D}\to\mathscr{R}$, for an arbitrary domain *D*, the random algorithm $\mathscr{M}$ provides ε-DP, if $\mathscr{M}$ satisfies the following:

$$\mathscr{M} = Q(D) + (\mathrm{Lap}(\mathrm{GS}/\varepsilon)) \tag{5}$$

where the noise $\mathrm{Lap}(\mathrm{GS}/\varepsilon)$ is drawn from a Laplace distribution.

Theorem 2. (the exponential mechanism): Given a sequence of counting queries $\mathscr{Q}$ and the generation of an entity object $r\in\mathscr{R}$, let $q(D,r)$ be a score function to assign each output r a score. The mechanism $\mathscr{M}$ maintains ε-DP, if:

$$\mathscr{M}(r,q) = \left\{ \text{return}\, r \,\text{with probability} \propto \exp\left(\frac{\varepsilon Q(D,r)}{2\mathrm{GS}}\right) \right\} \tag{6}$$

2.4.2. Homomorphic encryption

In standard encryption methods, a receiver must decrypt the data using private keys and convert it from ciphertext into plaintext before being used. Unlike other encryption methods, HE provides the possibility of conducting algebraic operations directly on the encrypted data without decryption. HE methods are divided into partially, somewhat, and fully homomorphic schemes, based on the operations they can support. If the cryptosystem enables either addition or multiplication operations only, the encryption schemes are referred to as "partially homomorphic;"examples include the ElGamal cryptosystem and the Paillier systems, among others. Rivest et al. [35] were the first to discover that the Rivest–Shamir–Adleman (RSA) public key encryption algorithm is partially homomorphic. By contrast, if the cryptosystems can support both the addition and the multiplication of ciphertexts, they are called "fully HE" schemes. The definition of HE is given as follows:

Let $(G,*)$ and $(H,^\circ)$ be two groups. $f:G\to H$ is a map. If for $\forall a,b\in G$ (where a and b are any values), $f(a*b)=f(a)^\circ f(b)$, then f is said to be a homomorphic map from G to H.

Definition 3. Let E be an encryption algorithm and let $E(c,q)$ denote the encryption of q with cryptographic key c. O denotes an operation on n variables. n denotes the number of input. An encryption $E(.)$ is homomorphic with respect to the operation O:

$$E(c,F(q_1,\ldots,q_n)) = O(E(c,q_1),\ldots,E(c,q_n))) \tag{7}$$

If Eq. (7) is only true for $O(q_1,\ldots,q_n)=\sum_{i=1}^{n}q_i$, then the encryption scheme is an additively HE scheme.

If Eq. (7) is only true for $O(q_1,\ldots,q_n)=\prod_{i=1}^{n}q_i$, then the encryption scheme is a multiplicatively HE scheme.

If Eq. (7) holds for both addition and multiplication, the encryption scheme is called "fully homomorphic." If the encryption only supports addition and a small number of multiplications, it is called "somewhat homomorphic."

2.4.3. Secure MPC

Secure MPC is the collaborative computation of an agreed-upon function by multiple participants without a trusted third party. It is made secure by ensuring certain security properties, such as privacy and correctness, so that each party only obtains its own calculation results and cannot infer the input and output data of any other party from the interaction data during the calculation. MPC was first proposed by Yao [29] in 1986 through Yao's "millionaire problem", which was later extended to secure MPC. MPC is important in digital signatures, electronic auctions, and secret sharing scenarios.

To be more formal, let g be a public function of n variables, where there are n participants with private inputs $v_1,v_2,\ldots,v_n$. The goal of MPC is to compute the common function value $g(v_1,v_2,\ldots,v_n)$ by the participants such that no non-trivial information on the individual inputs can be revealed from the computation and the output. However, MPC is not a single technology; it is a protocol stack composed of a series of different technologies. The details of the composition of MPC are provided in Fig. 3.

As shown in Fig. 3, an MPC structure often consists of two layers: the supporting technology layer and the scheme layer.

(1) The supporting technology layer provides the fundamental technology implementation for building MPC, including standard encryption and decryption, hash function, key exchange, HE, pseudorandom function, and many others. It also contains the essential tools in MPC, such as secret sharing, oblivious transfer, and oblivious pseudorandom.

(2) The scheme layer can be further divided into two main categories: specific schemes and general schemes. They aim to solve different privacy computing problems. Specific algorithms are designed for specific privacy computing logic and are more efficient but can only support a single computing logic; general-purpose frameworks can support most privacy computing logic.

- A general scheme is supported by garbled circuits and can theoretically support any computational task. This is done by compiling the computational logic into a circuit and then obfuscating the execution. However, for complex computational logic, the efficiency of the garbled circuits is reduced to varying degrees, and there can be a significant difference in efficiency compared to specific schemes.
- Specific schemes are constructed to solve specific problems. Due to the targeted construction and optimization, the efficiency of dedicated algorithms is much higher than the general framework, including arithmetic operations, comparison operations, matrix operations, the intersection of privacy sets, privacy data query, DP, and many others.

From a security perspective, MPC should meet three requirements: decentralization, privacy, and correctness, as detailed below.

- **Decentralization:** There is no participation of privileged third parties.
- **Privacy:** The data input of each party in the process of secure MPC is independent, and no local raw data is leaked during the computation.
- **Correctness:** The results obtained by the secure MPC algorithm are consistent with the local computation results of the original plaintext data.

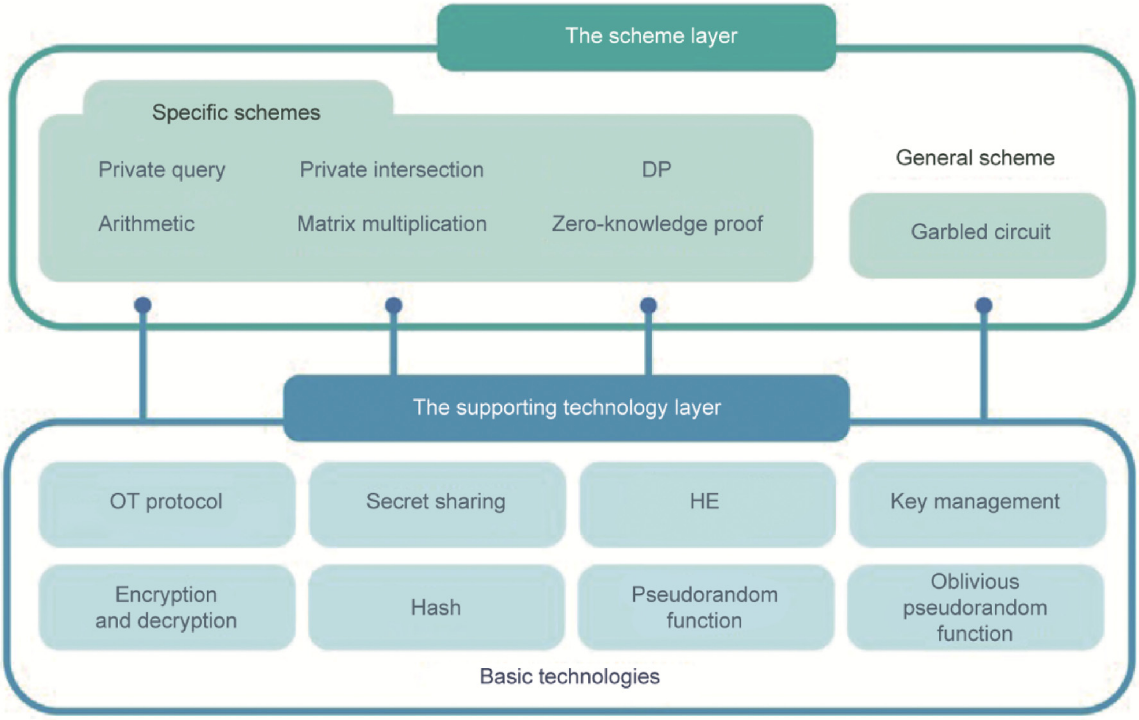

Fig. 3. An illustrative MPC composition structure. OT: oblivious transfer.

3. Federated learning

This section provides a detailed description of FL frameworks, which serve as the fundamentals for secure federated optimization. Federated optimization is an emerging field that integrates the strengths of FL, BO, and evolutionary optimization. Given that federated optimization fundamentally builds on a federated framework and incorporates surrogate modeling within each client, as well as the suggestion of newly infilled solutions, it is crucial to delve into FL before discussing federated optimization.

3.1. Background

FL emerged from privacy-preserving and trustful machine learning. The traditional way to increase training accuracy is centralized learning, in which a trusted third party or data center aggregates data from different participants and performs model training centrally. However, this approach requires a trusted third party, which is not privacy preserving. Moreover, data controlled by different organizations cannot be combined for privacy reasons, which presents barriers to data sharing and collaborations from different domains. This is known as the "data silos" problem.

Several existing approaches to multi-institution distributed training for alternative centrally hosting information have been proposed in the past decade. These solutions include model averaging [36], large-scale synchronous gradient descent [37], cyclical weight transfer [38], FL [1], cyclic institution incremental learning [39], and split learning [40]. Some of these solutions share the basic idea of a distributed learning structure and are very similar to FL.

FL is a distributed machine learning method in which participants train local models and upload the updated model parameters to the server; the server then aggregates them to obtain the parameters of a global model. Compared with traditional machine learning techniques, FL can not only improve learning efficiency but also solve the problem of data silos and protect local data privacy [41]. Since the raw data does not leave the owner's local device, FL is almost the only option for cross-border model training in data-sensitive scenarios such as medical records, personal photo albums, and personal voice recordings.

A typical FL training process is given as follows. The participants in FL include a server, clients, and outside users. The server controls the learning and communicates with the clients. The clients are the data owners; they perform the local training and update the trained model. Outside users are the owner of the model after the models are aggregated and published by the server.

The classical FL training process is described in Fig. 4; it consists of the following steps:

(1) **Initialization:** The server defines the problem to solve and prepares an initial global model. It chooses the clients and sends the initialized parameters of the global model to the clients to start the training process.

(2) **Repetition:** The following steps are repeated until the training process is converged or the maximum accuracy has been achieved:

- **Step 1. Broadcasting:** The server forwards the parameters of the initialized or aggregated global model to the clients;
- **Step 2. Local model training and uploading:** Each client trains the model downloaded from the server using the local data, and the updated model is uploaded to the server;
- **Step 3. Model aggregation:** The server collects the updated local models and aggregates them to obtain an updated global model;
- **Step 4. Repeating until convergence:** Repeat step 1 to step 4 if the convergence condition is not met.

3.2. Challenges in privacy-preserving and secure FL

It has been found that basic FL schemes are not strictly privacy preserving, since they are vulnerable to gradient leakage attacks

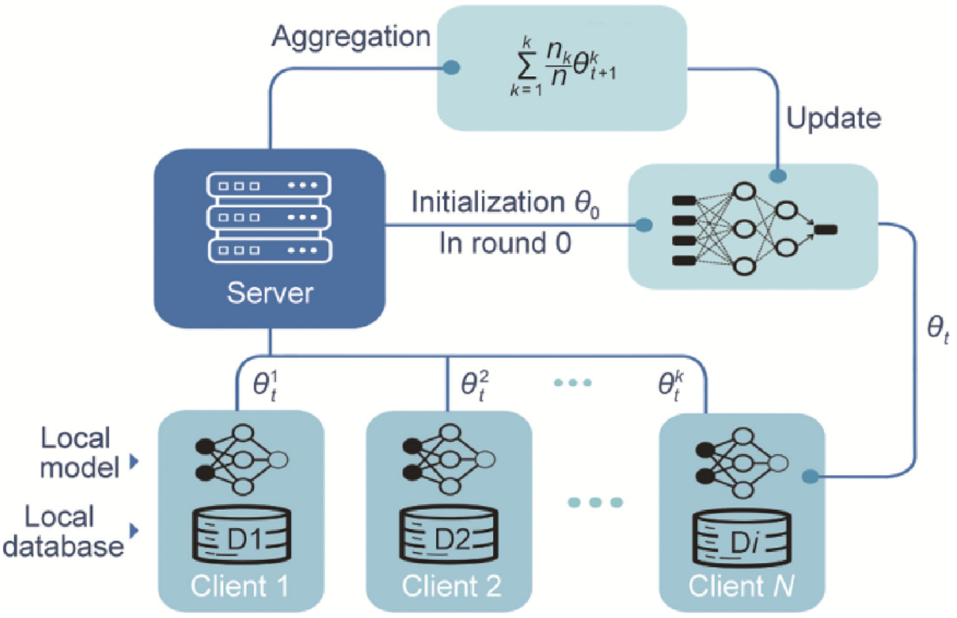

Fig. 4. The FL training process. An FL process includes four main steps: local model training, uploading the local model weights to the server, aggregating the model weights on the server, and downloading the aggregated models. In this plot, there are k participating clients at round t.θ_t^k is the model parameters of client k at round t. θ_t is the aggregated model parameters at round t. N is the number of clients. n_k is the number of training data in client k and n is the number of all training data in all clients. θ_0 is the initial gradient parameter.

