2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

Using Synthetic Data to Reduce Model Convergence Time in Federated Learning

Fida K. Dankar
Computer Science program
NYUAD
Abu Dhabi, UAE
fd2242@nyu.edu

Nisha Madathil
Information Systems and Security
UAEU
Al Ain, UAE
201990156@uaeu.ac.ae

***Abstract*— Federated Learning (FL) is a hot new topic in collaborative training of machine learning problems. It is a privacy-preserving distributed machine learning approach, allowing multiple clients to jointly train a global model under the coordination of a central server, while keeping their sensitive data private. The problem with FL systems is that they require intense communication between the server and clients to achieve the final machine learning model. Such complexity increases with the number of clients participating and the complexity of the model sought. In this paper, we introduce synthetic data generation into FL systems with the intention of reducing the number of iterations required for model convergence. In this novel method, clients generate synthetic datasets modeling their private data. The synthetic datasets are then sent to the central server and are used to generate a cognizant initial model. Our experiments show that such conscious method for generating the initial model lowers the number of iterations by a factor of more than 4 without affecting the model accuracy. As such it enhances the overall efficiency of FL systems.**

Keywords—synthetic data, federated learning, privacy preserving technologies.

I. Introduction

Research and decision-making are increasingly dependent on the analysis of huge amounts of sensitive and personal data. At the same time, growing concerns about the privacy implications of data sharing call for approaches that preserve fundamental rights to privacy. Federated learning (FL) is a fairly recent approach that permits the collaborative analysis of data from several sources simultaneously without the need to share the data across the sources, thus keeping the control with the data custodian [1]. This is believed to increase privacy and lower data breach instances.

FL enables data custodians to build a high-quality machine learning models on their distributed data with the help of a central server. Briefly, the server chooses an initial global model and sends it to the distributed custodians or clients. The clients train local models using the initial model and their private data and send their results back to the server. The server aggregates the received local models into one global model and sends it to all clients. This process repeats until the model converges, or a termination condition is met.

The problem with FL systems is that they require intense communication between the server and clients to achieve the final model (in terms of communication rounds as well as message size), which increases with the number of clients participating (such number can reach hundreds of organizations or millions of mobile phones). For example, Mandal and Gong use 1000 iterations to achieve a global regression model [2], and Chen et al use 200 iterations to train a neural network model [3].

In this paper, we introduce synthetic data generation into FL mechanisms to reduce the number of iterations required for model convergence. The method, referred to as FL_SD, proceeds as follows: each client uses their raw data to generate a synthetic dataset of the same size. The synthetic datasets are then sent to the central server and used to generate an initial model. Our experiments show that such conscious method for generating the initial model lowers the number of iterations by a factor of more than 4 in all experiments, and more than 7 in 80% of the experiments, without affecting accuracy. Thus, this new method enhances considerably the overall efficiency of FL systems.

The paper is designed as follows: Section 2 presents the algorithm FL_SD in detail, Section 3 presents the experimental set-up and Section 4 presents the results. The paper concludes in Section 5 with a summary and future work.

II. Methods

The problem can be formally described as follows: s clients ($p_1, p_2, \dots, p_s$), with private data ($d_1, d_2, \dots, d_s$, respectively), wish to train a machine learning model M collectively without sharing their own data. We assume that the data is horizontally distributed across the clients, in other words, all clients have the same features collected for different individuals [4]–[6]. In traditional FL, the clients collaborate to train the model, with the help of a central server, without exposing their personal data. The FL process can be described as follows:

Part 1. Initially, the server generates initial model parameters (IMP), often composed of zero values, and sends it to the different clients.

Part 2. Each client p_i $i \in \{1, \dots, s\}$, uses the received model as initial parameters to train their local data. The resulting local model parameters, $LMPs$, are then sent back to the server. The server aggregates the received results into a global model, GMP, which is sent back to the clients. Many aggregation methods exist in the literature, readers are referred to [3], [7], [8] for examples. Part 2 is repeated until a loss function converges or a predefined stopping criteria is met.

