On the Privacy of Federated Pipelines

Reza Nasirigerdeh
Technical University of Munich
Freising, Germany
reza.nasirigerdeh@tum.de

Reihaneh Torkzadehmahani
Technical University of Munich
Freising, Germany
reihaneh.torkzadehmahani@tum.de

Jan Baumbach*
University of Hamburg & University of Southern Denmark
Hamburg, Germany
jan.baumbach@uni-hamburg.de

David B. Blumenthal*
Technical University of Munich
Freising, Germany
david.blumenthal@wzw.tum.de

ABSTRACT

Federated learning (FL) is becoming an increasingly popular machine learning paradigm in application scenarios where sensitive data available at various local sites cannot be shared due to privacy protection regulations. In FL, the sensitive data never leaves the local sites and only model parameters are shared with a global aggregator. Nonetheless, it has recently been shown that, under some circumstances, the private data can be reconstructed from the model parameters, which implies that data leakage can occur in FL. In this paper, we draw attention to another risk associated with FL: Even if federated algorithms are individually privacy-preserving, combining them into pipelines is not necessarily privacy-preserving. We provide a concrete example from genome-wide association studies, where the combination of federated principal component analysis and federated linear regression allows the aggregator to retrieve sensitive patient data by solving an instance of the multidimensional subset sum problem. This supports the increasing awareness in the field that, for FL to be truly privacy-preserving, measures have to be undertaken to protect against data leakage at the aggregator.

CCS CONCEPTS

• **Security and privacy**; • **Computing methodologies** → *Machine learning*; • **Applied computing** → Life and medical sciences;

KEYWORDS

Federated Learning; Privacy; Genome-Wide Association Studies; Multidimensional Subset Sum; Integer Linear Programming

ACM Reference Format:
Reza Nasirigerdeh, Reihaneh Torkzadehmahani, Jan Baumbach, and David B. Blumenthal. 2021. On the Privacy of Federated Pipelines. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3404835.3462996

*Joint senior authors.

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3462996

1 INTRODUCTION

Over the past years, machine learning models have become ubiquitous in various fields, including healthcare and biomedicine [6, 9]. In order for machine learning models to yield robust and reliable results, it is often necessary to have access to large-scale datasets. This, however, constitutes a challenge in scenarios where sensitive data is scattered across various local sites (e. g. hospitals) and cannot liberally be pooled without violating privacy-protection regulations such as the EU General Data Protection Regulation [4, 7, 17].

A popular approach for learning from sensitive distributed data while addressing the privacy concerns is federated learning (FL) [13, 14]. In FL, the sensitive raw data does not leave the local sites and only model parameters are shared with an untrusted central aggregator. This, however, does not imply that FL is proof against data leakage, because it has recently been shown that, under certain circumstances, the sensitive raw data can be reconstructed from the exchanged model parameters [16, 24].

In practice, federated systems might consist of a pipeline of several federated algorithms, where the output of one federated algorithm is employed as part of the input of another one. For instance, a federated system may consist of a federated pipeline including steps such as federated data normalization, federated dimensionality reduction, and federated linear regression. In this paper, we show that, even under the assumption that the individual federated algorithms are indeed privacy-preserving, leakage of sensitive data can occur in such federated pipelines.

To establish our result, we provide a concrete example from the field of genetics. More precisely, we show that if genome-wide association studies (GWAS) are carried out in a federated manner using state-of-the-art federated implementations of the involved algorithms, the data necessary to reconstruct the sensitive patient data is available at the aggregator. Moreover, we empirically show that, although the reconstruction problem is weakly *NP*-complete, we can solve it for up to around 80 patients on a standard notebook.

The remainder of this article is organized as follows: In Section 2, we present the federated GWAS pipeline. In Section 3, we show how the aggregator can reconstruct the sensitive raw data from the model parameters. In Section 4, we report computational results. In Section 5, we discuss the implications of our findings.

2 FEDERATED GWAS PIPELINE

Over the past decade, GWAS [2] have resulted in enormous advances in the field of complex disease genetics [22, 25]. The goal of

GWAS is to associate variants observed in sequenced genomes of individuals with a phenotype of interest (typically, a disease) and thereby individuate genetic risk factors. GWAS have been used to discover thousands of unique associations, and have opened up new approaches for the prevention or treatment of diseases.