[5], inference attacks [42], and data-poisoning attacks [43]. Sensitive information in FL include the raw data of each client and intermediate information in learning, such as the gradients. To defend against attacks and protect this sensitive information, several studies have focused on improving the privacy of FL. Current approaches fall into two main categories: encryption methods, such as secure MPC [44] and HE [35], and data-scrambling methods, such as DP [7]. Encryption methods encode data from plaintexts into ciphertexts by specifying the access control to provide data security. However, such methods often require significant computational overhead and are more challenging to apply in practical scenarios, while data-perturbation methods are relatively lightweight. Randomized noise is added to the data to ensure that an attacker cannot infer sensitive information about an individual based on the differences in output [45]. However, noise reduces the accuracy of the models. Thus, the privacy-robustness tradeoff must be balanced when choosing privacy encounter measurements. The main research directions in privacy-preserving and secure FL focus on techniques with DP [46], HE [47], MPC, secure aggregation, k-anonymity [48], and so on.

Most schemes rely solely on cryptography (represented by MPC and HE) or perturbation techniques (represented by DP). Secure aggregation is an alternative strategy aiming at enhancing data security. For example, Song et al. [49] proposed EPPDA, an efficient fault-tolerant and privacy-preserving data-aggregation FL scheme. Still, none of the abovementioned schemes can fully resist privacy attacks. Encryption effectively hides the client's upload but cannot resist an inference attack on the output side; perturbation ensures that the output model satisfies DP while the upload model remains plaintext. For this reason, Truex et al. [50] proposed an FL scheme combining HE and DP, in which the client perturbs the data locally and then aggregates the perturbed data via Paillier encryption. Similarly, Xu et al. [51] combined function encryption and DP and proposed the Hybrid Alpha FL scheme. However, this scheme only supports linear function operations such as summation, and the paper does not provide rigorous proof that the definition of DP is satisfied. Zhu et al. [52] combined HE and DP in training XGBoost decision trees in vertical FL. In their work, labels are assumed to be distributed on different clients, which is more realistic. Similarly, Lian et al. [53] introduced a decentralized, efficient, and privacy-enhanced federated edge learning system (DEEP-FEL) framework, which integrates DP with a ring-based signature scheme to offer an efficient and privacy-enhanced federated edge learning solution for healthcare cyber-physical systems.

4. Secure and privacy-preserving optimization

Following the discussion on secure and privacy-preserving FL in the previous section, this section introduces the concepts and related work on privacy-preserving and secure optimization. First, we point out the differences between learning and optimization in terms of their objectives and methodologies. Then, we cover related work on privacy-preserving evolutionary optimization, privacy-preserving BO, privacy-preserving fully distributed optimization, and secure federated optimization to provide a relatively complete picture of the field. This section ends with a discussion on interesting points, such as the unique privacy-preserving techniques needed for optimization, what strategies in FL can be directly applied to federated optimization, and the additional requirements for secure federated optimization. These questions can provide inspiration for the development of secure and privacy-preserving optimization techniques.

4.1. Differences between learning and optimization

The main differences between learning and optimization can be discussed in terms of their final targets and applied strategies. Table 1 [15,54–61] elaborates the differences between FL, distributed optimization, evolutionary optimization, and BO in terms of problem assumptions, sensitive information in tasks, and classic algorithms.

4.1.1. Aims

As given in the problem definitions in Eqs. (1) and (2), the objectives of a learning problem differ from those of an optimization

Table 1
The differences between FL, distributed optimization, evolutionary optimization, and BO.

Task type	Aim	Method	Sensitive information	Problem assumption	Classic algorithms examples
A learning task	Prediction classification	FL	• Training/validation data • Gradients	The loss function is a sum of local loss functions	SGD [54], Adam [55] (loss function optimizer for optimizing model weights)
An optimizationtask	Obtain global optima	Distributed optimization	• Initial state values • Local objective functions • (Sub-)gradients	The global function is a sum of local objective functions Objective functions are usually convex and differentiable	ADMM [15], PDMM [56] (common optimizers in distributed optimization)
		Evolutionary algorithms	• All evaluated solutions • Global best optima	No explicit expression is required	GA [57], NSGA-II [58], RVEA [59] (inspired by biological evolution, including operators like reproduction, recombination and environmental selection)
		Bayesian optimization	• Local training data • Next query points • Global best optima	No explicit expression is required	GP-UCB [60], MES [61] (three core components: model construction, model management strategy and an optimizer)

SGD: stochastic gradient descent; PDMM: primal–dual method of multipliers; GA: genetic algorithm; NSGA-II: non-dominated sorting genetic algorithm II; RVEA: reference vector guided evolutionary algorithm; GP-UCB: Gaussian process-upper confidence bound; MES: max-value entropy search.

problem. In general, machine learning is a process in which a model is trained on the available training data and updates its parameters to accurately predict outputs for given inputs. In contrast, optimization targets aim to achieve solutions that minimize or maximize given objective functions. In fact, a machine learning problem aims to minimize a loss function (typically prediction accuracy, among others) based on the training data. Optimization problems may also include a variety of engineering optimization problems [62] or decision-making problems [63,64]. It should be noted that a model is involved in handling both machine learning and optimization tasks. However, the models in these different tasks serve different purposes. In machine learning, the model is trained to minimize the loss function, which is also an optimization task. In solving a data-driven optimization task, a surrogate model is typically also trained to approximate the objective functions, usually for expensive black-box optimization problems.

4.1.2. Solution strategies

The solution strategies in learning and optimization tasks are different. In machine learning, gradient-based methods are often used as the optimizer, since the loss function is usually derivative. However, in handling optimization tasks, it can be the case that there are no analytic mathematical expressions of the objective function, or there are multiple conflicting objectives. Thus, heuristic methods such as evolutionary algorithms are often adopted.

The strategies for solving machine learning tasks are often known as learning algorithms; in general, learning algorithms can be divided into supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, there are two main tasks: a regression task and a classification task. In unsupervised learning, there are two main research lines: dimension reduction, such as principal component analysis [65] and *t*-distributed stochastic neighbor embedding [66], and clustering [67,68]. Both supervised and unsupervised learning tasks are commonly seen in various applications, such as image recognition [69], speech recognition [70,71], and natural language processing [72]. Over the past decade, many new variants of machine learning algorithms, such as semi-supervised learning [73], transfer learning [74], and self-supervised learning [75], have been proposed. Regardless of which learning task it is, gradient-based methods are the most widely used for minimizing a loss function in machine learning; such methods are usually conjugated with heuristic derivative-free optimization methods, such as coordinated descent for solving learning tasks such as variational inference [76].

The strategies for solving learning and optimization problems may also differ. For an optimization task, the strategies for optimization heavily depend on the nature of the objective functions and the number of objectives. For example, traditional mathematical programming methods such as gradient-based methods can be adopted for solving single-objective problems whose objective function is differentiable, whereas meta-heuristic approaches such as evolutionary algorithms [77] and simulated annealing [78], or model-based algorithms such as BO [12], have been proven to be more effective for solving non-differentiable or black-box optimization problems. For the multi-objective optimization problem in Eq. (1), the objectives often conflict with each other, resulting in a set of Pareto optimal solutions (called a "Pareto set"), instead of a single optimal solution, which achieves the best possible tradeoffs among the objectives. The mapping of a Pareto set in the objective space is called a "Pareto front." Evolutionary algorithms are particularly well-suited for solving multi-objective optimization problems, as they can obtain a set of optimal solutions in a single run due to their population-based nature. The existing multi-objective evolutionary algorithms can be roughly categorized into four groups: dominance-based [58,79], decomposition-based [59,80–82], performance-indicator-based [83–85], and preference-based approaches [86]. Most multi-objective evolutionary algorithms aim at well-converged and well-distributed approximations of the whole Pareto front/Pareto set. However, in real-world applications, decision-makers are interested in a preferred subset of the whole Pareto front. Hence, preference-based multi-objective evolutionary algorithms incorporate decision-makers' preferences to guide the search toward the region of interest, in which the preference can be involved before (a priori) [87], after (a posteriori) [80], or during (interactive) [88] the optimization.

In general, multi-objective evolutionary algorithms require a large number of objective function evaluations to generate satisfying approximations to Pareto fronts, which is unaffordable for computationally expensive multi-objective optimization problems such as aerodynamic design optimization and structural optimization [11]. Surrogate-assisted evolutionary algorithms have emerged as a powerful solution to overcome the computational obstacle of applying multi-objective evolutionary algorithms to computationally expensive multi-objective optimization problems [89]. Among various surrogate-assisted evolutionary algorithms, BO has shown promising performance in many applications due to its sample efficiency [26]. Typically, BO constructs a Gaussian process (GP) that defines a distribution over the objective function. Then, conditioned on the observed data and the prior, the posterior can be calculated using the Bayes rule, which quantifies the updated belief about the unknown objective function. As a result, the next sample can be identified by leveraging the posterior, which is achieved by optimizing acquisition functions.

4.1.3. Differences between secure FL and federated optimization

Due to the close relationship between learning and optimization, especially data-driven optimization, it is very natural to extend ideas in FL to federated optimization. Due to the differences in their aims and strategies, the requirements for security and privacy preservation are also different. FL differs from federated optimization in terms of its aim, sensitive information to protect, model framework, and strategy for obtaining optimal learning/optimization performance.

- **Aim:** FL aims to approximate a function $\hat{F}$ from all possible hypotheses to minimize the expected value of loss over the whole dataset composed of the attributes (z) and the corresponding labels (l). Federated optimization aims to find the global optima—that is, the best decision variable $\boldsymbol{X}$, of an optimization problem by fitting the objective function of the problem using the dataset composed of the individuals $(\boldsymbol{X}, \boldsymbol{y})$ that have been evaluated using real function evaluations. Each party in FL and federated optimization only holds a portion of the whole dataset.
- **Sensitive information:** In secure FL, the local training data require protection. By contrast, in federated optimization, not only the local data collected on each client before optimization but also the final optimization results—that is, the best global optima—require protection.
- **Model frameworks:** In classical FL, the server conducts secure aggregation of model parameters to obtain the global model. In federated optimization, however, the server (or trusted clients) must also implement a surrogate-model management process, such as the optimization of an acquisition function in BO, while the classical averaging method in FL no longer works; newly sampled data during the optimization must also be protected.
- **Strategy:** Given the distinct objectives of FL and federated optimization, the approaches to achieving these aims differ significantly. In FL, the primary goal is to develop a global model tailored for effectively fitting datasets from all clients. In essence, a model that yields more accurate predictions is strongly preferred. However, in federated optimization, there is an additional emphasis on model management strategies alongside model fitting, since the final aim is to effectively find the global optima of an optimization problem.

4.2. Privacy-preserving optimization

This subsection discusses privacy-preserving optimization, including evolutionary optimization and BO. Before introducing the details of these privacy-preserving schemes, we first discuss attacks and defenses in BO and evolutionary optimization. Only by understanding the security issues in the optimization algorithm can we better provide security protections.

4.2.1. Attacks and defenses in optimization

Attacks and defenses in optimization can be divided into inference attacks and active attacks.

(1) An inference attack aims to infer sensitive information, such as whether a specific data point is involved in the training data or what the sensitive optimization results are. In existing secure scalar product protocols, such as the protocols in Ref. [90], the vectors in one party can be easily inferred by using a vector containing only one element whose value is one. To defend against such probing attacks, in Ref. [91], the scalar product of two binary vectors is expressed as the sum of two component values using the Du-Atallah private scalar product protocol [92], where each vector is privately held by each party. In Ref. [93], when solving the distributed graph coloring problem using a Tabu search, there is a potential inference risk of guessing the vertex color of another party by comparing the difference of the total number of conflicts between one solution and its neighboring solution. Hong et al. [93] proposed a mechanism called the synchronous move to overcome this issue by choosing to possibly change the color of one of the vertices of another party.

(2) In an active attack, it is mainly necessary to consider the case of conducting optimization with malicious adversaries. Considering that the final returned input point after conducting BO may be perturbed by an adversary, Bogunovic et al. [94] proposed a robust GP assisted algorithm by seeking to obtain the optimal ε-stable solution via a confidence-bound-based acquisition function. As a result, even though the final returned point is perturbed, the obtained solution is still applicable, since the obtained point is located in a relatively wider region. In Ref. [94], it is assumed that only the final input result adversary is reachable to an adversary. However, in fact, an adversary can access any intermediate results, such as corrupting each sample in terms of the input or output space at each step of an optimization process, as discussed in Ref. [95]. For example, in Ref. [96], the output—that is, the reward in the multi-arm bandit setting—is corrupted by an adversary during optimization, posing considerable challenges under an infinite action space—that is, the input space. To overcome this problem, Bogunovic et al. [96] proposed the inclusion of the corruption value in the weighting parameter in the upper confidence bound function under the known corruption setting and attempted to strike a balance between a fast or slow shrinkage of uncertainty under the unknown corruption setting. The abovementioned work discusses optimization under adversarial attacks from the perspective of algorithm designers; however, very recently, Han and Scarlett [97] were the first to discuss optimization from the perspective of the attackers and proposed several targeted attack strategies, such as the subtraction attack and the clipping attack, regardless of whether the objective function is known or not. The effectiveness of the proposed attacks was demonstrated by being applied to the GP bandit algorithms mentioned above, such as those in Ref. [96].

Surprisingly, only a handful of studies conduct optimization with adversaries. This could be because most BO or evolutionary algorithms are currently conducted in one device. However, with the development of federated BO and federated data-driven evolutionary optimization, there will be a surge of interest in conducting optimization with malicious adversaries.

4.2.2. Privacy-preserving evolutionary optimization

In privacy-preserving FL methods, it is not only the final result but also the intermediate results during the training process that can reveal users' privacy. Similarly, in evolutionary optimization, the course of the optimization process in the service provider can disclose private information such as the raw data, real objective functions, and current best optima. Hence, we group privacy-preserving evolutionary optimization into two categories by considering whether or not the whole optimization process is protected.

