International Journal of Artificial Intelligence, Data Science, and Machine Learning
Grace Horizon Publication | Volume 7, Issue 1, 1-7, 2026
ISSN: 3050-9262 | https://doi.org/10.63282/3050-9262.IJAIDSML-V7I1P101

Original Article

Federated Learning with Smartphone Sensor Data

Dheeraj Vaddepally
Independent Researcher, USA.

Received On: 12/11/2025 ***Revised On: 15/12/2025*** ***Accepted On: 21/12/2025*** ***Published On: 02/01/2026***

Abstract *- Federated learning (FL) is emerging as a promising approach for training machine learning models on distributed devices without violating the data privacy of these devices. In this paper, we examine federated learning for smartphone sensor data applications that involve both significant challenges related to privacy and data heterogeneity. Our primary interest lies in techniques such as differential privacy, secure aggregation, and homomorphic encryption that will ensure privacy over user-sensitive information during model training. We also pose and discuss heterogeneous data across devices, particularly non-independent and identically distributed (non-IID) data, by investigating methods such as normalization of data, personalized learning, and federated transfer learning. Using real-world smartphone sensor datasets, we demonstrate experimentally that federated learning is effective in training robust models while preserving privacy and accounting for device-specific data variations. Our findings highlight that federated learning can be regarded as a way to scale-up and privacy-preserve mobile-based machine learning, which may open new avenues for building real-time AI systems on top of devices themselves.*

Keywords *- Federated learning, smartphone sensor data, privacy-preserving techniques, differential privacy, secure aggregation, homomorphic encryption, heterogeneous data, non-IID data, personalized learning, federated transfer learning.*

1. Introduction

Smartphone proliferation has led to a massive increase in the creation of sensor data that encompasses such wide-ranging information as movement, location, environmental conditions, and activity on the device. This creates immense potential to train machine learning models to further improve user experience in health monitoring, fitness tracking, and other applications like recommendations. However, centralized techniques of traditional machine learning carry the risk of huge privacy exposure because sensitive data are collected, stored, and processed on cloud servers. Therefore, with the increase in the threat of data privacy, the necessity for techniques that allow model training on users' devices without any violation of their data's confidentiality has emerged.[1] This is solved using federated learning, which has emerged as one of the solutions. Federated learning has enabled training machine learning models in decentralized settings across multiple devices. Only the updates of the model, not the raw data, are sent to a central server in FL, thereby retaining data on the device. Although FL holds great benefits in terms of privacy, it poses issues when applied to heterogeneous data produced by smartphone sensors. These devices often generate non-independent and identically distributed (non-IID) data, meaning that the data collected by one device is quite different from that of another, making training complicated.[1]

This paper focuses on the application of federated learning to smartphone sensor data, emphasizing two main challenges: privacy preservation and handling heterogeneous data across devices. The paper discuss privacy-preserving techniques like differential privacy, secure aggregation, and homomorphic encryption to make sure that the data collected from the user remains private.[2] It also study approaches for handling heterogeneity, like data normalization and personalized federated learning, that can deal with the issue of non-IID data distribution. We present through a set of experiments how federated learning can train models effectively while keeping privacy and device-specific variations in sensor data intact. Addressing these important issues, this paper contributes to the emerging research in federated learning and the potential for its realization in the enabling of privacy-preserving decentralized machine learning on smartphones.[2] Results of this work give insights into the feasibility of federated learning as a practical solution for mobile-based AI applications, paving the way for real-time, privacy-conscious machine learning on resource-constrained devices.

2. Related Work

Federated learning (FL) has received much attention as a promising approach to decentralized machine learning, especially for applications involving sensitive data, such as data generated by smartphone sensors. This section reviews relevant research on federated learning, privacy-preserving techniques, and methods for handling heterogeneous data across devices.

