POLITECNICO DI TORINO
Repository ISTITUZIONALE

SGDE: Secure Generative Data Exchange for Cross-Silo Federated Learning

Original
SGDE: Secure Generative Data Exchange for Cross-Silo Federated Learning / Lomurno, E., Archetti, A., Cazzella, L., Samele, S., Di Perna, L., Matteucci, M.. - ELETTRONICO. - 0:(2022), pp. 205-214. (AIPR 2022, 5th International Conference on Artificial Intelligence and Pattern Recognition Xiamen (CHN) September 23-25, 2022) [10.1145/3573942.3573974].

Availability:
This version is available at: 11583/2971578 since: 2023-09-11T15:33:28Z

Publisher:
ACM

Published
DOI:10.1145/3573942.3573974

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
ACM postprint/Author's Accepted Manuscript

(Article begins on next page)

27 November 2026

SGDE: Secure Generative Data Exchange for Cross-Silo Federated Learning

Eugenio Lomurno*, Alberto Archetti†, Lorenzo Cazzella‡,
Stefano Samele§, Leonardo Di Perna¶, Matteo Matteucci‖

Politecnico di Milano, Italy

Email: eugenio.lomurno@polimi.it*, alberto.archetti@polito.it†, lorenzo.cazzella@polimi.it‡,
stefano.samele@polimi.it§, leonardo.diperna@polimi.it¶, matteo.matteucci@polimi.it‖

***Abstract*—Privacy regulation laws, such as GDPR, impose transparency and security as design pillars for data processing algorithms. In this context, federated learning is one of the most influential frameworks for privacy-preserving distributed machine learning, achieving astounding results in many natural language processing and computer vision tasks. Several federated learning frameworks employ differential privacy to prevent private data leakage to unauthorized parties and malicious attackers. Many studies, however, highlight the vulnerabilities of standard federated learning to poisoning and inference, thus raising concerns about potential risks for sensitive data. To address this issue, we present SGDE, a generative data exchange protocol that improves user security and machine learning performance in a cross-silo federation. The core of SGDE is to share data generators with strong differential privacy guarantees trained on private data instead of communicating explicit gradient information. These generators synthesize an arbitrarily large amount of data that retain the distinctive features of private samples but differ substantially. In this work, SGDE is tested in a cross-silo federated network on images and tabular datasets, exploiting beta-variational autoencoders as data generators. From the results, the inclusion of SGDE turns out to improve task accuracy and fairness, as well as resilience to the most influential attacks on federated learning.**

I. Introduction

The formal definition of strict privacy regulation laws, such as the European GDPR [3] and the Chinese Cyber Security Law [39], raised the need to impose the fundamental right of privacy as a design pillar for data elaboration algorithms. Today, it is of utmost importance for AI researchers and developers to provide sound and secure algorithms that minimize the risks for data owners while providing value and knowledge.

Federated Learning (FL) [28] is among the most popular frameworks for distributed machine learning, born with a strong commitment to privacy preservation. In FL, a set of clients, each holding private data samples, collaborate to train a single machine learning model with the help of a central server. In the original algorithm, FedAvg, developed by McMahan *et al.* [32], the central server initializes and broadcasts a shared model to a set of clients. Then, each of these clients performs a small number of Stochastic Gradient Descent (SGD) epochs on their private data. The updated weights of the model are then sent back to the central server and averaged proportionally to the number of samples involved in the local optimization steps. Finally, the central server broadcasts the aggregated model and repeats the process until convergence.

During the training procedure, private data samples never leave the clients; therefore, in principle, user privacy is preserved. However, the iterative exchange of model weights exposes an attack surface that can be exploited by members inside the federation with malicious intents. Indeed, a set of poisoning and inference techniques based on generative deep learning methods have been developed to retrieve secret information. For instance, many inference attacks manipulate the gradients sent to the server at each iteration [31] to reconstruct private data or alter the central model with carefully designed updates.

The growing popularity of attacks on the secrecy of private data has questioned the practical security level of federated learning. Thus, defense against attacks that threaten privacy is one of the biggest open problems for FL research. To this end, many defensive techniques have been developed to mitigate threats, e.g., robust model aggregation [40], model pruning [6], and gradient encryption [53]. Unfortunately, a complex and large-scale system, like the FL framework, is exposed to unique vulnerabilities and risks, and often, the adopted countermeasures are not adequate to ensure robust security standards [5].

To improve safety and grant privacy of user data in a federated context, we propose SGDE, a framework for secure data exchange through differentially private data generators (Figure 1). SGDE operates in three phases: *Subscribe*, *Push*, and *Pull*. In the *Subscribe* phase, the client communicates to the server its intention to take part in the data exchange process. In the *Push* phase, the client trains a set of data generators with a high differential privacy level following the constraints prescribed by the server and sends the data generators to the server. Finally, in the *Pull* phase, the client may access the set of generators stored by the server and train any machine learning model on generated data locally. This study is focused on the cross-silo federated learning setting, where clients are institutions, e.g., hospitals, universities, or companies. Cross-silo federated learning is characterized by a low number of clients – hundreds, at most – and each client has access to the computational power needed to train a generative model locally.

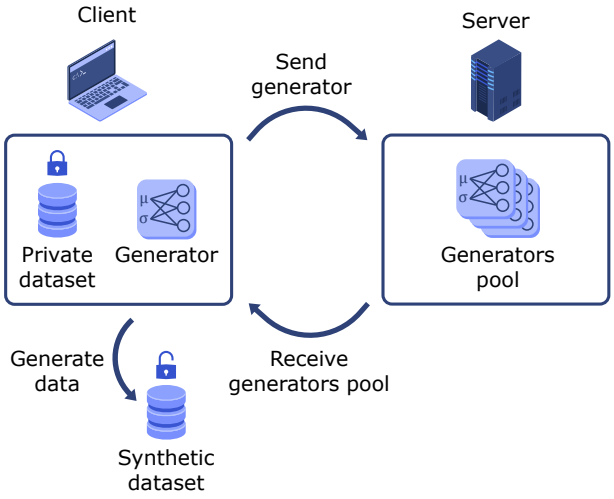

Fig. 1. The general scheme of SGDE from a single client perspective. In exchange for a generator trained on private data, the server grants the client access to the entire generators pool. Then, the client can generate an arbitrarily large synthetic dataset for offline use.