To enhance the time complexity of FL, we propose to generate an initial model that is mindful of the characteristics of the distributed dataset without violating the privacy of

IEEE/ACM ASONAM 2022, November 10-13, 2022
978-1-6654-5661-6/22/$31.00 © 2022 IEEE

clients' local data. The postulation is that a conscious initial model will reduce the number of iterations required to achieve convergence. In short, synthetic datasets are generated by the different clients to simulate their own data. The union of these synthetic datasets is then used to generate an initial model.

Synthetic data (SD) is artificial data that is simulated from real data to mimic its statistical properties. It is considered a safe approach for the wider release of sensitive data as it contains no identifiable information about the dataset it was generated from [9]–[11]. Synthetic datasets are generated from a model that is fit to real data using either statistical or machine learning methods [12]. Statistical methods use a family of predefined distributions to fit the real dataset (such as Gaussian models or Bayesian Networks), while machine learning methods build a machine learning model to fit the real dataset (such as classification and regression trees or generative adversarial networks). The generation process is stochastic, which implies that a different synthetic dataset is generated each time the model is used. It is generally accepted that synthetic data can be shared widely as it is considered non-identifiable and falls (to date) outside the scope of privacy regulations [13].

Our method, FL_SD, involves modification to the initial step of the FL algorithm as follows:

Part 1. Initially, each client generates a synthetic dataset modeling their private data. The generated synthetic datasets are then sent to the server which aggregates all results to generate an initial model, IMP. The model is then sent to all participating clients

Part 2. Same as Part 2 above.

Figure 1 describes the difference in the first step (Part 1) between FL and FL_SD, and Figure 2 describes the remaining iterative steps (part 2) which are identical in both methods.

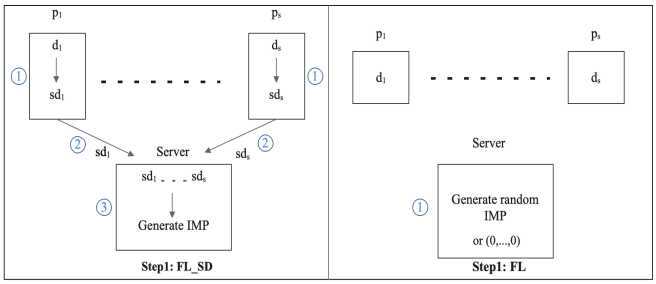

Fig. 1. Part 1 for FL and SD

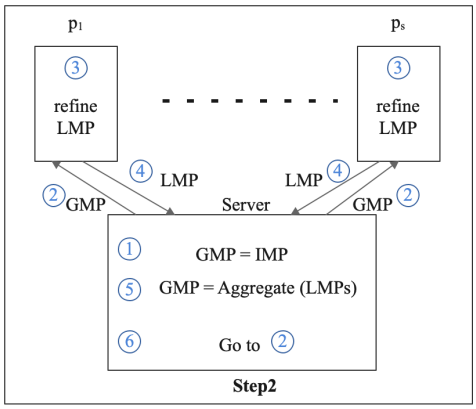

Fig. 2. Part 2 for FL and SD

III. Experiments

A. Algorithms' Overview

In this paper, we use linear regression as our machine learning model as a proof of concept. The FL algorithm for linear regression, FL(LR), is implemented using Python. The initial model (IMP) is set to zero (a common initial model in the literature [7], [8], [14], [15]). For local model aggregation we use the FedAvg method [1]. It uses averaging to calculate the global model. The stopping criteria for the algorithms is determined by two criteria: (i) the number of iterations performed and (ii) the change in the mean square error value (MSE) [16]. More specifically, the stopping criteria is as follows:

Stop if: change in $|MSE| \leq \epsilon$ or maximum iterations value (I) is reached (for some predefined ϵ and I).

On the other hand, the FL_SD algorithm for linear regression, FL_SD(LR), is similar to the FL(LR) algorithm with the exception of the first step that entails the generation of local synthetic datasets ($sd_1, \ldots, sd_s$). Once generated, the synthetic datasets, $sd_1, \ldots, sd_s$, are sent to the central server. In turn, the server runs linear regression on the union of these datasets ($sd_1 \cup \ldots \cup sd_s$), the generated linear regression parameters are set as the IMP.