2.1. Federated Learning in Mobile Applications

The first is the idea from McMahan et al. to Federated Learning in 2016, for an investigation of feasibility by training machine learning models on many mobile devices leaving data on each device local for privacy purposes, where the device sends the updated version of the model to

the centralized server instead of submitting raw data itself[1]. Ever since its creation, this has been much explored and optimized to improve both on efficiency and scalability. Hard et al. (2018) proposed the method Federated Averaging as an optimization mechanism that diminishes overhead from devices in communications to achieve more than one training local iterations before transmitting models to servers. [2] Various works show that federated learning is very effective in preserving users' privacy at the same time training models in relation to data such as readings from sensors like smartphones. Wang et al. (2023) applied federated learning for mobile health data in predicting chronic diseases, which showed its potential on maintaining privacy as well as giving meaningful results. [3] Similarly, Zhang et al. (2021) discussed federated learning for gesture recognition with smartphone accelerometer data with an emphasis on the model's performance and privacy-preserving benefits.[4].

2.2. Privacy-Preserving Techniques in Federated Learning

The foremost major concern with using smartphone sensor data is privacy. Several techniques regarding privacy preservation have been proposed with the intent to mitigate risks involved due to leakage during the training process of the model.

Differential Privacy: One of the promising techniques proposed with federated learning toward preserving privacy may be called differential privacy. McMahan et al. (2018) have introduced differential privacy into the federated learning framework such that data contribution at the individual level is private even in the training of the model. The technique incorporates noise in a way that prevents any information leakage about the individual data points. Very recently, with the development of differential privacy, for example, adaptive mechanisms, it has fine-tuned the application to federated learning and brought it to an interesting balance between the protection of breach of privacy and model accuracy.[5] Secure aggregation protocols enable devices to calculate and then transmit aggregated updates of the models in such a manner that no one would be able to find out what every individual is actually contributing. Recently, Bonawitz et al. published a method of secure aggregation in 2019 that protected the central server from receiving all sensitive information regarding model updates by individuals through aggregated models. The technique is heavily used within federated learning systems for augmenting privacy and security without losing on collaborative training [6].

Homomorphic Encryption: HE is the technique by which computations are performed on encrypted data without allowing the central server to access raw data in the training process. Shokri et al. (2024) discussed the use of homomorphic encryption in federated learning while allowing secure model training without exposing sensitive data. [7] However, high overhead in computation and complexity persists, especially in resource-constrained mobile devices.

3. Federated Learning in the Context of Smartphone Sensor Data

3.1. Smartphone Sensors

A smartphone is provided with several sensors, where data of various types can be collected. In this regard, the data received is used within different applications; these include health monitoring and analysis of user behavior. The most frequent sensors on the smartphone are as follows:

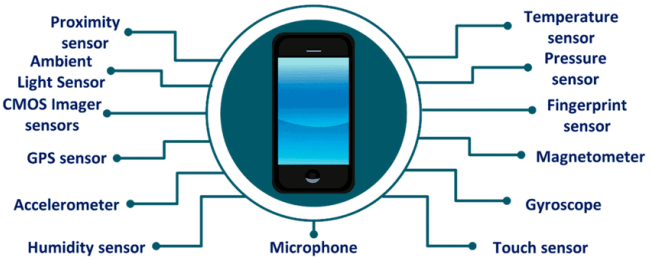

Fig 1: Smartphone Sensors

- Accelerometer: It calculates acceleration forces acting in three dimensions on the device, usually for use in activity recognition, step counting, and motion tracking.
- Gyroscope: The rate of rotation about axes of the device, usually it is combined with the accelerometer in gesture recognition, device orientation, and motion analysis.
- GPS: This gives location-based data. Thus, applications like navigation, geofencing, and location-based services use it.
- Camera: This captures images and videos, hence applications like face recognition, AR, and object detection.
- Magnetometer: It measures magnetic fields and is used to determine compass direction and orientation.
- Barometer: This measures atmospheric pressure, which can be read to determine the elevation or weather changes.

