Privacy Assessment of Federated Learning using Private Personalized Layers

Théo Jourdan, Antoine Boutet, Carole Frindel

HAL Id: hal-03354722
https://hal.inria.fr/hal-03354722

Submitted on 26 Sep 2021

Privacy Assessment of Federated Learning using Private Personalized Layers

Théo Jourdan
Univ Lyon, INSA Lyon, Inria, CITI, CREATIS, Inserm
theo.jourdan@insa-lyon.fr

Antoine Boutet
Univ Lyon, INSA Lyon, Inria, CITI
antoine.boutet@insa-lyon.fr

Carole Frindel
Univ Lyon, INSA Lyon, CREATIS, Inserm
carole.frindel@creatis.insa-lyon.fr

ABSTRACT

Federated Learning (FL) is a collaborative scheme to train a learning model across multiple participants without sharing data. While FL is a clear step forward towards enforcing usersâĂŹ privacy, different inference attacks have been developed. In this paper, we quantify the utility and privacy trade-off of a FL scheme using private personalized layers. While this scheme has been proposed as local adaptation to improve the accuracy of the model through local personalization, it has also the advantage to minimize the information about the model exchanged with the server. However, the privacy of such a scheme has never been quantified. Our evaluations using motion sensor dataset show that personalized layers speed up the convergence of the model and slightly improve the accuracy for all users compared to a standard FL scheme while better preventing both attribute and membership inferences compared to a FL scheme using local differential privacy.

KEYWORDS

Privacy, Federated Learning, Inference Attacks

1 INTRODUCTION

The development of Internet of Things (IoT) and wearables contributed to the progress of a wide range of quantified-self applications related to activity recognition. In this context, motion data extracted from connected devices (e.g., smartwatches) equipped with motion sensors (e.g., accelerometer and gyroscope) are sent to a central server which processes these data through machine learning models. According to the considered application, these learning models can classify which activity is performed by the user or compute other information such as the number of steps or burned calories.

While quantitative analysis of daily activities can bring benefits from the health perspectives [15, 16], transferring all this data to a third-party server raises important privacy concerns. Indeed, data breaches, compromised servers or any unwanted exploitation of the data expose users to personal and sensitive information leakage such as health-related attributes [9].

To mitigate this risk, a Federated Learning (FL) architecture (also known as collaborative learning) has been proposed [12]. In this scheme, the personal data of the user stays locally on its device and only a learning model is exchanged with the server. Iterativelly, the server sends a model to devices, this model is trained and refined with the local data on each device. Model updates are sent to the server which aggregates them to maintain a global learning model which will be disseminates back a devices. While this iterative process works well in case of data sharing similar distribution, heterogeneity of data across user devices can severely degrade performance of standard federated averaging for ML learning applications, especially for atypical users. Indeed, one unique model cannot cope with the heterogeneity of data and provide the best utility for all users [5]. To address this data heterogeneity, several local adaptation schemes have been proposed such as fine-tuning of personalized layers, multi-task learning, and knowledge distillation [1, 22] which depict a benefit for all participants in terms of accuracy.

While FL improves privacy by reducing the exposition of the personal data, it remains vulnerable to threats. For instance, FL is not robust to model poisoning which aims to destroy the convergence of the central model [3, 4]. Privacy leakages may also occur through membership inference attacks [17] which consist of inferring the presence of an individual data record in the training data, or attribute inference attack where the adversary is able to infer sensitive information about individuals [8]. The adversary can be passive [20] (i.e., only observing exchanges) or active [14] (i.e., modifying the protocol), and can control users or the server. To mitigate the risks, several approaches have been proposed from using Differential Privacy locally at user level or server level [13], Homomorphic Encryption (HE) and Secure Multiparty Computation (SMC) [7].