In the first group, only the fitness calculation is secured. For example, in Ref. [98], the user attempts to protect the real fitness values from being disclosed by just sending the relative ranking between every two fitness values to the service provider for conducting the optimization process. However, the disclosure of ranking in the optimization will inevitably disclose the best global optima, which is usually unacceptable, as the target of any optimization is to obtain the final global optima. In Ref. [98], only one user is involved, meaning that the decision variables and objective function calculation are only held by one user. However, it may be the case that the optimization of one objective function involves interaction between multiple users. In Ref. [91], a secure fitness evaluation protocol is proposed to securely calculate the fitness values of a drug rule based on private arbitrarily partitioned data held by two parties. The protocol is demonstrated to be able to prevent probing attacks. Another scenario, called the master–slave framework, involves multiple agents and one server. To protect privacy under this scenario, Zhao et al. [99] applied the Paillier cryptosystem to a distributed particle swarm optimization algorithm, with each slave client holding one particle, to protect the position of the current global best solution from being exposed to a master server.

In the second group, driven by the fact that most privacy-preserving approaches only secure the calculation of the fitness evaluation, which may reveal sensitive information, an intuitive idea is to secure the whole optimization process. However, for different optimization algorithms, the optimization procedures are quite different. For example, genetic algorithms involve reproduction operators, objective value calculation, and environmental selection, while particle swarm optimization updates the velocity and positions of particles based on the local best and global best optima. It can also easily be found that, apart from the algorithm type, different kinds of problems may require different primitive protocols for encrypting the optimization process. For example, in Ref. [100], the calculation of the objective value only involves addition and multiplication in solving a linear programming problem, such as a subset cover problem, so encrypting the whole optimization process requires addition, multiplication, and comparison primitive protocols. Another work [101] proposes solving the subset cover problem in an outsourcing manner by utilizing a masked bloom filter and Diffie–Hellman data exchange protocol. In Ref. [102], two parties—the supplier and the producer—collaboratively optimize their production planning to minimize their costs separately.

To ensure that no intermediate results are revealed during the optimization process, every step of a multi-objective evolutionary algorithm is secured by cryptography protocols. To be specific, Yao's protocol [29] is for the security of calculating the objective function, HE for the environmental selection, and secret sharing for individual mutation in the population. Since the environmental selection is based on Pareto non-dominated sorting, a specific primitive protocol for sorting is designed in this work for handling a multi-objective optimization problem. Very recently, Zhao et al. [103] proposed a framework only for handling combinatorial optimization problems. The main idea is to outsource the encrypted problem to the cloud server. It is notable that, in combinatorial optimization problems such as traveling salesman problems, the sensitive content owned by users is the city list and traveling cost between each city pair. Sending the encrypted content obtained by the hash function to the server means that the optimization must be conducted based on the encrypted data, which poses great challenges to traditional evolutionary algorithms. Thus, a secure comparison protocol and a secure division protocol are proposed. Similarly, in Ref. [104], the optimization of the double digest problem is outsourced to a cloud server to utilize its powerful computational resources. When an order-preserving homomorphic index scheme is proposed, the optimization procedure of the quantum-inspired genetic algorithm, which is conducted on the cloud, does not need any decryption process with encrypted double-digest data as the input. Experimental results show that this approach is very effective and can also protect the valuable double-digest data of the user.

In addition to privacy-preserving evolutionary algorithms, other privacy-preserving heuristic algorithms such as Tabu search and simulated annealing have been studied over the past two decades, especially in handling distributed combinatorial optimization problems and linear programming problems. For example, in Refs. [93,105], simulated annealing and Tabu search algorithms are applied to handle the traveling salesman and distributed graph coloring problem, respectively, with the help of HE and a secure comparison protocol to secure the calculation of objective values and fitness comparison between one solution and its neighbors.

Apart from single-/multi-objective problems and combinatorial optimization problems, some studies aim to solve distributed linear programming and quadratic programming problems. To efficiently handle such problems, the researchers proposed using transformation-based and DP-based methods to protect users' privacy and solve the distributed problems. Under distributed scenarios, the objective function and constraints of the linear programming problem are distributed on different clients. A situation can occur in which each client holds only parts of the constraints via horizontal partitioning, vertical partitioning, or arbitrary partitioning. Most approaches consider the horizontal partitioning case. In transformation-based methods, such as those in Refs. [106,107] for linear programming problems and that in Ref. [108] for quadratic programming problems, the main idea is to protect the problem parameters and decision variables of each user from being disclosed to the server by transforming the problem parameters. Hong and Vaidya [106] proposed solving horizontally partitioned linear programming problems with an arbitrary number of constraints, regardless of equality or inequality constraints, by letting every user generate artificial constraints and applying a random matrix to transform the problem parameters. Aside from transformation-based approaches, DP-based methods such as that in Ref. [109] are also studied because DP can provide rigorous proof. However, unlike cryptography or transformation methods, DP-based methods suffer from some extent of performance degradation.

In summary, the general privacy-preserving approaches applied in FL, such as HE, have also been adopted in privacy-preserving heuristic algorithms for the same purpose. Cryptography methods such as HE will inevitably be computationally demanding, as each component of the optimization process must be encrypted. Moreover, an optimization process usually contains multiple varying operators, such as Pareto non-dominated sorting, crossover, and environmental selection. The issue of how to efficiently protect the whole optimization process using different schemes such as secret sharing and HE remains a huge challenge.

4.2.3. Privacy-preserving BO

BO begins with the construction of a surrogate model, usually a GP model, based on the training data available before the optimization. Next, it optimizes an acquisition function to suggest the next query input—that is, a new data point to be sampled. Therefore, the sensitive information in BO includes the training data for constructing the surrogate and the new query points. An example of the framework of outsourcing BO is given in Fig. 5. On the experimenter side, the users' local training data can be perturbed by adding noise to the objective values or transforming the dataset into another new dataset before sending it to the service provider. On the service provider side, BO is conducted, and the query input suggested by the BO is sent back to the experimenter. Kusner et al. [110] were the first to propose protecting the best query point using DP at the end of a BO process by adding Laplace noise to the true objective values of the best query point. However, if a communication attack occurs during the BO process, intermediate results such as the predictions of a GP model can also be sensitive information, since they may disclose information on solutions that have been evaluated using real functions. For example, the real objective values of an individual can be inferred based on the kernel matrix, as discussed in Refs. [111,112]. Nguyen et al. [113] addressed this leakage risk by adding noise to every objective value corresponding to every next query input. A new question brought about is whether the constructed surrogate model based on the obfuscated objective values will degrade the performance of the optimization of the acquisition function. Moreover, the error in model construction will accumulate with an increase in the number of newly suggested query points. Apart from directly adding noise to the objective values, such as in Ref. [113] in regression tasks, DP has been extended to handle expensive classification problems for data protection. In Ref. [114], the Laplace noise generation mechanism is adopted and implemented using the expected value of the classification probability; classification tasks are then accomplished with an embedded squash function.

Apart from DP, a few attempts have been made along the line of transformation and HE, which is more accurate than DP-based approaches. Given an experimenter that is unwilling to directly send either decision variables or true objective values to an outsourcing party, Kharkovskii et al. [115] proposed transforming the decision variables by means of a defined matrix, which is only private for the experimenter, while the optimization process of the acquisition function is based on the transformed decision variables. At the end of the optimization process of the acquisition function, the next query point is converted back by the experimenter. Another approach without degrading the accuracy is the HE-based method. GP model construction and prediction are secured by using a modular approach [111], ensuring that any clients can still make predictions of the mean and standard deviation of any observed points, even though no local data in the service provider is transmitted. Compared with DP and transformation-based methods, HE-based approaches are relatively time-consuming. Very recently, Luo et al. [112] adopted a secret-sharing strategy in model construction and predictions that can be extended to three scenarios—horizontal data sharing, vertical data sharing, and prediction data sharing—with a relatively high model-training efficiency.

To sum up, the three main approaches—namely, DP, transformation, and HE-based privacy-preserving methods—mainly attempt to protect the training data, the optimization of the acquisition function, or the next query point. However, due to the model construction errors, existing DP-based approaches usually perform much worse than traditional BO. Transformation-based approaches seem to be a promising direction, as they are much more time-efficient than HE-based approaches and do not introduce as much uncertainty into the optimization process of the acquisition function as DP-based methods.

4.3. Privacy-preserving fully distributed optimization

A situation in which all agents work collaboratively to solve a global optimization problem is formulated as the sum of local functions private to all participating agents via a fully connected network. This framework requires each agent to explicitly exchange its state to its neighboring agents over many iterations to obtain the best global optima of the global optimization problem. It should be noted that there are no servers in this scenario, and each agent does not transmit the information to the servers but to the neighboring clients, in what is called fully distributed optimization. To handle this class of fully distributed problems, researchers have proposed many algorithms, such as ADMM [15] and many variants of the ADMM, over the past decade. However, the mechanisms of these approaches all rely on exchanging the state of one agent with neighboring agents, which fails to protect the privacy of each agent, considering that the exchanged states may leak sensitive information about the agents.

One intuitive way is to encrypt the exchanged states. Because of the interacting message passing among the agents in a fully decentralized setting, cryptographic techniques usually require a trusted third party to avoid the state values of one agent being inferred from the transmitted information. Motivated by the fact that a trusted third party is not always available in real-world applications, researchers have made attempts to incorporate cryptographic techniques into ADMM [116] and a projected subgradient algorithm optimizer [117], assuming that there are no third parties or aggregators, to address fully distributed optimization problems. For a dual-variable update, the method of incorporating the Paillier cryptosystem is the same as that used in Refs. [116,117]. Average consensus problems are also important problems in distributed computing. In Ref. [118], considering that the coupling weights for the undirected graph are symmetric, the coupling weights are also separated into the product of two random positive numbers. Gao et al. [119] successfully implemented a privacy-preserving push sum algorithm by introducing different time-varying coupling weights and using partially homomorphic cryptography to reach an average consensus on directed graphs. Since the coupling weights for out-neighbors are unknown, one node cannot infer the actual state values by decrypting the

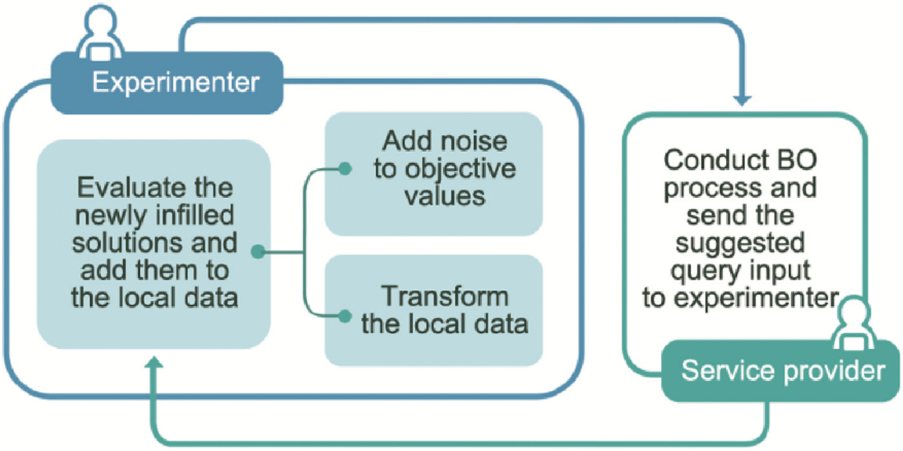
Experimenter
Evaluate the newly infilled solutions and add them to the local data
Add noise to objective values
Transform the local data
Conduct BO process and send the suggested query input to experimenter
Service provider

Fig. 5. A framework for privacy-preserving BO.

weighted state values. It is well known that Paillier cryptosystem approaches suffer from heavy computational and communication costs, making them impractical for dealing with real-time optimization problems. Apart from the Paillier cryptosystem, secret sharing as another popular cryptographic approach that has been widely used in fully distributed optimization because of its computational-efficient properties. Representative work is included in Refs. [120,121].

With a sound theoretical proof, DP has also been applied to fully distributed optimization by adding noise to the primal and/or dual variables in ADMM iterations [122,123] or the local functions [124], according to what must be protected. It is notable that, for a distributed optimization problem, either the objective functions or the constraints may contain sensitive information. Hence, most studies attempt to protect these two pieces of information. With regard to DP-based approaches, research has been conducted along two lines: providing a DP guarantee for each iteration or for all iterations. In the first line, for example, in Ref. [122], dynamic α-DP is achieved at each iteration, including the final iteration, by adding noise to primal variables or dual variables at every ADMM iteration. However, only ensuring the DP of the local objective function at each iteration will result in error accumulation over all iterations, breaking the convergence property of the iteration process and degrading the optimization performance. A promising idea is to achieve the DP guarantee over all iterations. In Ref. [125], in order to ensure a certain level of privacy budget over all iterations for iteration-distributed algorithms, Laplace noise with a decaying noise rate is added to the estimated global optima before broadcasting to neighboring agents. It should be noted that, when studies such as Refs. [122,123,126] rely on a decaying step size over the iterations to ensure convergence, a DP-based approach has been proposed [127] to achieve linear convergence in the mean under a constant step size. This work shows that the noise added to the state and direction values is indispensable, with each value accounting for privacy and convergence, respectively.