Positions in the genome where genetic variants have been observed in the population are called *single nucleotide polymorphisms* (SNPs). The predominant genetic variant is called *major allele*; other variants are called *minor alleles* [22]. Let n and m be the numbers of patients and SNPs. In GWAS, we are given an n-dimensional vector of phenotypes $\mathbf{y}$ and a mutation matrix $\mathbf{A} = (a_{i,j}) \in \{0,1,2\}^{n \times m}$ whose entry $a_{i,j}$ contains the number of minor alleles of patient i at SNP j. Each SNP j is tested in an individual association test

$$\mathbf{y} \sim \beta_0 + \beta_1 \cdot \mathbf{A}_{\bullet,j} + \sum_{r=1}^{R} \beta_{r+1} \cdot \mathbf{c}_r, \tag{1}$$

where $\mathbf{A}_{\bullet,j}$ denotes the j^{th} column of $\mathbf{A}$ and $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_R]$ contains n-dimensional vectors of confounders such as age or sex. Typically, simple models such as linear or logistic regression are used for the association tests [15, 25].

Since GWAS deal with possibly large cohorts of individuals with mixed ancestry or cryptic relatedness, these factors should be controlled for to avoid false positives [22]. This is typically done via principal component analysis (PCA) [3]. More precisely, PCA is used to compute the first K eigenvectors $\mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_K] \in \mathbb{R}^{n \times K}$ of the sample by sample covariance matrix $\mathbf{A}\mathbf{A}^\top$ ($K = 20$ is an often used choice in GWAS). Subsequently, these eigenvectors are included into the association test as covariates [8, 18]:

$$\mathbf{y} \sim \beta_0 + \beta_1 \cdot \mathbf{A}_{\bullet,j} + \sum_{r=1}^{R} \beta_{r+1} \cdot \mathbf{c}_r + \sum_{k=1}^{K} \beta_{k+R+1} \cdot \mathbf{g}_k \tag{2}$$

Note that this use of PCA is different from standard feature reduction PCA, where the leading eigenvectors of the feature covariance matrix $\mathbf{A}^\top\mathbf{A}$ are computed. In the sequel, $\mathbf{X}^j := [\mathbf{1}, \mathbf{A}_{\bullet,j}, \mathbf{C}, \mathbf{G}]$ denotes the design matrix of the association test for SNP j.

The vector of phenotypes $\mathbf{y}$, the mutation matrix $\mathbf{A}$, and the matrix of confounders $\mathbf{C}$ contain extremely sensitive patient data. However, jointly analyzing data available at different sites is often necessary to reach the sample sizes required for obtaining statistically significant results in the association tests. Consequently, techniques are needed which allow for joint GWAS and at the same time preserve the privacy of the patients.

Arguably, the most promising privacy-preserving technique for GWAS is FL [15, 23]. Alternative privacy-preserving techniques such as secure multi-party computation that have been used for GWAS are computationally very expensive and hence do not scale to large cohorts [3]. FL approaches are distributed algorithms which work on either horizontally (i. e., sample-wise) or vertically (i. e., feature-wise) partitioned datasets and only share model parameters with a global aggregator. In the context of GWAS, the data is horizontally partitioned: Each local site holds a subset of the rows of the phenotype vector $\mathbf{y}$, the mutation matrix $\mathbf{A}$, and the matrix of confounders $\mathbf{C}$ (cf. top panel in Figure 1).

For carrying out the association test described in eq. (2) in a federated manner, two ingredients are required: Firstly, a federated

