This project has received funding from the European's Union Horizon 2020 research innovation programme under Grant Agreement No. 957258

Architecture for Scalable, Self-*, human-centric, Intelligent, Secure, and Tactile next generation IoT

ASSIST-IoT Technical Report #2

Sunday-FL – Developing Open Source Platform for Federated Learning

Piotr Niedziela, Anastasiya Danilenka, Dominik Kolasa, Maria Ganzha, Marcin Paprzycki, Kumar Nalinaksh

2021 Emerging Trends in Industry 4.0 (ETI 4.0)

Sunday-FL – Developing Open Source Platform for Federated Learning

Piotr Niedziela, Anastasiya Danilenka, Dominik Kolasa
Department of Mathematics and Information Science
Warsaw University of Technology

Maria Ganzha, Marcin Paprzycki
Systems Research Institute
Polish Academy of Sciences
firstname.lastname@ibspan.waw.pl

Kumar Nalinaksh
Institute of Management and Technical Sciences
Warsaw Management University

***Abstract*—Since its inception, in approximately 2017, federated learning became an area of intensive research. Obviously, such research requires tools that can be used for experimentation. Here, the biggest industrial players proposed their own platforms, but these platforms are anchored in tools that they "promote". Moreover, they are mainly "all-in-one" solutions, aimed at facilitating the federate learning process, rather than supporting research "about it". Taking this into account, we have decided to start developing an open source modular flexible federated learning platform. The aim of this contribution is to briefly summarize key aspects of federated learning and, in this context, to introduce our platform.**

I. Introduction

Federated learning (FL) can be traced back to 2017 [1]. Hence, it is a relatively new research area. However, it has roots in research conducted at least at the end of last century. Here, one of classic contributions is the work of Cantu Paz [2], devoted to parallel genetic algorithms. In this work, separate nodes of a parallel computer evolved their own, independent, genetic models and, from time to time, exchanged best (local) genomes. Later, this approach became known as island model of genetic algorithms. In general, this work belongs to the class of approaches known as distributed machine learning, which can be summarized as follows. A computational task (involving some form of machine learning) is split into sub-tasks that are independently executed on separate "nodes". Next, results are combined to create a "common model". Combination of local "results" can be applied at the end of the process, or can be performed repeatedly, after a certain amount of local work is completed. Moreover, delivery of intermediate results, to the shared model, can be facilitated using a "master-slave" approach, where selected node is tasked with model aggregation. It can also be completed using some form of a "peer-to-peer" approach, where selected groups of (or all) nodes exchange their updates (without a centralized "manager"). In the latter case, information exchange continues until the global model "materializes" at each node. Both approaches have advantages and disadvantages, but discussing them is out of scope of this work (for more detials, see [3].

To understand why FL was proposed, and became an instant success, one should notice that majority of "old-style" distributed learning is based on "restricting" assumptions. The most important of them are: (a) all data belongs to a single owner, and (b) all data is processed within a "single computer". The latter should be understood as follows. While it is possible that computing takes place within a network of workstations, loosely connected using PVM [4], or Condor middleware [5], all nodes havw a single owner and should be treated jointly as a part of single "virtual computer".

Keeping this in mind, let us reflect on changes that took place within last few years (and are still ongoing). Obviously, this list is not exhaustive. (1) Smartphones contain multiple (typically more than ten) sensors. Each of them can generate a data stream, while billions of smartphones are in use (and this number is still growing). Here, on the one hand, users may want to control data generated by their devices (this is "their private data"). On the other, they may want to use this data for their advantage, e.g. by deriving from it actionable knowledge. Here, note that although users "have data", usually they lack capabilities required for processing it. Moreover, local data covers a limited "fragment" of knowledge, and does not generalize. For instance, single person mobility patterns will not help improving smart city services. For this, mobility patterns of a majority of citizens are needed.

(2) With technological advancements, the size and prices of sensors continue decreasing, while the number of aspects that can be measured is growing. This enables heterogeneous sensors to be easily placed at "any location". Moreover, such sensors (sensor clusters) can belong to multiple stakeholders, even if they measure the same parameter. Hence, delivery of user-centered applications requires use of data belonging to different owners that may not be willing/able to share it.