4.4. Federated data-driven evolutionary optimization

It is intuitive to borrow ideas from FL for privacy-preserving federated data-driven optimization since both methods assume raw data are decentralized in different clients and all participating clients collaboratively attempt to finish learning or optimization tasks. Despite their similarities, there are notable distinctions between the frameworks of FL and federated data-driven optimization, as illustrated in Fig. 6. Two special properties of the framework of federated data-driven optimization are: suggestion of next query input by optimizing an acquisition function and update of training data after evaluating newly infilled solutions. Although federated data-driven evolutionary optimization, as a newly emerging and promising topic, is effective in solving optimization tasks under the premise of privacy protection, only a few studies have explored this direction. A pioneering work is presented in Ref. [128], which proposes optimizing the hyperparameters of deep neural networks using BO, where the deep neural networks are locally trained on different clients. To reduce the transmitted parameters and prevent the local training data from being disclosed, a federated Thompson sampling acquisition function is proposed and optimized by conducting optimization of the acquisition function on a randomly selected local client at each round of the GP model update. Motivated by the fact that user-level privacy may not be ensured in Ref. [128], DP is applied to add noise to the model weights in Ref. [129], as in DP-assisted FL. It is notable that a central server is introduced in Ref. [129] in order to implement the DP. The optimization of the acquisition function is conducted on a randomly selected client in both Refs. [128] and [129] at each round of GP update.

While most existing data-driven surrogate-assisted evolutionary optimization algorithms assume that the data is centrally stored and sampled [27], it is of paramount practical importance to extend these optimization algorithms to the federated setting. Xu et al. [130,131] made the first attempt to achieve this by adapting the FL framework to federated data-driven evolutionary optimization. It should be noted, however, that GPs are nonparametric; therefore, the canonical FedAvg algorithm cannot be directly introduced in evolutionary BO. To address this issue, a radial basis function neural network model is built as the surrogate model on each client, based on their own data, to ensure that only model parameters rather than the local training data are uploaded to the server [130,131]. In addition, a federated lower confidence bound is designed as the acquisition function based on both the global surrogate model aggregated on the server and all local surrogate models. Inspired by these ideas, cloud edge model management was proposed [132] by weighting predicted local objective values based on coverage function values, and a blocking- and unblocking-based communication scheme was introduced to avoid deadlock during the optimization process.

Beyond the goal of federated optimization through shared model weights, an alternative strategy involves the direct sharing of insensitive data, such as model predictions or hyperparameters, between clients and the server. For example, particle swarm optimization can be integrated with a federated framework by sharing the velocity update directions of one client's particles instead of the real positions of these particles with a central server, thereby safeguarding the original raw data [133]. However, the convergence speed of this approach is somewhat compromised, due to the incorporation of a DP mechanism to increase the privacy protection. By contrast, Ref. [134] uses the predictions of a GP model to guide the velocity update of particle swarm optimization within a federated setting. Notably, this research is driven by the efficiency in tuning GP models rather than privacy concerns. Cheng et al. [135] individualized the optimization of the hyperparameters of the surrogate model of each client under a federated framework. This was accomplished by sharing transformed client encoding through random Fourier features, ensuring that each client's raw data remained confidential. It is essential to recognize that the federated optimization techniques discussed above assume complete honesty from all clients. However, in practice, a subset of these agents might be Byzantine. With this in mind, Zhang et al. [136] enabled the amalgamation of local GP models by aggregating the GP model predictions from all clients. This process accounts for arbitrary predictions from Byzantine clients. To mitigate Byzantine attacks, the largest and smallest β fractions of local predictive means and variances are strategically discarded. Furthermore, in Ref. [137], each client achieves the aim of merging the information of all clients by sharing the weighted local and global function values of a decision. The weights in each client are randomly sampled to preserve the actual local function values and to incorporate insights from other clients. Lastly, in a recent publication, Zhu et al. [138] employed the hyperparameters of GP models to infuse global information. A set of pseudo-data was introduced to gauge similarities between clients. This ensures that only the GP models of akin tasks are merged, representing a promising direction for federated BO.

In decentralized/federated optimization, the contributions from each client can vary significantly, depending on the volume and quality of the raw data they provide. Consequently, it might be unfair to suggest the same number of newly infilled solutions for all clients. For this reason, researchers have attempted to consider fairness, in addition to performance, in federated optimization. This direction holds considerable promise for concurrently gleaning insights from fairness-aware decentralized optimization when designing federated optimization frameworks. For example, in

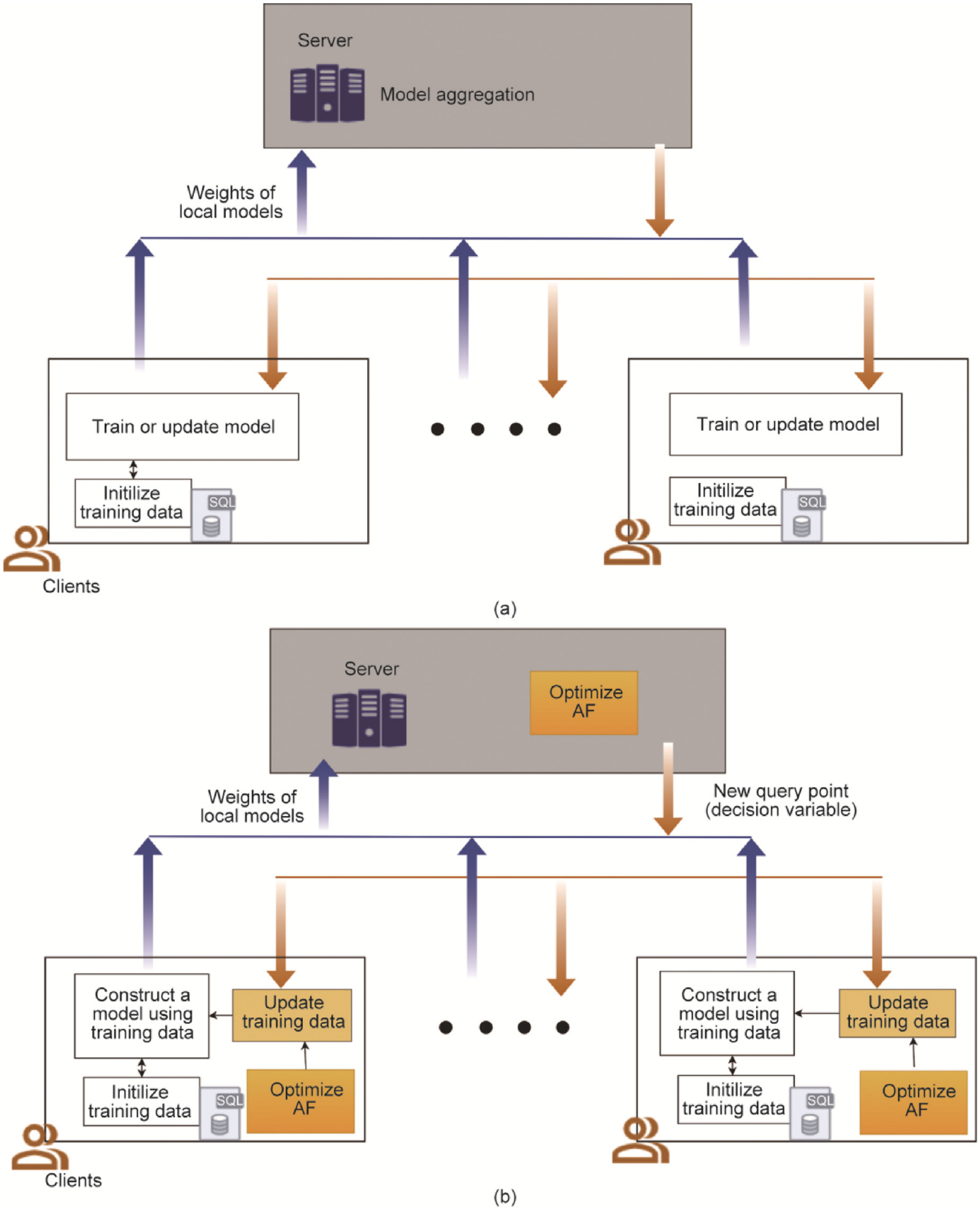

Fig. 6. Comparison between (a) the frameworks for FL and (b) federated data-driven optimization. AF: acquisition function; SQL: structured query language.

collaborative BO, assigning input queries to different parties with different information gains and rewards may hinder the collaboration [139]. Although an increasing number of studies have been dedicated to achieving fairness in machine learning and resource allocation [140], only a few attempts have been made to take fairness into consideration in decentralized optimization [141]. Perrone et al. [142] adopted a constrained acquisition function to satisfy the fairness constraint in BO. In Ref. [143], both statistical parity difference and the equality of opportunity difference are used as fairness metrics, resulting in a multi-objective optimization problem. Interestingly, a fair regret inspired by a cumulative utility is defined in batch BO, so that fairness is ensured by minimizing the difference between the cumulative utilities of all parties [139]. While the abovementioned work shed some light on fairness treatment in decentralized optimization, most of these studies focus on fairness issues in single-objective optimization problems and ignore the privacy concerns in federated optimization. Given the recent development in fairness-aware FL, performance fairness [140], which encourages uniform performance across devices, and collaboration fairness [144], which allows participants with a higher contribution to gain higher rewards, can be adopted in federated optimization as well. Moreover, the presence of multi-objective optimization problems in federated optimization poses additional fairness issues. For example, training data with sensitive attributes may lead to each local surrogate model having varying accuracy. In this case, biased optimization on different objectives and clients may occur, such as obtaining only a limited region of the Pareto front that satisfies certain groups of people. Furthermore, as different types of optimization problems occur with different clients (i.e., the number of objectives may vary), it is challenging to ensure fairness when constructing a global model. Therefore, the notion of model fairness [145] from FL can be adopted to address fairness-aware federated optimization problems, in order to ensure that no discrimination of the model against specific groups of people will occur. Interestingly, the preference of each client in federated optimization can be used to measure the reward when addressing fairness issues. This, in turn, results in a fairness issue when there is a tradeoff between conflicting preferences.

To conclude, the choice of federated optimization frameworks in real-world scenarios hinges on the type of sensitive information to be protected and the permissible performance tradeoffs. Regardless of whether the information fusion of clients is achieved through model weights, predictions, or hyperparameters, a significant disparity remains between optimization performance and security enhancement in these federated approaches. Continued efforts in this domain are essential. Moreover, the design of federated optimization should not just focus on improving the optimization performance; fairness should also be a critical factor in promoting collaborative optimization among multiple clients.

4.5. Discussion

To better understand the new challenges in federated optimization compared with FL, we summarize below the similarities and differences of privacy preservation in learning and optimization.

4.5.1. Unique challenges in privacy-preserving optimization

Although cryptographic techniques such as HE, MPC, and DP can be applied to both machine learning and optimization processes to ensure privacy preservation, the following mechanisms can be applied to privacy-preserving optimization but not to learning, due to the special requirements in optimization.

- **Subspace perturbation.** Inspired by the fact that dual variables will not converge in a certain subspace, noise is only added into the nonconvergent subspace of the dual variables in Ref. [56].
- **Function decomposition or state decomposition.** The local objective functions or state variables are decomposed into two parts—one accounting for passing messages with the neighboring clients and the other remaining local. This mechanism is specially designed for fully distributed optimization [146,147].
- **Transformation of training datasets.** Training samples are transformed in a such a way as to preserve the pairwise distance between samples, and the optimization of the acquisition function is based on the transformed datasets on the outsourcing party so that the privacy of the original datasets can be preserved [115].

4.5.2. Common strategies for FL and federated optimization

Federated optimization is still in its very early stage, and many research questions remain open. Thus, there are numerous research opportunities in introducing and adapting security and FL strategies to federated optimization.

- **FL and optimization structure.** The general structures of FL and federated optimization are very similar. It is notable that most FL structures belong to the horizontal FL framework, which is also commonly seen in federated optimization.
- **Privacy-preserving schemes.** The basic FL algorithm is not privacy preserving, even if it does not share the clients' raw data. There are vulnerabilities of data leakage from gradients and inference attacks, among others, if the server or clients are semi-honest. Many cryptographic primitives that have been embedded in FL to achieve privacy preservation and trustworthiness are also applicable to federated optimization. HE, MPC, DP, and secure aggregation are the most popular schemes in FL, among which HE can be directly applied to federated optimization. As in secure FL, the clients in federated data-driven optimization can encrypt sensitive data with private keys. The server calculates and returns the optimum to all clients. Clients can then decrypt the message with their private keys and obtain the plaintext value. Similarly, DP can be adopted by clients to add noise to the local dataset, the gradients, and the final model parameters to prevent inference attacks. Finally, secure aggregation as a special protocol of MPC can also be applied to the federated optimization, as discussed in Ref. [148].
- **Personalization.** In the classic FL structure, there is a common output for all clients. As a result, some local features may be missing from the global model. Moreover, not independent and identically distributed (non-IID) data distribution may lead to model divergence and therefore degrade the learning performance. To reflect the needs of individual clients, several personalized FL schemes have been proposed, including client clustering, local fine-tuning, meta-learning, multi-task learning, and the usage of local parameters, to name just a few [149]. In principle, these techniques can also be applied to federated optimization.
- **Incentive mechanisms.** FL loses its appeal to users with sufficient data for training, since local model training is even better than global training [150]. Many incentive mechanisms have been proposed to encourage more clients to join in federated training. However, applying incentive mechanisms directly into FL is difficult, as there is no standard approach to evaluate user behaviors. The most popular strategies for incentive mechanisms in FL are game theory, contract theory, and reinforcement learning, including Shapley value-based contribution evaluation and Stackelberg game-based contribution evaluation. These strategies can be applied to federated optimization to encourage more clients.