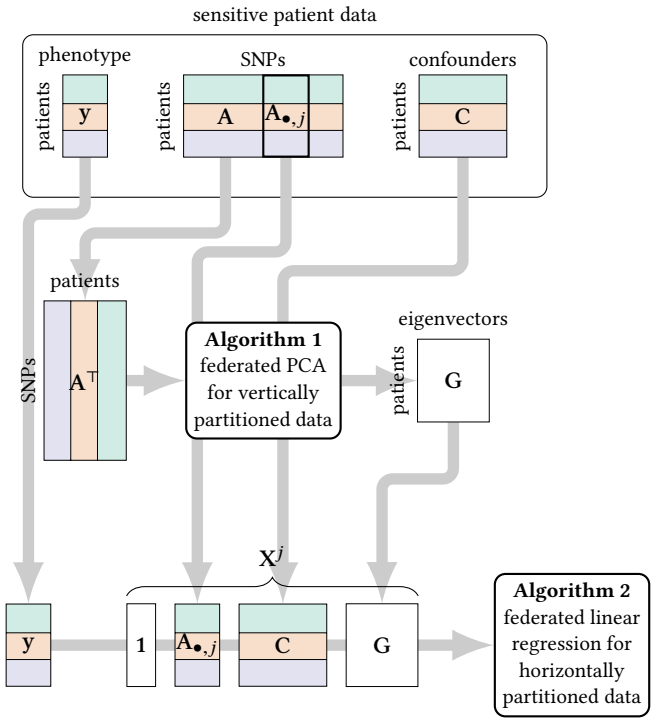

Figure 1: Federated GWAS pipeline. The colors indicate how the data is distributed across the local sites.

PCA algorithm for vertically partitioned data is needed to compute the eigenvector matrix $\mathbf{G}$. Note that the algorithm for computing $\mathbf{G}$ needs to be designed for vertically partitioned data, because $\mathbf{G}$ contains the leading eigenvectors of the sample covariance matrix $\mathbf{A}\mathbf{A}^\top = (\mathbf{A}^\top)^\top(\mathbf{A}^\top)$. From a technical point of view, samples and features hence switch roles in the PCA (i. e., SNPs are viewed as samples and patients as features). Secondly, we must have access to a federated linear regression algorithm that works on horizontally partitioned design matrices $\mathbf{X}^j$ and response variables $\mathbf{y}$. Figure 1 visualizes the entire federated GWAS pipeline.

3 DATA LEAKAGE AT THE AGGREGATOR

For assessing possible privacy leakage in the pipeline shown in Figure 1, we need to know which data the federated PCA and linear regression algorithms share with the aggregator. This depends on how exactly these algorithms are implemented. For this paper, we base our analysis on the federated PCA algorithm presented in [10] and the federated linear regression suggested in [15]. To the best of our knowledge, these are the only federated PCA and linear regression algorithms which meet the requirements of federated GWAS. The algorithms share the following data with the aggregator:

- In the federated PCA algorithm presented in [10], the matrix of eigenvectors $\mathbf{G}$ is computed at the aggregator.
- The federated linear regression suggested in [15] shares the matrix products $\mathbf{X}^{j\top}\mathbf{X}^j$ and $\mathbf{X}^{j\top}\mathbf{y}$ with the aggregator.

Henceforth, we assume that, individually, the algorithms presented in [10] and [15] do not leak any sensitive data. We will show

that, even if this is the case, data leakage can occur if the algorithms are used jointly in the pipeline visualized in Figure 1. The main building block for this result is the following Proposition 3.1:

Proposition 3.1. *Assume that the algorithms presented in [10] and [15] are used in the federated GWAS pipeline shown in Figure 1. Moreover, let* $\mathbf{d}$ *be a column of the sensitive data* $\mathbf{D} \coloneqq [\mathbf{y}, \mathbf{A}_{\bullet,j}, \mathbf{C}]$ *and let* $\mathbb{D}$ *be the domain of* $\mathbf{d}$*. Then* $\mathbf{d} \in \mathcal{S}$*, where* $\mathcal{S}$ *is the solution space to a constrained system of linear equations*

$$\mathbf{H}\boldsymbol{\chi} = \mathbf{b} \tag{3}$$

$$\boldsymbol{\chi} \in \mathbb{D}^n \tag{4}$$

whose left-hand side $\mathbf{H} \in \mathbb{R}^{(K+1)\times n}$ *and right-hand side* $\mathbf{b} \in \mathbb{R}^{K+1}$ *are known to the aggregator.*

Proof. Let $\mathbf{H} \coloneqq [\mathbf{1}, \mathbf{G}]^\top$ and $\mathbf{b} \coloneqq \mathbf{H}\mathbf{d}$. Then $\mathbf{d} \in \mathcal{S}$ holds by construction. $\mathbf{H}$ is known to the aggregator, because the aggregator computes $\mathbf{G}$ itself [10]. If $\mathbf{d} = \mathbf{y}$, then $\mathbf{b}$ is known to the aggregator, because it appears as a block in the matrix product ${\mathbf{X}^j}^\top \mathbf{y}$ shared with the aggregator [15]. Otherwise, $\mathbf{b}$ is known to the aggregator because it is an element of the matrix product ${\mathbf{X}^j}^\top \mathbf{X}^j$, which, again, is shared with the aggregator [15]. □

