2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence) | 978-1-6654-1451-7/20/$31.00 ©2021 IEEE | DOI: 10.1109/Confluence51648.2021.9377066

Heralding the Future of Federated Learning Framework: Architecture, Tools and Future Directions

1st Saneev Kumar Das
Department of Computer Science and Engineering
College of Engineering and Technology
Bhubaneswar, India
saneevdas.061995@gmail.com
0000-0002-0097-5102

2nd Sujit Bebortta
Department of Computer Science and Engineering
College of Engineering and Technology
Bhubaneswar, India
sujitbebortta1@gmail.com
0000-0002-7316-5063

***Abstract*—In today's era, the exponential growth of data and its management is a matter of concern. Machine learning has shown its efficacy in multiple application areas. But machine learning on decentralized data was a hectic task since last decade. A novel technology has gained much importance in recent days i.e., federated learning which deals with training on decentralized and distributed data along with preservation of its privacy. Smartphone data being privacy-sensitive is used for locally training a global model which further is aggregated to generate an updated global model which again is distributed among multiple clients. This paper focuses on presenting the efficacy of federated learning by epitomizing an architecture showing the working mechanism of the technology. Further, this paper exhibits an intersection of on-device machine learning, privacy preservation technology and edge computing i.e., federated learning. Also, we have used TensorFlow Federated, an open source platform to simulate federated learning tasks for MNIST and extended MNIST (E-MNIST) datasets. Further, the results contain the loss and accuracy parameters for ten iterations repeated for six optimizer states (Opt_{st}) for each dataset. The peak accuracy that we achieved for MNIST and E-MNIST datasets are 0.843 and 0.853 respectively by using federated averaging algorithm. Further, the minimum loss value that we obtained for MNIST and E-MNIST datasets are 0.652 and 0.646 respectively. The execution time for implementing the algorithm for each dataset is presented in a graphical manner. Finally, certain application areas where federated learning technology has aided are scrutinized.**

***Index Terms*—Federated Learning, Federated Averaging, MNIST, E-MNIST, TensorFlow Federated (TFF), Edge Computing**

I. Introduction

Generalized machine learning approaches are focused on training data which exist in a centralized fashion. One of the most promising cloud infrastructures created by Google provides a platform for training secure data using machine learning. Google claims to introduce a novel approach named federated learning, towards safeguarding data from user interactions over mobile phones and training corresponding models with the aid of machine learning [1]. The approach provides mobile devices, the competency to learn a shared predictive model in a collaborative environment. Federated learning corresponds to a distributed approach of machine learning where the training data residing within diverse end-devices is of decentralized type [2]. Conventional machine learning technology is purely a centralized concept where all the training data needs to be aggregated over a single machine. Huge AI companies, train and store large volume of aggregated data in their data centres. Due to the privacy-intrusive nature of centralized training approaches, mobile device users are a matter of concern [3]. Mobile computing devices in today's era are becoming more and more privacy sensitive. Thus, federated learning offers a decentralized training approach which in turn shall fetch training data from these devices without hampering the privacy concern of the users [4]. To augment user experiences and provide personalized features without interfering with the privacy concern is one of the greatest achievements of federated learning approach. Data in a centralized environment is dependent on a trusted third party but nowadays data is becoming more and more privacy sensitive. Edge computing requires integration with this novel technology due to storage constraints which prevail while working on distributed and decentralized data [5]. Secure aggregation technology has proven itself to be one of the most secure platforms that uses cryptography at its backend. The combination of secure aggregation protocol with on-device machine learning models shall lead to a never-ending growth of applicability in the field of machine learning [6].