4.5.3. Common challenges in FL

Prior to delving into the newly stipulated requirements for secure federated optimization, it is helpful to provide a concise overview of the prevalent challenges that are typically encountered in the domain of FL. Given that federated optimization borrows many ideas from FL, it is logical to expect a certain degree of overlap in the context of their structural frameworks and the challenges they present. These challenges may include performance deterioration on non-IID data, increased computational and communication expenses, the question of client-level fairness, and the complexities related to federated neural architecture search.

- **Model divergence in the presence of non-IID data.** In machine learning, non-IID data refers to a scenario in which data distributions are not IID. A previous study [151] provided a comprehensive summary of five different non-IID forms. To investigate the impact of this non-IID data on the model, experimental results [152] present the difference in the model performance between IID data and non-IID data and show a significant drop in performance. It has also been demonstrated that federated averaging (FedAvg) suffers from a deterioration in accuracy of up to 9 % in non-IID datasets. As indicated by Zhao et al. [153], the performance degradation of FL can mainly be attributed to model divergence. Lian et al. [154] proposed a blockchain-based secret-sharing method to enhance the model's accuracy in non-IID settings without compromising user privacy. A detailed discussion of techniques for handling non-IID data in FL can be found in Ref. [149].

- **Communication efficiency.** FL relies on the transmission of model parameters from clients to the server and from the server to the clients. Insufficient FL bandwidth can result in inefficient communication, latency, and slow learning processes. Furthermore, such communication overhead increases with the number of participants and the number of iteration rounds. Much effort has been dedicated to designing communication-efficient FL. Strategies for reducing communication costs can be roughly divided into five categories: reducing the number of clients via client selection and rescheduling [155,156]; reducing the size of the parameters to be transmitted by means of model parameter compression [157,158]; parameter reduction through compression [158,159]; parameter transmission reduction through heterogeneous updating [160,161]; or model size reduction via neural architecture search [162,163].
- **Fairness.** In standard FL, all clients receive the same reward (e.g., the aggregated model), which may be unfair to clients who contribute significantly to the model performance [164,165]. To alleviate possible unfairness in FL, several fairness metrics have been proposed. For example, in order to protect weak clients, selection fairness has been proposed to increase their chances of participation [166]. Model fairness [167,168] aims to ensure that the federated trained model has no discrimination against specific individuals' or groups' data, regardless of whether the data distribution across clients is IID or heterogeneous. On the contrary, performance fairness [169] or accuracy parity [140] aims to encourage a uniformly standard accuracy parity across participants or clients. Collaboration fairness [144] and contribution fairness [170] were developed to ensure the long-term stability of the FL system, so that clients or participants with higher contributions to the model performance will receive higher rewards or incentives. Although a great deal of progress has been made in fairness-aware FL, it is still in its infancy. Different fairness metrics may conflict with each other [171], and too much emphasis on fairness will harm the model accuracy [172,173]. As a result, the question of how to balance the interests of the clients and the performance of the model remains an issue [174].
- **Federated neural architecture search.** Incorporating a neural architecture search into FL to collaboratively search for optimal models across clients in an automatic and privacy-preserving way has recently attracted increasing attention [163,175]. However, few federated neural architecture search approaches have been proposed. Most existing methods focus on model penalization in FL. For example, in Ref. [176–179], a neural architecture search is utilized to search for a personalized neural architecture to mitigate data heterogeneity in FL. Pan et al. [180] proposed a general framework for a federated neural architecture search to allow both the search and aggregation at the neural-operator-based micro level and the cell-based macro level. Multi-device environments—that is, when there is heterogeneity in the devices per user—are considered in Ref. [181], in which the server selects devices by utility values in terms of accuracy, efficiency, and convergence time. As the gradient information exchanged in FedAvg may leak privacy, Singh et al. [182] utilized DP to further enhance the privacy protection by adding Gaussian noise to the gradient before it is sent to the server for aggregation. Based on an evolutionary optimization-based neural architecture search, Zhu et al. [183] proposed a double-sampling method that randomly samples the sub-model and clients to reduce computational and communication costs. Interestingly, a federated evolutionary algorithm is adopted in Ref. [184] to accelerate the automatic design of graph convolutional network architectures in FL scenarios.

4.5.4. New requirements for secure federated optimization

Here, we present the additional requirements for secure federated optimization in terms of model management strategies, special operators in optimization, improving the performance of federated acquisition function optimization, and special defense schemes in optimization.

- **Model management strategies.** One critical issue in the implementation of secure federated optimization is the design and optimization of the model management strategies—that is, the federated acquisition function. For example, in Ref. [130,132], the new query points are fully exposed to the server, since the decision variables to be queried, their approximated objective values, and the local model parameters are all known to the server, making it more likely to disclose sensitive information. As a result, it is highly desirable to design additional privacy-preserving measures for surrogate model management strategies in federated data-driven optimization.
- **Special operators in optimization.** In genetic algorithms [57], operators such as crossover and mutation involve probability and division calculations, which are not commonly seen in secure FL. Hence, special secure protocols for these operators are required, as indicated in Ref. [103]. Environmental selection is also an essential component in an evolutionary optimization process. In single-objective evolutionary algorithms, a comparison of two fitness values can use existing comparison protocols, as in Ref. [103]. However, environmental selection operators in multi-objective evolutionary optimization algorithms such as Pareto non-dominated relationships are difficult to encrypt efficiently [102]. Therefore, there is great demand for the design of efficient and effective encryption schemes for special operators in optimization in order to securely conduct the whole optimization process.
- **Optimization of federated acquisition function.** In privacy-preserving BO, noise is added to the predicted objective values before they are sent to the server for constructing surrogate models. Consequently, optimizing an acquisition function based on the perturbed objective values may result in a failure to find the optimal solution. Hence, it is expected that the optimizer is able to alleviate the influence of noise on the optimization results in order to obtain an acceptable new query input.
- **Special defense schemes in optimization.** Since attackers in optimization attempt to mislead the optimization process rather than confuse the model training as in FL, it is expected that appropriate defense schemes for the optimization process can be designed for secure optimization.

5. Challenges and opportunities

Despite the encouraging progress that has been made in secure and privacy-preserving optimization, many challenges remain open. In this section, we outline challenges in secure optimization, based on which promising future work is suggested. One major open question is the definition of privacy and security in optimization. First of all, it is essential yet non-trivial to provide a quantitative definition of security and privacy, regardless of whether the context is FL or federated optimization. In addition, knowledge of how security and privacy-preservation measures may influence the optimization performance remains elusive. Other issues, similar to those in FL, include how to deal with non-IID, how to balance between privacy and security and between security and accuracy, and how to deal with the relationship between preference and fairness. In the following, we enlarge on these points.

5.1. Definition of privacy and security in optimization

It is always tricky to define privacy and security separately, as they are usually close and related to each other. For example, a security issue may be connected to privacy leakage. However, the tendencies of privacy and security always differ. It is challenging to determine how to define privacy and security in optimization. Here, we define the terms of security and privacy in optimization.

According to Ref. [18], privacy protection generally refers to preventing the public exposure of sensitive personal information. However, since the sensitive information in optimization is quite different from that in FL, we define privacy in optimization as follows: In optimization, privacy protection refers to the protection of sensitive information such as parameters in objective functions and constraints, individuals that have been evaluated using real function evaluations, and relative ranking among individuals during and after the optimization process. Security refers to defenses against possible attacks that may mislead the optimization process and give an inaccurate optimization result.

5.2. What are the limitations of different technical strategies in optimization?

Different strategies such as DP, HE, and MPC have both limitations and advantages, which we summarize below.

- **DP.** Although DP-based approaches have a solid theoretical background, applying them to optimization does not always result in satisfactory convergence performance. This is because an optimization process usually has a large number of iterations; thus, DP-based approaches must consider the total privacy leakage over the iterations. For example, in Ref. [185], the DP noise over the iterations decays to zero to improve the opportunity to converge to the global optimum. In DP-assisted BO, even though a very small ε value is adopted, it is not possible to ensure sufficient protection of the global optimum [113], since the perturbed solutions in the previous rounds will still have a direct influence on the surrogate modeling and acquisition function optimization in later rounds.
- **HE.** HE has been adopted for GP modeling, in which the intermediate data is encrypted for making privacy-preserving predictions [111]. In this work, GP prediction involves the calculation of a kernel function composed of pairwise distances between observed and unobserved data. Considering that the intermediate distance transmissions between the service provider and the client can contain sensitive information, adding a small amount of noise to the distances is suggested to prevent the dishonest party from knowing the exact distance. Therefore, it is concluded that HE approaches do not ensure complete privacy protection in optimization, and it is necessary to design HE schemes according to the properties of an optimization process. Moreover, computing on ciphertext requires a huge computation cost for the server side. The clients must encrypt and decrypt the values with their private keys, which also increases the computation cost. The communication cost also increases sharply, as the ciphertext is usually much larger than the plaintext.
- **Secure MPC.** MPC involves more than two subjects in a computing task without a trusted third party. Ultimately, each party can complete the computing task securely without knowledge of the data held by the other parties. As discussed earlier, the main techniques used in MPC are secret sharing, oblivious transfer, and garbled circuits. The popular secure aggregation protocol in FL is based on secret sharing, which can only protect the user's privacy from semi-honest or malicious servers. Moreover, it involves encryption and decryption, which will increase the communication and computation costs. For other threats, such as malicious clients and poisoning attacks, MPC is inadequate.

It can be concluded that privacy-protection techniques such as DP and HE cannot achieve complete protection for an optimization task. Thus, it is desirable for hybrid techniques to be adopted, according to the task and the user-specific privacy-protection requirements.

5.3. What makes privacy-preserving techniques in federated learning hard to implement in optimization?

As previously discussed, optimization involves more sensitive data and has more complex privacy-preservation requirements than learning. Thus, some of the privacy-preserving techniques applied to FL cannot be directly implemented in optimization. Among others, the protection of newly queried data and the influence of DP noise on the optimization performance are two predominant additional challenges that must be taken into account.

- **Protection of newly queried data.** As discussed in Ref. [148], it is necessary to prevent newly infilled solutions from being disclosed, in addition to protecting the local training dataset collected before the optimization starts. In other words, it is necessary for the privacy-preserving techniques in federated optimization to consider the protection of both raw data and newly infilled solutions.
- **Influence of DP noise on the optimization performance.** In FL, DP noise is added to the weights of the model, which should not heavily influence the model accuracy while being able to protect the local training data. In an optimization task, adding noise to the predicted objective values will influence not only the GP surrogate accuracy but also the optimization performance of the acquisition function, which may eventually seriously degrade the optimization performance.

5.4. Promising directions for federated optimization

Based on our discussions above, we consider the following six research directions to be promising and important for federated optimization.

(1) **Heterogeneity in local optimization problems.** In FL, one widely investigated heterogeneity between clients is the non-IID training data on different clients in the horizontal FL setting. FL becomes more challenging when different clients have different attributes and some clients do not have labeled data. By contrast, one important heterogeneity in federated optimization is that different clients have different subsets of decision variables, different constraints, or different objective functions. It should be noted that heterogeneity in decision variables, objectives, and constraints may need to be handled differently.

   - **Decision variables.** Even if all clients have the same decision variables, their operation conditions may differ. As a result, the data on the clients will become increasingly non-IID as the optimization proceeds, as discussed in Ref. [130]. It can also be the case that different clients have different decision variables, making it more challenging to perform federated optimization.

- **Objective functions.** It can be the case that different clients have different objectives in multi-task optimization [186–188], multi-scenario optimization [189,190], or multi-objective optimization [82,191].
- **Constraints.** In distributed optimization, it is common for each client to hold a different subset of constraints. Similarly, in federated optimization, different clients may have different constraint functions, as studied in Ref. [100,192].

(2) **Privacy preservation.** In federated optimization, the sensitive information includes the global optimum, the intermediate solutions during the optimization process, or even the rankings, decision variables, and objective values of any solutions that have been evaluated using the real local function evaluations. Thus, appropriate strategies must be designed according to the specific settings in federated optimization.

- **Protection of the ranking of the solutions in a population.** For an optimization problem, the ranking of individuals in the population may reveal sensitive information, since many evolutionary algorithms rely on the rank for performing optimization. Thus, unlike in FL and distributed optimization, it is considered that the design of corresponding encryption approaches for protecting the ranking of individuals is necessary and critical in federated optimization, as stated in Ref. [148].
- **Hybrid privacy-preserving techniques.** An optimization process usually includes several steps, such as the generation of offspring and selection, in which many operators such as multiplication, addition, and comparison are included. Hence, it is expected that various effective and efficient encryption approaches can be introduced into the optimization process without significantly increasing the communication cost.

(3) **Balancing among privacy and security, efficiency, and accuracy.** Although there is always a balance between privacy and accuracy in FL, the performance in distributed optimization does not have to be compromised, with careful design. For example, in Refs. [56,193], privacy can only be preserved while reaching the distributed average consensus if the dual variables in a primal–dual method of multipliers (PDMM) optimizer are initialized with random numbers with sufficiently large variances. In Ref. [194], an adaptive differential quantization method is proposed to achieve a low communication cost without compromising privacy, considering that the inserted noise will increase the communication costs. In FL, it is possible to strike a balance between privacy preservation and efficiency. For example, in Ref. [195], privacy preservation and efficiency are balanced by ternary gradient quantization and ElGamal encryption, significantly reducing the quantities of transmitted ciphertext. Similarly, it is believed that a balance between privacy and efficiency can be achieved in optimization, which is a promising topic.