(3) Proliferation of wireless networks, including first deployments of 5G networks, allows efficient implementation of sensor networks. This, in turn, makes it easy to establish communication channels between sensors, actuators, edge devices, computing nodes, gateways, cloud services, etc.

(4) The later supports progress in research, development and deployment of Internet of Things (IoT) ecosystems. To deliver services to the users, such ecosystems need, broadly understood, (5) machine learning (ML). Since popular ML approaches are based on large-scale neural networks (NN), (6) hardware dedicated to NN-based ML has been developed. It includes extra small devices that can be placed "almost anywhere", and may belong to different owners. Examples of such devices are Intel Movidius [21], or NVIDIA Jetson Nano [22], but number of such devices is large, indeed.

Finally, (7) data that is generated locally, often cannot be transferred to the cloud for processing. On the one hand, data elements may be too large and too many (e.g. images generated behind a narrow-bandwidth network), or they cannot be released (e.g. medical data, which is legally protected).

This means that the vision of a single owner of data used to train the model to realize individual goals is not the only one. Thus, it should be obvious why, federated learning has been proposed. Here, the simplest metaphor describing FL is: distributed machine learning process, where participating nodes use local data to train private (sub-)model, and share parameters used to modify the common model. In FL, local data is not shared at any stage of training of the shared model.

Interestingly, immediate success of FL, denoted by the number of published results (see, [35], Figure 1) shows that it addresses actual needs of the real-world [35]. Moreover, success of federated learning can be seen through involvement of the largest IT companies, such as Google or NVIDIA.

Here, it can be noted that the ongoing FL research proceeds in two general directions. First, FL is applied in individual domains. The main point of this work is to consider realistic scenarios, take existing, domain specific, data sets, and apply FL. This is possible, among others, since a number of platforms materialized (see, Section II-B). Second, research questions concerning open issues in FL, as an approach to machine learning, need to be studied (see, Section II).

It is the latter that suggests that a different kind of FL platform can be useful. Let us consider the case of unbalanced data sets (see, Section II-C). Here, one of possible solutions, proposed in [32], requires that additional "modules" (called *Mediators*) be instantiated. Obviously, such *Mediators* cannot be easily added to standard FL platforms. Hence, researchers, who explore any ideas out of boxes of existing FL platforms have to implement private FL engines.

To address this issue we have started developing the Sunday-FL platform. The main idea is to provide the skeleton infrastructure for FL, and allow users to develop their own modules that can be put together, to run FL scenarios. Obviously, we assume that, over time, a library of open source modules will be created and made available to the public. In this context the aim of this contributions is to summarize the state of development of Sunday-FL.

II. State-of-the-art

Let us start from a snapshot of research, as represented in contributions stored in the arXiv repository. During Summer of 2020, we have studied content of 250 papers, related to federated learning. Out of these, 24 contained theoretic/analytical results only (no implementation). Approximately 20 had realistic simulations, involving distributed hardware. This means that, among 250 early FL-related publications, almost 80% was based on some form of "internal simulations". Moreover, use of FL platforms was negligible.

A. Application areas

Let us now present few application areas, where federated learning is considered. Let us start from FL in IoT in healthcare (see, [6]). Here, it is claimed that standard FL will not work, because medical IoT devices, have lower computing power and battery capacity, compared to smartphones. Also, the network speed is lower, compared to mobile devices. Hence, a shallow NN was trained on the devices, and a deep NN in the cloud. They experimented with arrhythmia prediction, and tests showed slightly worse (about 2%) accuracy. The second examined aspect was the limitation of network traffic, where savings of over 90% were reported.

Authors of [7] propose application of FL for classifying electroencephalograph signals. Here, FL is used due to the need of personal data protection, and lack of large data sets. Specifically, existing local (small) data sets cannot be combined due to privacy regulations. The proposed solution used covariance-based NN and averaging of weights to update the model. The proposed algorithm achieved 63% accuracy, compared to 66% reported in the best centralized solutions.

NVIDIA Clara delivers AI to customers [8], [9]. Here, NVIDIA offers an edge computing platform, which uses FL in radiology. Participating hospitals label patient data and train the global model. Platform preserves privacy by sharing only partial model weights. There are already partners (mostly in UK and US) who use this solution.

