Evaluation of Federated Learning Aggregation Algorithms Application to Human Activity Recognition

Sannara Ek
Grenoble Alpes University
sannara.ek@etu.univ-grenoble-alpes.fr
Grenoble, France

François Portet
Grenoble Alpes University
francois.portet@imag.fr
Grenoble, France

Philippe Lalanda
Grenoble Alpes University
philippe.lalanda@imag.fr
Grenoble, France

German Vega
Grenoble Alpes University
german.vega@imag.fr
Grenoble, France

Abstract

Pervasive computing promotes the integration of connected electronic devices in our living spaces in order to assist us through appropriate services. Two major developments have gained significant momentum recently: a better use of fog resources and the use of AI techniques. Specifically, interest in machine learning approaches for engineering applications has increased rapidly. This paradigm seems to fit the pervasive environment well. However, federated learning has been applied so far to specific services and remains largely conceptual. It needs to be tested extensively on pervasive services partially located in the fog. In this paper, we present experiments performed in the domain of Human Activity Recognition on smartphones in order to evaluate existing algorithms.

KEYWORDS
Federated Learning, Edge Comp, Human activity recognition.

Reference:

Sannara Ek, François Portet, Philippe Lalanda, and German Vega. 2020. Evaluation of federated learning aggregation algorithms. UbiComp/ISWC '20 Adjunct, September 12–16, 2020, Virtual Event, Mexico

1 Introduction

Pervasive computing promotes the integration of connected electronic devices in our living spaces in order to assist us in our daily activities. We are already surrounded by such smart devices providing a number of services. These services are today still limited but they are nonetheless raising huge economical and societal expectations in many domains including industry and healthcare. In particular, advances in pervasive computing have the potential to provide non-intrusive services enhancing health monitoring of elderly and patients. Recently, there has been a strong movement to better use resources close to devices in order to store data and run services [1]. This evolution is known as fog or edge computing. Today, most pervasive applications are based on cloud infrastructures. In practice, cloud computing limits the number and type of services that can be implemented because of unpredictable delays, lack of security, privacy issues, and sometimes insufficient bandwidth or excessive costs. The use of edge resources makes it possible to envisage a greater variety and quality of service [2].

Another major change is the urge for AI-based services. Interest in machine learning (ML) approaches for engineering applications has increased rapidly. The goal of an ML system is to train an algorithm to automatically make a decision (prediction, classification) by identifying 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: efficient algorithms, 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. It is then not surprising that there is today an increasing demand to apply AI techniques in pervasive domains where traditional solutions cannot be used for lack of modeling tools and excessive algorithmic complexity.

Dealing with this evolution is very challenging. On the one hand, engineers in most pervasive fields are used to build models of dynamic phenomena and are not proficient in data-intensive approaches. On the other hand, most AI based solutions heavily rely on cloud infrastructures and cannot be easily implemented in fog devices for lack of resources. However, Google recently proposed federated learning (FL) [3,4,5] for distributed model training in the edge with an application of personalized type-writing assistance. Federated learning, in design, is supposed to save com-

munication costs and forward security and privacy by preventing data collected at the terminal level to be sent through the network. It has immediately attracted attention as a new machine learning paradigm promoting the use of fog-level 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 order to assess the interest of FL, we conducted a set of experiments with promising algorithms recently proposed in the literature. These experiments were conducted in the field of Human Activity Recognition (HAR) with smartphones. HAR is a pervasive application particularly suited to FL since activities tend to have generic patterns (e.g., walking involves the same movement sequence 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 to the network. This paper is organized as it follows. First, some background about federated learning and HAR is provided, then the task and experimental settings are presented. Experimental results are finally discussed.

2 Background

2.1 Federated Learning

Federated Learning proposes a distributed machine learning strategy that enables training on decentralized data residing on terminal devices such as mobile smartphones. Federated learning is well in line with the objectives of fog computing in the sense that data and computing are distributed on local devices. This clearly can address problems related to performance, privacy and data ownership. As illustrated hereafter by figure 1, federated learning relies on a distributed architecture made of a server located in a cloud-like facility and a number of devices, called clients. The number of clients is variable and can be dynamic; clients can appear and disappear without notice.