(4) **Fairness.** Recently, fairness in BO [139,142] has attracted considerable attention. As pointed out in Ref. [174], the study of fairness-aware optimization mainly focuses on the following three aspects: fairness in decision-making under a multi-objective scenario, the tradeoff between the fairness and performance of federated optimization, and the modeling of fairness in data-driven optimization.

(5) **Designing new test benchmark problems and performance indicators.** As an emerging topic, there are no specific test benchmark problems at present that are designed for federated optimization, especially for the non-IID scenario mentioned above. Furthermore, the measurement of privacy is difficult to quantify, and different users may have different acceptance levels of privacy leakage. Thus, it is desirable to design new test benchmark problems and performance indicators for the evaluation of secure, privacy-preserving federated optimization.

(6) **Asynchronous infill sampling.** In asynchronous BO, considering that the function evaluation of each infilled solution in a batch may not be completed at the same time due to problems such as communication interruption or simulation errors, researchers have proposed either penalizing the ongoing infilled solution [196] or utilizing the randomness feature of the Thompson sampling acquisition function [128,197] to explore more regions. Under the federated optimization framework, it is also promising to consider asynchronous batch infill sampling, since the computation power of each client will vary considerably, resulting in infilled solutions with different evaluation times.

6. Conclusions

This survey aimed to provide a comprehensive literature review of privacy-preserving optimization, including fully distributed optimization, evolutionary optimization, BO, and data-driven evolutionary optimization. In addition to an introduction to the fundamentals of privacy-preserving and secure computing methods, including FL and cryptography techniques, we focused on discussing the common and differing requirements in privacy-preserving learning and in optimization, based on which we outlined promising future research topics.

We hope that this survey will help scholars recognize the importance of privacy protection and security protection in optimization, thereby promoting research interest in developing privacy-preserving and secure optimization algorithms and their real-world applications.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (62136003, 62302147, 62103150, 62006053, and 62306097), in part by the China Postdoctoral Science Foundation (2021M691012), in part by the Natural Science Foundation of Guangdong Province (2022A1515010443), in part by the National Research, Development and Innovation Fund of Hungary under the Establishment of Competence Centers, Development of Research Infrastructure Programme funding scheme (2019-1.3.1-KK-2019-00011), in part by an European Research Council (ERC) Advanced Grant "ERMiD", and in part by an Alexander von Humboldt Professorship for Artificial Intelligence awarded to Y. Jin by the German Federal Ministry of Education and Research.

Compliance with ethics guidelines

Qiqi Liu, Yuping Yan, Yaochu Jin, Xilu Wang, Peter Ligeti, Guo Yu, and Xueming Yan declare that they have no conflict of interest or financial conflicts to disclose.

References

[1] McMahan B, Moore E, Ramage D, Hampson S, Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics; 2017 Apr 20–22; Ft. Lauderdale, FL, USA; 2017.

[2] Jin Y, Zhu H, Xu J, Chen Y. Federated learning: fundamentals and advances. Singapore: Springer; 2022.

[3] Voigt P, von dem Bussche A. The EU General Data Protection Regulation (GDPR): a practical guide. 1st ed. Cham: Springer; 2017.

[4] Yang Q, Liu Y, Cheng Y, Kang Y, Chen T, Yu H. Federated learning—synthesis lectures on artificial intelligence and machine learning. Kentfield: Morgan & Claypool Publishers; 2019.

[5] Zhu L, Liu Z, Han S. Deep leakage from gradients. In: Proceedings of the 33rd Conference on Neural Information Processing Systems; 2019 Dec 8–14; Vancouver, BC, Canada; 2019.

[6] Lyu L, Yu H, Yang Q. Threats to federated learning: a survey. 2020. arXiv:2003.02133.

[7] Dwork C. Differential privacy: a survey of results. In: Agrawal M, Du DZ, Duan ZH, Li AS, editors. Theory and applications of models of computation. Berlin: Springer; 2008.

[8] Truong N, Sun K, Wang S, Guitton F, Guo Y. Privacy preservation in federated learning: an insightful survey from the GDPR perspective. Comput Secur 2021;110:102402.

[9] Jin Y, Wang H, Sun C. Data-driven evolutionary optimization: integrating evolutionary computation, machine learning and data science. Cham: Springer; 2021.

[10] Shahriari B, Swersky K, Wang Z, Adams RP, De Freitas N. Taking the human out of the loop: a review of Bayesian optimization. Proc IEEE 2015;104 (1):148–75.

[11] Jin Y. Surrogate-assisted evolutionary computation: recent advances and future challenges. Swarm Evol Comput 2011;1(2):61–70.

[12] Jones DR, Schonlau M, Welch WJ. Efficient global optimization of expensive black-box functions. J Glob Optim 1998;13(4):455–92.

[13] Yu X, Gen M. Introduction to evolutionary algorithms. London: Springer Science & Business Media; 2010.

[14] Back T. Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. Oxford: Oxford University Press; 1996.

[15] Boyd S, Parikh N, Chu E, Peleato B, Eckstein J. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach learn 2011;3(1):1–122.

[16] Mothukuri V, Parizi RM, Pouriyeh S, Huang Y, Dehghantanha A, Srivastava G. A survey on security and privacy of federated learning. Future Gener Comput Syst 2021;115:619–40.

[17] Yin X, Zhu Y, Hu J. A comprehensive survey of privacy-preserving federated learning: a taxonomy, review, and future directions. ACM Comput Surv 2021;54(6):131.

[18] Zhang K, Song X, Zhang C, Yu S. Challenges and future directions of secure federated learning: a survey. Front Comput Sci 2022;16:165817.

[19] Li Q, Wen Z, Wu Z, Hu S, Wang N, Li Y, et al. A survey on federated learning systems: vision, hype and reality for data privacy and protection. IEEE Trans Knowl Data Eng 2023;35(4):3347–66.

[20] Cao L, Chen H, Fan X, Gama J, Ong YS, Kumar V. Bayesian federated learning: a survey. 2023. arXiv:2304.13267.

[21] Weeraddana PC, Athanasiou G, Jakobsson M, Fischione C, Baras J. Per-se privacy preserving distributed optimization. 2012. arXiv:1210.3283.

[22] Yang T, Yi X, Wu J, Yuan Y, Wu D, Meng Z, et al. A survey of distributed optimization. Annu Rev Contr 2019;47:278–305.

[23] Li Q, Gundersen JS, Heusdens R, Christensen MG. Privacy-preserving distributed processing: metrics, bounds and algorithms. IEEE Trans Inf Forensics Secur 2021;16:2090–103.

[24] Molzahn DK, Dörfler F, Sandberg H, Low SH, Chakrabarti S, Baldick R, et al. A survey of distributed optimization and control algorithms for electric power systems. IEEE Trans Smart Grid 2017;8(6):2941–62.

[25] Zhao B, Chen WN, Li X, Liu X, Pei Q, Zhang J. When evolutionary computation meets privacy. 2023. arXiv:2304.01205.

[26] Wang X, Jin Y, Schmitt S, Olhofer M. Recent advances in Bayesian optimization. ACM Comput Surv 2023;55(13s):287.

[27] Jin Y, Wang H, Chugh T, Guo D, Miettinen K. Data-driven evolutionary optimization: an overview and case studies. IEEE Trans Evol Comput 2019;23 (3):442–58.

[28] Gentry C. A fully homomorphic encryption scheme. Palo Alto: Stanford University; 2009.

[29] Yao AC. Protocols for secure computations. In: Proceedings of 23rd Annual Symposium on Foundations of Computer Science (SFCS 1982); 1982 Nov 3–5; Chicago, IL, USA; 1982.

[30] Shamir A. How to share a secret. Commun ACM 1979;22(11):612–3.

[31] Gollmann D. Computer security. Wiley Interdiscip Rev Comput Stat 2010;2 (5):544–54.

[32] Sweeney L. *k*-anonymity: a model for protecting privacy. Int J Uncertain Fuzziness Knowl Based Syst 2002;10(05):557–70.

[33] Machanavajjhala A, Kifer D, Gehrke J, Venkitasubramaniam M. *L*-diversity: privacy beyond *k*-anonymity. ACM Trans Knowl Discov Data 2007;1(1):3.

[34] Li N, Li T, Venkatasubramanian S. *t*-closeness: privacy beyond *k*-anonymity and *l*-diversity. In: Proceedings of 2007 IEEE 23rd International Conference on Data Engineering; 2007 Apr 15–20; Istanbul, Turkey; 2007.

[35] Rivest RL, Adleman L, Dertouzos ML. On data banks and privacy homomorphisms. Found Secur Comput 1978;4:169–80.

[36] Su H, Chen H. Experiments on parallel training of deep neural network using model averaging. 2015. arXiv:1507.01239.

[37] Dean J, Corrado G, Monga R, Chen K, Devin M, Mao M, et al. Large scale distributed deep networks. NV, USA: Lake Tahoe; 2012.

[38] Chang K, Balachandar N, Lam C, Yi D, Brown J, Beers A, et al. Distributed deep learning networks among institutions for medical imaging. J Am Med Inform Assoc 2018;25(8):945–54.

[39] Sheller MJ, Reina GA, Edwards B, Martin J, Bakas S. Multi-institutional deep learning modeling without sharing patient data: a feasibility study on brain tumor segmentation. In: Proceedings of International MICCAI Brainlesion Workshop; 2022 Sep 18; Singapore; 2018.

[40] Gupta O, Raskar R. Distributed learning of deep neural network over multiple agents. J Netw Comput Appl 2018;116:1–8.

[41] Custers B, Sears AM, Dechesne F, Georgieva I, Tani T, Van der Hof S. EU personal data protection in policy and practice. Hague: Springer; 2019.

[42] Rahman MA, Rahman T, Laganière R, Mohammed N, Wang Y. Membership inference attack against differentially private deep learning model. Trans Data Priv 2018;11:61–79.

[43] Zhang X, Zhu X, Lessard L. Online data poisoning attacks. In: Proceedings of the 2nd Conference on Learning for Dynamics and Control; 2020 Jun 11–12; Berkeley, CA, USA; 2020.

[44] Du W, Atallah MJ. Secure multi-party computation problems and their applications: a review and open problems. In: Proceedings of the 2001 Workshop on New Security Paradigms; 2001 Sep 10-13; Cloudcroft New Mexico , pp. 13–22.

[45] Shokri R, Shmatikov V. Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security; 2015 Oct 12-16; Denver Colorado USA, pp. 1310–1321.

[46] Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, et al. Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security; 2016 Oct 24–28; Vienna, Austria; 2016.

[47] Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, et al. Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security; 2017 Oct 30– Nov 3; Dallas, TX, USA; 2017.

[48] Choudhury O, Gkoulalas-Divanis A, Salonidis T, Sylla I, Park Y, Hsu G, et al. Anonymizing data for privacy-preserving federated learning. 2020. arXiv:2002.09096.

[49] Song J, Wang W, Gadekallu TR, Cao J, Liu Y. EPPDA: an efficient privacy-preserving data aggregation federated learning scheme. IEEE Trans Netw Sci Eng 2023;10(5):3047–57.

[50] Truex S, Baracaldo N, Anwar A, Steinke T, Ludwig H, Zhang R, et al. A hybrid approach to privacy-preserving federated learning. In: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security; 2019 Nov 15; London, UK; 2012.

[51] Xu R, Baracaldo N, Zhou Y, Anwar A, Ludwig H. Hybrid Alpha: an efficient approach for privacy-preserving federated learning. In: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security; 2019 Nov 15; London, UK; 2019. p. 13–23.

[52] Zhu H, Wang R, Jin Y, Liang K. PIVODL: privacy-preserving vertical federated learning over distributed labels. IEEE Trans Artif Intell 2023;4(5):988–1001.

[53] Lian Z, Yang Q, Wang W, Zeng Q, Alazab M, Zhao H, et al. DEEP-FEL: decentralized, efficient and privacy-enhanced federated edge learning for healthcare cyber physical systems. IEEE Trans Netw Sci Eng 2022;9 (5):3558–69.

[54] Zhang S, Choromanska AE, LeCun Y. Deep learning with elastic averaging SGD. In: Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS 2015); 2015 Dec 11–12; Montreal, QC, Canada; 2015.

[55] Kingma DP, Ba J. Adam: a method for stochastic optimization. 2014. arXiv:1412.6980.

[56] Li Q, Heusdens R, Christensen MG. Privacy-preserving distributed optimization via subspace perturbation: a general framework. IEEE Trans Signal Process 2020;68:5983–96.

[57] Kramer O. Genetic algorithms. In: Kramer O, editor. Genetic algorithm essentials. Cham: Springer; 2017.

[58] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 2002;6(2):182–97.

[59] Cheng R, Jin Y, Olhofer M, Sendhoff B. A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 2016;20 (5):773–91.

[60] Auer P. Using confidence bounds for exploitation–exploration trade-offs. J Mach Learn Res 2002;3:397–422.

[61] Wang Z, Jegelka S. Max-value entropy search for efficient Bayesian optimization. In: Proceedings of the 34th International Conference on Machine Learning; 2017 Aug 6–11; Sydney, NSW, Australia; 2017.