In this paper, we quantify the utility and privacy of a FL scheme using private personalized layers [1]. In such a scheme, only the lower layers of the model (capturing coarse grain information) are exchanged with the server while the upper layers of the model (capturing fine grain information) are personalized and kept private on each user. This scheme is known to improve the accuracy of the model in presence of heterogeneous data across users. However, the privacy impact of sharing only a sub part of the model has never been measured. To assess privacy leakage of this scheme, we consider both an attribute and a membership inference attack. Evaluations have been conducted using two datasets of motion sensor data collecting in real-life conditions. Results show that FL with personalized layers speeds up the convergence compared to vanilla FL and slightly increases the activity accuracy between 1% and 5%, while decreasing the gender and the overweight inference between 10% and 20% and 15% on average for membership inference. This utility and privacy trade-off is better than a defense scheme using local differential privacy which decreases the inference of the gender and the overweight up to 12% but at the cost of the activity accuracy which reduces up to 10%. These results tend to show that minimizing the information exchanged with the server is an interesting avenue for both personalizing the model

(i.e., improving accuracy) while limiting potential inferences (i.e., improving privacy).

The outline of the paper is as follows. First, background and related work are described in Section 2. The exhaustive evaluation is then presented in Section 3 before concluding in Section 4.

2 BACKGROUND AND RELATED WORK

In this section, we review background and related work on FL (Section 2.1) and inference attacks and mitigation schemes (Section 2.2).

2.1 Federated Learning

Deep Neural Networks are now the most effective algorithms for a lot of machine learning tasks. A new paradigm has been proposed by training Machine Learning (ML) models on the user devices, named Federated Learning (FL).

In a FL scheme following [12], participants keep their data locally on their device and exchange a model – targeting a specific learning task – with a server. The main objective of this algorithm is to iteratively train a learning model M maintained by the server by aggregating this model trained locally on each participant. At each learning round i, each client k trains its local model m_k with its own data using Stochastic Gradient Descent (SGD) during several iterations j. In its synchronous version, once all the participants send their model update m to the server, this server then aggregates all these model updates using the following equations before to disseminate back this aggregated model to all devices:

$$M_{i+1} = \sum_{c=1}^{C} \frac{n_c}{n} m_c^{i+1},$$

with n_c the set of indexes of all the data points n on client c, m_c^{i+1} the local update of a client c, calculated with the following equation:

$$m_c^{i+1} = m_c^i - \eta g_c^i,$$

with η a fixed learning rate (i.e., hyperparameter which controls the step size of the optimization) for each client and g_c^i the average gradient on the local data of the client c at the epoch i. Those learning rounds continue until the convergence of the central model.
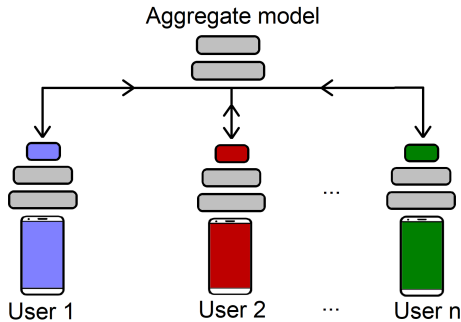

Figure 1: Personalized FL approach: only the upper layers (colored in grey) are shared with the server while the personalization layers are kept private on the device.

The development of FL highlighted other challenges [10] such as the heterogeneity of data across user devices leading to a degraded accuracy for less represented users. To overcome this limitation, [1] studied an FL scheme using personalization layers. In this scheme, the local model on each participant is composed of *lower layers* trained following classical FL learning round, and *upper and personalization* layers trained locally and which stay private on the device and not exchanged with the server (Figure 1).

2.2 Inference Attacks and Defenses

Privacy concerns are another main challenge of FL. Even though the raw data is not shared but kept on each local device, the model updates exchanged between participants and the server can leak sensitive information. Specifically, two types of attack can be considered. Poisoning attacks from malicious users aim at preventing the convergence of the learning model [4], or implanting a backdoor to control its behaviour [2]. Inference attacks, in turn, attempt to infer sensitive information on the users through model updates exchanged during the training process. These attacks can be conducted by participants or by the server. In addition, they can be passive or active. For instance, a malicious server can infer sensitive information based on the local updated parameters w^i sent by each user. To increase the observation, the server can send to users a fudged model in order to amplify the potential inference of sensitive information [14]. There are mainly two types of inference attacks: the attribute and the membership inference attacks. Attribute inference attack consists of inferring a sensitive information of the user [8] while membership inference attack consists of determining if a data record has been used for the training of a specific model [17]. To apply these attacks on a FL scheme, a supervised classification model (i.e., Random Forest) was trained to infer the sensitive information from the model's parameters shared as input.