To generate synthetic data, we use Synthpop non-parametric (SP-np) synthetic data generator [17]. SP-np generates the model by synthesizing the different attributes sequentially. The first attribute is synthesized after estimating its marginal distribution from the raw data, and following attributes are synthesized after estimating their conditional distribution using all prior attributes as predictors. SP-np uses classification and regression trees, CART [18], for estimating the probabilities. Once the model is generated, it is used to produce the synthetic data. SP-np was recognized as a leading synthesizer in multiple prior investigations [19]–[21].

B. Datasets

We use six datasets in our experiments, two contained within the UCI repository[22], two contained within Kaggle [23], one contained within data.world [24], and one contained within Cerner clinical database [25]. The datasets are summarized in Table I.

TABLE I. DATASET USED FOR THE EXPERIMENTS. n DENOTES THE TOTAL NUMBER OF RECORDS, p NUMBER OF FEATURES AND s NUMBER OF CLIENTS.

Datasets	symbol	Description	n	p	s	Response var	Source
Student	D_1	Students performance in Portuguese and math in two schools	1044	30	2	Final grade	UCI
Diabetics	D_2	clinical diabetic dataset	9926	41	5	Length of stay in hospital	Cerner
AutoMPG	D_3	car fuel consumption	392	9	3	Mile per gallon	UCI
Covid	D_4	covid 19 dataset	7264	22	5	Number of deaths	Kaggle
ExcessDeath	D_5	covid 19 dataset	4271	17	5	Excess death	data.world
Weather	D_6	Weather data	37625	16	5	Temperature	Kaggle

C. Algorithm Specifics

Each client divides their datasets into 70% training and 30% testing. An initial model is generated by the server as follows:

- for the FL algorithm parameters are all set to zero
- for the FL_SD, sites generate SD from their training sets. Then the initial model is generated from the SDs as explained previously.

Then, for each iteration i, the algorithm proceeds as follows:

- Each client j generates the local model parameters LMP_j^i using the training dataset and sends the value to the server.
- The server aggregates the local models ($LMP_1^i, \dots, LMP_s^i$) into one global model GMP_i and sends the model to all clients
- Each client j calculates the mean square error MSE_j^i for model GMP_i using the testing set. The local MSE's are then sent to the server for aggregation
- The server aggregates the local MSE's ($MSE_1^i, \dots, MSE_s^i$) into one global error MSE_i using averaging [26].
- The server then checks whether the stopping criteria is satisfied as follows:
  1. If $MSE_{i-1} - MSE_i \leq 0.01$ or $i = 1000$, for the datasets (Student, Diabetics, AutoMPG, Covid, and Weather), then abort.
  2. If $MSE_{i-1} - MSE_i \leq 0.001$ or $i = 1000$, for the dataset (ExcessDeath) then abort (the MSE value for this dataset is very small as is the change in MSE so we opted to allow smaller changes in the MSE values).

As the generation of synthetic data is stochastic, multiple instances generated from the same model will exhibit utility variations. To lower the variability effect on our experiments, we repeated the synthetic data generation 20 times for each dataset. Thus, for each dataset D_i:

1. We run FL(LR) on D_i and report on the number of iterations required (k), the final MSE value (MSE) and the total running time (t)
2. In FL_SD(LR), for each D_i, we repeat the synthetic data generation 20 times. For each repetition we calculate the number of iterations required (k), the MSE value (MSE) and the total running time (t).
3. For each D_i, we denote by:
   a. $E(k)$, $E(MSE)$, and $E(t)$ the averages over the 20 repetitions of the number of iterations, the MSE values, and the computation time respectively.
   b. $R(k)$, $R(MSE)$, and $R(t)$ the ranges over these 20 repetitions, and
   c. $C(k)$, $C(MSE)$, and $C(t)$ the the confidence intervals over the 20 repetitions calculated to 95%.

FL(LR) and FL_SD(LR) where implemented in Python3 on top of two security-PowerEdge-R640 machines with Intel(R) Xeon(R) Silver 4116 CPU @ 2.10GHz family, 251GB RAM size, 480GB disk size with 48 cores each. We used Ubuntu 16.04 LTS as our operating system. Python was augmented with the Scikit-learn library, NumPy, and pandas. For generating synthetic data, we implemented SP-np in python as the available implementation is in R [27].