SGDE enables each client to generate an arbitrarily large set of synthetic data that preserves the distinctive features of real data. Synthetic data is available offline for client-side machine learning, combining the flexibility and transparency of a centralized dataset while reducing communication costs. Also, in a supervised classification setting, SGDE enables equal generation of samples for each class, increasing fairness with respect to underrepresented classes in the real dataset.

We argue that SGDE is secure against the most prominent attacks to federated learning, namely, poisoning and inference. In fact, from the security standpoint, an offline synthetic dataset is more advantageous than a privacy-protected distributed dataset, as poisoning attacks are much easier to detect [16]. Also, since there is no single model to be poisoned nor a global task to learn, the attack surface for malicious agents is strongly reduced, mitigating attacks by design, and enhancing the framework security.

This work brings the following contributions:

- we propose the SGDE framework and extensively discuss the advantages of its inclusion in a federated setting;
- we discuss how its implementation through differential-privacy-compliant data generators translates into remarkable improvements from a security standpoint;
- our claims about SGDE are validated by experimentally testing the framework with tabular and images datasets, demonstrating performance improvements from the federation members point of view;
- the SGDE framework demonstrates capabilities to deal with distribution biases and discriminations.

The rest of this work is organized as follows. Section II collects the most recent works related to federated networks, differential privacy, and generative models, highlighting the motivations behind the development of SGDE. Section III presents SGDE from a formal point of view, providing a detailed description of the protocol and analyze its security advantages. In Section IV collects the experiments to validate the claims about SGDE and extensively discuss the results. Finally, Section V concludes this work by summarizing the main concepts presented.

II. Related Work

This section collects the most influential works related to federated learning and privacy preservation in the machine learning context. In particular, Section II-A presents Federated Learning (FL) as the target scenario for the application of SGDE. Then, Section II-B describes Differential Privacy (DP) and its relation to FL security alongside an overview of the main techniques in generative deep learning and how they relate to DP. Finally, Section II-C describes the most prominent attacks to the FL paradigm and the secrecy of user data.

A. Federated Learning

Federated Learning (FL) [28], [24] is a privacy-compliant framework for distributed machine learning, scalable to millions of devices. In the general setting, K clients, each equipped with a local dataset of private samples, collaborate with a central server to minimize a global loss function F on a model with parameters w:

$$\min_w F(w) = \min_w \sum_{k=1}^{K} p_k F_k(w). \tag{1}$$

In (1), F_k is the loss function of client k evaluated on n_k samples from the local dataset. p_k is a weighting factor, such that $p_k \geq 0$ and $\sum_{k=1}^{K} p_k = 1$. The values of p_k depend on the application, but the most common assignments, given $n = \sum_{k=1}^{K} n_k$, are $p_k = \frac{n_k}{n}$ and $p_k = \frac{1}{K}$.

The main challenge of FL is dealing with the many faces of heterogeneity to be found in a massive network of clients. First, computational capabilities and connectivity may vary a lot between different clients. Indeed, the federated network may comprise devices with different hardware architectures and memory constraints. Also, the communication channel may be unreliable, leading to long pending periods and lost updates. In this setting, naively ignoring dropped updates may introduce a bias in the trained model towards the features of clients with more reliable connectivity. To this end, many efforts have been devoted to reducing the communication cost in FL systems and to dealing with dropped updates [26], [43]. As stated in [30], the main techniques to reduce the required bandwidth in mobile networks are increased edge computation, model compression, and importance-based updating. Energy efficiency is another topic of interest, as wireless communication is extremely power-hungry, especially in network federations including IoT and battery-powered devices [51], [18].

Another dimension of heterogeneity is the statistical diversity between the data distribution of each client, making the IID assumption unrealistic in real-world FL scenarios.

Many techniques have been introduced to deal with federated data heterogeneity. Agnostic FL [36] optimizes the central model with respect to any mixture of client updates, naturally increasing the fairness of the trained model. FedProx [29] generalizes the original FedAvg algorithm [32] by adding a regularization term that guarantees convergence in the non-IID setting.

The common FL framework provides a single model for each user, limiting the performance on local inference. Many researchers advocate for personalization of local models as a method to deal with non-IID data distributions in federated networks [8]. In [48], the authors address heterogeneity in a network of IoT devices and provide the main techniques to implement personalized on-device learning, namely, transfer learning, meta learning, federated multi-task learning, and federated distillation. Concerning meta learning, most works [13], [23] rely on the MAML framework [14] to provide a personalized model for each user. In [22] attentive message passing facilitates personalization by aggregating clients with similar features.

B. Differential Privacy and Generative Models

Differential Privacy (DP) [11] is a mathematically rigorous procedure to measure the security of a system against the disclosure of sensitive information related to individual samples. In particular, a randomized mechanism $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ is (ε)-differentially private if for any pair of adjacent inputs $x, y \in \mathcal{X}$ and any subset of outputs $S \subseteq \mathcal{Y}$ the following holds:

$$\Pr[\mathcal{M}(x) \in S] \leq \exp^{\varepsilon} \Pr[\mathcal{M}(y) \in S]. \tag{2}$$

In (2), ε represents the privacy budget, i.e., the theoretical amount of information that could leak from the system. The lower the ε value, the stronger the privacy guarantee is. Moreover, DP exhibits three convenient properties that make its inclusion natural in iterative optimization procedures such as stochastic gradient descent:

- **Composability**: a system composed by several differentially private mechanisms is still differentially private; this property holds for sequential and parallel composition.
- **Group privacy**: privacy guarantees never degrade abruptly, even if the adjacent inputs are strongly correlated or belong to the same individual.
- **Robustness to auxiliary information**: the privacy level is theoretically granted regardless of the knowledge the attacker has about the mechanism.

However, the constraints of ε-DP are too strict to make it viable for real world applications. To answer this issue, many relaxed variants have been developed, such as *f*-DP [9], concentrated DP [12] and Rényi DP [34]. The most common relaxation of ε-DP is (ε, δ)-DP [1]. In (ε, δ)-DP, the randomized mechanism $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ is (ε, δ)-differentially private if for any pair of adjacent inputs $x, y \in \mathcal{X}$ and any subset of outputs $S \subseteq \mathcal{Y}$ the following holds:

$$\Pr[\mathcal{M}(x) \in S] \leq \exp^{\varepsilon} \Pr[\mathcal{M}(y) \in S] + \delta. \tag{3}$$

In (3), the additive term δ represents the (possibly small) probability that ε-DP could be violated.