These are sensor devices that create highly useful data for modeling very diverse and varied applications. Some of the applications include health monitoring, activity recognition, environmental sensing, and personalized services.

3.2. Data Collection Challenges

Although smartphone sensors carry rich and diverse data, some difficulties are present while collecting this data from such devices:

- Noise: Sensor data is noisy because of environmental factors, device limitations, and sensor calibration issues. For instance, accelerometers may be sensitive to environmental vibrations around the user, causing them to register incorrect readings.
- Variability: A different model used by different models of smartphones would cause inconsistency

in the data. Differing qualities, placements, and calibrations of sensors may produce a different distribution of data, hence making it impossible to construct models that are generally applicable.

- Inconsistencies: Some of the factors that are liable for the inconsistency of sensor data include user behavior or environmental conditions. For example, GPS readings are not very periodic because it has obstacles like buildings and high roofs. This is primarily because of spasmodic and sporadic use of sensors to serve the needs of an application that results in some amount of data gaps.
- Data Privacy: In general, sensor data involves personal information. That may be a location or even activity about a user. So, it's important that the design of an application be privacy-preserving and that the data cannot leave the device for anybody.

3.3. Federated Learning for Sensor Data

It is a powerful solution in machine learning models to be trained using smartphone sensor data but keep data on the devices, thus alleviating privacy and data consistency concerns. In FL, instead of sending raw sensor data to the central server, models are trained locally on each device using data available on that device, whereas model updates, aggregated in a privacy-preserving way, are sent to the central server for the improvement of the global model.

Benefits of the application of FL on smartphone sensor data are:

- Privacy preservation: FL has the property where sensitive data would never leave a device. Rather, only aggregate model updates shared ensure user privacy.
- Handling Heterogeneity: FL could handle the heterogeneity of sensor data that may originate from different sources. Even heterogeneous types, qualities, and configurations of sensors FL enables every sensor device to contribute in updating global models without data exchange.
- Efficiency: FL enables training the model on the target device, which reduces the amount of data upload needed, thereby reducing network resource usage and minimizing the power the devices consume. The devices may periodically update their models locally, with the help of any resources that can be accessed.

FL is a suitable and promising framework toward tapping the abundance of sensor data collected by a smartphone, in an efficient, real-time manner that overcomes issues from data variability, noise, as well as associated privacy.

4. Privacy-Preserving Techniques for Federated Learning

The most relevant issue here would be privacy concern by the sensitive nature of data captured, such as personal activity, location, and behavior patterns. Preserving federation learning's ability to preserve privacy is important; for this reason, several techniques have been designed to protect user data during model training and aggregation.