To prevent these attacks, several approaches have been considered. Homomorphic Encryption (HE) and Secure Multiparty Computation (SMC) [7] approaches have been adapted to FL but are difficult to apply at scale due to overhead. Local Differential Privacy (LDP) [13], in turn, consists of introducing a random perturbation on SGD during the learning phase on the user device. The method provides statistical guarantees on the inference capability of an adversary to infer information on an individual's data. A differential-private SGD can be written as:

$$w_{t+1} = w_t - \eta_t(\nabla l(w_t, x_t, y_t) + N_t),$$

with η_t the learning rate, w the parameter that minimizes the objective function l for a data point (x,y) at time t, and N_t a noise value that follows a Gaussian distribution.

3 EVALUATION

We exhaustively evaluate the utility and privacy of a FL scheme using private personalized layers in the context of activity recognition (details of the methodology are given Section 3.1). In this section, we show that personalized layers improves the utility (Section 3.2) and privacy (evaluated through attribute inference Section 3.3 and membership inference Section 3.4) compared to both a vanilla FL and a defense scheme using local differential privacy.

3.1 Experimental setting

System model: We consider a FL scheme using Stochastic Gradient Descent (SGD) addressing activity recognition. The learning model is based on 2 convolutional layers, and 3 fully connected layers. Only the 2 lower convolutional layers are exchanged with the server which aggregates and disseminates model updates to devices, the 2 upper fully connected layers stay private on the device and are personalized with the user data. The devices of users are considered as trusted but it is not the case of the server which is considered as an adversary trying to infer personal information of participants from their model updates.

Datasets: Two real-life condition datasets are used for the evaluation. They are both publicly available and heavily used in the literature. These datasets come from the extraction of motion sensor data during gait activities (*i.e.*, based on step patterns) of different subjects.

- **MotionSense** [11] contains motion data captured from an accelerometer (*i.e.*, acceleration and gravity) and gyroscope of an iPhone 6s kept in the front pocket at a frequency rate of 50Hz. Overall, six activities (*i.e.*, walking, jogging, going upstairs, going downstairs, sitting and standing) have been made by 24 users during 15 trials in the same conditions and environment.
- **MobiAct** [19] records the motion data from 58 subjects during more than 2500 trials, all captured with a smartphone also in the front pocket. This dataset includes signals recorded from the accelerometer and gyroscope of a Samsung Galaxy S3 smartphone. Nine different activities of daily living are performed by the users. We only used the trials corresponding to the same activities as MotionSense in order to do the evaluation with the exact same settings.

Both datasets contain an equal number of men and women, and each activity is performed according the same conditions by all subjects. However, the walking activity is more represented than the others. For each user, we also have access to physical information (e.g., the gender, weight, height and age).

Baselines: We considered two baseline approaches to compare FL scheme using private personalized layers (**FedPer**) [1]:

- **Standard FL (Vanilla)** [12] This is the most common FL scheme using SGD training on the device and average aggregation of all models at each learning round on the central server.
- **Local Differential Privacy (LDP)** [13] We consider an implementation based on an introduction of noise following a Gaussian distribution ($\mathcal{N}(0,0.01)$) to the model updates computed through a classical learning phase (i.e., Vanilla).

Evaluation metrics: We evaluated FedPer and the different baselines along both utility and privacy metrics.