This paper has its prime focus in providing an overview for securing the involved data in multiple applications of machine learning. The architecture for practically realizing federated learning environment is presented. Also, we scrutinize certain applications of federated learning and try to reveal the future of machine learning necessitating data security. The practicability of federated learning is of utmost importance and this paper endeavours the adoption of latest tools in order to visualize it. We have used TensorFlow Federated API in order to evaluate certain performance parameters of two non-IID datasets i.e., MNIST and extended MNIST (E-MNIST). The highest accuracy obtained for MNIST and E-MNIST datasets are 0.843

and 0.853 and minimum loss achieved are 0.652 and 0.646 respectively. Also, the execution time required for both the datasets for six optimizer states (Opt_{st}) are presented in a graphical manner.

II. Related Works

In this section, a comprehensive account of some recent works using federated learning are presented. Yang et al. [7] focused on the two major issues faced by current day artificial intelligence that needs to be tackled i.e., the isolated form of existing data in AI applications and the aspects of data privacy and also safeguarding data. This paper also presented the federated learning framework with an in-depth knowledge through performed literature survey. This survey article also provided proper definitions, relevant architectures and federated frameworks in detail. Furthermore, this paper suggested some inevitable applications of federated learning in tackling many current challenges. The focus was on constructing a data alliance among enterprises using federated learning. This article clearly promoted federated machine learning towards improving the investigation of techniques in order to integrate data in such a way that data privacy as well as security are complied with. Smith et al. [8] focused on challenges faced while training any machine learning model in a federated paradigm. This paper proposed multi-task learning as an efficient solution to handle the issues faced in a federated setting. Furthermore, the article proposed a novel optimization technique acronymed MOCHA. The issues addressed in this article were: stragglers, cost of communication and fault-tolerance over a distributed network of devices for multi-task learning. This article considered some federated datasets for simulating multi-task learning and recorded prediction errors along with a comparison of global model, local model and multi-task learning model. Furthermore, the paper proved the efficiency of multi-task learning model to be the best fit. The considered datasets were Human Activity dataset, Google Glass dataset and Vehicle Sensor dataset. Furthermore, this paper presented the use of MOCHA optimization to deal with statistical and system challenges faced in current federated setting.

Konečný et al. [9] considered some specific algorithms of federated machine learning to train distributed data in such a way that individual clients can independently work over a centralized machine learning model. Furthermore, this paper proposed several brilliant ways in order to diminish uplink communication cost. This article presented an experimental setup created with the aid of convolutional neural network (CNN) and recurrent neural network (RNN) to prove that the proposed ways undeniably try diminishing the uplink communication costs in an efficient manner. The datasets used for experimentation included CIFAR-10 Image Classification and the public Reddit post data from Google Big Query. Two generalized cost optimizing techniques for uplink communication were used and validated through deep neural networks and those were: Structured Updates and Sketched Updates where the former provided a direct learning approach for restricted domains whereas the latter signified the learning of the complete model followed by compression for restrained environments.

McMahan et al. [10] focused on the practical visualization of federated learning over deep neural networks with the convergence of a federated technique called iterative model averaging. Furthermore, this article presented empirical results by considering four significant datasets over the proposed model. The prime perspectives of the proposed model were the communication cost optimization techniques indulged within. This paper provided a suitable research scope towards training of decentralized data over mobile devices. The algorithm used named Federated Averaging Algorithm was proved to act efficiently when exposed to unbalanced and non-IID data. Almost the most popular deep networks were used in the experimental setup such as multi-layer perceptron, CNNs and long short-term memory (LSTM) networks. The outcomes showed that the Federated Averaging algorithm was competent enough to train high-dimensional models along with reduced rounds of communication. Kang et al. [11] performed a novel research towards overcoming the challenges based on privacy such as incentive mechanisms. This paper tried to measure the reliability of mobile devices with the aid of reputation as a parameter. Furthermore, this article proposed a reputation-based worker selection scheme assisted by a multi-weight subjective logic model. Qu et al. [12], proposed a blockchain-based federated learning approach for management of big data in industrial and data-driven manufacturing applications. Here, federated learning approach was employed to facilitate efficient processing and privacy preservation of data, whereas the blockchain technology was employed for providing decentralization and implementation of incentive mechanism in the cognitive framework. The convergence of these technologies leveraged advanced functionalities for industrial environments like verification and membership selection. Further, the model's efficacy in addressing adversaries was improved by using an optimized Markovian decision process which was observed to fortify the architecture against poisoning attacks in a decentralized environment.