FL is to be used also in drug discovery [33]. Ten pharmaceutical companies agreed to build the common platform in partnership with NVIDIA. Authors claim that fewer than 12% of all drugs entering clinical trials end up in pharmacies, and it takes at least 10 years for medicines between discovery and marketplace. FL is expected to accelerate this process.

In [27] researchers describe use of FL in the domain of mobile devices. They present the high-level system design. The proposed solution aggregates updates from clients without high communication delays and in scale. Data is taken from over 10M daily active devices. Training should be fitted to the users schedule. It is done only when the devices are on unmetered network and charging. Some potential problems are elaborated, such as: low memory devices and bandwidth causes delays.

Federated learning makes its way to image processing. Here, let us mention *FedVision* [10], [11], an application for recognizing objects in photos. The main advantage of FL is the reduction of data transmission (images remain stored locally). Here, each node performs local training and only parameters are communicated. Note that this allows training the model also when data sets belong to different owners. FedVIsion has been used by three corporate customers. It helped them to improve their operational efficiency, and reduce costs, while eliminating the need to transmit sensitive data.

Let us now consider support of maintenance of industrial machines (see, [12]). In the classical approach, machines belonging to a single owner train models relying on their data. Thanks to FL, data of business partners can be included. In referenced work not only FL is considered, but also steps that needed when multi-owner FL is to be used. For instance, input data used to train the model must be properly specified. This means, that semantic interoperability needs to be established

to facilitate the learning process.

An interesting connection between FL and distributed ledgers is discussed in [13]. Here, authors address two problems of standard FL. First is server's susceptibility to failures. Second is lack of reward for clients that put work into training the model. The authors observe also that clients with larger data sets may be less dependent (and willing) to collaborate. In the proposed solution a blockchain network provides rewards for data samples used in training and for the verification process. The reward is proportional to the data sample size.

Final example of applying FL was developed by We-Bank [14], and is dedicated to credit rating. Here, the authors claim that fintech companies need to cooperate, but they can't share their data. Proposed model is restricted to measuring the credit risk of small and micro-enterprises. It is stated that the approach has halved the number of defaults.

B. Federated learning platforms

There exist a number of FL platforms. Let us briefly describe four of them, which we deem most interesting. Let us start from TensorFlow Federated (TFF, current version is 0.18.0, see [15]). It is an open-source platform for decentralised ML and other computations [16]. It uses its own framework to facilitate FL. The Federated Learning (FL) API and the Federated Core (FC) API are its two layers. The (FL) API enables developers to apply FL and evaluation to TensorFlow models already in use. The Federated Core (FC) API is the FL cornerstone. It is a collection of low-level interfaces for implementing federated learning. It supports Mac and Linux platforms only.

PySyft, the second framework, is a Python library (current version is 0.3.0, see [17]) for deep learning [18]. It supports federated learning, Differential Privacy, and Multi-Party Computation (MPC) in PyTorch. It supports also Keras, and Tensorflow. It runs on MacOS, Linux and Windows platforms.

The third framework, is FATE (current version 1.6.0, see [34]). It is an open-source project, initiated by Webank. It implements multiple secure computation protocols to enable collaboration, while supporting data protection regulation compliance [35]. Available ML models include NNs, GBDTs, and logistic regression. It runs on Linux and MacOS. Moreover, it provides MPC and homomorphic encryption. FATE-Flow allows users to define their pipelines of the FL process. It may include data preprocessing, federated training, federated evaluation, model management, and model publishing.

Finally, Flower [36] is another FL platform, currently under development. It is designed to handle large number of clients (10,000 or more) and is platform independent. It supports Keras, TensorFlow, MXNet, and PyTorch. It allows running FL on Android, Nvidia Jetson, MacOS, and the Raspberry Pi. According to its creators, it follows the following core principles: (A) customizability – it allows individual configuration depending on the needs; (B) extensibility – for creation of modern state-of-the-art architectures, elements can be expanded and overridden.

C. Sample research directions

In Section I we have claimed that use of existing all-in-one platforms is not conducive to researching open FL questions. Moreover, analysis of literature, reported in Section II, showed that majority of current work involves "homemade FL" implementations. Let us provide sample reasons, why Sunday-FL platform may be useful in FL research.