Many Federated learning systems include (ε, δ)-DP-based techniques in the distributed learning process to increase the privacy level of private data stored in the client devices. A common technique is to define a (ε, δ)-differentially-private optimizer, such as differentially-private SGD [1], where DP standards are met by clipping and adding Gaussian noise, at each iteration, to the current gradient. PATE [38] is a DP-aware training framework based on the student-teacher model, where a central model, the student, learns from a set of black-box private models, the teachers, by predicting outputs chosen with noisy voting.

DP countermeasures to data disclosure are also adopted in synthetic data generation through deep generative models, such as Generative Adversarial Networks (GANs) [17] and Variational Autoencoders (VAEs) [25]. Differentially-private VAEs [7] and GANs [50] achieve DP standards by adding carefully designed perturbations to the gradients during training. Advanced GAN architectures have been extended to meet DP standards, such as InfoGAN [37] and DP-Conditional GAN [47], relying on the Rényi-DP model [34], [35].

DP-auto-GAN [46] is a framework for synthetic data generation, applicable to unlabeled, mixed-type data, that combines the flexibility of GANs with the dimensionality-reduction capabilities of autoencoders. PATE-GAN [52] is an extension of the PATE framework in which a set of private teacher discriminators train a student discriminator against a common generator. In this way, synthetic data is differentially private with respect to the original, private data. In [2], the authors separate data synthesis in two steps. First, they perform K-means clustering with a differentially private kernel over sensitive data. Then, they train K generative models, one for each cluster, achieving higher data utility than a single-model architecture.

C. Threats to federated learning

A federated learning pipeline usually involves differential privacy techniques to protect clients data against unintended disclosure. However, many authors show that targeted attacks can retrieve sensitive information, even when differential privacy is involved. The most influential attacks to federated learning are collected in [31] and categorized as poisoning attacks and inference attacks. During a poisoning attack, a malicious agent deliberately modifies the training data or the parameters of the local model to deviate the learning procedure away from the true objective.

Inference attacks, instead, aim at guessing whether a specific data sample took part in the training procedure (membership inference [44], [19]) or reconstructing the input sample given a model and its corresponding output (input inference or model inversion [15]).

Another attack surface regards the backbone of FL training, the iterative gradient exchange. These inference methods are called gradient leakage attacks and allow malicious agents to

obtain information about raw private data samples with GAN-based gradient reconstruction [33]. It has been shown that even small portions of intermediate updates can lead to sensitive local data disclosure [4].

III. SGDE: Secure Generative Data Exchange

This section introduces SGDE, a data exchange framework based on generative models. The goal of SGDE is to guarantee strong privacy levels and face the major security issues related to the Federated Learning (FL) domain. It is important to note that SGDE is not a *learning* protocol, as there is no target model to be optimized nor a predefined task to be solved. Instead of sharing models gradients, SGDE provides to each client in a cross-silo federation a set of differentially private data generators able to synthesize an arbitrarily large number of samples. Each data generator embodies the features of its corresponding private dataset without disclosing any explicit information to curious or malicious agents. Once the generators are shared, each client can freely generate synthetic samples for any local machine learning task. In fact, the purpose of the dataset is entirely up to the client. Machine learning can occur privately, directly on generated data, or the client may still participate in a federated iterative procedure, training a common model in a distributed fashion using synthetic data, with clear privacy advantages, since synthetic data does not retain any sensitive information.

This work is focused on supervised classification, but the SGDE framework can be easily extended to other machine learning tasks as well. We argue that SGDE brings the following advantages to a cross-silo federation:

- **Flexibility**: SGDE gives full control of synthetic data to the client, both from the generation and usage aspects. The client can choose the best generators to build the dataset and arbitrarily decide the dataset cardinality. Moreover, synthetic data samples from SGDE are task-agnostic and can be involved in different machine learning applications, without the need for a central authority to coordinate the federation parties.
- **Security**: in SGDE, private data never leave the client, and the message exchange involves only generators. Also, private training parameters remain local, except for the generator privacy level. In this way, the attack surface with respect to poisoning and inference is severely limited.
- **Fairness**: each client can generate an arbitrarily large number of samples for each predefined label. In this way, each class can populate the dataset in equal proportion. This means that a synthetic dataset is fair with respect to class representation so that, as shown in Section IV, it is possible to attenuate distribution biases.
- **Communication efficiency and robustness**: SGDE does not involve any iterative exchange of data, as training occurs entirely on the device with no internet communication involved. Instead, generators are exchanged once between clients and the central server, providing a huge advantage in terms of bandwidth usage.

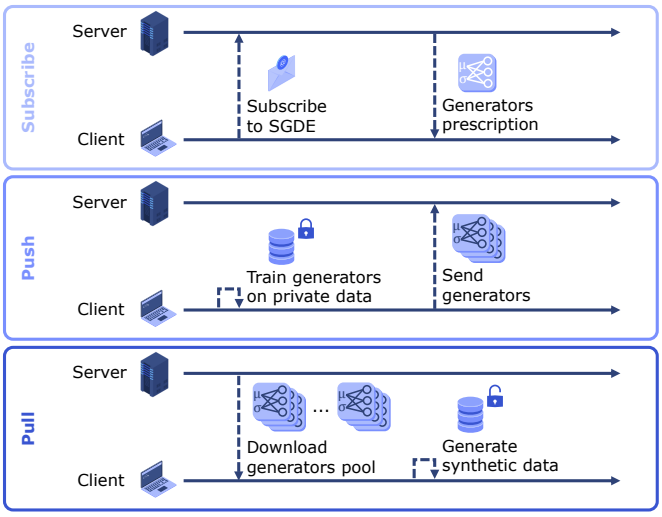

Fig. 2. The three steps of SGDE. First, during the *Subscribe* phase, client and server exchange preliminary information. Then, in the *Push* phase, the client trains a generator on private data according to the server prescriptions and sends it to the server. Finally, in the *Pull* phase, the client accesses the pool of generators available on the server.