IV. Results

A. Overall Results

Tables II, III and IV below summarize the results of the experiments. For FL, the tables report the values k, MSE and t. For FL_SD, the tables report on averages ($E(k)$, $E(MSE)$, $E(t)$), ranges ($R(k)$, $R(MSE)$, $R(t)$) and confidence intervals ($C(k)$, $C(MSE)$, $C(t)$).

TABLE II. ITERATION RESULTS FOR EACH DATASET

Datasets	FL_SD			FL
	$E(k)$	$R(k)$	$C(k)$	$\boldsymbol{k}$
Student	12	9 - 15	(11,13)	232
Diabetics	11	8 - 14	(11,12)	141
AutoMPG	6	4 - 6	(5,6)	98
Covid	112	96 - 125	(107,116)	524
ExcessDeath	5	3 - 6	(4,5)	45
Weather	39	36 - 43	(38,40)	291

TABLE III. COMPUTATION (TIME) RESULTS FOR EACH DATASET

Datasets	**FL_SD**			**FL**
	$E(t)$	$R(t)$	$C(t)$	t
Student	37.83579	29.10453 - 47.20548	(35.02719, 40.64439)	697.08685
Diabetics	73.98391	54.86124 - 90.94465	(69.83495, 78.13286)	848.20894
AutoMPG	20.38407	15.25875 - 23.46244	(19.29359, 21.47455)	392.20926
Covid	673.51832	581.67656 - 755.71417	(646.73534, 700.30118)	3144.46725
ExcessDeath	32.74525	22.52476 - 41.91229	(29.84487, 35.64553)	270.43454
Weather	243.43054	222.57572 - 264.97768	(237.67181, 249.18910)	1746.67862

TABLE IV. MSE RESULTS FOR EACH DATASET

Datasets	FL_SD			FL
	$E(MSE)$	$R(MSE)$	$C(MSE)$	MSE
Student	4.976842	4.900642 - 5.019173	(4.963099, 4.990586)	4.984437
Diabetics	6.936350	6.845748 - 7.023188	(6.913138, 6.959564)	6.946724
AutoMPG	14.812964	14.658648 - 14.985132	(14.770867, 14.855057)	14.829121
Covid	225.392476	222.9240 - 227.68818	(224.815976, 225.968732)	225.357623
ExcessDeath	0.034884	0.017374 - 0.05166	(0.028531, 0.038238)	0.037137
Weather	64.141664	62.664778 - 65.29248	(63.792674, 64.490654)	64.137991

Table II shows a significant reduction in number of iterations in FL_SD. k is reduced by a factor bigger than 4.6 in all cases (for all datasets) and bigger than 10 in 50% of the cases.

Table III shows a significant reduction in the total time required for FL_SD. This is an expected result as the number

of iterations were reduced. The total time t is reduced by a factor bigger than 7 in all datasets except one (Covid), in which case, it is reduced by a factor of 4.67.

The reductions in the values of k and t does not affect the accuracy of the result obtained from FL_SD. Table IV reports on these accuracy results. These results indicate a match between $E(MSE)$for FL_SD and MSE for FL in all datasets.

B. *Confidence Intervals*

The results corresponding to the confidence intervals are included in tables II, III and IV and displayed in Figures 3, and 4.