Specifically, let us look into research in two areas. First, let us consider the fact that separate devices, participating in FL, may have data distributions different from the global data distribution. In other words, they do not belong to the category of Identically and Independently Distributed (IID) data sets (see, [26]). Here, multiple situations may result in non-IID distributions. (a) Feature distribution skew – individual users with different handwriting. (b) Label distribution skew — mavrud grapes grow only in Bulgaria. (c) Same label, different features — images of village homes vary around the world. (d) Same features, different label -– labels reflecting emotions have regional variation. (e) Quantity skew -– clients hold vastly different amounts of data. Using non-IID data sets can result in significant loss of accuracy. Obviously, globally imbalanced data is another source of accuracy loss.

To mitigate this effect, several techniques were proposed, e.g.: (i) client selection, based on the degree of non-IID data [29], (ii) sharing proxy IID dataset [30], (iii) clustering clients and using multiple models (instead of a single common model, [31]). Finally, (iv) self-balancing FL [32], which involves two steps: (1) data augmentation and down sampling; and (2) client rescheduling and data rebalancing performed by *Mediators* — intermediaries that coordinate the FL process by grouping clients and applying modified/sequential updates to local models (before generating shared model update).

Regardless of the approach, exploring it reaches beyond services offered by standard platforms (summarized in Section II-B). Obviously, each time a private platform could be implemented, but this is leads to waste of effort.

Let us now look into another research area. In [23] address one of known FL problems – malicious clients updates. Aggregating data from such clients in harming the global model, e.g. by lowering the global model accuracy. The three types of attacks are mentioned: sign-flipping attack [24], additive noise attack [24], and backdoor attack [25]. The Anomaly Detection Model algorithm [23] uses the reference data to test local models in order to find abnormal, attacking clients. This method can also detect non-intentional erroneous updates.

In [20] the malicious updates problem is considered from a different angle. The article focuses on three types of attacks: direct attack, indirect attack and a combination of the two. A direct attack is when a device that can participate in training, sends erroneous data. An indirect attack is when an attacker accesses a device on the network and then sends erroneous data. The attack consists of selecting the network coefficients in such a way as maximize harm to the learning process. Considered text discusses a method of creating such update. The conducted research showed that the mean error increased

over 4 times from 5% to 23.88%. The amount of data that has been infected was 20%.

The topic of privacy-preserving data aggregation has attracted a lot of interest. Authors of research on Secure Aggregation [28] proposed a secure way to exchange data within FL, using cryptography primitives and sharing keys between clients. In this way they achieve practical data aggregation for privacy-preserving machine learning. Two variants of the protocol where introduced. First, proven secure against honest but curious adversaries, while the second guarantees privacy against active adversaries. The protocol can tolerate failing devices. Authors say it will be ideal to use with mobile applications.

Obviously, also in this research area existing all-in-one platforms (including Flower, which is the most flexible of them) are not very useful. What is needed, is a solution that is much more amenable to modifications and experimenting.

D. ASSIST-IoT application areas

Let us now look into one more set of potential FL applications. This time they originate from an EU-funded project. The Architecture for Scalable, Self-*, human-centric, Intelligent, Secure, and Tactile next generation IoT (ASSIST-IoT) project started on November 1, 2020. It aims at design, implementation and validation of an open, decentralized reference architecture, associated enablers, services and tools, to assist human-centric applications in multiple verticals (see, [19]). Within its scope, ASSIST-IoT will explore use of federated learning in IoT ecosystems. In this context, the initial assessment of the three pilots that the project will use to validate the developed technologies, indicates that in two of them there is an identifiable immediate need for federated learning.

Smart safety of workers pilot. The aim of this pilot is, among others, to demonstrate use of smart IoT devices functioning in a closed feedback loop of an Occupational Safety Hazard (OSH) risk management. Here, sensors will provide streams of measurements of key workers' health parameters (e.g. heart rate, galvanic skin response, skin temperature, and acceleration) performed by personal health trackers. Moreover, environmental factors (e.g. ambient temperature or UV radiation) will be measured by sensors embedded into protective clothing, or a helmet. Obviously, in this pilot, concerns similar to these described above materialize. Data from a single worker is not sufficient to build high quality health risk assessment models. However, private "medical-type" data is involved, and it has to be protected. Moreover, moving all data to the cloud may be problematic also from the technical point of view. For instance, construction sites may be located in areas with limited network availability.

