# Evaluating Federated Learning for human activity recognition

**Sannara Ek, Francois Portet, Philippe Lalanda and German Vega**

Grenoble-Alpes University, France
Firstname.name@imag.fr

## Abstract

Pervasive computing promotes the integration of connected electronic devices in our living environments in order to deliver advanced services. Interest in machine learning approaches for engineering pervasive applications has increased rapidly. Recently federated learning (FL) has been proposed. It has immediately attracted attention as a new machine learning paradigm promoting the use of edge servers. This new paradigm seems to fit the pervasive environment well. However, federated learning has been applied so far to very specific applications. It still remains largely conceptual and needs to be clarified and tested. Here, we present experiments performed in the domain of Human Activity Recognition (HAR) on smartphones which exhibit challenges related to model convergence.

## 1 Introduction

Pervasive computing promotes the integration of smart devices in our living spaces in order to provide advanced services. These services use information collected by the devices, perform some computation and, in some cases, act on the environment. Today, computation is essentially done in the cloud where powerful, flexible and pay-per-use infrastructures are made available to service providers. However, cloud-based architectures have important limitations mainly related to security issues and lack of reactivity. In practice, such architectures limit the number of services that can be implemented because of unpredictable communication delays, privacy concerns regarding data transferred over the Internet and, in some cases, insufficient bandwidth or excessive costs. A better use of edge resources would allow the implementation of new, high quality services [Becker et al., 2019]. The notion of edge was mentioned in 2009 [Satyanarayanan et al., 2009] and generalized by Cisco Systems in 2014 as a new operational model. The main idea is to place computing and storage functions as close as possible to data sources, that is in resources located in direct physical environments.

Using edge resources to run services is however challenging. Most current services based on such approaches heavily rely on cloud infrastructures and cannot be easily implemented in edge devices for lack of resources. Google recently proposed Federated Learning (FL) [McMahan *et al.,* 2017] [Bonawitz *et al.,* 2019] [Konecny *et al.*, 2016] for distributed model training in the edge with an application of personalized type-writing assistance. Rather than gathering data from remote devices on a centralized server, Federated Learning fuses several models that have been learned locally in one, more generic, model to be redistributed to the local devices as a bootstrap model for the next local learning iteration. In theory, the new model provides more genericity while the local learning provides more adaptation. Moreover, FL is supposed to save communication costs and protect security and privacy by preventing data collected at the terminal level from being sent through the network. It has immediately attracted attention as a new machine learning paradigm promoting the use of edge resources. This new paradigm seems to fit the pervasive environment well. Nevertheless, federated learning is still largely conceptual and needs to be clarified and tested extensively.

In this paper, we present several experiments aiming at assessing the interest and the limits of FL with respect to the centralized deep learning approach. The experiments were conducted in the field of Human Activity Recognition (HAR) on smartphones. HAR is a pervasive application that is well suited to FL since activities tend to have generic patterns (*e.g.,* walking involves the same sequence of movements for anybody) while being highly idiosyncratic (*i.e.,* data depends on the person, the device and the environment). Furthermore, the collected data is private and should not be sent over the network.

This paper is organized as it follows. First, some background about machine learning and federated learning is provided. Section 3 gives information about HAR and presents our experimental settings. Then, in section 4, experimental results are presented. They are discussed in detail in section 5. Finally, this paper is ended by a conclusion presenting some future work.

## 2 Federated learning

The goal of a machine learning system is to induce a decision model (prediction, classification) from data which will be able to make automatic decisions on new unseen data. Machine learning algorithms identify patterns that may be hidden within massive data sets whose exact nature is unknown and therefore cannot be programmed explicitly. The growing attention towards machine learning stems from different sources: the emergence of deep learning [Lecun *et al.,* 2015], the availability of massive amounts of data, advances in high-performance computing, broad accessibility of these technologies, and impressive successes reported by industry, academia, and research communities, in fields such as in vision, natural language processing or decision making. In pervasive, many AI-based applications have been designed and have demonstrated to be effective in proof-of-concept trials. Nevertheless, the deployment of large-scale machine learning applications on devices in practice is still limited due to multiple reasons.