By Proposition 3.1, we know that $\mathbf{d} \in \mathcal{S}$ and that $\mathbf{H}$ and $\mathbf{b}$ are known to the aggregator. Therefore, if it also holds that $|\mathcal{S}| = 1$, the aggregator can retrieve the sensitive data $\mathbf{d}$ by solving the constrained system of linear equations given in eqs. (3) and (4). Although, in theory, it might be the case that $|\mathcal{S}| > 1$, in practice, we usually have $|\mathcal{S}| = 1$. This is because $\mathbf{G}$, and hence also $\mathbf{H}$ and $\mathbf{b}$, are real-valued, whereas we have $\mathbb{D} = \{0, \dots, \kappa\}$ for some constant $\kappa \in \mathbb{N}$ in the most relevant cases. In particular, we have $\mathbb{D} = \{0, 1, 2\}$ if $\mathbf{d}$ is the SNP column $\mathbf{A}_{\bullet,j}$ and $\mathbb{D} = \{0, \dots, \kappa\}$ if $\mathbf{d}$ is a binary or categorical phenotype or confounder with κ categories. In Section 4, we empirically show that, for a real GWAS dataset, $|\mathcal{S}| = 1$ indeed holds both for the SNP column and for the binary confounder sex.

The next question is how the aggregator can solve the constrained system of linear equations given in eqs. (3) and (4). For answering this question, we focus on the situation where $\mathbb{D} = \{0, \dots, \kappa\}$, since this covers the most important cases as mentioned above. The most straightforward approach for solving the constrained system of linear equations is to view it as an instance of integer linear programming (ILP) and solve it via ILP solvers such as Gurobi [11] or CPLEX [12]. Although general ILP is *NP*-hard, these commercial solvers often exhibit good performance in practice.

Proposition 3.2 states that, if $\mathbb{D} = \{0, \dots, \kappa\}$, the problem specified in eqs. (3) and (4) is equivalent to the d-dimensional subset sum problem (d-SSP) [5] with $d = K + 1$. For constant d, d-SSP is only weakly *NP*-complete and can be solved in pseudo-polynomial time via dynamic programming (cf. [5] for details). Since the number of eigenvectors K can be assumed to be constant (as mentioned above, $K = 20$ is a typical choice in GWAS), it follows that our constrained system of linear equations is weakly *NP*-complete and can be solved in pseudo-polynomial time. Note, however, that this result is relevant only from a theoretical point of view: Although the pseudo-polynomial dynamic programming algorithm for d-SSP has better asymptotic complexity than general ILP solving, it is by far too slow for practical usage (again, we refer to [5] for details).

Proposition 3.2. *If* $\mathbb{D} = \{0, \dots, \kappa\}$ *for some constant* $\kappa \in \mathbb{N}$*, the constrained system of linear equations given in eqs.* (3) *and* (4) *is polynomially equivalent to d-SSP with* $d = K + 1$*, where* K *is the number of eigenvectors contained in* $\mathbf{G}$*.*

Proof. Given a integer-valued matrix of coefficients $\mathbf{S} \in \mathbb{Z}^{d\times n}$ and an integer-valued target vector $\mathbf{t} \in \mathbb{Z}^d$, d-SSP asks to decide whether there is an index set $I \subseteq \{1, \dots, n\}$ such that $\sum_{i\in I} \mathbf{S}_{\bullet,i} = \mathbf{t}$, where $\mathbf{S}_{\bullet,i}$ denotes the i^{th} column of $\mathbf{S}$. For reducing d-SSP with $d = K + 1$ to the problem specified in eqs. (3) and (4), it suffices to set $\mathbf{H} \coloneqq \mathbf{S}$, $\mathbf{b} \coloneqq \mathbf{t}$, and $\mathbb{D} \coloneqq \{0, 1\}$, and interpret the decision variable $\boldsymbol{\chi}$ as the indicator vector of the index set I.