Cohesive vehicle monitoring and diagnostics pilot. Here, two scenarios related to FL have been identified. First scenario includes vehicle monitoring by the OEM (Ford GmbH, is the project partner), and its repair department, for in-service conformity verification, including regulating emission footprint, in the context of vehicle use phase. Here, note that vehicles my belong to individual owners, as well as fleet owners, including leasing scenarios. The main reasons why FL should be applied are: (a) owners not willing to share driving style describing data, (b) fleet owners not willing to share data with other fleet owners (e.g. car rental companies), or (c) car leasing companies not willing to share data with competitors.

The second scenario deals with vehicle inspection using TwoTronic' (project partner) vehicle scanning solution. Here, multiple vehicles (i.e. 40-50 vehicles per day) visit repair shop for typical maintenance services. The aim of the work is to identify mechanical malfunctions and to monitor vehicle's aesthetic condition, in order to record deterioration of external body, facilitate driver/insurance liability, and to schedule (preventive) maintenance and repair interventions. The diagnostics involves image processing, while FL is needed for the reasons identified above. (i) Sending all data to the cloud is not feasible due to its volume, and (ii) different stakeholders may not be willing to share their data (while being interested in using a shared model to perform inspections on their vehicles).

Overall, within the scope of the ASSIST-IoT project there will be at least three scenarios where FL should be explored. Moreover, these scenarios involve industrial partners. Hence, full control over the data has to be assured. This is one more reason for initiating development of the Sunday-FL platform.

III. SUNDAY-FL – DESIGN CONSIDERATIONS

Based on the approach, described in [27], we extend it and propose an architecture of an FL platform. The key aspects of the design result from the above discussion. There we have argued that for FL research, availability of flexible configuration of FL workflows is necessary. Hence, the client program should be able to realize different ML algorithms. Moreover, approach should not only be platform agnostic, it should also accommodate platform-optimized modules that the client may want to use (instead of generic ones). Finally, it should be possible to define workflows (pipelines) consisting of multiple modules responsible for different functions.

Therefore, we propose design where the client can download a module (in the future, multiple modules) according to the FL scenario it will be involved in. Module that is downloaded can represent any machine learning approach, run on any platform, and be optimized for any hardware. It can also be generic and run on "any system". To realize such platform, an actor-inspired approach is used. Figure 1 depicts the birds-eye view of the system and main actors.

In the proposed system we recognize the following actors. (1) *Cooperation Manager* – facilitates interactions of *Personal Agent*s, within the learning process. (2) *Injector Agent* is responsible for providing needed learning modules. In the future it will manage library of available modules. (3) *Coordinator* is the top manager of a single federated learning process. (4) *Master Aggregator* facilitates aggregation of parameters into the shared model during given FL process. (5) *Regional Aggregator* is a short-lived actor, which takes care of model updates within a single round (those actors are spawned for each round, separately). (6) *Verificator* in the future this optional actor will be available to verify the FL