A. The Steps of SGDE

Consider a federated network of K clients, such that each client k is equipped with a private set of data samples such that $\mathcal{D}_k = \{x_1, x_2, \ldots, x_{n_k} | x_i \in \mathcal{X}\}$, where $\mathcal{X}$ corresponds to the data domain. From now on, we assume to work in a supervised classification setting, but SGDE can be extended to other machine learning tasks. In this case, each sample x_i is paired with a class label y_i belonging to a finite set $\mathcal{Y}$. Furthermore, we assume each client k to be interested in accessing a larger set of data samples than $\mathcal{D}_k$, and willing to join a federation with other clients with the same intent.

In this context, SGDE is a highly secure solution for the federation described above In SGDE, each client, starting from its private dataset $\mathcal{D}_k$, is required to build a set of generators $\mathcal{G}_k$ composed of one data generator g_k^y for each class $y \in \mathcal{Y}$. Then, the set of generators $\mathcal{G}_k$ is collected by the central server and published among the other clients.

SGDE, as shown in Figure 2, is articulated in three steps, namely, *Subscribe*, *Push*, and *Pull*:

- *Subscribe*: client k communicates to the central server its intention to join the protocol. The server responds with a set of requirements for the generators. These may include specifications regarding synthetic data, the internal structure of the generators, or the minimum level of (ε, δ)-DP to guarantee.
- *Push*: for each class $y \in \mathcal{Y}$, client k trains a generator $g_k^y : \mathcal{Z} \to \mathcal{X}$, such that g_k^y is able to synthesize samples $\hat{x}$ corresponding to label y, starting from a noise vector $z \in \mathcal{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. Each g_k^y is collected in a generator set $\mathcal{G}_k$ and sent to the central server, alongside the (ε, δ)-DP level measured at the end of the training. Any other

information related to the training procedure that occurred on client k is not required by the server and should remain private.

- *Pull*: client k is granted access to the generators pool stored in the central server, containing generators from different clients. Client k may access the parameters, structure, and privacy level of the generators. At this point, client k may select which generators to download and start building a synthetic dataset. Each generator can be used to sample an unbounded number of synthetic instances and to produce arbitrarily large datasets.

B. Threat Analysis

This section analyzes how SGDE resists to the most common attacks to federated learning. Considering the topology of attacks described in [31], the attention is focused is on attacks run by malicious agents inside the federation.

The first family of attacks to be analyzed is poisoning. In SGDE, the attack surface for a malicious agent resides only in the construction of the generators. Since there is no central model nor a single point of failure, an attacker cannot compromise the whole system, as in a centralized FL setting. At most, the attacker may build a set of generators that synthesize poisoned data with the intent of having local models produce wrong estimations. However, from an honest user perspective, a poisoning attack can be detected by performance drops and solved by discarding malicious data generators during the synthetic data sampling process. More generally, as argued in [16], a centralized and accessible dataset, such as a synthetic one generated with SGDE, is less prone to poisoning, as direct heuristic analysis makes poisoning attacks easier to detect. This is not possible in a standard FL scenario, where no data information is transparent to the clients, other than their private datasets.

The second family of attacks to be considered is inference. Again, the attack surface of SGDE is much smaller with respect to a standard FL setting. The absence of iterative gradient exchange among the parties prevents attackers from carrying out gradient leakage attacks and reconstructing private data.

We argue that other inference threats over data generators, such as membership inference and model inversion attacks, would not be effective either with SGDE. In fact, differential privacy was introduced in machine learning scenarios precisely as a countermeasure to these kinds of attacks. As discussed in [41], training models with strict DP levels guarantees high resilience against membership inference attacks. At the same time, current model inversion attacks against differentially private data generators exchanged through SGDE turn out to be ineffective. Since the data generators map random noise to synthetic samples, a model inversion attack would reconstruct the latent noise, at most.

Outsider attacks are generally considered a threat to the network infrastructure rather than a threat to the learning protocol. However, SGDE narrows the attack surface concerned with system availability too. Reducing the communication to the exchange of generators decreases the amount of information to be sent over the internet, with an immediate benefit in terms of bandwidth usage. This data flow reduction allows the SGDE framework to evade a broad set of server availability attacks concerning the continuous gradient exchange.

As a final note, we argue that traditional FL and SGDE can be symbiotically exploited to further improve security guarantees for individuals. In fact, a potentially distributed learning system, training on a dataset generated via SGDE, should not be concerned about the secrecy of synthetic data. Private users samples never leave the clients, while new machine learning algorithms can run on public synthetic data retaining the distinctive features of the private counterparts.

IV. Experiments

This section collects the set of experiments involving SGDE to validate our claims. In particular, we argue that the inclusion of SGDE in a machine learning scenario with private distributed data is beneficial from the utility, security, and fairness standpoints. To this end, well-known machine learning models are evaluated according to the accuracy, F1 score, and AUC metrics in three different scenarios called *Local*, *Federated*, and *Synthetic*. In the *Local* scenario, clients have no access to public generators and rely only on their private data to produce machine learning models locally. In the *Federated* scenario, a single machine learning model is jointly trained using the FedAvg algorithm [32] starting from private data. Finally, in the *Synthetic* scenario, clients take part in the SGDE protocol and exchange data generators to access a larger synthetic dataset. In this case, local models are trained on synthetic data coming from generators provided by different clients.

Our experiments include five tabular datasets from the UCI Machine Learning repository [10] (Titanic, Breast Cancer Wisconsin - Diagnostic, Mushroom, Adult, and Wine Quality) and two image datasets (MNIST [27] and Fashion MNIST [49]). Datasets with already defined training and test splits are kept unchanged, while for the other datasets a 90%-train-10%-test random split is applied. The federation network is composed of 20 clients, each equipped with a 5% non-overlapping random split of the original training dataset.

A. Baseline Experiments

The first set of baseline experiments, identified with the *Local* keyword, measures the classification accuracy, F1 score, and AUC of a machine learning model over different classification tasks using 10-fold cross-validation. Results are available in Table I. Instead, the second set of baseline experiments measures the performance of locally-trained models on external data. For each client, a machine learning model is trained on the entire local dataset, evaluating its performance on the test set. The best number of training epochs is found via cross-validation. The results are collected in Table III.