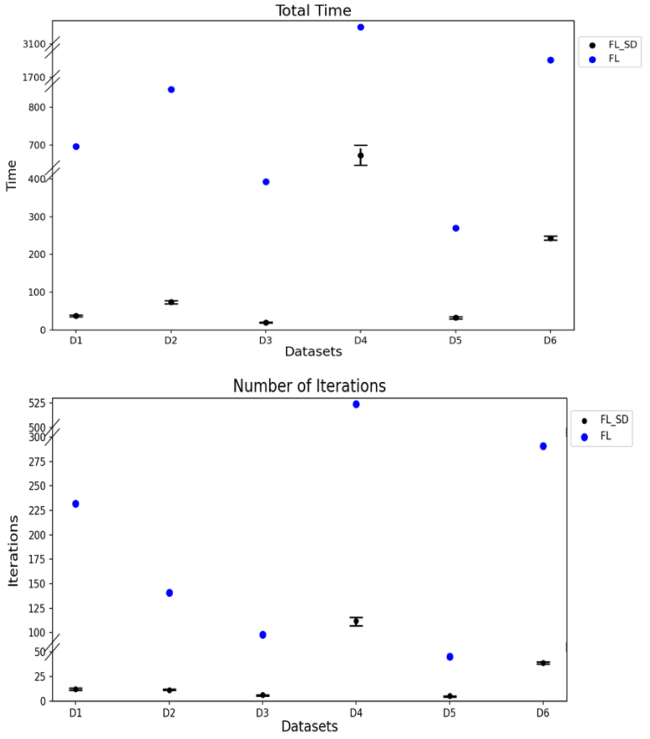

Fig. 3. Upper figure: t and $C(t)$ for FL and FL_SD respectively. Lower figure: k and $C(k)$ for FL and FL_SD respectively

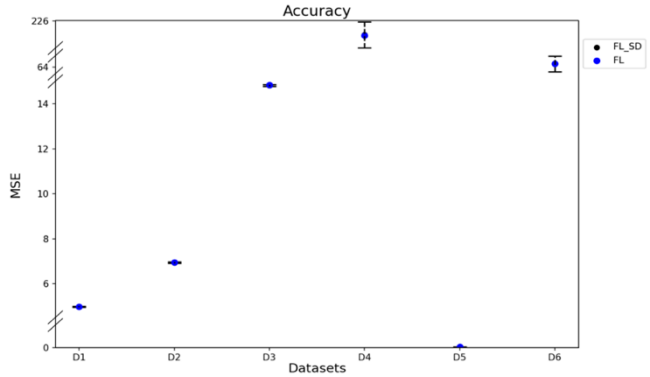

Fig. 4. MSE and $C(MSE)$ for FL and FL_SD respectively.

It is clear from the figures (and the tables) that the variations in the values of t and k across the synthetic data repetitions (as indicated by the confidence intervals and ranges) are small relative to the difference with the corresponding FL values. This indicates the stability of the results. In other words, the variability produced by the synthetic data generation is negligible and has no effect on our conclusions.

Figure 4 displays the values for MSE and $C(MSE)$ for FL and FL_SD respectively. They show that accuracy values are comparable in both algorithms as the values overlap in all dataset instances.

To show the stability of the FL_SD further, we look at the results per repetition in the next section.

C. *Results per Repetition*

Figure 5 displays values of t and k for each of the 20 repetitions against the same values for FL. The figure indicates that for every synthetic dataset generated, there is a significant improvement in terms of total time and number of iterations (which stresses again the stability of the results). In other words, despite the variations in the quality of the synthetic data generated at each iteration, the improvement in terms of computation complexity are always significant when using FL_SD.

Figure 6 displays the accuracy results for every iteration of FL_SD versus the accuracy for FL.

Note that the variations in the MSE values over the 20 repetitions are very small, less than 4.2% for all datasets except ExcessDeath. The variations for ExcessDeath (D_5) are more significant as the MSE value is very small (the highest MSE value obtained in FL_SD is 0.052 versus a value 0.037 for FL).

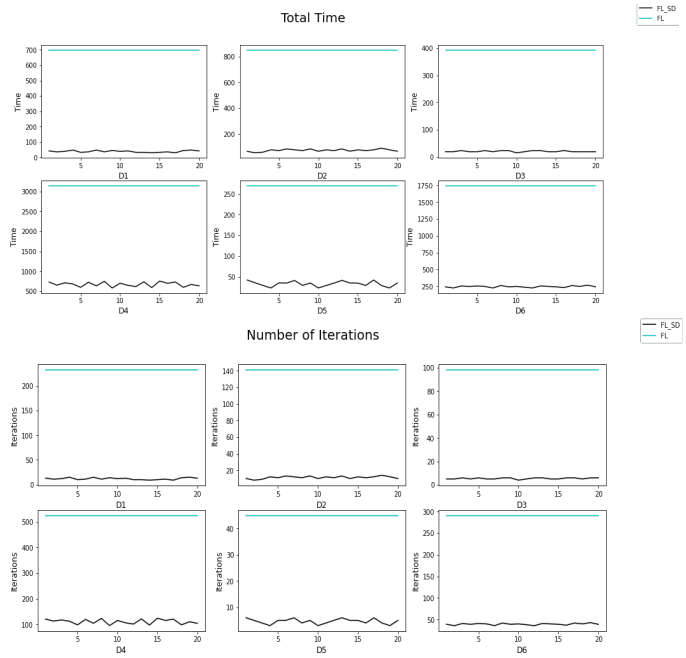

Fig. 5. The figure displays the total time for FL and for all repetitions of FL_SD on top, and the total number of iterations for FL and for all repetitions of FL_SD (bottom).