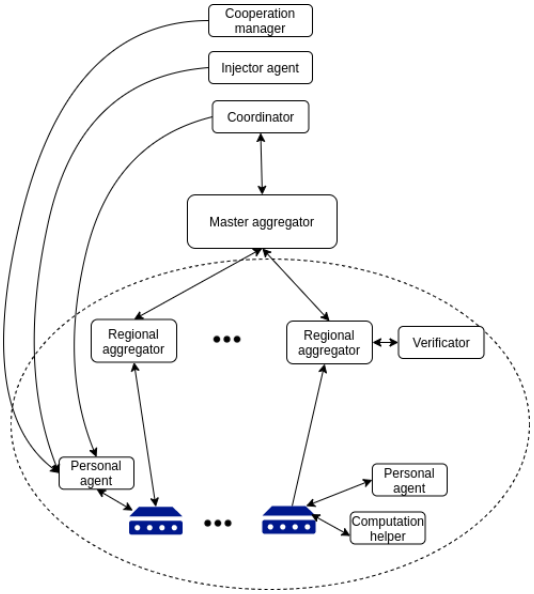

Fig. 1. System-design

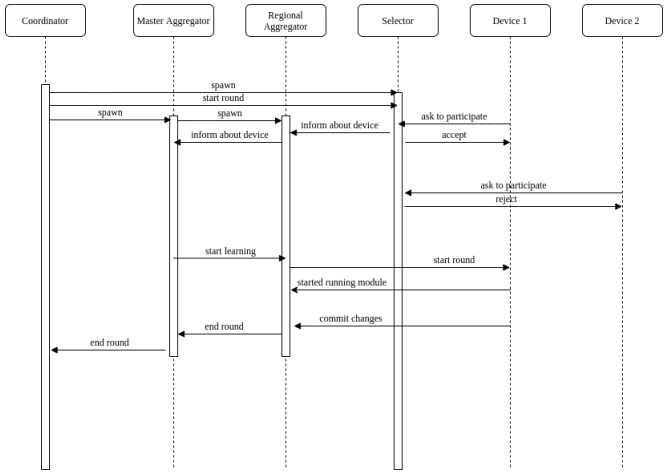

Fig. 2. Process flow – sequence diagram

process. (7) *Personal Agent* represents each device. Finally, (8) *Computation Helper* is planned (in the future) to facilitate realization of complex workloads.

Note that the process of downloading modules is very simple. In the beginning, the *Personal Agent* actor checks if it has a needed module for the task it wants to perform. If it already has the module it can join learning. If not. *Personal Agent* asks the *Injector Agent* actor for the list of modules. When the *Personal Agent* receives the list, it chooses the module needed for training, and downloads it.

IV. Running Sunday-FL

Running the Sunday-FL platform has been depicted in Figure 2. Here we can see that when the project's main module is executed, it spawns the *Coordinator* actor, which in turn spawns *Selector* and *Master Aggregator* actors. The *Master Aggregator* spawns the *Regional Aggregator* actor. Once a device reaches out to the *Selector*, requesting participation in the learning, it further informs the same to the *Regional Aggregator*, which conveys the same to the *Master Aggregator*. The device receives a positive or negative reply from the *Selector*, which decides whether it will be part of learning process or not. Next, *Master Aggregator* starts the learning round. Once the learning is completed, it commits the changes to the *Regional Aggregator* actor. Here, *Regional Aggregator* signals to *Master Aggregator* that the learning round has been successfully completed. *Master Aggregator* conveys the same to the *Coordinator* actor.

For describing the federated learning process, using the Sunday-FL platform, we used a simple example. Specifically, the well-known digit recognition problem, using the MNIST data set.

For training we used a simple backpropagation neural network, with 1 hidden layer, with 128 neurons. We have used standard approach, with basic activation functions. The same module was used in each node. Each experiment consisted of 20 rounds of training and was conducted 10 times.

In addition we run experiments based on suggestions found in [32]. In addition to the base data set, an augmented one was used (where additional images, have been generated by shifting the original ones by one pixel in each direction). Moreover, we have experimented with grouping of clients, so that in a given round the sum of classes was minimized. This was done using the standard Kullback–Leibler divergence. We have also applied both techniques. Results (average accuracy) are summarized in Table I.

Here, let us stress that running reported experiments was not a research objective in itself. Results are presented only to illustrate that the platform actually works.

TABLE I
Results

Used method	Result
Base	72.92 %
Data augmentation	79.99 %
Grouping clients	73.08 %
Both	76.06 %

The Base method applied to basic data set was 72.92% effective. The best result was obtained with data augmentation (80%). Client grouping did not improve performance (73.08%). Interestingly, combination client grouping and data augmentation gave result worse than data augmentation itself, but perhaps this occurred because the classes for the customer data were not diverse enough. As noted, since investigating FL is not a part of this contribution, we leave further study of reasons for this observation for another report.