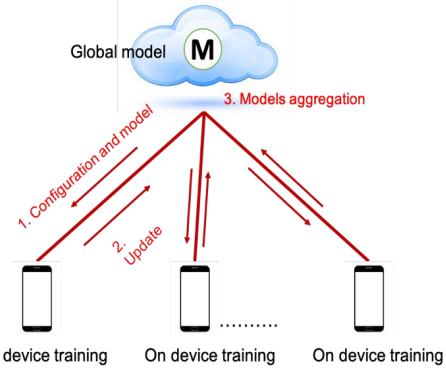

Figure 1. Federated learning architecture

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 models' aggregation. In the first publications related to federated learning, aggregation was implemented as an average function. We call this method FedAvg [3]. This means that the weights of the different local models are averaged to provide new weights and, thus, a new model. New aggregation algorithms have been very recently proposed including FedPer [9], a federated learning algorithm incorporating a base and personalized layer with transfer learning methodologies, and FedMA [10], a federated layer-wise learning scheme which incorporates the match and merging of nodes with similar weights.

Federated learning has been tested and validated on simulated data and on a few domains only, which leaves a number of open questions. 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. It is not clear that on-device training made by different subjects leads to a robust, accurate model at the server level.

Studies on FL with regards to the state-of-the-art in majority focus on solving computer vision problems [3,9,10] and using the results as an open benchmark to compare different algorithms. Among the state-of-the-art FL aggregation algorithms, the FedMA algorithm, layer-wise training and matching scheme paradigm, has shown the most prominent capability.

Only a few studies have applied FL on the HAR domain [12, 13], with yet missing analysis regarding the performance of the global and local models on generalization and personalization with different FL approaches.

2.2 Human Activity Recognition

Human Activity Recognition (HAR) based on wearable sensors, often provided by smartphones, has prompted numerous research works, be they academic or industrial [6]. 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. HAR is useful for health monitoring, senior care and personal fitness training. It also provides high level contextual information that can be used by a large set of personal services. Physical human activities are generally classified from recorded sensor data (e.g. accelerometers, GPS, audio, etc.) which are embedded into wearable devices like smartphones or smart watches.

We focus on research works leveraging machine learning techniques. Regarding classification models, an important number of techniques has 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 do so. Let us cite Bayesian Network, Decision Trees, Random Forest, Neural Network, and Support Vector Machines [7]. 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, however, it turns out that the most popular and effective technology is undoubtedly deep neural networks (as investigated in this paper [14]).

As previously introduced, machine learning is however highly dependent on datasets. It is even more the case with deep learning. The survey conducted by [6] 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 [8], the "stairs" activity represents 22% of the data while the "jumping" one is limited to 2%. In this case, the learning approach should consider the class imbalance problem.

For all these reasons, we believe that HAR is an excellent domain to test and better understand federated learning. Specifically, we appreciate the availability of diverse datasets, diverse approaches with good performances (accuracy, time, etc.) that can be used for comparison, the imbalanced nature of data, and the availability of "natural" clients: smartphones.

3 Experiments

3.1 Settings

We evaluated the performance of the FedAvg, FedPer and FedMa algorithms against a centralized training approach using the REALWORLD dataset [8], a very heterogeneous dataset that closely represents in-the-wild data amongst other known HAR datasets. The REALWORLD dataset contains accelerometer and gyroscope time-series data obtained with Samsung Galaxy S4 and LG G Watch R with a sampling rate of 50 Hz. Data was collected from 15 subjects from 7 different devices/body positions and consisted of 8 activities as shown in table 1. We use a window-frame size of 128 samples with a 50% overlap of 6 channels. To respect the deep learning approach (features should be learned and not hand-crafted), no preprocessing was applied except channel-wise z-normalization. Final size of the dataset was 6.98GB. Our experiments were done using a Convolution Neural Network (CNN) to compare the federated learning results against traditional centralized training in deep learning. Our 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.