First, most current learning systems are based on offline training and online predictions. The learning process is implemented on powerful servers and is run periodically, out of sync with data generation. Typically, models are updated every day or every week (depending on the servers' cost and availability) while predictions (model execution) are needed at a much higher pace. This is not appropriate for most pervasive environments where important changes can occur anytime, generally in unexpected ways. Here, applications must adapt immediately to those asynchronous changes to remain relevant. Also, AI-based applications used in pervasive computing make the implicit assumptions that 1) all data is stored in a cloud and 2) data ownership belongs to the company providing the services. These assumptions are no longer true in some pervasive environments, including smartphones for instance. Huge amount of data is generated in the edge devices, and sending all of it to cloud servers is not practical. Data can have additional security and privacy requirements that must be taken into consideration (to follow regulations like GDPR in Europe or Cyber Security Law in China for instance). Pervasive environments may change, sometimes rapidly and unexpectedly, and often in non-reproducible ways. Handling such environments will require AI systems that can react quickly and safely even in scenarios that have not been encountered before. Finally, is to be noted that a high level of resilience is expected in pervasive environments. Wrong predictions in the physical world may end up in tragic issues.

As introduced before, federated learning has been recently introduced by the Google company. This new approach proposes a distributed machine learning strategy that enables training on decentralized data residing on devices like mobile smartphones. Federated learning is in line with the objectives of fog computing in the sense that data and computing are distributed on smart devices. This clearly can address problems of performance, privacy and data ownership.

As illustrated hereafter by Figure 1, federated learning [McMahan *et al.,* 2017] relies on a distributed architecture made of a server located in a cloud-like facility and a number of devices, called clients. Number of clients is variable and dynamic; clients can appear and disappear without notice.

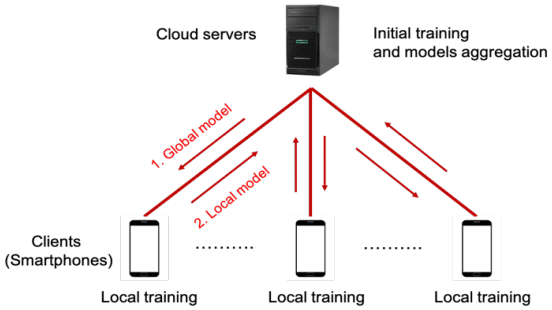


Figure 1. Federated learning architecture.

The theoretical architectural behavior is the following. First, a randomized global model, a convolutional neural network for instance, is generated at the server site and sent to the clients. Then, selected clients collect data and on-device training is performed. After some pre-defined time, local models built by the clients are sent back to the server. The server aggregates these models into a new global model which is, again, sent to the clients and the cycle is repeated. It can be also noted that in theory new clients are allowed to join at any time which may prolong training indefinitely.

A key point in this new paradigm model is the model's aggregation. In the first publications related to federated learning, aggregation was implemented as an average function [McMahan *et al.,* 2017]. This means that the weights of the different local models are averaged to provide new weights and, thus, a new model. New aggregation functions have been very recently proposed including FedPer [Arivazhagan *et al.,* 2019], a federated learning algorithm incorporating a base and personalized layer with transfer learning methodologies, and FedMA [Wang *et al.,* 2020], a federated layer-wise learning scheme which incorporates the match and merging of nodes with similar weights.

In theory, FL has thus the features to fit pervasive constraints. It can adapt to change since each client is supposed to constantly learn from its own experience and from the others. It should preserve privacy and efficiency since large data is not transferred through the network but are kept on the client. It should be adapted to edge computing since the merging part has not necessarily to be performed on the cloud.

However, Federated learning has been tested and validated on simulated data and on a few domains only, which leaves a number of open questions. In fact, it is unclear whether a federated approach will always lead to superior performances and robustness than a purely centralized or decentralized one. Specifically, we believe that data distribution and heterogeneity is a major aspect that needs more investigation and testing. In the pervasive domain, data can be very different depending on subjects, environments and conditions. Shedding light on that question is our goal here.