V. Concluding remarks

While federated learning attracts a lot of attention, as usually in the early stage of development, it is missing tools to support research. Even though there are multiple FL platforms

under development, their use implies reliance on existing tools (e.g. ML libraries). In contrast, Sunday-FL project aims at development of a flexible modular FL platform that can be used, extended, and customised depending on user needs. Sunday-FL is actively developed, and can be accessed from the GitHub repository [37]. Maria Ganzha and Marcin Paprzycki are the primary points of contacts for inquiries, and can be reached at their respective emails. Sunday-FL will be also explored within the context of the ASSIST-IoT project.

ACKNOWLEDGMENT

Work of Maria Ganzha and Marcin Paprzycki was sponsored by the ASSIST-IoT project, which received funding from the EU's Horizon 2020 research and innovation programme under grant agreement No. 957258.

REFERENCES

[1] H. Brendan McMahan et al.,"Communication-Efficient Learning of Deep Networks from Decentralized Data" in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS) 2017. JMLR: W&CP volume 54.

[2] Erick Cantu-Paz. 2000. Efficient and Accurate Parallel Genetic Algorithms. Kluwer Academic Publishers, USA.

[3] Joost Verbraeken et al.,"A Survey on Distributed Machine Learning" in ACM Comput. Surv. 53, 2, Article 30 (June 2020), 33 pages. DOI:https://doi.org/10.1145/3377454

[4] Geist A.,"PVM (Parallel Virtual Machine)" in Padua D. (eds) Encyclopedia of Parallel Computing. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-09766-4_455

[5] J. Frey et al.,"Condor-G: a computation management agent for multi-institutional grids" in Proceedings 10th IEEE International Symposium on High Performance Distributed Computing, 2001, pp. 55-63, doi: 10.1109/HPDC.2001.945176.

[6] Binhang Yuan, Song Ge, Wenhui Xing, "A Federated Learning Framework for Healthcare IoT devices", ArXiv:2005.05083, 2020, [online] Available: https://arxiv.org/abs/2005.05083.

[7] C. Ju, D. Gao, R. Mane, B. Tan, Y. Liu and C. Guan, "Federated Transfer Learning for EEG Signal Classification," 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2020, pp. 3040-3045, doi: 10.1109/EMBC44109.2020.9175344.

[8] AI privacy enabled with Clara federated learning — NVIDIA blog. (2020, July 8). The Official NVIDIA Blog. Retrieved May 3, 2021, from https://blogs.nvidia.com/blog/2019/12/01/clara-federated-learning

[9] NVIDIA Clara. (2021, April 12). NVIDIA Developer. Retrieved May 3, 2021, from https://developer.nvidia.com/clara

[10] Y. Deng, T. Han and N. Ansari, "FedVision: Federated Video Analytics With Edge Computing," in IEEE Open Journal of the Computer Society, vol. 1, pp. 62-72, 2020, doi: 10.1109/OJCS.2020.2996184.

[11] Computer vision platform powered by federated learning. FedAI.org. Retrieved May 2, 2021, from https://www.fedai.org/cases/computer-vision-platform-powered-by-federated-learning/

[12] Hiessl T., Schall D., Kemnitz J., Schulte S. (2020) Industrial Federated Learning – Requirements and System Design. In: De La Prieta F. et al. (eds) Highlights in Practical Applications of Agents, Multi-Agent Systems, and Trust-worthiness. The PAAMS Collection. PAAMS 2020. Communications in Computer and Information Science, vol 1233. Springer, Cham. https://doi.org/10.1007/978-3-030-51999-5_4

[13] H. Kim, J. Park, M. Bennis and S. Kim, "Blockchained On-Device Federated Learning," in IEEE Communications Letters, vol. 24, no. 6, pp. 1279-1283, June 2020, doi: 10.1109/LCOMM.2019.2921755.

[14] Ku, L. (2019, July 29). Tencent's WeBank applying "federated learning" in A.I. Digital Finance. Retrieved May 4, 2021, from https://www.digfingroup.com/webank-clustar/

[15] TensorFlow federated. (2016). TensorFlow. Retrieved May 3, 2021, from https://www.tensorflow.org/federated

[16] Martin Abadi et al., "TensorFlow: A system for large-scale machine learning", In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16) (pp. 265–283), 2016.