B. Federated Learning Experiments

In the set of experiments identified with the *Federated* keyword, a single machine learning model is trained on a collection of private client datasets using the FedAvg algorithm [32].

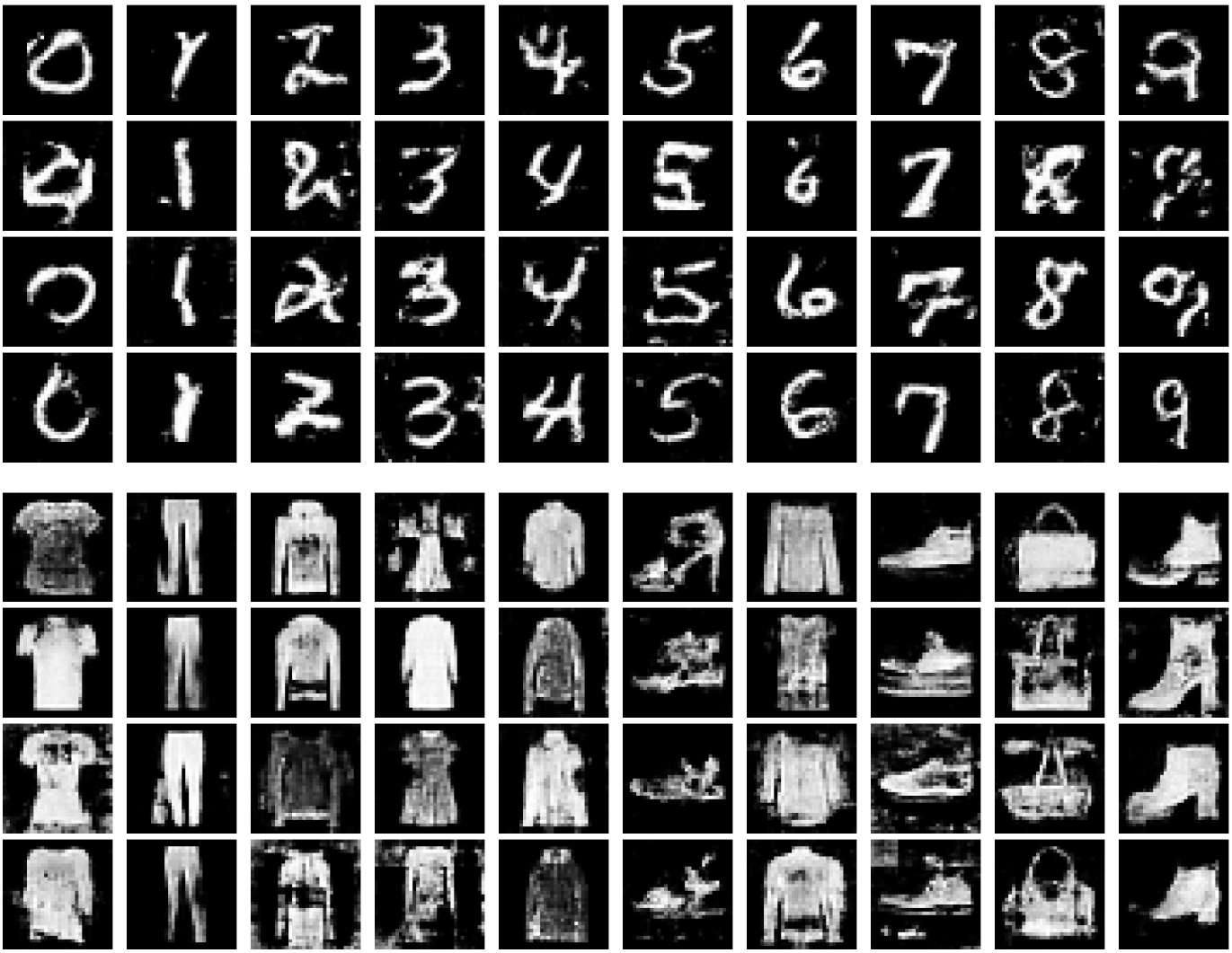

Fig. 3. Examples of synthetic data generated from SGDE generators. The first four rows are related to MNIST, and the remaining four to Fashion MNIST. For each dataset, every column contains images from a single generator trained for that specific class. It is noticeable the presence of noise in the background and the content distortion related to the differential privacy. The images are perceptually identifiable as not real and, therefore, not related to any privacy-protected real sample from a client dataset.

Table I collects the average model metrics evaluated on private validation splits from client datasets. Instead, Table III reports the metrics of the same model evaluated on the test set. Federated averaging was run until convergence, imposing the average between accuracy, F1, and AUC as validation score for the early stopping criterion.

C. SGDE Experiments

The experiments related to data generated using SGDE are identified with the *Synthetic* keyword. Given a specific dataset, according to the SGDE requirements, each client must build, train, and upload a data generator for each class to a trusted central server. Subsequently, the client can access all the available generators associated with the requested dataset. Assuming that all the federation members are collaborative and equally interested in any available data generator, each client can access only and exclusively their private data and the set of public data generators. At this point, clients can train machine learning models using synthetic data from all the available generators. Thus, local machine learning models are not limited to exploit only the private data on the same device.

In the experiments marked as Synthetic, each client trains a machine learning model locally using the optimal number of generated samples produced by the generators exchanged with SGDE. The synthetic dataset is constructed from samples uniformly picked from each available generator. In Table I, the average metrics of local models trained on synthetic samples and evaluated on private client datasets are collected under the *Synthetic* column. In Table III, instead, are reported the average metrics of the same models evaluated on the test set.

TABLE I

EXPERIMENTS EVALUATED ON LOCAL DATA SPLITS. THE *Local* COLUMNS REFER TO THE AVERAGE PERFORMANCE OF LOCAL MODELS TRAINED ON LOCAL DATA AND EVALUATED WITH 10-FOLD CROSS-VALIDATION. THE *Federated* COLUMNS REFER TO THE PERFORMANCE OF A SINGLE GLOBAL MODEL, TRAINED WITH FEDAVG [32] AND EVALUATED ON PRIVATE VALIDATION SPLITS. THE *Synthetic* COLUMNS REFER TO THE AVERAGE PERFORMANCE OF LOCAL MODELS TRAINED WITH SYNTHETIC DATA COMING FROM THE SGDE PROTOCOL. MODELS IN THE *Synthetic* COLUMNS ARE EVALUATED ON THE ENTIRE LOCAL DATASETS. WE HIGHLIGHT THE AVERAGE IMPROVEMENT OF FEDERATED LEARNING (*Federated*) AND SGDE (*Synthetic*) ON THE EVALUATION METRICS WITH RESPECT TO LOCAL (*Local*) TRAINING.