# 3 Experiments

## 3.1 Human activity recognition

Human Activity Recognition (HAR) based on wearable sensors, often provided by smartphones, has prompted numerous research works [Lara and Labrador, 2013]. Many approaches have been investigated to identify and classify physical human activities such as running or walking, and also interactive and social activities like chatting, talking, or playing. In this section, we focus on research works leveraging machine learning techniques. Regarding classification models, many techniques have indeed been investigated to deal with HAR based on wearable sensors. The most common approach is to process windows of data streams in order to extract a vector of features which, in turn, is used to feed a classifier. Many instance-based classifiers have thus been used to so. Let us cite Bayesian Network, Decision Trees, Random Forest, Neural Network, and Support Vector Machines [Lara and Labrador, 2012]. Since human activities can be seen as a sequence of smaller sub-activities, sequential models such as Conditional Random Fields, Hidden Markov Model or Markov Logic Network have also been applied. Today, the most popular and effective technology is clearly deep neural networks [Ignatov, 2018].

Deep learning is however highly dependent on access to large amounts of data. This is why FL is also seen as a way to leverage the ability of DL to benefit from a growing number of data with actually circulating them to the network. Another problem is that machine learning algorithms must face is the heterogeneity in data. A survey conducted by [Lara and Labrador, 2013] presents a large number of datasets acquired from smartphones, worn in different ways. It clearly highlights the lack of uniformity in tasks, sensors, protocols, time windows, etc. It is worth noticing that some datasets are very imbalanced because activity distributions among classes are very different. For instance, in the REALWORLD dataset [Sztyler et al., 2017], the "*stairs*" activity represents 22% of the data while "*jumping*" is limited to 2%. The learning approach should consider the class imbalance problem. In our experiment, the loss was weighted by the class weights to counter balance the non-uniform distribution of classes.

## 3.2 Experiments description

Our purpose is to evaluate the performance of the FedAvg algorithm against a centralized training approach using the UCI [Davide *et al.*, 2013] and REALWORLD dataset, which contain accelerometer and gyroscope time-series data obtained with Android devices. The choice of these datasets is to present the performance of FL with homogeneous clients (UCI) and with heterogeneous clients (REALWORLD). Data was collected from 30 and 15 subjects and consists of 6 and 8 activities, respectively. We use a window-frame size of 128 with a 50% overlap of 6 channels. To respect the deep learning approach (*i.e.,* features should be learned and not hand-crafted), no preprocessing was applied except channel-wise z-normalization. The final size of the datasets is 202 MB and 6.98GB, respectively, in a csv format.

Our experiments were done using a Dense Neural Network (DNN) and a Convolution Neural Network (CNN) to compare the federated learning results against traditional centralized training in deep learning and as well as to observe the effects of the 2 different architectures in federated learning. Our DNN model has hidden layers composed of 400 and 100 neural units. The CNN model has 192 convolutional filters of size 1x16 followed by a max-pooling layer of 1x4 where the outputs are then flattened and fed to a fully-connected layer of size 1024. We emphasize the use of shallow neural network models for the context of usage on edge devices with limited processing power and the reduction of communication cost in federated learning. The models are trained using a mini-batch SGD of size 32 and to counter over-fitting, a dropout rate of 0.50 is used. The models were developed using TensorFlow for our implementations.

For the FL test, we split the data of each subject into an 80% train-set and 20% test-set based on the classes uniformly and then another instance randomly to simulate a balanced and an unbalanced dataset test scenario for clients.

Due to the small size and lack of activity-subject correspondence in the UCI dataset, we partitioned the train and the test-set into five artificial clients. On the other hand, each subject of REALWORLD dataset is treated as a client with its own respective data, leading to 15 clients. We used 50 communication rounds for our test and each client trained for a total of 10 local epochs with 0.005 as the learning rate.