- **Utility:** To measure the utility, we considered the accuracy of the predicted activity. More precisely, we produce a confusion matrix based on the output of the classifier and measure the number of correct predictions made by this classifier over all predictions made. The value of the accuracy ranges from 0 to 1, in which 1 corresponds to perfect accuracy.
- **Privacy:** To assess the level of privacy, we rely on the accuracy of both the inference of sensitive attributes and the

inference to be a member of the training set. These inference attacks implement the solution proposed by [8] which leverages an invariant permutation representation of nodes at each layer to classify model updates received by the server through a random forest of 1000 trees with a maximum depth of 10. We consider the gender and the Body Mass Index (BMI) of the users as sensitive attributes. The BMI is a value defined by the weight of the user divided by the square of her height. This value allows to categorize a person as underweight, normal weight, overweight or obese. In our case we only focus on a binary classification: overweight (BMI > 25) or not (BMI < 25) for the sake of class balance. For the membership inference, the accuracy refers to the percentage of correct prediction (that a participant has been involved in the training of the model) over all predictions made. In both attacks, an accuracy of 0.5 corresponds to a random guess as our dataset is balanced.

Implementation details: For each experiment, we run 10 times of 5-fold cross validation where each fold is tested based on the training of the other four. We considered 200 learning rounds and an early stopping that stops the learning process if the average test loss of the aggregated model sent locally on the user data does not decrease during 30 learning rounds. During each learning round, the training with SGD is done locally at user's level during 10 epochs. A constant learning rate is used with $\eta = 0.001$ for all the users. These parameters have been optimized independently for the tasks of activity recognition, gender and BMI inference.

3.2 Utility Evaluation

We measure the accuracy of the activity detection of FedPer and the baselines. Figure 2 reports the Cumulative Distribution Function (CDF) of this accuracy over the population of users for MotionSense and MobiAct dataset. First, results show that the local adaptation of FedPer on the upper layers sightly increases the accuracy compared to the Vanilla approach (from 1% to 7% of increase on average for MotionSense and MobiAct, respectively). Second, results show that LDP baseline degrades significantly the accuracy for both datasets (10% on average of MotionSense and 6% on average for MobiAct). Indeed, by introducing noise, the convergence of the model is greatly degraded leading to a loss of prediction for all users. This result comforts previous results [18, 22].

We also measured the convergence speed of the learning. Figure 3 depicts the accuracy of the activity detection as a function of learning rounds for FedPer and the Vanilla approach. Results show that FedPer drastically speeds up the convergence. For instance, FedPer achieves 90% of accuracy after 12 learning rounds on MotionSense where the Vanilla approach achieves the same accuracy after 100 learning rounds. For MobiAct, FedPer achieves 90% of accuracy after 35 learning rounds where the Vanilla approach only reaches 86% of accuracy after 200 learning rounds. By using its personalized layers at each learning round instead of starting the learning from the aggregate model sent by the server, the accuracy increases faster. For LDP, we can observe that the noise introduced prevents the model from converging.
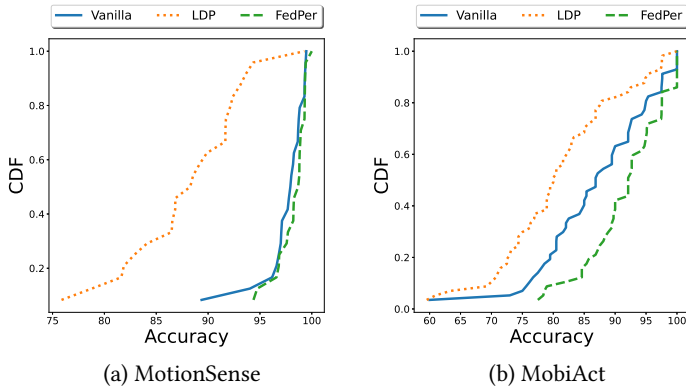
(a) MotionSense   (b) MobiAct

Figure 2: By personalizing upper layers of the model, FedPer sightly increases the accuracy of the activity prediction compared to a FL vanilla approach; local differential privacy, in turn, greatly degrades the accuracy.
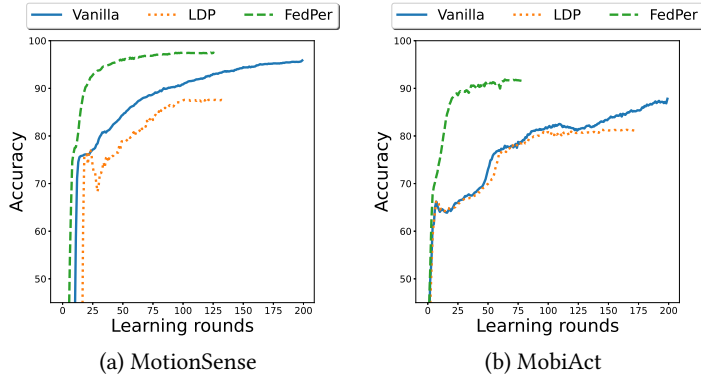

(a) MotionSense   (b) MobiAct

Figure 3: By using personalized layers instead of aggregated information, the learning is drastically speeds up.