[62] Rodemann T. A many-objective configuration optimization for building energy management. Brazil: Rio de Janeiro; 2018.

[63] Ye X, Chen B, Li P, Jing L, Zeng G. A simulation-based multi-agent particle swarm optimization approach for supporting dynamic decision making in marine oil spill responses. Ocean Coast Manage 2019;172:128–36.

[64] Schmitt T, Hoffmann M, Rodemann T, Adamy J. Incorporating human preferences in decision making for dynamic multi-objective optimization in model predictive control. Inventions 2022;7(3):46.

[65] Abdi H, Williams LJ. Principal component analysis. Wiley Interdiscip Rev Comput Stat 2010;2(4):433–59.

[66] Belkina AC, Ciccolella CO, Anno R, Halpert R, Spidlen J, Snyder-Cappione JE. Automated optimized parameters for T-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. Nat Commun 2019;10:5415.

[67] Ntelemis F, Jin Y, Thomas SA. Image clustering using an augmented generative adversarial network and information maximization. IEEE Trans Neural Netw Learn Syst 2022;33(12):7461–74.

[68] Ntelemis F, Jin Y, Thomas SA. Information maximization clustering via multi-view self-labelling. Knowl Base Syst 2022;250:109042.

[69] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. NV, USA: Las Vegas; 2016.

[70] Gaikwad SK, Gawali BW, Yannawar P. A review on speech recognition technique. Int J Comput Appl 2010;10:16–24.

[71] Yu D, Deng L. Automatic speech recognition. Cham: Springer; 2015.

[72] Chowdhary KR. Natural language processing. In: Chowdhary KR, editor. Fundamentals of artificial intelligence. Cham: Springer; 2020.

[73] Van Engelen JE, Hoos HH. A survey on semi-supervised learning. Mach Learn 2020;109(2):373–440.

[74] Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H, et al. A comprehensive survey on transfer learning. Proc IEEE 2020;109(1):43–76.

[75] Jaiswal A, Babu AR, Zadeh MZ, Banerjee D, Makedon F. A survey on contrastive self-supervised learning. Technologies 2020;9(1):2.

[76] Sun S, Cao Z, Zhu H, Zhao J. A survey of optimization methods from a machine learning perspective. IEEE Trans Cybern 2019;50(8):3668–81.

[77] Bäck T, Schwefel HP. An overview of evolutionary algorithms for parameter optimization. Evol Comput 1993;1(1):1–23.

[78] Van Laarhoven PJ, Aarts EH. Simulated annealing: theory and applications. Cham: Springer; 1987.

[79] Yuan Y, Xu H, Wang B, Yao X. A new dominance relation-based evolutionary algorithm for many-objective optimization. IEEE Trans Evol Comput 2016;20(1):16–37.

[80] Zhang Q, Li H. MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 2007;11(6):712–31.

[81] Liu Q, Jin Y, Heiderich M, Rodemann T. Adaptation of reference vectors for evolutionary many-objective optimization of problems with irregular Pareto fronts. In: Proceedings of 2019 IEEE Congress on Evolutionary Computation (CEC); 2019 Jun 10–13; Wellington, New Zealand; 2019.

[82] Liu Q, Jin Y, Heiderich M, Rodemann T, Yu G. An adaptive reference vector guided evolutionary algorithm using growing neural gas for many-objective optimization of irregular problems. IEEE Trans Cybern 2020;52(5):2698–711.

[83] Sun Y, Yen GG, Yi Z. IGD indicator-based evolutionary algorithm for many-objective optimization problems. IEEE Trans Evol Comput 2018;23(2):173–87.

[84] Bader J, Zitzler E, Hyp E. An algorithm for fast hypervolume-based many-objective optimization. Evol Comput 2011;19(1):45–76.

[85] Zhou Y, Zhu M, Wang J, Zhang Z, Xiang Y, Zhang J. Tri-goal evolution framework for constrained many-objective optimization. IEEE Trans Syst Man Cybern Syst 2020;50:3086–99.

[86] Yu G, Ma L, Jin Y, Du W, Liu Q, Zhang H. A survey on knee-oriented multi-objective evolutionary optimization. IEEE Trans Evol Comput 2022;26(6):1452–72.

[87] Said LB, Bechikh S, Ghédira K. The r-dominance: a new dominance relation for interactive evolutionary multicriteria decision making. IEEE Trans Evol Comput 2010;14(5):801–18.

[88] Deb K, Sinha A, Korhonen PJ, Wallenius J. An interactive evolutionary multiobjective optimization method based on progressively approximated value functions. IEEE Trans Evol Comput 2010;14(5):723–39.

[89] Coello CAC, Brambila SG, Gamboa JF, Tapia MGC, Gómez RH. Evolutionary multiobjective optimization: open research areas and some challenges lying ahead. Complex Intell Syst 2020;6(2):221–36.

[90] Sakuma J, Kobayashi S. A genetic algorithm for privacy preserving combinatorial optimization.In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation; 2007 Jul 7–11; London, UK; 2007.

[91] Han S, Ng WK. Privacy-preserving genetic algorithms for rule discovery. In: Proceedings of International Conference on Data Warehousing and Knowledge Discovery; 2007 Sep 3–7; Regensburg, Germany; 2007.

[92] Goethals B, Laur S, Lipmaa H, Mielikäinen T. On private scalar product computation for privacy-preserving data mining. In: Proceedings of International Conference on Information Security and Cryptology; 2004 Dec 2–3: Seoul, Republic of Korea; 2004. p. 104–20.

[93] Hong Y, Vaidya J, Lu H. Securely solving the distributed graph coloring problem. 2018. arXiv:1803.05606.

[94] Bogunovic I, Scarlett J, Jegelka S, Cevher V. Adversarially robust optimization with Gaussian processes. In: Proceedings of Advances in Neural Information Processing Systems; 2018 Dec 3-8; Montréal, Canada.

[95] Cai X, Scarlett J. On lower bounds for standard and robust Gaussian process bandit optimization. In: Proceedings of International Conference on Machine Learning; 2021 Jul 18–24; online; 2021. p. 1216–26.

[96] Bogunovic I, Krause A, Scarlett J. Corruption-tolerant Gaussian process bandit optimization. In: Proceedings of International Conference on Artificial Intelligence and Statistics; 2020 Aug 26–28;online; 2020. p. 1071–81.

[97] Han E, Scarlett J. Adversarial attacks on Gaussian process bandits. In: Proceedings of International Conference on Machine Learning; 2022 Jul 17–23; Baltimore, Maryland; 2022. p. 8304–29.

[98] Zhan ZH, Wu SH, Zhang J. A new evolutionary computation framework for privacy-preserving optimization. In: Proceedings of International Conference on Advanced Computational Intelligence; 2021 May 14–16; Wanzhou, China; 2021. p. 220–6.

[99] Zhao B, Liu X, Song A, Chen WN, Lai KK, Zhang J, et al. PriMPSO: a privacy-preserving multiagent particle swarm optimization algorithm. IEEE Trans Cybern 2023;53(11):7136–49.

[100] Bogdanov D, Emura K, Jagomägis R, Kanaoka A, Matsuo S, Willemson J. A secure genetic algorithm for the subset cover problem and its application to privacy protection. In: Proceedings of International Workshop on Information Security Theory and Practice; 2014 Jun 30 - July 2; Crete, Greece, pp.108–123.

[101] Yan Y, Han D, Shu T. Privacy preserving optimization of participatory sensing in mobile cloud computing. In: Proceedings of International Conference on Distributed Computing Systems; 2017 Jun 5–8; Atlanta, GA, USA; 2017. p. 1084–93.

[102] D. Funke, F. Kerschbaum. Privacy-preserving multi-objective evolutionary algorithms. In: Proceedings of International Conference on Parallel Problem Solving from Nature; 2010 Sep 11–15; Krakow, Poland; 2010. p. 41–50.

[103] Zhao B, Chen WN, Wei FF, Liu X, Pei Q, Zhang J. Evolution as a service: a privacy-preserving genetic algorithm for combinatorial optimization. 2022. arXiv:2205.13948.

[104] J. Suo, L. Gu, X. Yan, S. Yang, X. Hu, L. Wang. PP-QIGA: a privacy-preserving quantum inspired genetic algorithm for the double digest problem. 2022. reseachsquare:10.21203/rs.3.rs-1941096/v1.

[105] Hong Y, Vaidya J, Lu H, Wang L. Collaboratively solving the traveling salesman problem with limited disclosure. In: Proceedings of the 4th ACM Conference on Data and Application Security and Privacy; 2014 Mar 3–5; San Antonio, TX, USA; 2014. p. 179–94.

[106] Hong Y, Vaidya J. An inference–proof approach to privacy-preserving horizontally partitioned linear programs. Optim Lett 2014;8(1):267–77.

[107] Hong Y, Vaidya J, Rizzo N, Liu Q. Privacy-preserving linear programming. In: Goel S, Hong Y, Giboney J, Atrey P, editors. World scientific reference on innovation: volume 4: innovation in information security. Singapore: World Scientific; 2018.

[108] Borden AR, Molzahn DK, Lesieutre BC, Ramanathan P. Power system structure and confidentiality preserving transformation of optimal power flow problem. In: Proceedings of Annual Allerton Conference on Communication, Control, and Computing; 2013 Oct 2–4; Monticello, IL, USA; 2013. p. 1021–8.

[109] Gupta A, Ligett K, McSherry F, Roth A, Talwar K. Differentially private combinatorial optimization. In: Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms; 2010 Jan 17–19; Austin, TX, USA; 2010. p. 1106–25.

[110] Kusner M, Gardner J, Garnett R, Weinberger K. Differentially private Bayesian optimization. In: Proceedings of International Conference on Machine Learning; 2015 Jul 6–11; Lille Grand Palais, France; 2015. p. 918–27.

[111] Fenner P, Pyzer-Knapp E. Privacy-preserving Gaussian process regression–a modular approach to the application of homomorphic encryption. In: Proceedings of Thirty-Fourth AAAI Conference on Artificial Intelligence; 2020 Feb 7–12; New York City, NY, USA; 2020. p. 3866–73.

[112] Luo J, Zhang Y, Zhang J, Qin S, Wang H, Yu Y, et al. Practical privacy-preserving Gaussian process regression via secret sharing. 2023. arXiv:2306.14498.

[113] Nguyen TD, Gupta S, Rana S, Venkatesh S. A privacy preserving Bayesian optimization with high efficiency. In: Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining; 2018 Jun 3–6; Melbourne, VIC, Australia; 2018. p. 543–55.

[114] Xiong Z, Li L, Yan J, Wang H, He H, Jin Y. Differential privacy with variant-noise for Gaussian processes classification. In: Proceedings of Pacific Rim International Conference on Artificial Intelligence; 2019 Aug 26–30; Yanuca Island, Fiji; 2019. p. 107–19.

[115] Kharkovskii D, Dai Z, Low BKH. Private outsourced Bayesian optimization. In: Proceedings of International Conference on Machine Learning; 2020 Jul 12–18; Vienna, Austria; 2020. p. 5231–42.

[116] Zhang C, Ahmad M, Wang Y. ADMM based privacy-preserving decentralized optimization. IEEE Trans Inf Forensics Secur 2018;14(3):565–80.

[117] Zhang C, Wang Y. Enabling privacy-preservation in decentralized optimization. IEEE Trans Control Netw Syst 2018;6(2):679–89.

[118] Ruan M, Ahmad M, Wang Y. Secure and privacy-preserving average consensus. In: Proceedings of the 2017 Workshop on Cyber–Physical Systems Security and Privacy; 2017 Nov 3; Dallas, TX, USA; 2017. p. 123–9.

[119] Gao H, Zhang C, Ahmad M, Wang Y. Privacy-preserving average consensus on directed graphs using push-sum. In: Proceedings of the 6th Annual IEEE Conference on Communications and Network Security (CNS); 2018 May 30–Jun 1; Beijing, China; 2018.

[120] Tian N, Guo Q, Sun H, Zhou X. Fully privacy-preserving distributed optimization based on secret sharing. TechRxiv; 2021.

[121] Li Q, Cascudo I, Christensen MG. Privacy-preserving distributed average consensus based on additive secret sharing. In: Proceedings of the 27th European Signal Processing Conference; 2019 Sep 2–6; A Coruña, Spain; 2019.

[122] Zhang T, Zhu Q. Dynamic differential privacy for ADMM-based distributed classification learning. IEEE Trans Inf Forensics Secur 2016;12(1):172–87.

[123] Huang Z, Hu R, Guo Y, Chan-Tin E, Gong Y. DP-ADMM: ADMM-based distributed learning with differential privacy. IEEE Trans Inf Forensics Secur 2019;15:1002–12.

[124] Zhang X, Khalili MM, Liu M. Improving the privacy and accuracy of ADMM-based distributed algorithms. In: Proceedings of International Conference on Machine Learning; 2018 Jul 10–15; Stockholmsmässan, Sweden; 2018. p. 5796–805.

[125] Z. Huang, S. Mitra, N. Vaidya, Differentially private distributed optimization. In: Proceedings of the 16th International Conference on Distributed Computing and Networking; 2015 Jan 4–7; Goa, India; 2015.

[126] Gauthier F, Gratton C, Venkategowda NK, Werner S. Privacy-preserving distributed learning with nonsmooth objective functions. In: Proceedings of the 54th Asilomar Conference on Signals, Systems, and Computers; 2020 Nov 1–4;online; 2020. p. 42–6.