Table 1. Activity distribution of all subjects of the REALWORLD dataset (global dataset)

Activities	Instances
Climbing Down	32047
Climbing Up	37520
Jumping	6183
Lying	40843
Running	45581
Sitting	40747
Standing	40672
Walking	41555

For the federated learning experimentation, each subject of the REALWORLD dataset is treated as a client with its own respective data, leading to 15 different clients. Each client's dataset is in turn partitioned into an 80% – 20% ratio to obtain local train and test datasets, used to evaluate performance of the client's model. We also combine all the local train and test sets to respectively generate unified global train and test sets, used to evaluate the behavior of the client's model on unseen data. We used 200 communication rounds for a majority of our experiments and each client trained for a total of 5 local epochs with 0.01 as the learning rate. For the trainings without federated learning, we employed 200 epochs for each model with the same learning rate.

Respecting the same metric as used in the original study of the REALWORLD dataset, the findings on the centralized approach are reported using the F-measure. While the findings for our federated/local learning approaches are presented using their accuracies.

Our studies for the FedAvg and FedMa algorithms scenario consist of 3 different evaluations. The first evaluation uses the aggregated model on the server to test against the global dataset. As the model on the server has, indirectly, seen all train datasets, thus the results of the test can be best used to assess the performance difference between federated learning and the centralized learning approach. In the second evaluation, we test each client's model on its own local dataset, to understand the client's ability to personalize. The third evaluation uses again the client's model, but tests it against the combined global dataset from all the clients. We deduce that this evaluation provides crucial insight on how the client model can perform on data not seen, which underlines one of Federated learning's main benefits.

For the FedPer algorithm, the server lacks a global model and only has the base layer's weights. Naturally, only the two local evaluations are performed (with the client models tested against their own and the combined datasets).

To further the comparison with the traditional learning approaches, we also trained 15 independent models on the individual local datasets, separately without utilizing any federated learning techniques. We refer to this training scenario as the *local learning*. All the results for our tests with the local models are

shown as the mean and standard deviation of the measured metric for all clients.

3.2 Centralized training approach

The original study on the REALWORLD dataset [8] proposed a Random Forest Classifier (RFC) based solution for an activity-independent method, and reached an F-measure of 81%. Recently, the majority of current state-of-the-art HAR solutions incorporate approaches based on some form of CNNs [6].

Our own work displays similar performance results when using a 2-layered DNN, as well as our proposed CNN model (see table 2). The experiment shows that our DNN/CNN models are well tuned and in-line (superior) with the state-of-the-art for this dataset. The shallow CNN model can perform better than the DNN. So for the rest of this paper, As our studies focus on how the different aggregation method of federated learning algorithms affects neural networks in their own way, we will exclusively focus on presenting the results with the CNN.

Table 2. State-of-the-art performances of classical centralized approaches on the REALWORLD dataset.

Literature	Models	F-Measure (%)
[8]	RFC	81.00
Our study	DNN	84.76
Our study	CNN	91.84

The proposed CNN model, with the *centralized learning* approach, achieved an F-measure of 91.84% on the global test-set. When we used the same model to evaluate individually on each local test-set, we obtained a mean accuracy score of 91.99%. The model rapidly converges (as can be seen in Figure 2 and 3.) at around 10 epochs, with little gains afterwards.

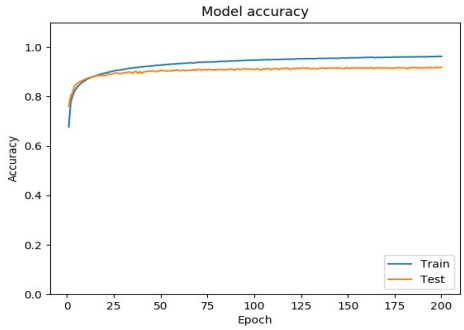

Figure 2. Model accuracy (centralized approach).