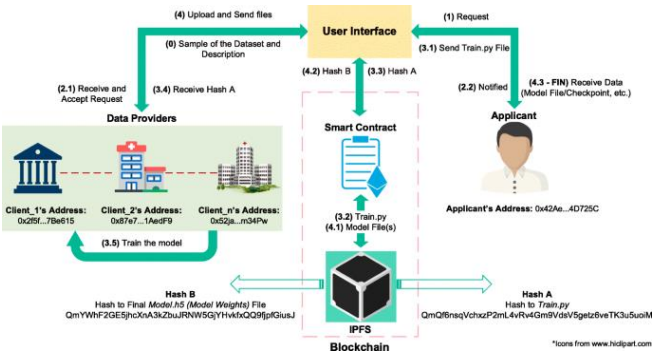

Fig 2: FL privacy Techniques

- The most popular technique applied to FL is differential privacy. In this method, contributions of the individual data points cannot be reconstructed or even traced from the updates of the aggregated model. The differential privacy method injects noise into the updates of the models so that it is impossible for traceability to updates of single user data to occur. The noise is calibration noise; this means that an amount added is balanced between preserving privacy and providing good model accuracy. This guarantees that even if an adversarial party gains access to the aggregated model updates, he will not be able to extract any private information regarding the individual users or their data.
- Another key privacy-preserving technique is Secure Aggregation. It does not allow the central server to see any raw data during model training. The model updates are not sent from each device to the server. Instead, secure aggregation protocols forward the aggregate of the updates to the server without revealing which device contributed which part of the data. This is because encryption techniques ensure that the model updates remain encrypted until they reach the server. The aggregation process will keep individual updates from the devices secret so that sensitive information about the users involved in the federated learning process will not leak out.
- Homomorphic encryption plays a major role in ensuring data confidentiality in model training and aggregation. It enables computations on encrypted data without decryption. This means in federated learning that the central server can aggregate model updates from devices without ever accessing the raw updates themselves. Such a method is especially useful for maintaining privacy of sensitive applications like health monitoring or location tracking when even intermediate data processing needs to be confidential. Though there is some computation overhead associated with homomorphic encryption, recent developments in

encryption algorithms make this method even more practicable to deploy on mobile devices.

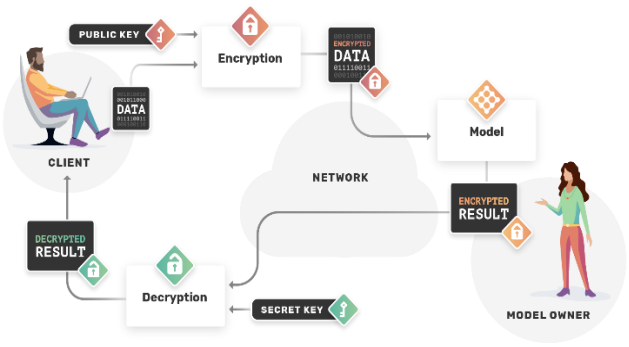

Fig 3: Homomorphic encryption

- Federated Transfer Learning: This is a very interesting method that combines the benefits of both transfer learning and federated learning. This is yet another method to enhance the protection of privacy while improving the performance of the model across heterogeneous devices. This minimizes the sensitive data that has to be transferred again since federated transfer learning will enable other devices to modify their model depending on pre-trained models or knowledge from another device, which will make a device use knowledge already acquired in other devices rather than only its local data. In this way, federated transfer learning transfers knowledge between devices and enhances the generalization of the models while minimizing privacy risks that come with training models on limited or sensitive data.

All these privacy-preserving techniques are combined to collectively ensure that federated learning is applicable to data from smartphone sensors without compromising user privacy. All of differential privacy, secure aggregation, homomorphic encryption, and federated transfer learning enhance the model training efficiency and security and therefore apply in privacy-sensitive applications.

4. Handling Heterogeneous Data across Devices

One of the biggest challenges federated learning will face when applied to smartphone sensor data is how to handle heterogeneity in data across devices. Because of differences in user behaviors, device configurations, and environmental conditions, the data collected by smartphones are often non-independent and identically distributed (non-IID). This non-IID nature creates several challenges to the convergence and performance of models since it becomes hard for the model to generalize across different devices while trained on data not drawn from the same distribution. For example, accelerometer data captured by one user is different from another user because their movement patterns and placement of the device might differ. These differences lead to lower convergence rates, lower precision models, and inability to have a model that would fit on all devices.

To solve this problem, the application of Data Normalization and Standardization techniques is of extreme importance. It allows standardizing or normalizing sensor data to be used for making the input data uniform on devices despite having a raw data set with very large variations. It means a series of steps focusing on rescaling readings from the sensor to an applicable range like [0, 1] in normalization; with standardization values are further normalized to zero mean and a unit standard deviation. Such effects reduce ones that will appear due to nonuniform sensor readings; a federated learning model would adjust better around devices having these. This will normalize the sensor values of features that are acceleration, location, or orientation and the same on diverse smartphones for useful model training aggregation.