3.3 Privacy evaluation through attribute inference

We conducted an attribute inference attack to infer the gender and the BMI of users from their model updates sent to the server. In this attack, participants train their local model on 80% of their data. Once all the models are sent to the server, only the models from one class of the targeted attribute are aggregated (in our case, models from women for gender inference, and models from overweight users for BMI inference). Then the server sends back the aggregated model to all the users that fine-tune locally on the remaining 20% of their data (e.g., training from a model aggregating model updates from women) before returning the update to the server. The adversary then trains an RF classifier on these model updates to infer the sensitive attribute. This training exploits 80% of all the updates and the testing is done on the remaining 20%, with cross validation.

Figure 4 evaluates, for both datasets, the accuracy of these both sensitive attribute inferences over the epochs of local learning. Firstly, results show that without any protection (i.e., the Vanilla approach), all sensitive attributes can be inferred with high accuracy for both datasets (e.g., around 90% of accuracy for the gender on
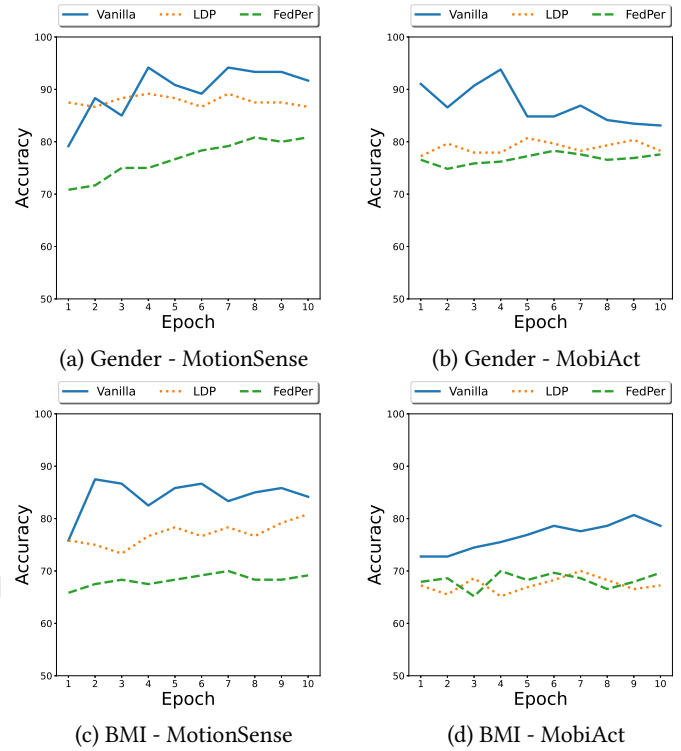

(a) Gender - MotionSense   (b) Gender - MobiAct

(c) BMI - MotionSense   (d) BMI - MobiAct

Figure 4: The increase of number of learning epochs per user increases the accuracy of the attack on both sensitive attributes.

MotionSense). FedPer reduces this accuracy between 10% and 20% according to the dataset and the sensitive attribute. Results also show that FedPer better protects users against inference attack compared to LDP regardless of dataset and sensitive attributes (from 5% to 10% of accuracy loss for MotionSense).

Secondly, results show that the inference accuracy tends to increase over the epochs for all approaches. This is explained by the fact that attribute inference attack is closely related to overfitting [21], the more the model learns on user's data, the more it adjusts the parameters to data structure and the more it may incorporate sensitive information.

Figure 5 reports the CDF of the inference accuracy over the participants. Results show that while each attribute can be inferred with high accuracy for a large part of the users, this accuracy drops for few percent of users. FedPer and LDP increase the percentage of users with a small inference accuracy.