[127] Ding T, Zhu S, He J, Chen C, Guan X. Differentially private distributed optimization via state and direction perturbation in multiagent systems. IEEE Trans Automat Contr 2021;67(2):722–37.

[128] Dai Z, Low BKH, Jaillet P. Federated Bayesian optimization via Thompson sampling. In: Proceedings of the 34th Conference on Neural Information Processing Systems; 2020 Dec 6–12; Vancouver, BC, Canada; 2020. p. 9687–99.

[129] Dai Z, Low BKH, Jaillet P. Differentially private federated Bayesian optimization with distributed exploration. In: Proceedings of the 35th Conference on Neural Information Processing Systems; 2021 Dec 6–14; online; 2021. p. 9125–39.

[130] Xu J, Jin Y, Du W, Gu S. A federated data-driven evolutionary algorithm. Knowl Base Syst 2021;233:107532.

[131] Xu J, Jin Y, Du W. A federated data-driven evolutionary algorithm for expensive multi-/many-objective optimization. Complex Intell Syst 2021;7(6):3093–109.

[132] Guo XQ, Chen WN, Wei FF, Mao WT, Hu XM, Zhang J. Edge–cloud co-evolutionary algorithms for distributed data-driven optimization problems. IEEE Trans Cybern 2023;53(10):6598–611.

[133] Torra V, Galván E, Navarro-Arribas G. Pso+ fl= paaso: particle swarm optimization + federated learning = privacy-aware agent swarm optimization. Int J Inf Secur 2022;21(6):1349–59.

[134] Kathen MJT, Johnson P, Flores IJ, Reina DGE. Aquafel-PSO: a monitoring system for water resources using autonomous surface vehicles based on multimodal PSO and federated learning. 2022. arXiv:2211.15217..

[135] Cheng A, Wang Z, Li Y, Cheng J. HPN: personalized federated hyperparameter optimization. 2023. arXiv:2304.05195.

[136] Zhang X, Yuan Z, Zhu M. Byzantine-tolerant federated Gaussian process regression for streaming data. Adv Neural Inf Process Syst 2022;35:13499–511.

[137] Salgia S, Vakili S, Zhao Q. Collaborative learning in kernel-based bandits for distributed users. 2023. arXiv:2207.07948.

[138] Zhu H, Wang X, Jin Y. Federated many-task Bayesian optimization. IEEE Trans Evol Comput. In press.

[139] Sim RHL, Zhang Y, Low BKH, Jaillet P. Collaborative Bayesian optimization with fair regret. In: Proceedings of International Conference on Machine Learning; 2021 Jul 18–24; online; 2021. p. 9691–701.

[140] Li T, Sanjabi M, Beirami A, Smith V. Fair resource allocation in federated learning. 2019. arXiv:1905.10497.

[141] Candelieri A, Ponti A, Archetti F. Fair and green hyperparameter optimization via multi-objective and multiple information source Bayesian optimization. 2022. arXiv:2205.08835.

[142] Perrone V, Donini M, Zafar MB, Schmucker R, Kenthapadi K, Archambeau C. Fair Bayesian optimization. In: Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society; 2021 May 19–21; online; 2021. p. 854–63.

[143] Mehrabi N, de Lichy C, McKay J, He C, Campbell W. Towards multi-objective statistically fair federated learning. 2022. arXiv:2201.09917.

[144] Lyu L, Xu X, Wang Q, Yu H. Collaborative fairness in federated learning. In: Yang Q, Fan L, Yu H, editors. Federated learning. Cham: Springer; 2020.

[145] Liu C, Fan Z, Zhou Z, Shi Y, Pei J, Chu L, et al. Achieving model fairness in vertical federated learning. 2021. arXiv:2109.08344.

[146] Zhang C, Gao H, Wang Y. Privacy-preserving decentralized optimization via decomposition. 2018. arXiv:1808.09566.

[147] Wang Y. Privacy-preserving average consensus via state decomposition. IEEE Trans Automat Contr 2019;64(11):4711–6.

[148] Liu Q, Yan Y, Ligeti P, Jin Y. A secure federated data-driven evolutionary multi-objective optimization algorithm. IEEE Trans Emerg Top Comput Intell., in press.

[149] Zhu H, Xu J, Liu S, Jin Y. Federated learning on non-IID data: a survey. Neurocomputing 2021;465:371–90.

[150] Yan Y, Ligeti P. A survey of personalized and incentive mechanisms for federated learning. In: Proceedings of IEEE 2nd Conference on Information Technology and Data Science. 2022 May 16–18; Debrecen, Hungary; 2022. p. 324–9.

[151] Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Bhagoji AN, et al. Advances and open problems in federated learning. Found Trends Mach learn 2021;14(1–2):1–210.

[152] Chai D, Wang L, Chen K, Yang Q. Fedeval: a benchmark system with a comprehensive evaluation model for federated learning. 2020. arXiv:2011.09655.

[153] Zhao Y, Li M, Lai L, Suda N, Civin D, Chandra V. Federated learning with non-IID data. 2018. arXiv:1806.00582.

[154] Lian Z, Zeng Q, Wang W, Gadekallu TR, Su C. Blockchain-based two-stage federated learning with non-IID data in IoMT system. IEEE Trans Comput Soc Syst 2023;10(4):1701–10.

[155] Nishio T, Yonetani R. Client selection for federated learning with heterogeneous resources in mobile edge. In: Proceedings of IEEE International Conference on Communications; 2019 May 20–24; Shanghai, China; 2019.

[156] Deng Y, Lyu F, Ren J, Wu H, Zhou Y, Zhang Y, et al. Auction: automated and quality-aware client selection framework for efficient federated learning. IEEE Trans Parallel Distrib Syst 2021;33(8):1996–2009.

[157] Konecný J, McMahan HB, Yu F, Richtárik P, Suresh AT, Bacon D. Federated learning: Strategies for improving communication efficiency. 2017. arXiv:1610.05492v2.

[158] Sattler F, Wiedemann S, Müller KR, Samek W. Robust and communication-efficient federated learning from non-IID data. IEEE Trans Neural Netw Learn Syst 2019;31(9):3400–13.

[159] Xu J, Du W, Jin Y, He W, Cheng R. Ternary compression for communication-efficient federated learning. IEEE Trans Neural Netw Learn Syst 2022;33(3):1162–76.

[160] Chen Y, Sun X, Jin Y. Communication-efficient federated deep learning with layer-wise asynchronous model update and temporally weighted aggregation. IEEE Trans Neural Netw Learn Syst 2020;31(10):4229–38.

[161] Guo Q, Qi Y, Qi S, Wu D, Li Q. FedMCSA: personalized federated learning via model components self-attention. 2022. arXiv:2208.10731.

[162] Zhu H, Jin Y. Multi-objective evolutionary federated learning. IEEE Trans Neural Netw Learn Syst 2020;31(4):1310–22.

[163] Liang X, Liu Y, Luo J, He Y, Chen T, Yang Q. Self-supervised cross-silo federated neural architecture search; 2021. arXiv:2101.11896.

[164] Lyu L, Yu J, Nandakumar K, Li Y, Ma X, Jin J, et al. Towards fair and privacy-preserving federated deep models. IEEE Trans Parallel Distrib Syst 2020;31(11):2524–41.

[165] Shi Y, Yu H, Leung C. A survey of fairness-aware federated learning; 2021. arXiv:2111.01872.

[166] Zhou P, Fang P, Hui P. Loss tolerant federated learning; 2021. arXiv:2105.03591.

[167] Chouldechova A, Roth A. The frontiers of fairness in machine learning. 2018. arXiv:1810.08810.

[168] Mehrabi N, Morstatter F, Saxena N, Lerman K, Galstyan A. A survey on bias and fairness in machine learning. ACM Comput Surv 2021;54(6):1–35.

[169] Yue X, Nouiehed M, Kontar RA. GIFAIR-FL: an approach for group and individual fairness in federated learning; 2021. arXiv:2108.02741.

[170] Cong M, Yu H, Weng X, Yiu SM. A game-theoretic framework for incentive mechanism design in federated learning. In: Yang Q, Fan L, Yu H, editors. Federated learning: privacy and incentive. Cham: Springer; 2020.

[171] Zhang Q, Liu J, Zhang Z, Wen J, Mao B, Yao X. Fairer machine learning through multi-objective evolutionary learning. In: Proceedings of the 30th International Conference on Artificial Neural Networks; 2021 Sep 14–17; Bratislava, Slovakia; 2021. p. 111–23.

[172] Speicher T, Heidari H, Grgic-Hlaca N, Gummadi KP, Singla A, Weller A, et al. A unified approach to quantifying algorithmic unfairness: Measuring individual & group unfairness via inequality indices. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2018 Aug 19–23; New York City, NY, USA; 2018. p. 2239–48.

[173] Chouldechova A. Fair prediction with disparate impact: a study of bias in recidivism prediction instruments. Big Data 2017;5(2):153–63.

[174] Yu G, Ma L, Du W, Du W, Jin Y. Towards fairness-aware multi-objective optimization; 2022. arXiv:2207.12138.

[175] Mushtaq E, He C, Ding J, Avestimehr S. Spider: searching personalized neural architecture for federated learning; 2021. arXiv:2112.13939.

[176] He C, Annavaram M, Avestimehr S. FedNAS: federated deep learning via neural architecture search. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020 Jun 13–19; Seattle, WA, USA; 2020.

[177] Garg A, Saha AK, Dutta D. Direct federated neural architecture search; 2020. arXiv:2010.06223.

[178] Xu M, Zhao Y, Bian K, Huang G, Mei Q, Liu X. Federated neural architecture search; 2020. arXiv:2002.06352.

[179] Zhang C, Yuan X, Zhang Q, Zhu G, Cheng L, Zhang N. Toward tailored models on private AIoT devices: federated direct neural architecture search. IEEE Internet Things J 2022;9(18):17309–22.

[180] Pan Z, Hu L, Tang W, Li J, He Y, Liu Z. Privacy-preserving multi–granular federated neural architecture search a general framework. IEEE Trans Knowl Data Eng 2023;35:2975–86.

[181] Cho H, Mathur A, Kawsar F. FLAME: federated learning across multi-device environments. Proc ACM Interact Mob Wearable Ubiquitous Technol 2022;6(3):107.

[182] Singh I, Zhou H, Yang K, Ding M, Lin B, Xie P. Differentially-private federated neural architecture search; 2020. arXiv:2006.10559.

[183] Zhu H, Jin Y. Real-time federated evolutionary neural architecture search. IEEE Trans Evol Comput 2022;26(2):364–78.

[184] Wang C, Chen B, Li G, Wang H. FL-AGCNS: federated learning framework for automatic graph convolutional network search; 2021. arXiv:2104.04141.

[185] Gratton C, Venkategowda NK, Arablouei R, Werner S. Privacy-preserving distributed zeroth-order optimization; 2020. arXiv:2008.13468.

[186] Swersky K, Snoek J, Adams RP. Multi-task Bayesian optimization. In: Proceedings of the 26th International Conference on Neural Information Processing Systems; 2013 Dec 5–10; New York City, NY, USA; 2013.

[187] Lin X, Zhen HL, Li Z, Zhang QF, Kwong S. Pareto multi-task learning. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems; 2019 Dec 8–14; Vancouver, BC, Canada; 2019.

[188] Smith V, Chiang CK, Sanjabi M, Talwalkar AS. Federated multi-task learning. In: Proceedings of the 30th International Conference on Neural Information Processing Systems; 2017 Dec 4–9; Long Beach, CA, USA; 2017.

[189] Zhu L, Deb K, Kulkarni S. Multi-scenario optimization using multi-criterion methods: a case study on byzantine agreement problem. In: Proceedings of IEEE Congress on Evolutionary Computation; 2014 Jul 6–11; Beijing, China; 2014.p. 2601–8.

[190] Deb K, Zhu L, Kulkarni S. Multi-scenario, multi-objective optimization using evolutionary algorithms: Initial results. In: Proceedings of IEEE Congress on Evolutionary Computation; 2015 May 25–28; Sendai, Japan; 2015. p. 1877–84.

[191] Hua Y, Liu Q, Hao K, Jin Y. A survey of evolutionary algorithms for multi-objective optimization problems with irregular Pareto fronts. IEEE/CAA J Autom Sin 2021;8(2):303–18.

[192] Wei FF, Chen WN, Li Q, Jeon SW, Zhang J. Distributed and expensive evolutionary constrained optimization with on-demand evaluation. IEEE Trans Evol Comput 2023;27(3):671–85.

[193] Li Q, Heusdens R, Christensen MG. Convex optimisation-based privacy-preserving distributed average consensus in wireless sensor networks. In: Proceedings of the 45th International Conference on Acoustics, Speech, and Signal Processing; 2020 May 4–8; online; 2020. p.5895–9.

[194] Li Q, Heusdens R, Christensen MG. Communication efficient privacy-preserving distributed optimization using adaptive differential quantization. Signal Process 2022;194:108456.

[195] Zhu H, Wang R, Jin Y, Liang K, Ning J. Distributed additive encryption and quantization for privacy preserving federated deep learning. Neurocomputing 2021;463:309–27.

[196] Alvi AS, Ru B, Calliess J, Roberts SJ, Osborne MA. Asynchronous batch Bayesian optimisation with improved local penalization. 2019. arXiv:1901.10452.

[197] Garcia-Barcos J, Martinez-Cantin R. Fully distributed Bayesian optimization with stochastic policies. 2019. arXiv:1902.09992.