Personalized Federated Learning builds upon federated learning while focusing on improved model performance given heterogeneous data across different devices to customize models according to specific characteristics at an individual's device. Rather than having one model to be trained on all the devices, federated learning, which is personalized, allows each device to have a unique model to help it better serve its local data distribution and usage patterns. For instance, a smartphone primarily used for fitness tracking will probably have different sensor data than one primarily used for navigation. Personalized models are fine-tuned on local data from each device, so the model's predictions are more accurate and relevant to the user's context. Personalization can be achieved by maintaining a global model that serves as a starting point, while each device adjusts its model locally based on its specific data. This will let every device utilize its unique sensor data, but still benefit from the general knowledge that is present in the global model. Non-IID challenges can be dealt with and personalized federated learning as well as normalization of data techniques support effective handling of the heterogeneity of smartphone sensor data by FL. These assure the proper usage of data that are unique to a device results in higher accuracy in models, quicker convergence, and practical federated learning.

4. Methodology

4.1. Federated Learning Framework

The federated learning (FL) framework follows server-client architecture, as it is a smartphone sensor data design. Several devices, under the coordination of a central server, collaboratively train a model; the server keeps track of the global model and coordinates the model updates from clients, while local training of the model is done at the clients themselves using their sensor data. The data never leaves the device; hence, it is not transmitted directly to the server and maintains privacy.

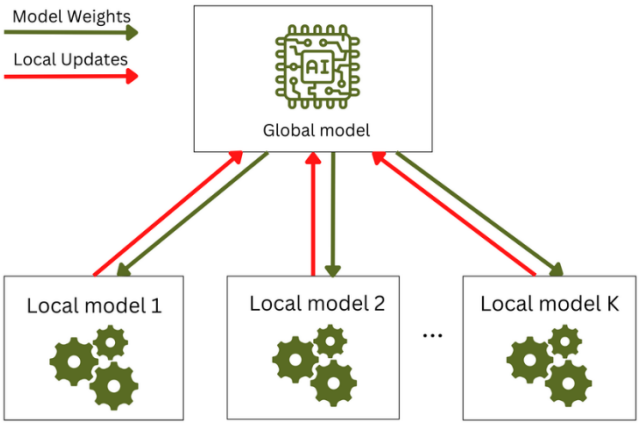

Fig 4: FL Framework

Data partitioning is a crucial aspect of federated learning because every client trains the model with its local dataset. This local dataset usually consists of sensor data obtained from the sensors of the device itself, such as an accelerometer, GPS, camera, etc. Because of the non-IID nature of the data across devices, data partition should be carried out in a manner that captures the heterogeneity in devices. It is coordinated through the server in its training process by aggregating the local model updates from each client. Communication protocols in the form of secure aggregation and differential privacy ensure that only aggregated model updates are shared, thus protecting user data. Additionally, synchronization techniques like periodic synchronization and protocols on communication efficiency (for example, model compression) reduce the impact of network latency and the number of rounds of communication between the clients and the server.

4.2. Preprocessing of Sensor Data

To be able to train the model on sensor data collected, several preprocessing steps have to be performed in order to ensure data quality and consistency. Sensor data are noisy and often inconsistent; therefore, in preprocessing, noise removal techniques such as filtering out extreme outliers, for example, or smoothing algorithms like moving averages usually form part of the preprocessing pipeline. The feature extraction part of the preprocessing pipeline is concerned with how to identify meaningful features from raw sensor data. This might range from extracting mean, standard deviation, signal magnitude area features from accelerometer data to represent different user motion patterns. This is more meaningful for model training than raw time-series data. Therefore, it will improve the model accuracy and reduce the computational cost.

Handling missing values is another very important preprocessing step. The sensor data are incomplete for a number of reasons, such as inactivity on the device or even sensor failure. Imputation is an important technique, either mean imputation or regression imputation. The other one is interpolation of data, which can also be used for filling in the missing data points. If missing data occurs too frequently or is critical for training, then the data will be discarded so that the model trains properly.

Table 1. Accelerometer data.