Dataset	Accuracy			F1 score			AUC		
	Local	*Federated*	*Synthetic*	*Local*	*Federated*	*Synthetic*	*Local*	*Federated*	*Synthetic*
Titanic	75.67	76.67	**80.87**	19.43	**69.89**	63.37	75.70	69.38	**78.35**
Breast Cancer	89.67	89.50	**97.09**	93.37	84.16	**97.81**	99.17	98.75	**99.27**
Mushrooms	92.93	91.51	**93.49**	92.43	91.32	**93.14**	96.23	95.84	**96.61**
Adult	**80.64**	76.01	79.65	49.69	47.95	**61.64**	83.30	79.81	**83.73**
Wine Quality	93.46	90.44	**98.54**	82.98	85.81	**97.10**	99.44	99.10	**99.49**
MNIST	98.20	**99.40**	98.72	98.16	**99.39**	98.71	99.02	**99.66**	99.31
Fashion MNIST	88.47	**91.75**	89.30	88.32	**91.65**	89.22	93.87	**95.76**	94.76
Avg. Improvement		-0.54	**+2.66**		+6.54	**+10.94**		-1.20	**+0.68**

TABLE II

HYPERPARAMETERS OF THE β-VAE ARCHITECTURE. FOR DENSE LAYERS, WE HIGHLIGHT THE NUMBER OF NEURONS, WHILE FOR CONVOLUTIONAL LAYERS, WE REPORT THE NUMBER OF FILTERS, THE KERNEL SIZE, AND THE STRIDE VALUE.

Architecture	Layer	Tabular data	Image data
Encoder	1st Layer	`Dense(64)`	`Conv2D(128,3,2)`
	2nd Layer	`Dense(32)`	`Conv2D(256,3,2)`
	3rd Layer	-	`Conv2D(512,3,2)`
Decoder	1st Layer	`Dense(64)`	`Conv2DT(128,3,2)`
	2nd Layer	`Dense(128)`	`Conv2DT(256,3,2)`
	3rd Layer	-	`Conv2DT(512,3,2)`

D. Classifiers and Infrastructure

To ensure a fair and robust comparison, only well-known models from machine learning literature are used as classifiers. In particular, the experiments on tabular datasets involve logistic regression as classifier, while the ones on image datasets include the first eight pre-trained layers of VGG16 [45] adapted with transfer learning. In the latter case, the only trainable component of the VGG16 architecture is the last 256-neuron dense layer with LeakyReLU as activation function, followed by the Softmax classifier. VGG16 is trained using the Adam optimizer with a learning rate of 0.001. All experiments are executed on a system equipped with an Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz and an Nvidia Quadro RTX 6000 GPU.

E. Differentially Private Generators

Data generators are the core component behind SGDE, as they need, on the one hand, to guarantee strict security levels and, on the other hand, to produce valuable synthetic data. In this sense, generated samples do not need to produce perceptively realistic data as long as they provide high utility in machine learning tasks. In fact, as shown in Figure 3, which depicts some synthetic images coming from class-specific generators of a single client, subjects are quite noisy and do not closely resemble any real sample. Nevertheless, the experimental results that will be presented in Section IV-F highlight a high machine learning utility for synthetic images, sometimes even higher than real data.

Concerning the level of security, data generators must not leak information related to the private data used during their training phase, thus preserving client privacy. SGDE allows the sharing of any data generator that meets the security requirements.

In our experiments, we implemented a custom version of the β-VAE [21] architecture trained with the differentially private implementation of the Adam optimizer [1]. In order to achieve the highest resilience level from inference attacks, i.e., the attack is not more effective than random guessing [41], [20], each client must train its generators so that their final (ε, δ)-DP level is characterized by $\varepsilon \leq 1.5$, $\delta \ll \frac{1}{|\mathcal{D}_c|}$ and $RDP \geq 9$, where $|\mathcal{D}_c|$ is the number of samples belonging to class c of dataset $\mathcal{D}$ and RDP is the Rényi-DP value. Each member must cut off from every model its initial part, the encoder, and share only the portion meaningful to generate synthetic samples, the decoder.

The ablation study conducted on the hyperparameters of the β-VAE architecture maximizes the generation performance against the noise introduced with differential privacy. The ablation study consists of a hyperparameter grid search on two architectural configurations for tabular and image data. The results are shown in Table II. In the model architecture for tabular data, each dense layer is followed by a LeakyReLU activation function. Instead, in the model architecture for the image data, each convolutional layer is followed by a Swish [42] activation function. Finally, the latent space dimension and the β value are tuned for each dataset.

F. Results

From an individual client perspective, the most compelling question is whether joining the SGDE protocol is beneficial from a utility standpoint. In Table I, each member of the federated network improved the classification average accuracy and AUC by 2.66% and 0.68%, respectively over all

TABLE III

EXPERIMENTS EVALUATED ON TEST SETS. THE *Local* COLUMNS REFER TO THE AVERAGE PERFORMANCE OF LOCAL MODELS TRAINED ON LOCAL DATA. THE *Federated* COLUMNS REFER TO THE PERFORMANCE OF A SINGLE GLOBAL MODEL, TRAINED WITH FEDAVG [32]. THE *Synthetic* COLUMNS REFER TO THE AVERAGE PERFORMANCE OF LOCAL MODELS TRAINED WITH SYNTHETIC DATA COMING FROM THE SGDE PROTOCOL. ALL THE MODELS ARE EVALUATED ON A HOLD-OUT SET. WE HIGHLIGHT THE AVERAGE IMPROVEMENT OF FEDERATED LEARNING (*Federated*) AND SGDE (*Synthetic*) ON THE EVALUATION METRICS WITH RESPECT TO LOCAL (*Local*) TRAINING.