For the reduction in the other direction, we first transform our instance $\mathcal{I} \coloneqq (\mathbf{H}, \mathbf{b}, \{0, \dots, \kappa\})$ into an equivalent instance $\mathcal{I}' \coloneqq (\mathbf{H}', \mathbf{b}, \{0, 1\})$. If $\kappa = 1$, we set $\mathbf{H}' \coloneqq \mathbf{H}$. Otherwise, we obtain $\mathbf{H}'$ by adding $\kappa - 1$ additional copies of each column of $\mathbf{H}$. Clearly, $\mathcal{I}$ and $\mathcal{I}'$ are equivalent: If $\boldsymbol{\chi} \in \{0, \dots, \kappa\}^n$ solves $\mathcal{I}$, then we can obtain a solution $\boldsymbol{\chi}' \in \{0, 1\}^{\kappa n}$ for $\mathcal{I}'$ by setting $\chi'_{i'} \coloneqq 1$ for χ_i of the κ columns $i' \in I'(i)$ of $\mathbf{H}'$ obtained from the column i of $\mathbf{H}$. Conversely, we can transform a solution $\boldsymbol{\chi}'$ for $\mathcal{I}'$ into a solution $\boldsymbol{\chi}$ for $\mathcal{I}$ by setting $\chi_i \coloneqq \sum_{i'\in I'(i)} \chi'_{i'}$. Moreover, the size of $\mathcal{I}'$ is polynomial in the size of $\mathcal{I}$, because κ is a constant.

To finish the prove, we pick a constant $\lambda \in \mathbb{R}$ such that we have $\lambda\mathbf{H}' \in \mathbb{Z}^{\kappa n\times d}$ and $\lambda\mathbf{b} \in \mathbb{Z}^d$. Since computers support only finite precision, we can assume w. l. o. g. that such a constant λ exists. Since $\boldsymbol{\chi}'$ solves $\mathcal{I}'$ if and only if it solves the instance $(\lambda\mathbf{H}', \lambda\mathbf{b}, \{0, 1\})$, the proposition follows from setting $\mathbf{S} \coloneqq \lambda\mathbf{H}$ and $\mathbf{t} \coloneqq \lambda\mathbf{b}$, and defining $\boldsymbol{\chi}'$ as the indicator vector of the index set I. □

4 COMPUTATIONAL RESULTS

To assess whether the data leakage issue described in the previous section is relevant from a practical point of view, we tried to solve the constrained system of linear equations given in eqs. (3) and (4) with the ILP solver Gurobi [11]. Additionally, we tested if, in the considered scenarios, we indeed have $|\mathcal{S}| = 1$. Recall that in this case the aggregator has all information required to reconstruct the sensitive data $\mathbf{d}$. For testing whether this is the case, we add the cut

$$\sum_{i=1}^{n} |\chi_i - d_i| \geq 1 \tag{5}$$

to the ILP and use Gurobi to check if the resulting modified ILP is infeasible. For adding eq. (5) to the ILP, we transform it into one or several linear constraints using standard ILP modeling techniques [1]. For the binary confounder sex, we have $\mathbb{D} = \{0, 1\}$ and eq. (5) is hence equivalent to

$$\sum_{i\in I_0} \chi_i + \sum_{i\in I_1} (1 - \chi_i) \geq 1, \tag{6}$$

where $I_l = \{i \in \{1, \dots, n\} \mid d_i = l\}$ for $l \in \mathbb{D}$. For the SNPs, we have $\mathbb{D} = \{0, 1, 2\}$. In this case, we linearize eq. (5) as

$$\sum_{i\in I_0} \chi_i + \sum_{i\in I_1} \xi_i + \sum_{i\in I_2} (2 - \chi_i) \geq 1 \tag{7}$$

$$\xi_i - \chi_i - 2\eta_i \leq -1 \quad \forall i \in I_1 \tag{8}$$

$$\xi_i + \chi_i + 2\eta_i \leq 3 \quad \forall i \in I_1, \tag{9}$$

where $\boldsymbol{\xi} \in \mathbb{R}^{|I_1|}$ and $\boldsymbol{\eta} \in \{0, 1\}^{|I_1|}$ are additional auxiliary variables.

Table 1 shows the results for the binary confounder sex as well as for three randomly selected SNPs of the GWAS dataset presented in [19]. For each target variable, we randomly selected ten batches of, respectively, $n = 70$ and $n = 80$ patients, and recorded how often we were able to reconstruct the sensitive data and establish uniqueness of the solution. The tests were run on a laptop with an Intel Core i7 CPU (4 physical cores, 1.8 GHz) and 16 GB of main memory, and we set a time limit of one hour for each run.