3.4 Privacy evaluation through membership inference

Lastly, we conduct a membership inference attack to evaluate privacy. In this attack, 50% of the users follow a normal FL learning round with 80% of their data. The models are sent to the server which disseminates back the aggregated model to all the users. All of them fine-tune the aggregated model on their remaining 20% of data. The server then trains a RF to classify membership from model updates for all users (using 80% of all these updates for the
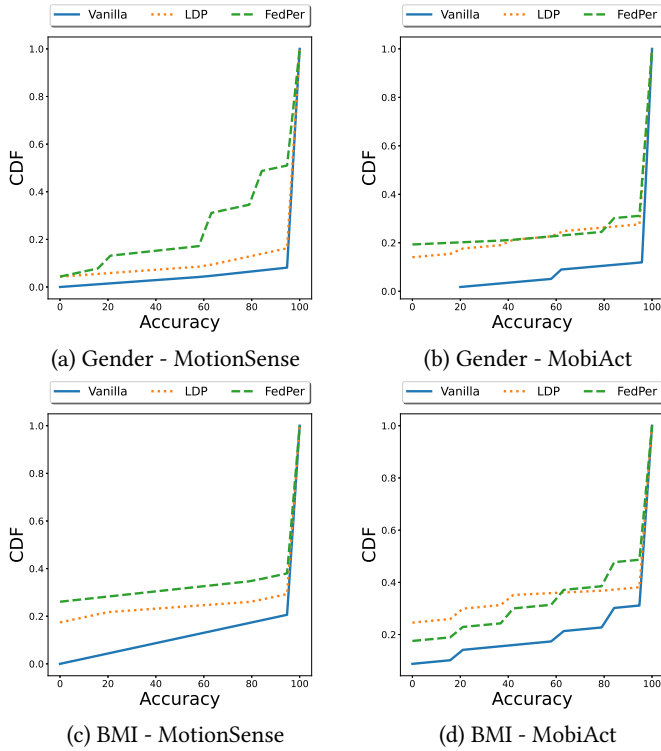
(a) Gender - MotionSense (b) Gender - MobiAct

(c) BMI - MotionSense (d) BMI - MobiAct

Figure 5: FedPer and LDP increase the number of users with a small inference accuracy.
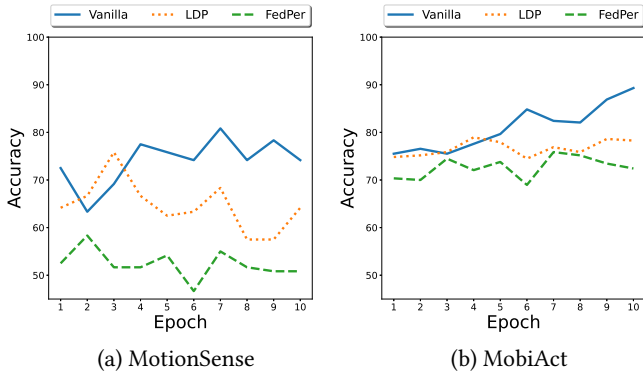

(a) MotionSense (b) MobiAct

Figure 6: FedPer and LDP significantly decrease the accuracy of the membership inference attack compare to Vanilla method

training and 20% for the testing with cross validation as described in section 3.1).

Figure 6 depicts the accuracy of this inference attack for both datasets and for all approaches. Similarly to the attribute inference attack, results show that the membership inference attack is more efficient on the Vanilla approach. FedPer provides the best protection compared to LDP (20% on average for MotionSense dataset). Interesting enough, FedPer depicts an accuracy close to 50% which correspond to a random guess (if the data of a specific user has been used to train the model) for MotionSense dataset.

4 CONCLUSION

We experimentally quantified the utility and privacy trade-off of FL using private personalized layers proposed by [1] in a context of activity recognition. We consider both an attribute and a membership inference attack to measure privacy leakage. Results show that using private personalized layers provides a better utility and privacy trade-off compared to a FL vanilla approach and a defense scheme using local differential privacy. These results tend to show that minimizing the information exchanged with the server is an interesting avenue for both improving accuracy and limiting privacy leakage. To comfort these results, it would be interesting to use other quantitative privacy metrics such as *average information leakage* and *maximum information leakage* [6].