Dataset	Accuracy			F1 score			AUC		
	Local	*Federated*	*Synthetic*	*Local*	*Federated*	*Synthetic*	*Local*	*Federated*	*Synthetic*
Titanic	71.83	**74.81**	74.01	29.70	**71.61**	56.00	77.14	70.85	**77.43**
Breast Cancer	89.42	91.86	**93.02**	92.25	90.82	**94.78**	99.60	99.36	**99.76**
Mushrooms	92.56	91.27	**93.49**	91.92	91.20	**93.14**	96.30	96.07	**96.61**
Adult	**80.87**	76.47	79.00	50.14	47.70	**60.21**	84.02	81.24	**84.08**
Wine Quality	92.57	90.23	**97.79**	82.42	85.10	**95.70**	98.63	98.50	**98.65**
MNIST	97.76	**99.08**	98.49	97.71	**99.08**	98.49	99.02	**99.50**	99.19
Fashion MNIST	85.97	87.94	**88.13**	85.81	87.99	**88.04**	92.65	93.68	**94.13**
Avg. Improvement		+0.10	**+1.85**		+6.22	**+8.06**		-1.17	**+0.36**

the datasets. This result allows us to assert that synthetic data coming from generators exchanged with SGDE is more effective than single-client local data to learn a classification task. Thus, synthetic data can effectively substitute privacy-protected local data in a machine learning procedure. In fact, in our experiments, SGDE is crucial to increase the amount of available information of a single client to train a machine learning model, without compromising the privacy of the other federation individuals.

Moreover, the F1 score improves by 10.94% on average over all the experiments. This means that generating data from each class uniformly lowers the distribution bias with respect to underrepresented classes in unbalanced datasets, as each client has access to more information about minority classes. Therefore, taking part to the SGDE protocol and training a machine learning model on synthetic data is not only beneficial from the accuracy standpoint, but leads to a more fair classification performance overall.

The results are confirmed in Table III too, where models are evaluated on the test set. By combining the *Local* and *Synthetic* experiments, there is an average improvement of accuracy and AUC of 1.85% and 0.36%, respectively. The F1 score increases by 8.06% on average, confirming the strong fairness advantage granted by taking part in the SGDE generator exchange.

So far, the discussion focused on the performance difference of training on synthetic samples with respect to training on real local data. Then, the next natural question is how training on synthetic samples compares to federated learning, the most prominent privacy-preserving training technique for a machine learning model on distributed data. The interesting result is that in most cases SGDE has still an advantage, not only by design in transparency communication costs, but in the final classification performance too. The experiments show that SGDE performs similarly, or even outperforms the standard FedAvg algorithm [32] in settings with a small number of clients and unbalanced data distributions. The result is more evident in the first five rows of Table I and Table III, where the experiments involve small tabular datasets and logistic regression as global machine learning model.

To recap, the experiments showed that SGDE provides a secure way of sharing knowledge embedded in private data by collecting data generators in a single pool and making them publicly accessible. With these results, SGDE has proven capable of improving the performance of machine learning tasks of individuals taking part in the generators exchange. Moreover, as SGDE allows the generation of a transparent, local dataset, it eliminates the need to join an iterative model exchange procedure, as in federated learning, alleviating the communication overload. Additionally, SGDE is beneficial from the accuracy, fairness, and transparency standpoints with respect to standard FL, while still guaranteeing strong protection for private user data.

More generally, we advocate for the development of secure technologies based on publicly accessible synthetic data, as we believe individuals cooperation in a secure environment to be the key to increasing value and knowledge in a privacy-compliant manner.

V. CONCLUSION

This work presents SGDE, a secure data exchange framework based on data generators with high privacy guarantees. The benefits of a generative-centric approach to data sharing are extensively discussed in a context where granting privacy is a hard constraint. Generators retain the distinctive features of private data while providing access to an arbitrarily large set of synthetic samples that are public, reproducible, and fair. Moreover, a centralized dataset is resilient against poisoning and inference attacks which pose a real threat to standard federated learning.

The effectiveness of SGDE is showcased in several experimental scenarios with high confidentiality levels, employing differentially private β-VAEs as generators. Training on synthetic data yielded better performance than true, private data, granting privacy protection for the individuals. SGDE consistently outperformed federated learning, one of the most influential techniques to train a machine learning model in a privacy-preserving way from distributed data. In fact, the inclusion of a generative protocol to share privacy-compliant

information is more communication-efficient, transparent, and effective than iteratively training a model with gradients exchange, especially when data distributions are skewed among clients.

Today, many researchers praise the benefits of a generative approach in privacy-critical federations. We believe that this is a research direction worth exploring, with a strong potential to lower the obstacles towards more user-centric, fair, and secure large-scale machine learning.

ACKNOWLEDGEMENT

The European Commission has partially funded this work under the H2020 grant N. 101016577 AI-SPRINT: AI in Secure Privacy-pReserving computINg conTinuum.

REFERENCES

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 308–318. [Online]. Available: https://doi.org/10.1145/2976749.2978318

[2] G. Acs, L. Melis, C. Castelluccia, and E. De Cristofaro, "Differentially private mixture of generative neural networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 6, pp. 1109–1121, 2018.

[3] J. Albrecht, "How the GDPR Will Change the World," *European Data Protection Law Review*, vol. 2, no. 3, pp. 287–289, 2016.

[4] Y. Aono, T. Hayashi, L. Wang, S. Moriai *et al.*, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2017.

[5] N. Bouacida and P. Mohapatra, "Vulnerabilities in federated learning," *IEEE Access*, vol. 9, pp. 63 229–63 249, 2021.

[6] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar, "Expanding the reach of federated learning by reducing client resource requirements," *arXiv preprint arXiv:1812.07210*, 2018.

[7] Q. Chen, C. Xiang, M. Xue, B. Li, N. Borisov, D. Kaarfar, and H. Zhu, "Differentially Private Data Generative Models," *arXiv:1812.02274 [cs]*, Dec. 2018, arXiv: 1812.02274.

[8] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive Personalized Federated Learning," *arXiv:2003.13461 [cs, stat]*, Nov. 2020, arXiv: 2003.13461.

[9] J. Dong, A. Roth, and W. J. Su, "Gaussian differential privacy," *arXiv preprint arXiv:1905.02383*, 2019.