In another training instance, where we treat each client's data separately, we trained 15 independent CNN client models on their own local dataset, without using any federated learning techniques. The mean accuracy of the client models with the *local learning* approach is 95.41%, when measured on their own local test-set. If the local models are tested on the global test-set (the combined test-sets of all the clients) the obtained mean accuracy is 52.05% (results are summarized here after in Table 3).

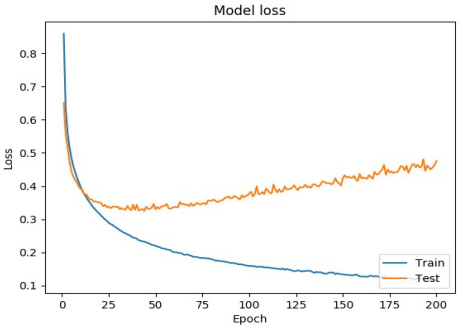

Figure 3. Model loss (centralized approach)

3.3 Federated Averaging (FedAvg)

In the *FedAvg approach*, the weights of the different local models are averaged by the server to provide new weights and, thus, a new aggregated model. Applying the FedAvg algorithm to our CNN model we obtained a model at the server level that generalizes well on the global test-set with 82.74% accuracy (which is far off to the *centralized learning* approach of 91.84%).

On the other hand, at the end of each communication round, clients receive the federated server model and independently train it with their own data. We observed that the obtained federated client models get a significant benefit on their ability to perform on data it has not yet seen: the mean accuracy of the client model when measured on the combined global test-set is 71.22% (compared with 52.05% for the *local learning* model without incorporating federated learning).

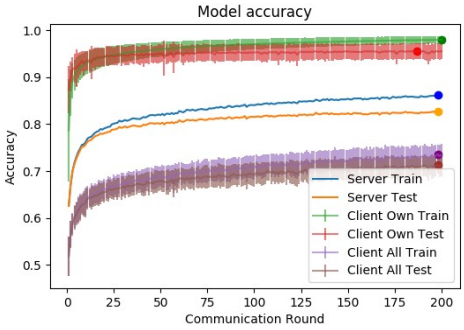

Figure 4. Model accuracy (FedAvg approach)

This benefit comes with no apparent penalty on the client's ability to personalize: the client models trained with FedAvg obtained an accuracy of 95.55% on their own local test-set (that is slightly better than the typical *local learning* accuracy of 95.41%).

As shown in Figures 4 and 5, the models are steadily and slowly improving with even more room to develop, even after 200 communication rounds. It can even be forecasted that with further training, the server model may well perform at the very least on

par with the *centralized learning* approach. Although, as each communication round goes through 5 iterations over the training set, the FedAvg approach ultimately requires a far larger number of epochs when compared with the centralized approach.

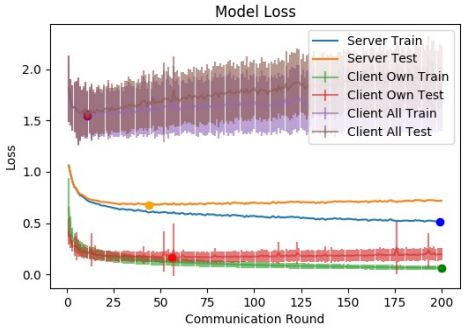

Figure 5. Model loss (FedAvg approach)

3.4 Federated Personalization (FedPer)

The principle of the *FedPer approach* [9] is that the model is split in *base* and *personalized* layers. Personalized layers are not communicated to the server, only the base layers are aggregated by the federated server, using transfer learning methodologies. For the two-layered CNN used in this study, the last dense layer is the *personalized* layer; this means that it is not communicated to the server, only the lower layers are trained using the FL approach.

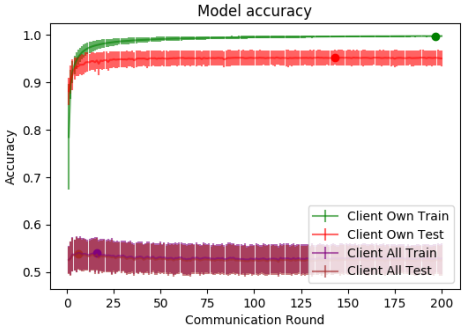

Figure 6. Model accuracy (FedPer approach)

We found that the clients are able to retain their ability to personalize and perform well, with 95.05% accuracy on their local test-sets (only lacking by a little the *local learning* accuracy of 95.41%). Nonetheless, the exhibited client's accuracy of 52.51% on the global test-set suggests that FedPer provides little advantage regarding the client's model ability to generalize.