[17] Theo Ryffel et al., "A generic framework for privacy preserving deep learning", In Privacy Preserving Machine Learning, NeurIPS 2018 Workshop, December, 2018

[18] PySyft, a Python library for secure and private Deep Learning. (2018). GitHub. https://github.com/OpenMined/PySyft

[19] Architecture for Scalable, Self-*, human-centric, Intelligent, Secure, and Tactile next generation IoT. (2020, November 1). ASSIST-IoT. Retrieved May 2, 2021, from https://assist-iot.eu/

[20] Gan Sun, Yang Cong, Jiahua Dong, Qiang Wang, Ji Liu, "Data Poisoning Attacks on Federated Machine Learning", arXiv:2004.10020, [online] Available: https://arxiv.org/abs/2004.10020 .

[21] Intel. Intel Movidius Myriad 2 VPU Specifications. Available online: https://newsroom.intel.com/wpcontent/uploads/sites/11/2017/06/Myriad-2-VPU-Fact-Sheet.pdf (accessed on 02 May 2021).

[22] A. A. Süzen, B. Duman and B. Şen, "Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN," 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2020, pp. 1-5, doi: 10.1109/HORA49412.2020.9152915.

[23] Suyi Li et al.,"Learning to Detect Malicious Clients for Robust Federated Learning". Available: arXiv.org, https://arxiv.org/abs/2002.00211. [Accessed April. 29, 2021].

[24] Liping Li et al.,"RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets" in 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019 (pp. 1544-1551). AAAI press.

[25] Arjun Nitin Bhagoji et al.,"Analyzing Federated Learning through an Adversarial Lens" in Proceedings of the 36th International Conference on Machine Learning, in Proceedings of Machine Learning Research 97:634-643 Available from http://proceedings.mlr.press/v97/bhagoji19a.html

[26] Peter Kairouz and H. Brendan McMahan,"Advances and Open Problems in Federated Learning" in Foundations and Trends in Machine Learning: Vol. 14: No. 1. http://dx.doi.org/10.1561/2200000083

[27] Keith Bonawitz et al.,"Towards federated learning at scale: system design" in Proceedings of Machine Learning and Systems 1 (MLSys 2019)

[28] Keith Bonawitz et al., "Practical Secure Aggregation for Privacy-Preserving Machine Learning" in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17). Association for Computing Machinery, New York, NY, USA, 1175–1191. DOI:https://doi.org/10.1145/3133956.3133982

[29] W. Zhang et al., "Client Selection for Federated Learning With Non-IID Data in Mobile Edge Computing," in IEEE Access, vol. 9, pp. 24462-24474, 2021, doi: 10.1109/ACCESS.2021.3056919.

[30] Y. Zhao et al., "Federated Learning with Non-IID Data" in arXiv e-prints, 2018. Available: arXiv.org, https://arxiv.org/abs/1806.00582. [Accessed April. 29, 2021].

[31] C. Briggs, Z. Fan and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-IID data," 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1-9, doi: 10.1109/IJCNN48605.2020.9207469

[32] M. Duan et al., "Astraea: Self-Balancing Federated Learning for Improving Classification Accuracy of Mobile Deep Learning Applications," 2019 IEEE 37th International Conference on Computer Design (ICCD), 2019, pp. 246-254, doi: 10.1109/ICCD46524.2019.00038.

[33] K. Wiggers, "Major pharma companies, including Novartis and Merck, build federated learning platform for drug discovery", VentureBeat, 2021. [Online]. Available: https://venturebeat.com/2020/09/17/major-pharma-companies-including-novartis-and-merck-build-federated-learning-platform-for-drug-discovery/. [Accessed: 29- Apr- 2021]

[34] An Industrial Grade Federated Learning Framework. Available online: https://fate.fedai.org/ [Accessed: 29- Apr- 2021].

[35] Li, Q., "A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection" in arXiv e-prints, 2019. Available: arXiv.org, https://arxiv.org/abs/1907.09693. [Accessed April. 29, 2021].

[36] Daniel J. Beutel et al., "Flower: A Friendly Federated Learning Research Framework" in arXiv e-prints, 2020. Available: arXiv.org, https://arxiv.org/abs/2007.14390. [Accessed April. 29, 2021].

[37] Sunday-FL. (2021). GitHub. Retrieved April 30, 2021, from https://github.com/pioek11111/Federated-learning