Table 1: Success rates for reconstructing sensitive patient data and proving uniqueness of the solution using the ILP solver Gurobi with a time limit of one hour.

	SNPs		Sex	
Number of patients	$n = 70$	$n = 80$	$n = 70$	$n = 80$
Reconstructed data	80 %	40 %	100 %	70 %
Reconstruction timeout	20 %	60 %	0 %	30 %
Established uniqueness	73 %	33 %	100 %	50 %
Uniqueness timeout	27 %	67 %	0 %	50 %

For $n = 70$, we could reconstruct the sensitive data and establish uniqueness in most cases. For $n = 80$, the success rates dropped since we reached the time limit more often. However, it never happened that we reconstructed a wrong solution $\boldsymbol{\chi} \neq \mathbf{d}$, or that the ILP with the additional constraints given in eqs. (7) to (9) terminated with a feasible solution. This is strong evidence for the conjecture that, in practice, the constrained system of linear equations given in eqs. (3) and (4) indeed has exactly one feasible solution, which implies that all information necessary to reconstruct the sensitive data is available at the aggregator.

Figure 2 visualizes the distributions of reconstruction times for the runs which terminated within the time limit. The most important observation is that the reconstruction times vary a lot. Even for the batches containing 80 samples, we could sometimes reconstruct the sensitive data in just a few seconds. As is often the case for *NP*-complete problems, instance size is hence a bad predictor for instance complexity [21]. This is particularly relevant here, because it implies that the weak *NP*-completeness of our constrained system of linear equations does not provide a strong security guarantee: Although the general problem is hard, individual instances might be easy even for larger numbers of patients.

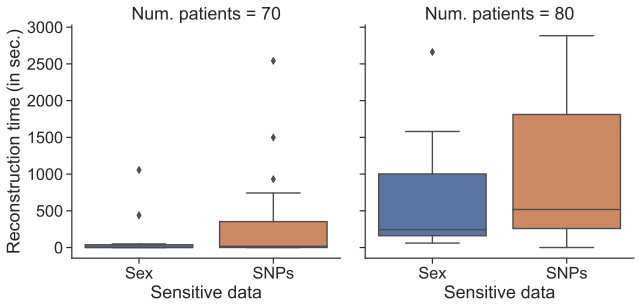

Figure 2: Reconstruction times for ILP runs which terminated within the time limit of one hour.

5 CONCLUSIONS

In this paper, we have shown that if state-of-the-art implementations of federated PCA and federated linear regression are used in a federated GWAS pipeline, the aggregator can reconstruct sensitive patient data by solving a constrained system of linear equations. We have proved that this reconstruction problem is equivalent to d-SSP, which implies that it is weakly *NP*-complete and solvable in pseudo-polynomial time via dynamic programming. Furthermore, we have demonstrated that, in practice, sensitive data for up to 80 patients can often be reconstructed in around an hour on a standard laptop using ILP techniques. Moreover, the reconstruction time varies a lot from instance to instance, which implies that the problem complexity does not provide a strong security guarantee, since easy instances can be solved quickly.

We conclude this paper by pointing out to three implications of our findings. Firstly, improved federated PCA algorithms for vertically partitioned data are needed for privacy-preserving federated GWAS. Recall that in the state-of-the-art solution [10], the eigenvectors of the patient by patient covariance matrix are finally computed by the aggregator, which leads to the privacy leakage described in Section 3. Consequently, federated PCA algorithms are required where the eigenvectors are computed at the local sites.

Secondly, our findings emphasize the evident but sometimes overlooked fact that, by itself, FL is not necessarily privacy-preserving. Although the raw data stays with the data holders, it can sometimes be reconstructed from the shared model parameters, and the scenarios where this can happen are often subtle and difficult to oversee at the time of implementation. Allegedly privacy-preserving FL approaches should hence always be analyzed very carefully for potential vulnerabilities before deployment.

Ultimately, however, we argue that for FL to be truly privacy-preserving, it should incorporate other privacy-preserving techniques such as differential privacy (DP) or homomorphic encryption (HE). In such hybrid approaches, noise is added to the model parameters before sending them to the aggregator (FL & DP) [24, 26], or the model parameters are encrypted at the clients and aggregated in the encrypted domain (FL & HE) [20, 27]. Especially in application domains such as genetics where long-term security is at stake, we strongly recommend using such hybrid schemes.