Timestamp	Unix Timestamp	Milliseconds	X	Y	Z
02–02–21 10:28:14 AM	1,612,261,680	1	−1.63639	−0.60269	9.899107
02–02–21 10:28:14 AM	1,612,261,680	12	−1.69412	−0.49502	9.731311
02–02–21 10:28:14 AM	1,612,261,680	21	−1.64596	−0.63858	9.702597
02–02–21 10:28:14 AM	1,612,261,680	30	−1.74915	−0.55005	9.968499
02–02–21 10:28:14 AM	1,612,261,680	41	−1.66271	−0.45912	9.853643

Table 2. Gyroscope data.

Timestamp	Unix Timestamp	Milliseconds	X	Y	Z
02–02–21 10:28:14 AM	1,612,261,680	1	0.003595	−0.00426	−0.0028
02–02–21 10:28:14 AM	1,612,261,680	12	−6.66E-04	−0.00213	−6.66E-04
02–02–21 10:28:14 AM	1,612,261,680	20	0.001465	−0.0032	3.99E-04
02–02–21 10:28:14 AM	1,612,261,680	31	0.003595	−0.00213	−0.00173
02–02–21 10:28:14 AM	1,612,261,680	41	0.00466	−0.00213	−6.66E-04

Fig 5: Sensor Data

4.3. Training Process

In federated learning, training is round-by-round, where model aggregation follows the local training. The server starts with an initial global model that it distributes to all clients. Each client then trains its local model on its sensor data. Traditionally, optimization techniques such as stochastic gradient descent or Adam, based on gradient descent, are used to minimize the loss function. This is known as updating parameters of the models based on locally available data that does not use the actual transferred data. Gradients or the weights returned back by the end of each round are aggregated by every client and relayed to the server. By aggregating each of these returned updates, it constructs a fresh global model on the server-side. The described process is also known as 'aggregation in the gradient layer'. It averages the updates, weighing them by the amount of data on each device so that clients with more data have a larger influence on the global model. More advanced techniques such as secure aggregation have been used in order to ensure that the server cannot get any individual client's model update without violating privacy.

Other techniques include adaptive gradient methods, learning rate adjustment, and regularization among others, in an effort to optimize a training procedure and the performance of the models. Others are communication efficiency protocols such as early stopping where the probability of overfitting does not occur, much more computationally expensive rounds of training would have been avoided. Others include protocols such as model compression or quantization which would reduce the size of updates to the models, hence decreasing the overhead of communication. This, in turn, trains a robust model that is deployable on smartphones while still preserving privacy, data efficiency, and model performance across heterogeneous devices.

5. Challenges and Open Issues

5.1. Scalability

One of the significant challenges with federated learning is scalability, especially in terms of extending the system to handle a large number of devices. The complexity of coordinating model training and aggregation increases with an increase in the number of participating devices. In this system, every device might have different computational powers, network bandwidth, and battery capacity, leading to inconsistent training processes. It involves more devices in the process and grows the volume of model updates being sent to a central server. Such an effect could lead to communication overhead due to increased information transmission. Techniques like efficient communication protocols and techniques of model aggregation help in tackling the scalability issues of the same. Strategies that involve model compression, asynchronous updates, and client sampling will help with relieving strain on network resources, ensuring it does not collapse at large scales.

5.2. Device Resource Constraints

Naturally, smartphones are devices that are very resource-constrained, and therefore, these resources pose a tremendous challenge to federated learning. Mobile devices normally have limited capacities in terms of computational power, memory, and battery life-all of which limit their ability to efficiently participate in federated learning processes. This is because substantial computational resources in training a machine learning model significantly consume energy; hence, making the device's usability for other tasks low. Moreover, the training process may be memory-intensive for a device since models have to store weights and gradients as well as other intermediate values. Thus, federated learning frameworks should implement lightweight model architectures, model quantization and more efficient optimization algorithms to take account of these constraints. Other techniques may involve local model update or offloading part of the computation to more powerful edge devices so that the smartphone balances the intensity of computation with energy consumption.