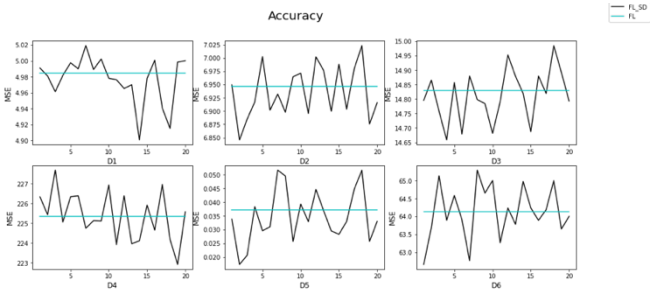

Fig. 6. Accuracy for FL and all repetitions of FL_SD

V. Concluding Remarks

Federated learning is a hot topic in private distributed computing. It is a technique that builds a high-quality, robust machine learning model where the data are distributed over a large number of clients (as opposed to storing on a single centralized server). However, designing an efficient FL system is a challenging task and requires attention to multiple parameters. One of the main challenges that requires attention is efficiency. The high operating cost for FL systems is driven by the number of iterations required for model convergence and the size of the messages communicated per iteration.

In traditional FL, the server starts the computation by generating random initial model parameters (often composed of zero values), the parameters are then refined over multiple iterations. In this paper, we designed a new method for calculating the initial model parameters. In our method, synthetic datasets are generated by the different clients to simulate their own private data and are sent to the server. The union of these synthetic datasets is used by the server to generate an initial model. The postulation is that a conscious initial model will reduce the number of iterations required to achieve convergence. We implemented this algorithm and compared it against traditional FL for linear regression. We use 6 different datasets in the experiments with varying number of clients. The results indicate a better performance in the number of iterations and consequently in the total time without affecting accuracy. The number of clients in the datasets used vary from 2 to 5. We believe that our method will display further reduction in complexity when the number of clients increases.

Going forward, we intend to run additional experiments with more datasets and different machine learning models. These experiments are needed to validate the above results, and make sure that they are independent of the particular machine learning algorithms used.

References

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," arXiv, arXiv:1602.05629, Feb. 2017. doi: 10.48550/arXiv.1602.05629.

[2] K. Mandal and G. Gong, "PrivFL: Practical Privacy-preserving Federated Regressions on High-dimensional Data over Mobile Networks," in Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop, New York, NY, USA, Nov. 2019, pp. 57–68. doi: 10.1145/3338466.3358926.

[3] Y. Chen, X. Sun, and Y. Jin, "Communication-Efficient Federated Deep Learning With Layerwise Asynchronous Model Update and Temporally Weighted Aggregation," IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 10, pp. 4229–4238, Oct. 2020, doi: 10.1109/TNNLS.2019.2953131.

[4] L. Kamm, "Privacy-preserving statistical analysis using secure multi-party computation," 2015.

[5] F. K. Dankar, N. Madathil, S. K. Dankar, and S. Boughorbel, "Privacy-preserving analysis of distributed biomedical data: designing efficient and secure multiparty computations using distributed statistical learning theory," JMIR Med. Inform., vol. 7, no. 2, 2019.

[6] F. K. Dankar, S. Boughorbel, and R. Badji, "Using Robust Estimation Theory to Design Efficient Secure Multiparty Linear Regression," 2016. Accessed: Sep. 09, 2016. [Online]. Available: http://ceur-ws.org/Vol-1558/paper33.pdf

[7] J. Xu, S. Wang, L. Wang, and A. C.-C. Yao, "FedCM: Federated Learning with Client-level Momentum," arXiv, arXiv:2106.10874, Jun. 2021. doi: 10.48550/arXiv.2106.10874.