# 4 Results and discussion

## 4.1 Evaluation measures

In this section, we present the results of the comparison between the classical centralized approach against the FedAvg algorithm with the mentioned DNN and CNN models for the task of HAR. The performance of the FedAvg is evaluated twice against the test-set, once by the aggregated model on the server and then by all the client models against their own test-set where the average accuracy is presented, which we note as the federated accuracy. We have adopted the F1 score, which is the harmonic mean of precision and recall, as our primary evaluation metric. It is defined here:

$$F_1\_Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

In line with standard practices in HAR research, the macro average of the individual F1 scores of all classes, is also reported. It is more resilient to class imbalances in the test dataset, *i.e.:*

$$macro\ F_1\_Score = (\sum_{ci=1} F_{i1}\_score)/|C|$$

Where $F_{i1}\_score$ is the F1 score of the ith class and $C$ is the set of classes. When reported, the *Accuracy* is computed as the number of correct classifications divided by the total number of instances. This measure is not insightful with regard to class imbalance.

## 4.2 Results on UCI Dataset

Table 1 presents the state-of-the-art accuracy when using a centralized approach on the UCI dataset, based on a standard train-test supervised learning. With a standard centralized training approach, our own CNN model was able to reach an accuracy of 94.64%, which is in the lower scale of the state-of-the-art results [94.61-97.62]. However, we did not use any preprocessing or aggressive hyper-parameter tuning. Our DNN model got a lower accuracy of 90.23%, which confirms that CNN is more suited for the task. It is important however to recall here that the aim of the study is not to improve the state-of-the-art performance on the HAR field but to evaluate FedAvg using credible and standard deep models.

| Studies | **Base Models** | **Accuracy %** |
|---|---|---|
| Ronao and Cho, 2016 | CNN | 94,61 |
| Jiang and Yin, 2015 | CNN | 95,18 |
| Ronao and Cho, 2015 | CNN | 94,79 |
| Ronao and Cho, 2015 | CNN | 90,00 |
| Almaslukh, 2017 | SAE | 97,50 |
| Ignatov, 2018 | CNN | 96,06 |
| Anguita *et al*, 2013 | SVM | 96,37 |
| Cho and Yoon, 2018 | CNN+Sharpen | 97,62 |
| Cho and Yoon, 2018 | CNN | 97,29 |
| Our study | DNN | 90,23 |
| Our study | CNN | 94,64 |

Table 1. State-of-the-art performances of classical centralized approaches on the UCI dataset.

Table 2 hereafter presents the server accuracy obtained on the UCI dataset using FedAvg. Results are compared to our centralized baseline models, presented in Table 1.

| **Models** | **Server - Accuracy** | **F1** | **Macro F1** | **Nb of Clients** | **Federated - Accuracy** |
|---|---|---|---|---|---|
| Baseline DNN | 90.23% | 90.23% | 89.98% | N/A | N/A |
| Baseline CNN | 94.64% | 94.64% | 94.69% | N/A | N/A |
| FedAvg DNN | 94.37% | 94.12% | 94.26% | 5 | 93.78% |
| FedAvg CNN | 97.47% | 97.13% | 97.28% | 5 | 96.11% |

Table 2. Classical vs. FedAvg at the server level (UCI).

The server accuracy of the aggregated DNN model on the UCI balanced dataset obtained an accuracy of 94.37% while the federated accuracy was 93.78%. The federated accuracy was computed as the average client test accuracy. Overall, the macro-F1 is similar to the global F1 for each model which means that models are not biased towards some majority classes. Hereafter, Figure 2 shows the accuracy and loss vs communication rounds for the DNN model. The server train and test accuracy and loss measures are reported after the aggregation of the communication round. The clients train and test accuracy and loss are evaluated before the communication round (i.e., before sending to the server). The figures show the curves of the average accuracy and loss as well as the standard deviation over the clients' one. On Figure 2, while the CNN aggregated model (server-side) has an accuracy of 97.47%, its federated accuracy (client-side) has reached an accuracy of 96.11%, see Figure 3 that provides the accuracy and loss over 50 communication rounds.

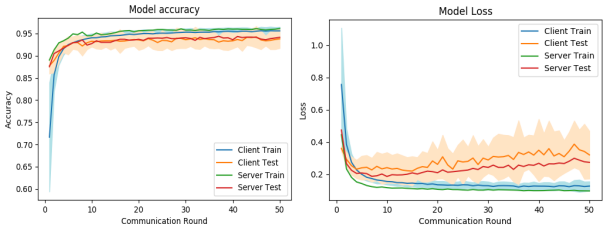


Figure 2. DNN performance vs communication rounds on UCI.