Zhou et al. [13], presented a recommender system based on federated learning approach for making personalized recommendations in online learning platforms. The fundamental objective behind this work was to concede educational institutions to present a conglomerated collection of academic content for promoting in-house learning. The authors used a 3A approach which signified the interaction between actors, assets, and activities. This model further extends the concept of REST APIs for facilitating acquisition of data in JSON, or XML formats from individual learning platforms. Huang et al. [14], proposed a secure loss-based adaptive boosting (LoAdaBoost) framework for sharing of healthcare data among diverse platforms. This addressed the privacy issues associated with sharing of sensitive intensive care data over multiple devices and data silos. The federated learning model was trained by considering data from different sources and performed aggregation upon the individual locally trained models.

Further, models with high cross-entropy loss were optimized prior to averaging of the models. The model was tested for both IID and non-IID data distributions for intensive care data collected from different hospitals. It was observed that the LoAdaBoost model achieved higher performance in terms of prediction accuracy and computational costs as compared to other baseline learners.

III. Materials and Methods

In this section, the architecture epitomizing the working mechanism behind federated learning along with certain predominant tools and techniques are presented.

A. Federated Learning Architecture

The architecture of federated learning is presented in Fig. 1 which entails a sequence of tasks. Initially, a pre-trained global machine learning model is designed which is stored in the server. The model is then distributed among millions of smartphones with the assistance of a cloud distributor. The smartphones then perform localized training on the global machine learning model. The prime concern here is the privacy of the smartphone users which is preserved with the use of federated learning approach. The locally trained model from each smartphone is collected in a cloud paradigm and with the assistance of a federated learning algorithm the model is updated. Several federated learning algorithms like federated averaging, federated stochastic gradient descent (Fed. SGD) exist to perform the task of aggregating the outputs from various smartphones. Also, the algorithms to aggregate the locally trained models can be customized as per the developer requirements. Further, the extensively trained model is sent to the server from where the model is distributed globally. The entire sequence of tasks is an iterative process and the process continues till the desired performance of the model is achieved. Thus, federated learning can perform training on privacy-preserved decentralized data by distributing the model among multiple clients.

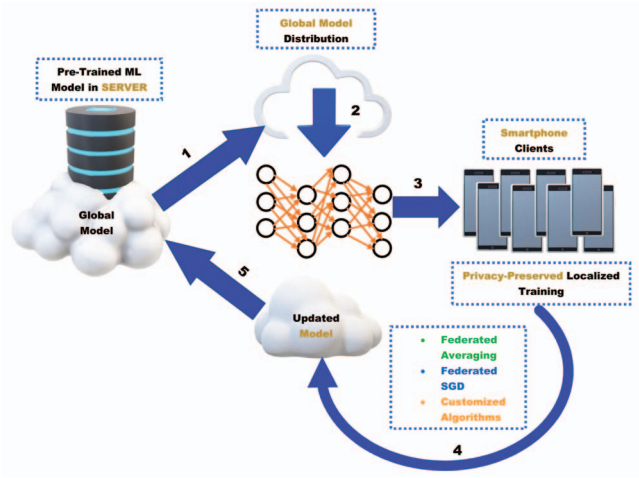

Fig. 1. Architecture of suggested federated learning framework.

B. Tools and Techniques

TensorFlow is an open-source library which aids in implementation of high-ended machine learning tasks. TensorFlow Federated acronymed TFF is a designed environment which offers interfaces consisting of three pivot elements viz., models, federated computation builders, and datasets. The namespace required to define the interfaces is "tff.learning" and to simulate data with scientific relevance and research purposes is "tff.simulation". A runtime environment is provided to practically realize the interfaces with the assistance of "Federated Core (FC)".