[8] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic Controlled Averaging for Federated Learning," in Proceedings of the 37th International Conference on Machine Learning, Nov. 2020, pp. 5132–5143. Accessed: May 20, 2022. [Online]. Available: https://proceedings.mlr.press/v119/karimireddy20a.html

[9] J. Hu, "Bayesian Estimation of Attribute and Identification Disclosure Risks in Synthetic Data," ArXiv180402784 Stat, Dec. 2018, Accessed: Sep. 01, 2020. [Online]. Available: http://arxiv.org/abs/1804.02784

[10] N. Park, M. Mohammadi, K. Gorde, S. Jajodia, H. Park, and Y. Kim, "Data synthesis based on generative adversarial networks," Proc. VLDB Endow., vol. 11, no. 10, pp. 1071–1083, Jun. 2018, doi: 10.14778/3231751.3231757.

[11] N. Ruiz, K. Muralidhar, and J. Domingo-Ferrer, "On the Privacy Guarantees of Synthetic Data: A Reassessment from the Maximum-Knowledge Attacker Perspective," in Privacy in Statistical Databases, Cham, 2018, pp. 59–74. doi: 10.1007/978-3-319-99771-1_5.

[12] X. Lei, "Sythesizing Tabular Data using Conditional GAN," 2020.

[13] K. El Emam, "Could Synthetic Data Be the Future of Data Sharing?," CPO Magazine, 2021. Accessed: Sep. 28, 2021. [Online]. Available: https://www.cpomagazine.com/data-privacy/could-synthetic-data-be-the-future-of-data-sharing/

[14] S. Truex et al., "A Hybrid Approach to Privacy-Preserving Federated Learning," in Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, New York, NY, USA, Nov. 2019, pp. 1–11. doi: 10.1145/3338501.3357370.

[15] S. Boughorbel, F. Jarray, N. Venugopal, S. Moosa, H. Elhadi, and M. Makhlouf, "Federated Uncertainty-Aware Learning for Distributed Hospital EHR Data," arXiv, arXiv:1910.12191, Oct. 2019. doi: 10.48550/arXiv.1910.12191.

[16] F. Wang, H. Zhu, R. Lu, Y. Zheng, and H. Li, "A privacy-preserving and non-interactive federated learning scheme for regression training with gradient descent," Inf. Sci., vol. 552, pp. 183–200, Apr. 2021, doi: 10.1016/j.ins.2020.12.007.

[17] B. Nowok, "Utility of synthetic microdata generated using tree-based methods," UNECE Stat. Data Confidentiality Work Sess., 2015.

[18] R. J. Lewis, "An introduction to classification and regression tree (CART) analysis," in Annual meeting of the society for academic emergency medicine in San Francisco, California, 2000, vol. 14.

[19] F. K. Dankar and M. Ibrahim, "Fake It Till You Make It: Guidelines for Effective Synthetic Data Generation," Appl. Sci., vol. 11, no. 5, Art. no. 5, Jan. 2021, doi: 10.3390/app11052158.

[20] F. K. Dankar, M. K. Ibrahim, and L. Ismail, "A Multi-Dimensional Evaluation of Synthetic Data Generators," IEEE Access, vol. 10, pp. 11147–11158, 2022, doi: 10.1109/ACCESS.2022.3144765.

[21] K. El Emam, L. Mosquera, and C. Zheng, "Optimizing the synthesis of clinical trial data using sequential trees," J. Am. Med. Inform. Assoc. JAMIA, vol. 28, no. 1, pp. 3–13, Jan. 2021, doi: 10.1093/jamia/ocaa249.

[22] "UCI Machine Learning Repository." https://archive.ics.uci.edu/ml/index.php (accessed Oct. 14, 2021).

[23] "Kaggle: Your Machine Learning and Data Science Community." https://www.kaggle.com/ (accessed Oct. 14, 2021).

[24] "data.world | The Cloud-Native Data Catalog," data.world. https://data.world/ (accessed May 20, 2022).

[25] "Real-World Data solution | Cerner." https://www.cerner.com/solutions/real-world-data (accessed Oct. 14, 2021).

[26] C. van Aarle and A. C. Peter-Bram, "Federated Regression Analysis on Personal Data Stores," 2021.

[27] B. Nowok, G. M. Raab, and C. Dibben, "synthpop: Bespoke creation of synthetic data in R," J Stat Softw, vol. 74, no. 11, pp. 1–26, 2016.