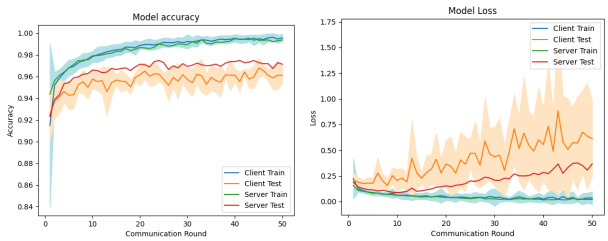


Figure 3. CNN performance vs communication rounds on UCI.

## 4.3 Results on REALWORLD Dataset

The F1-measures of other state-of-the-art centralized approaches on the REALWORLD dataset using a standard train-test supervised learning are presented below in Table 3.

| **Literature** | **Models** | **Accuracy %** |
|---|---|---|
| Sztyler *et al*, 2017 | RFC | 81,00 |
| Our study | DNN | 82,40 |
| Our study | CNN | 84,45 |

Table 3. State-of-the-art performances of classical centralized approaches on the REALWORLD dataset.

Training with the standard centralized approach and a DNN model, we got a test accuracy of 82.40%. The CNN model achieved an accuracy of 84.45%. These performances are

above the other results of the state-of-the-art which add credence to our baseline model choice. In table 4, we show comparative results between our baseline models and the models obtained through FedAvg trained on the REALWORLD dataset.

| Models | Server - Accuracy | F1 | Macro F1 | Nb of Clients | Federated - Accuracy |
|---|---|---|---|---|---|
| Baseline DNN | 82.40% | 82.40% | 82.50% | N/A | N/A |
| Baseline CNN | 84.45% | 84.45% | 84.50% | N/A | N/A |
| FedAvg DNN | 72.32% | 72.31% | 74.89% | 15 | 92.43% |
| FedAvg CNN | 76.08% | 75.75% | 77.72% | 15 | 95.10% |

Table 4. Classical vs FedAvg at the server level (REALWORLD)

Using the FedAvg approach on the DNN model, the server and clients had an accuracy of 72.32% and 92.43%, respectively. Figure 4 shows the corresponding accuracy and loss over 50 communication rounds. For the CNN model, FedAvg model held an accuracy of 76.08% and 95.10%, respectively on server and client. Figure 5 shows the corresponding accuracy and loss over 50 com rounds.

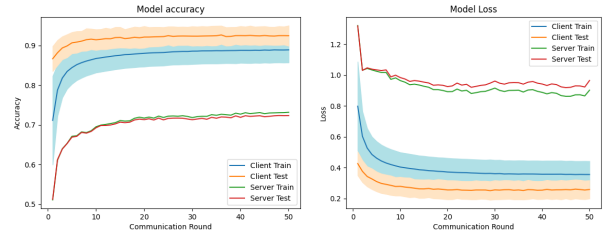


Figure 4. DNN performance vs com. rounds (REALWORLD).

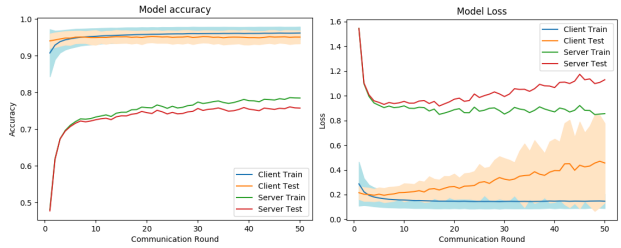


Figure 5. CNN performance vs com. rounds (REALWORLD).

## 5 Discussion

Our experiments show different behaviors for the same FL algorithm but with different datasets and models. The most striking difference is the one of the datasets. In the case of UCI, the FedAvg with the CNN model performs on par with the state-of-the-art compared to the centralized approach. However, with the REALWORLD dataset, FedAvg gives a lower performance. In that case, it can also be observed a large difference in accuracy between the server model and the clients' ones. This behavior might be due to the fact that we used one subject's data as a client and because the number of clients is higher (15) for REALWORLD than UCI (5).