ACKNOWLEDGMENTS

R. N., R. T., and J. B. received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement (grant no. 826078 and grant no. 777111). This publication reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains. J. B. was supported by the German Federal Ministry of Education and Research (BMBF) within the e:Med framework (grant no. 01ZX1908A and grant no. 01ZX1910D) and within the CLINSPECT-M framework (grant no. 031L0214A). J. B. was partially funded by his VILLUM Young Investigator Grant (grant no. 13154).

REFERENCES

[1] Dimitris Bertsimas and John N. Tsitsiklis. 1997. *Introduction to Linear Optimization.* Athena Scientific, Belmont.

[2] William S Bush and Jason H Moore. 2012. Chapter 11: Genome-wide association studies. *PLoS Comput. Biol.* 8, 12 (2012), e1002822. https://doi.org/10.1371/journal.pcbi.1002822

[3] Hyunghoon Cho, David J Wu, and Bonnie Berger. 2018. Secure genome-wide association analysis using multiparty computation. *Nat. Biotechnol.* 36, 6 (2018), 547–551. https://doi.org/10.1038/nbt.4108

[4] Aloni Cohen and Kobbi Nissim. 2020. Towards formalizing the GDPR's notion of singling out. *Proc. Natl. Acad. Sci. U.S.A.* 117, 15 (04 2020), 8344–8352. https://doi.org/10.1073/pnas.1914598117

[5] Ioannis Z. Emiris, Anna Karasoulou, and Charilaos Tzovas. 2017. Approximating Multidimensional Subset Sum and Minkowski Decomposition of Polygons. *Math. Comput. Sci.* 11, 1 (2017), 35–48. https://doi.org/10.1007/s11786-017-0297-1

[6] Gökcen Eraslan, Žiga Avsec, Julien Gagneur, and Fabian J Theis. 2019. Deep learning: new computational modelling techniques for genomics. *Nat. Rev. Genet.* 20, 7 (2019), 389–403. https://doi.org/10.1038/s41576-019-0122-6

[7] Yaniv Erlich and Arvind Narayanan. 2014. Routes for breaching and protecting genetic privacy. *Nat. Rev. Genet.* 15, 6 (2014), 409–421. https://doi.org/10.1038/nrg3723

[8] Kevin J Galinsky, Gaurav Bhatia, Po-Ru Loh, Stoyan Georgiev, Sayan Mukherjee, Nick J Patterson, and Alkes L Price. 2016. Fast Principal-Component Analysis Reveals Convergent Evolution of ADH1B in Europe and East Asia. *Am. J. Hum. Genet.* 98, 3 (2016), 456–472. https://doi.org/10.1016/j.ajhg.2015.12.022

[9] Jeremy Goecks, Vahid Jalili, Laura M Heiser, and Joe W Gray. 2020. How machine learning will transform biomedicine. *Cell* 181, 1 (2020), 92–101. https://doi.org/10.1016/j.cell.2020.03.022

[10] Yue-Fei Guo, Xiaodong Lin, Zhou Teng, Xiangyang Xue, and Jianping Fan. 2012. A covariance-free iterative algorithm for distributed principal component analysis on vertically partitioned data. *Pattern Recognit.* 45, 3 (2012), 1211–1219. https://doi.org/10.1016/j.patcog.2011.09.002

[11] Gurobi Optimization LLC 2018. *Gurobi Optimizer Reference Manual.* Gurobi Optimization LLC. http://www.gurobi.com

[12] IBM Coorporation 2016. *IBM ILOG CPLEX Optimization Studio CPLEX User's Manual.* IBM Coorporation. www.cplex.com

[13] Peter Kairouz and H. Brendan McMahan. 2021. Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.* 14, 1 (2021). https://doi.org/10.1561/2200000083

[14] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *AISTATS 2017 (PMLR, Vol. 54)*, Aarti Singh and Jerry Zhu (Eds.). PMLR, Fort Lauderdale, 1273–1282. http://proceedings.mlr.press/v54/mcmahan17a.html

[15] Reza Nasirigerdeh, Reihaneh Torkzadehmahani, Julian Matschinske, Tobias Frisch, Markus List, Julian Späth, Stefan Weiß, Uwe Völker, Dominik Heider, Nina Kerstin Wenke, Tim Kacprowski, and Jan Baumbach. 2020. sPLINK: A Federated, Privacy-Preserving Tool as a Robust Alternative to Meta-Analysis in Genome-Wide Association Studies. bioRxiv. https://doi.org/10.1101/2020.06.05.136382

[16] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In *SP 2019*. IEEE, 739–753. https://doi.org/10.1109/SP.2019.00065

[17] The European Parliament and the Council of the European Union. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Official Journal of the European Union* L 119 (2016), 1–88.