5.3. Data Privacy

Federated learning tries to protect the privacy of data by keeping the sensitive data on the local device, but there are challenges in fully maintaining privacy throughout the learning process. Differential privacy, secure aggregation, and homomorphic encryption help reduce the risk of leakage but do not completely eliminate it. For example, adversaries might use patterns in model updates for side-channel attacks or infer private information about individual users. Federated learning relies on local data updates, so one might also be concerned whether aggregated model updates may still leak private user behaviours or characteristics. Ongoing research into stronger privacy-preserving techniques, better secure aggregation protocols, and methods for detecting and preventing adversarial attacks is needed to enhance privacy guarantees in federated learning systems.

5.4. Data Quality

Data quality is the biggest problem with federated learning, particularly in sensor data from smartphones. Sensor data are generally noisy and frequently include inaccuracies. For example, this can be due to calibration errors on devices, environmental influences, or even malfunctions within the sensors themselves. Noise degrades the quality of the model, which may cause incorrect predictions or slower convergence. Missing values are another critical problem with sensor data. The causes of missing data may arise from the fact that a sensor has failed or the device was inactive, which resulted in insubstantial sampling. Handling the problems involves elimination of noise, such as filter and smoothing, detection of outliers, and imputation techniques to provide values for the missing data. Nonetheless, these are not very effective and are secondary to the root variation in readings from different sensor devices. This is one area where federated learning models have been researched; that is, how they can even perform well despite the presence of noisy and incomplete data.

This raises several challenges and open issues toward the deployment of federated learning systems on sensor-rich smartphones, and techniques including model compression, privacy-preserving methods, and data preprocessing have immense potential to further address some of these issues, hence a pressing need for ongoing research in that direction to mitigate the current barriers toward making federated learning scale-up, resource-aware, and safe for a whole host of realistic applications.

6. Conclusion

Federated learning might unlock the sensor data of phones with minimal deviation from privacy, and it is especially toward the limitation of scattered data. Thus, federated learning brings privacy issues down to their barest minimum as it allows training models to be conducted straight on devices without transferring sensitive data that may cause personal information. However, federated learning of smartphone sensor data will be successful only if the kind of host of challenges associated with heterogeneous data across the devices is mitigated; the limitation at the device level addresses resource limitations, ensures privacy in data collection, and addresses quality in data.

Some of the techniques applied in these directions include differential privacy, secure aggregation, and personalized federated learning. Other techniques that are equally important in helping to make the data reliable and consistent include normalization of data, feature extraction, and noise removal, which are often required for noisy and incomplete sensor data. In scaled federated learning systems, which involve a number of devices of various sensor capacities, communication protocols and model architectures need to strike the right balance in computational load with respect to energy consumption.

In summary, while promising in the space of privacy-preserved applications in machine learning applications, further innovation in its systems' scalability of processing

data together with improvement for privacy protection makes federated learning closer. Moving forward with growing technology and subsequent solutions that could emerge, federation learning would be a foundational aspect in widespread distributed machine applications in improving both efficacy and safety and personalization in countless domains.

References

[1] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[2] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[3] T. Wang, Y. Du, Y. Gong, K. K. R. Choo, and Y. Guo, "Applications of federated learning in mobile health: Scoping review," *Journal of Medical Internet Research*, vol. 25, p. e43006, 2023.

[4] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021.

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

[6] K. Bonawitz, F. Salehi, J. Konečný, B. McMahan, and M. Gruteser, "Federated learning with autotuned communication-efficient secure aggregation," *arXiv preprint arXiv:1912.00131*, 2019.

[7] A. Aminifar, M. Shokri, and A. Aminifar, "Privacy-preserving edge federated learning for intelligent mobile-health systems," *arXiv preprint arXiv:2405.05611*, 2024.

[8] L. Li, Y. Fan, M. Tse, and K. Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, 2020.

[9] P. M. Mammen, "Federated learning: Opportunities and challenges," *arXiv preprint arXiv:2101.05428*, 2021.

[10] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.