In the case of REALWORLD, clients tend to converge rapidly to very high accuracy on their own test set (95%) but get lower accuracy on the whole test set (around 65%). This explains why the server cannot get high performance, since clients' models are learning in different directions, hence averaging weights blinding is not an efficient way to benefit from each client invariants. The large standard deviation of the client losses supports this interpretation. Although, as the curves suggest, this could be called 'overfitting' this is a desirable behavior since the client's model is self-specializing to the device/user. In fact, the notions of overfitting/specialization in a federated learning setting must be clarified and the way to assess them systematized.

These experiments also show that designing a FL experiment in which clients are modeled with different datasets is much more realistic and challenging than settings where data is evenly distributed among clients. On this aspect, one comparable work was performed by [Sozinov *et al.,* 2018] on the Heterogeneity Human Activity Recognition Dataset [Stisen *et al.,* 2015] in which they report effects of device and subject heterogeneity and distribution on FedAvg.

It also appears that UCI data is much more uniform and less noisy than the REALWORLD one. The REALWORD dataset is more realistic since it involves 7 different smartphone positions and has been performed outdoors. This can be backed up by the federated accuracy on the REALWORLD dataset performing very well but is lacking with the server's aggregated model. To further check the ecological aspect of the dataset, we ported the trained models to a Google Pixel 2 android device using TensorFlow Lite. The model trained from the UCI dataset often fails to accurately predict activities in the wild, which contradicts the accuracy evaluated against its own test-set. On the contrary, the model trained from the REALWORLD dataset, when tested in the wild showed more satisfying results.

The second difference is related to models. The DNN and the CNN model seem to exhibit the same behavior with respect to the overall performances. However, the CNN model shows a much larger standard deviation than the DNN one along with the communication rounds. One problem of FedAvg is that it performs averaging coordinate-wise which might have significant detrimental effects on the performance of the averaged model. As raised by [Wang *et al.,* 2020], this issue arises due to the fact that during FL some invariants are captured by parameters that only differ in their ordering in different client models. Although it might be expected that these discrepancies get solved with more communication rounds, it is not desirable in terms of communication cost and it is not guaranteed if the clients' data is very different from each other. This is particularly acute with CNN which contains more features layer than the dense model and whose dropout layer might imply that invariants are learned by different sets of neurons in the client's model. To get a better analysis of this behavior, it would be necessary to analyze the divergence of model weight layer per layer along the communication round. This can be achieved using standard matrix similarity measures (*e.g.* Mantel Test) or dedicated deep learning analysis [Kornblith *et al.,* 2019].

## 6 Conclusion

Federated learning generates expectations and significant efforts have been made to find techniques to improve model accuracy and communication cost. However, research work is still needed to understand FL behaviors, define relevant evaluation methods, and bring experiments out of the simulation mode. Our experiments with two families of neural networks and two different datasets show that issues such as data heterogeneity challenge the FedAvg algorithm and necessitate better analysis strategies. From our experiments it seems that FedAvg is better suited to realistic heterogeneous and imbalanced datasets (REALWORLD) than carefully balanced and in-lab datasets (UCI) since experiments with REALWORLD demonstrate a high degree of personalization.

Our next step is to evaluate the performance of advanced FL algorithms such as FedMa [Wang *et al.,* 2020] and FedPer [Arivazhagan *et al.,* 2019] since it has been shown they can better tackle the heterogeneity problem in HAR [Li et al., 2019]. On the longer term, since FL studies have been mainly restricted to an extension of classical centralized off-line learning, it is necessary to evaluate FL in practice. In particular how to handle client personalization (server generalization vs client overfitting), asynchronous communication and identify biases in large scale deployment (users, devices, geographic and even cultural biases).

## References


[Almaslukh, 2017] Almaslukh, B. An Effective Deep Autoencoder Approach for Online Smartphone-Based Human Activity Recognition. Int. Journal of CS and Network Security. 2017.

[Anguita *et al.,* 2013] Anguita Davide, Ghio Alessandro, Oneto Luca, Parra Xavier and Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition using Smartphones. 2013.

[Arivazhagan *et al.,* 2019] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choud-hary. Federated learning with personalization layers, 2019.