REFERENCES

[1] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary. 2019. Federated Learning with Personalization Layers. *CoRR* abs/1912.00818 (2019).
[2] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov. 2018. How To Backdoor Federated Learning. *CoRR* abs/1807.00459 (2018).
[3] J. Bernstein, J. Zhao, K. Azizzadenesheli, and A. Anandkumar. 2018. signSGD with Majority Vote is Communication Efficient And Byzantine Fault Tolerant. *CoRR* abs/1802.04434 (2018).
[4] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer. 2017. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. *NIPS'17* 30 (2017), 118âĂŞ128.
[5] A. Boutet, C. Frindel, S. Gambs, T. Jourdan, and C. R. Ngueveu. 2021. DYSAN: Dynamically sanitizing motion sensor data against sensitive inferences through adversarial networks. *AsiaCCS'21* (2021).
[6] F. du Pin Calmon and N. Fawaz. 2012. Privacy Against Statistical Inference. *CoRR* abs/1210.2123 (2012).
[7] H. Fang and Q. Qian. 2021. Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning. *Future Internet* 13, 4 (2021), 94.
[8] K. Ganju, Q. Wang, W. Yang, C. A. Gunter, and N. Borisov. 2018. Property Inference Attacks on Fully Connected Neural Networks Using Permutation Invariant Representations. *ACM SIGSAC'18* (2018), 619âĂŞ633.
[9] J. L. Kröger, P. Raschke, and T. R. Bhuiyan. 2019. Privacy Implications of Accelerometer Data: A Review of Possible Inferences. *ICCSP'19* (2019), 81âĂŞ87.
[10] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. 2020. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Process. Mag.* 37, 3 (2020), 50–60.
[11] M. Malekzadeh, R. G. Clegg, A. Cavallaro, and H. Haddadi. 2019. Mobile Sensor Data Anonymization. *ACM IoTDI'19* (2019), 49–58.
[12] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas. 2016. Federated Learning of Deep Networks using Model Averaging. *CoRR* abs/1602.05629 (2016).
[13] M. Naseri, J. Hayes, and E. De Cristofaro. 2020. Toward Robustness and Privacy in Federated Learning: Experimenting with Local and Central Differential Privacy. *CoRR* abs/2009.03561 (2020).
[14] M. Nasr, R. Shokri, and A. Houmansadr. 2019. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. *IEEE SP'19* (2019), 739–753.
[15] E. Park, H.-J. Chang, and H. S. Nam. 2017. Use of Machine Learning Classifiers and Sensor Data to Detect Neurological Deficit in Stroke Patients. *J Med Internet Res* 19, 4 (2017), e120.
[16] J. Qi, P. Yang, D. Fan, and Z. Deng. 2015. A Survey of Physical Activity Monitoring and Assessment Using Internet of Things Technology. *IEEE CIT'15* (2015), 2353–2358.
[17] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models. *IEEE SP'17* (2017), 3–18.
[18] L. van der Maaten and A. Y. Hannun. 2020. The Trade-Offs of Private Prediction. *CoRR* abs/2007.05089 (2020).
[19] G. Vavoulas, C. Chatzaki, T. Malliotakis, M. Pediaditis, and M. Tsiknakis. 2016. The MobiAct Dataset: Recognition of Activities of Daily Living using Smartphones. *ICT4AWE'16* (2016), 143–151.
[20] M. Xu and X. Li. 2020. Subject Property Inference Attack in Collaborative Learning. *IHMSC'20* (2020), 227–231.
[21] S. Yeom, M. Fredrikson, and S. Jha. 2017. The Unintended Consequences of Overfitting: Training Data Inference Attacks. *CoRR* abs/1709.01604 (2017).
[22] T. Yu, E. Bagdasaryan, and V. Shmatikov. 2020. Salvaging Federated Learning by Local Adaptation. *CoRR* abs/2002.04758 (2020).