[18] Alkes L Price, Nick J Patterson, Robert M Plenge, Michael E Weinblatt, Nancy A Shadick, and David Reich. 2006. Principal components analysis corrects for stratification in genome-wide association studies. *Nat. Genet.* 38, 8 (2006), 904–909. https://doi.org/10.1038/ng1847

[19] Elizabeth A Regan, John E Hokanson, James R Murphy, Barry Make, David A Lynch, Terri H Beaty, Douglas Curran-Everett, Edwin K Silverman, and James D Crapo. 2010. Genetic epidemiology of COPD (COPDGene) study design. *COPD* 7, 1 (2010), 32–43. https://doi.org/10.3109/15412550903499522

[20] Md. Nazmus Sadat, Md Momin Al Aziz, Noman Mohammed, Feng Chen, Xiaoqian Jiang, and Shuang Wang. 2019. SAFETY: Secure gwAs in Federated Environment through a hYbrid Solution. *IEEE ACM Trans. Comput. Biol. Bioinform.* 16, 1 (2019), 93–102. https://doi.org/10.1109/TCBB.2018.2829760

[21] Christine Solnon. 2019. Experimental Evaluation of Subgraph Isomorphism Solvers. In *GbRPR 2019 (LNCS, Vol. 11510)*, Donatello Conte, Jean-Yves Ramel, and Pasquale Foggia (Eds.). Springer, Cham, 1–13. https://doi.org/10.1007/978-3-030-20081-7_1

[22] Vivian Tam, Nikunj Patel, Michelle Turcotte, Yohan Bossé, Guillaume Paré, and David Meyre. 2019. Benefits and limitations of genome-wide association studies. *Nat. Rev. Genet.* 20, 8 (2019), 467–484. https://doi.org/10.1038/s41576-019-0127-1

[23] Reihaneh Torkzadehmahani, Reza Nasirigerdeh, David B. Blumenthal, Tim Kacprowski, Markus List, Julian Matschinske, Julian Späth, Nina Kerstin Wenke, Béla Bihari, Tobias Frisch, Anne Hartebrodt, Anne-Christin Hausschild, Dominik Heider, Andreas Holzinger, Walter Hötzendorfer, Markus Kastelitz, Rudolf Mayer, Cristian Nogales, Anastasia Pustozerova, Richard Röttger, Harald H. H. W. Schmidt, Ameli Schwalber, Christof Tschohl, Andrea Wohner, and Jan Baumbach. 2020. Privacy-preserving Artificial Intelligence Techniques in Biomedicine. arXiv:2007.11621 [cs.CR]

[24] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. 2019. A Hybrid Approach to Privacy-Preserving Federated Learning. In *AISec@CCS 2019*, Lorenzo Cavallaro, Johannes Kinder, Sadia Afroz, Battista Biggio, Nicholas Carlini, Yuval Elovici, and Asaf Shabtai (Eds.). ACM, 1–11. https://doi.org/10.1145/3338501.3357370

[25] Peter M Visscher, Naomi R Wray, Qian Zhang, Pamela Sklar, Mark I McCarthy, Matthew A Brown, and Jian Yang. 2017. 10 Years of GWAS Discovery: Biology, Function, and Translation. *Am. J. Hum. Genet.* 101, 1 (2017), 5–22. https://doi.org/10.1016/j.ajhg.2017.06.005

[26] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farhad Farokhi, Shi Jin, Tony Q. S. Quek, and H. Vincent Poor. 2020. Federated Learning With Differential Privacy: Algorithms and Performance Analysis. *IEEE Trans. Inf. Forensics Secur.* 15 (2020), 3454–3469. https://doi.org/10.1109/TIFS.2020.2988575

[27] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. 2020. BatchCrypt: Efficient Homomorphic Encryption for Cross-Silo Federated Learning. In *USENIX ATC 2020*, Ada Gavrilovska and Erez Zadok (Eds.). USENIX Association, 493–506. https://www.usenix.org/conference/atc20/presentation/zhang-chengliang