[Bonawitz *et al*., 2019] Keith Bonawitz *et al.,* Towards federated learning at scale: system design, Proceedings of the 2nd SysML Conference, Palo Alto, CA, USA, 2019

[Becker *et al.,* 2019] C. Becker, C. Julien, P. Lalanda and F. Zambonelli, Pervasive Computing Middleware: Current Trends and Emerging Challenges, CCF Transactions on Pervasive Computing and Interaction, 1-14, 2019.

[Cho and Yoon, 2018] Cho, Heeryon & Yoon, Sang. Divide and Conquer-Based 1D CNN Human Activity Recognition Using Test Data Sharpening. Sensors (Basel, Switzerland). 2018.

[Davide *et al.,* 2013] A. Davide, G. Alessandro, O. Luca, P. Xavier, and R.-O. Jorge L, "A Public Domain Dataset for Human Activity Recognition Using Smartphones," 2013.

[Ignatov, 2018], Andrey Ignatov. Real-time human activity recognition from accelerometer data using con volutional neural networks.Applied Soft Computing, 62:915 – 922, 2018.

[Jiang and Yin, 2015] Wenchao J. and Zhaozheng Y. Human activity recognition using wearable sensors by deep convolutional neural networks. In Proc. of the 23rd ACM Int. Conf. on Multimedia, MM '15, pp 1307–1310, New York, 2015.

[Konecny *et al*., 2016] Konecny, J., McMahan, H. B., Ramage, D., and Richtarik, P. Federated optimization: Distributed machine learning for on-device intelligence. arXiv preprint arXiv:1610.02527, 2016

[Kornblith *et al.,* 2019] Kornblith, S., Norouzi, M., Lee, H., & Hinton, G. Similarity of neural network representations revisited. arXiv preprint arXiv:1905.00414, 2019.

[Lara and Labrador, 2013] O. D. Lara and M. A. Labrador, "A Survey on Human Activity Recognition using Wearable Sensors," IEEE Comm Surveys Tutorials, pp. 1192–1209, 2013.

[Lara and Labrador, 2012] O. D. Lara and M. A. Labrador, "A mobile platform for real-time human activity recognition," in IEEE CCNC, pp. 667–671, 2012.

[Lecun et al., 2015] Yann LeCun, Yoshua Bengio & Geoffrey Hinton Deep learning Nature volume 521, pages436–444(2015)

[Li *et al.,* 2019] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, and Bingsheng He. A survey on federated learning systems: Vision, hype and reality for data privacy and protection, 2019

[McMahan *et al.*, 2017] McMahan, H. B., Moore, E., Ramage, D., & Hampson, S. (2016). Communication-efficient learning of deep networks from decentralized data. International Conference on Artificial Intelligence and Statistics, USA, pp 1273–1282, 2017.

[Ronao and Cho., 2015] Ronao, Charissa & Cho, Sung-Bae. Deep Convolutional Neural Networks for Human Activity Recognition with Smartphone Sensors. 46-53, 2015

[Ronao and Cho., 2016] Ronao, Charissa & Cho, Sung-Bae. (2016). Human activity recognition with smartphone sensors using deep learning neural networks. Expert Systems with Applications. 59.2016. 10.1016/j.eswa.2016.04.032.

[Shi *et al.,* 2016] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges", IEEE Internet of Things Journal, Vol. 3, Issue 5, pp. 637–646, 2016.

[Silver *et al.*, 2016] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D., 2016, "Mastering the game of Go with deep neural networks and tree search," Nature, Vol. 529 (Jan. 28), pp. 484-503.

[Sozinov *et al.,* 2018] K. Sozinov, V. Vlassov and S. Girdzijauskas, "Human Activity Recognition Using Federated Learning," 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom), Melbourne, Australia, 2018, pp. 1103-1111.

[Stisen *et al*., 2015] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen.Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, SenSys '15, page 127–140, New York, 2015.

[Sztyler *et al.,* 2017] Timo Sztyler, Heiner Stuckenschmidt, and Wolfgang Petrich. Position-aware activity recognition with wearablee devices.Pervasive Mob. Comput., 38:281–295, 2017.

[Wang *et al.,* 2020] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and YasamanKhazaeni. Federated learning with matched averaging, 2020.