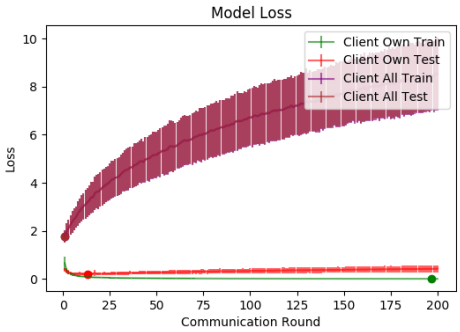

Figure 7. Model loss (FedPer approach)

The FedPer algorithm performed moderately well, with results very close to traditional training without noticeable gains. The algorithm reaches a plateau shortly after the first few couples of communication rounds with only little gains in model performance afterward, as shown in Figures 6 and 7.

3.5 Federated Match Averaging (FedMA)

The principle of the *FedMA approach* [10] is a layer-wise learning scheme, which incorporates the match and merging of nodes with similar weights. Layers are independently trained and communicated to the server.

Applying the *FedMA approach* to our two-layered CNN involves then two intermediate communications with the server (per communication round) to transmit layers weights and perform the matching. In our experimentation, five local epochs are used to train client models (this is a hyper-parameter of federated learning approaches), for FedMa this means in total 25 local epochs for each communication round (five local epochs for each of the first and second layer, and another 15 for the SoftMax layer). Given the substantial number of iterations over the train sets (compared to the other algorithms), we hence decided to limit our experiment to 100 communication rounds for FedMa.

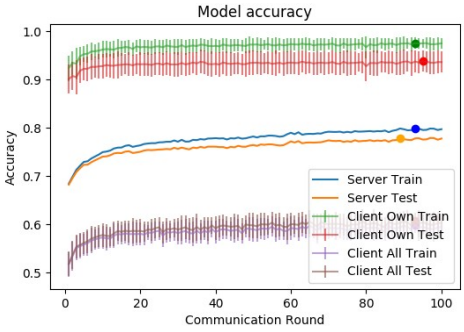

Figure 8. Model accuracy (FedMa approach)

With the *FedMA approach*, the achieved accuracy for the server model is 77.91% (which is behind the *centralized learning* method). The client's model retains its ability to personalize with some minor drawback (the accuracy on the local test-set is 93.80%) while being able to perform well on the global test-set (reaching an accuracy of 60.77%). Based on figure 8, we observe

that the FedMA demonstrates a steady growth rate as training proceeds.

We can remark in figure 9 that for the FedMa algorithm there is a notably large standard deviation for the loss of the client model, when tested against the global dataset, and that this deviation is not attenuated over time. This behavior can be attributed to local clients generating filters or neurons specific to their individual task, which do not contribute to the global task, but instead produce detrimental effects to a certain degree.

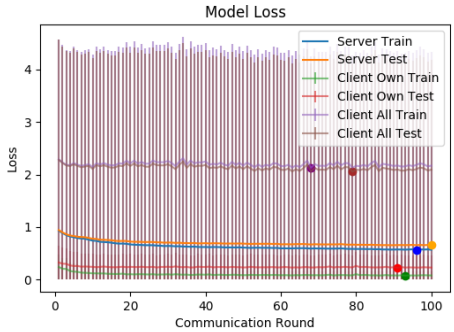

Figure 9. Model loss (FedMa approach)

Despite the higher complexity, training, and communication cost, the *FedMA approach* paled in results when compared with the *FedAvg approach*. As many sub-processes persist with FedMa, further fine-tuning and adaptation are required to achieve results that perform more satisfyingly.