[10] D. Dua and C. Graff, "UCI machine learning repository," 2017.

[11] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2013.

[12] C. Dwork and G. N. Rothblum, "Concentrated differential privacy," *arXiv preprint arXiv:1603.01887*, 2016.

[13] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized Federated Learning: A Meta-Learning Approach," *arXiv:2002.07948 [cs, math, stat]*, Oct. 2020, arXiv: 2002.07948.

[14] C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks," *arXiv:1703.03400 [cs]*, Jul. 2017, arXiv: 1703.03400.

[15] M. Fredrikson, S. Jha, and T. Ristenpart, "Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. Denver Colorado USA: ACM, Oct. 2015, pp. 1322–1333.

[16] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Mitigating Sybils in Federated Learning Poisoning," *arXiv:1808.04866 [cs, stat]*, Jul. 2020, arXiv: 1808.04866.

[17] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," *arXiv:1406.2661 [cs, stat]*, Jun. 2014, arXiv: 1406.2661.

[18] B. Guler and A. Yener, "Sustainable Federated Learning," *arXiv:2102.11274 [cs, math]*, Feb. 2021, arXiv: 2102.11274.

[19] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, "LOGAN: Membership Inference Attacks Against Generative Models," *arXiv:1705.07663 [cs]*, Aug. 2018, arXiv: 1705.07663.

[20] ——, "Logan: Membership inference attacks against generative models," in *Proceedings on Privacy Enhancing Technologies (PoPETs)*, vol. 2019, no. 1. De Gruyter, 2019, pp. 133–152.

[21] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *2017 International Conference on Learning Representations (ICLR)*, 2017.

[22] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang, "Personalized Cross-Silo Federated Learning on Non-IID Data," *arXiv:2007.03797 [cs, stat]*, Dec. 2021, arXiv: 2007.03797.

[23] Y. Jiang, J. Konečný, K. Rush, and S. Kannan, "Improving Federated Learning Personalization via Model Agnostic Meta Learning," *arXiv:1909.12488 [cs, stat]*, Sep. 2019, arXiv: 1909.12488.

[24] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, H. Eichner, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao, "Advances and Open Problems in Federated Learning," *arXiv:1912.04977 [cs, stat]*, Mar. 2021, arXiv: 1912.04977.

[25] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv:1312.6114 [cs, stat]*, May 2014, arXiv: 1312.6114.

[26] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated Learning: Strategies for Improving Communication Efficiency," *arXiv:1610.05492 [cs]*, Oct. 2017, arXiv: 1610.05492.

[27] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2, 2010.

[28] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, May 2020, arXiv: 1908.07873.

[29] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated Optimization in Heterogeneous Networks," *arXiv:1812.06127 [cs, stat]*, Apr. 2020, arXiv: 1812.06127.

[30] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[31] L. Lyu, H. Yu, and Q. Yang, "Threats to Federated Learning: A Survey," *arXiv:2003.02133 [cs, stat]*, Mar. 2020, arXiv: 2003.02133.

[32] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," *arXiv:1602.05629 [cs]*, Feb. 2017, arXiv: 1602.05629.

[33] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 691–706.

[34] I. Mironov, "Rényi differential privacy," in *2017 IEEE 30th computer security foundations symposium (CSF)*. IEEE, 2017, pp. 263–275.

[35] I. Mironov, K. Talwar, and L. Zhang, "Rényi differential privacy of the sampled gaussian mechanism," *arXiv preprint arXiv:1908.10530*, 2019.

[36] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic Federated Learning," *arXiv:1902.00146 [cs, stat]*, Jan. 2019, arXiv: 1902.00146.

[37] V. Mugunthan, V. Gokul, L. Kagal, and S. Dubnov, "DPD-InfoGAN: Differentially Private Distributed InfoGAN," *arXiv:2010.11398 [cs]*, Mar. 2021, arXiv: 2010.11398.

[38] N. Papernot, M. Abadi, Ú. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data," *arXiv e-prints*, p. arXiv:1610.05755, Oct. 2016.

[39] M. Parasol, "The impact of China's 2016 Cyber Security Law on foreign technology firms, and on China's big data and Smart City dreams," *Computer Law & Security Review*, vol. 34, no. 1, pp. 67–98, Feb. 2018.

[40] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *arXiv preprint arXiv:1912.13445*, 2019.

[41] M. A. Rahman, T. Rahman, R. Laganière, N. Mohammed, and Y. Wang, "Membership inference attack against differentially private deep learning model." *Trans. Data Priv.*, vol. 11, no. 1, pp. 61–79, 2018.

[42] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.
[43] F. Sattler, S. Wiedemann, K.-R. Muller, and W. Samek, "Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.
[44] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership Inference Attacks against Machine Learning Models," *arXiv:1610.05820 [cs, stat]*, Mar. 2017, arXiv: 1610.05820.
[45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[46] U. Tantipongpipat, C. Waites, D. Boob, A. A. Siva, and R. Cummings, "Differentially Private Synthetic Mixed-Type Data Generation For Unsupervised Learning," *arXiv:1912.03250 [cs, stat]*, Dec. 2020, arXiv: 1912.03250.
[47] R. Torkzadehmahani, P. Kairouz, and B. Paten, "DP-CGAN: Differentially Private Synthetic Data and Label Generation," *arXiv:2001.09700 [cs, stat]*, Jan. 2020, arXiv: 2001.09700.
[48] Q. Wu, K. He, and X. Chen, "Personalized Federated Learning for Intelligent IoT Applications: A Cloud-Edge Based Framework," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 35–44, 2020.
[49] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *CoRR*, vol. abs/1708.07747, 2017.
[50] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou, "Differentially Private Generative Adversarial Network," *arXiv:1802.06739 [cs, stat]*, Feb. 2018, arXiv: 1802.06739.
[51] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy Efficient Federated Learning Over Wireless Communication Networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.
[52] J. Yoon, J. Jordon, and M. van der Schaar, "PATE-GAN: Generating synthetic data with differential privacy guarantees," in *International Conference on Learning Representations*, 2019.
[53] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "{BatchCrypt}: Efficient homomorphic encryption for {Cross-Silo} federated learning," in *2020 USENIX Annual Technical Conference (USENIX ATC 20)*, 2020, pp. 493–506.