4 Conclusion

Federated Learning exhibits clear theoretical advantages over classical centralized learning from a pervasive computing perspective. But little is known about how these advantages are actually achieved in practice, and the behavior of such learning approaches. In this study we implemented 3 main FL algorithms, and evaluated them on the HAR task. We had expected that this learning scheme should lead to a high degree of adaptation to the device (high client accuracy on its own data) while keeping a high degree of generalization (e.g., prevent over-fitting, high client accuracy on global data).

The results, summarized in Table 3, indicate that the *FedAvg approach* indeed does exhibit such behavior. The results also evidence the limits of other more sophisticated FL algorithms on the HAR task. This calls for more research on FL algorithms considering the ecosystem in which such algorithms are supposed to evolve (tasks, communication, long-term learning).

We see that the baseline FedAvg algorithm obtained better performance against other complex algorithms. This can be said that as FedMA and FedPer have additional designs to improve the personalization of local models, they still incorporate the averaging of clients' property. The averaging of more personalized models leads to a decremental effect to the server model which reflects the lower accuracy when used to evaluate on the global testset.

Table 3. Centralized and Federated learning accuracy results on the REALWORLD dataset.

Approaches	Server accuracy (%)	Client own accuracy (%)	Client all accuracy (%)
Centralized Learning	91.84	91.99	N/A
Local learning	N/A	95.41	52.05
FedAvg	82.74	95.55	71.22
FedPer	N/A	95.05	52.51
FedMA	77.91	93.80	60.77

Although these results add credence to the interest of federated learning for pervasive computing, there still remain a lot of challenges ahead. Future work is needed to study robustness of FL: to asynchronous learning (devices come and go), to sudden change in client data, to communication issues, to heterogeneous population of devices (e.g., traveling device) and to mismatches between server data and clients (noisy acquisition). Furthermore, long term studies are needed to optimize communication schedule and lifelong learning effects such as catastrophic forgetting [11]. We also suggest the community to set up benchmarks for comparison and replication of research in this area and we believe that the study presented here is a stepping stone in this direction.

REFERENCES

[1] 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.

[2] 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.

[3] H.B. McMahan, E. Moore, D. Ramage and S. Hampson, "Communication-efficient learning of deep networks from decentralized data". International Conference on Artificial Intelligence and Statistics (AISTATS) Fort Lauderdale, Florida, 2017.

[4] K. Bonawitz et al., "Towards federated learning at scale: system design", Proceedings of Machine Learning and Systems 2019 (MLSys 2019), Palo Alto, CA, USA, 2019.

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

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

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

[8] T. Sztyler, H. Stuckenschmidt, and W. Petrich, "Position-aware activity recognition with wearablee devices". Pervasive and Mobile Computing, 38, pp. 281–295, 2017.

[9] M.G, Arivazhagan, V. Aggarwal, A.K. Singh, and S. Choudhary. "Federated learning with personalization layers", preprint arXiv:1912.00818, 2019.

[10] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging", preprint arXiv:2002.06440, 2020.

[11] Z. Chen, and B.Liu, "Life long Machine Learning". Morgan & Claypool Publishers, R. Brachman, P.Stone, and F. Rossi Series Editors, 2nd edition. ISBN 1681733021, 2018.

[12] Q. Wu, K. He and X. Chen "Personalized Federated Learning for Intelligent IoT Applications: A Cloud-Edge based Framework". IEEE Open Journal of the Computer Society, vol. 1, pp. 35-44, 2020, doi: 10.1109/OJCS.2020.2993259..

[13] K. Sozinov, V. Vlassov, and S. Girdzijauskas. "Human activity recognition using federated learning". Intl Conference 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, pp. 1103-1111, 2018, doi: 10.1109/BDCloud.2018.00164.

[14] A. Ignatov. "Real-time human activity recognition from accelerometer data using convolutional neural networks". Applied Soft Computing, Vol. 62, pp. 915 – 922, 2018. doi: https://doi.org/10.1016/j.asoc.2017.09.027.

[15] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. Deep learning for sensor-based activity recognition: A survey. CoRR , abs/1707.03502, 2017.