With the large-scale popularity of federated learning, several robust software frameworks are evolving rapidly which have been specifically developed for reducing performance overheads associated with training non-IID datasets [15]. Among these software considerations, PySyft is a popular python based library which can be implemented in TensorFlow as well as PyTorch environments for locally training non-IID data by leveraging sophisticated federated computations and encryption mechanisms. PaddleFL is also an open source framework which supports a rich collection of built-in federated libraries, and can be efficiently used for large-scale technical support for industrial platforms. Further, LEAF is a popular federated environment which allows modular code implementation and can be used with existing services. It also facilitates the users for building their codes upon some built-in non-IID datasets like Shakespeare, and E-MNIST.

C. Framework

The framework which has been used in order to implement the federated learning techniques is presented in Fig. 2. The framework is three-tiered and the tiers include "Dataset Selection", "Federated Learning API" and "Performance Evaluation". The dataset selected for implementation is of non-IID type i.e., neither independently nor identically distributed data. We have used MNIST and E-MNIST datasets for simulating federated learning. Further, the datasets are subjected individually to the federated learning API where it is simulated using federated aggregation and federated computation builders. After iterating multiple times for each dataset, accuracy and loss parameters are evaluated. Also, CPU time can be evaluated since it describes the execution time which should be desirably low.

IV. Emerging Application Areas of Federated Learning

The federated learning framework may comprise of dynamically trained machine learning models over several decentralized platforms by leveraging edge computing devices. According to [16], most smart IoT-based environments generate massive data perceived by sensor nodes. This gives rise to the big data issue, thus posing challenges for time and resource constraint tasks. Hence, federated learning approach could be suitable for such environments, in order to achieve a decentralized execution of tasks over multiple autonomous

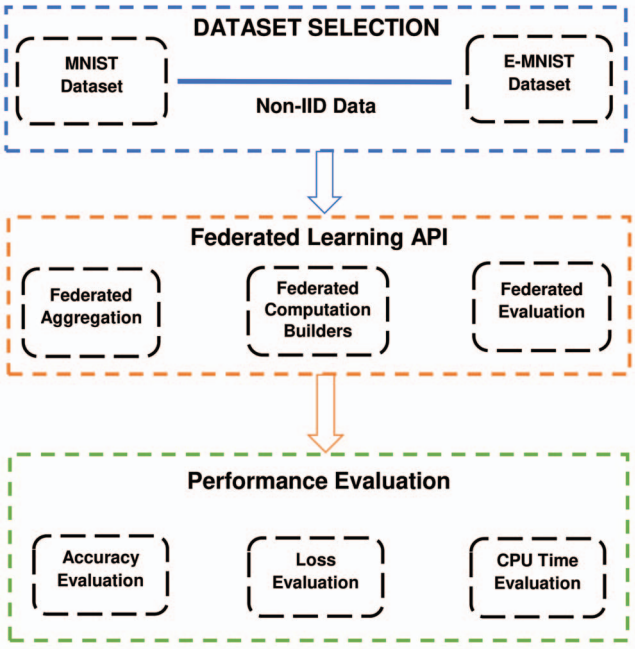

Fig. 2. Framework for real-time implementation and visualization of working mechanism behind federated learning.

edge computing devices and at the same time preserve privacy of sensitive information. In this, section a comprehensive account of some crucial application areas for deployment of federated learning framework is discussed. The four major applications areas considered for our study are as follows:

A. Healthcare Services

The present day healthcare systems involve a huge collection of demographical health data acquired from body borne sensors. The substantial health information obtained after processing such data, combined with other clinical attributes constitute of electronic health records (EHRs) [17]. In order to attain more pervasiveness in the present healthcare system, the EHRs may be transmitted over multiple heterogeneous networks across different geographic regions. However, some important hindrances associated with these systems arise mostly due to delays encountered in transmission and processing speed, costs associated with processing such data, and privacy concerns. Thus, the federated learning approach addresses most of these issues by facilitating computation of time critical tasks on the edge which also minimizes the bandwidth requirements as well as computing costs associated with processing of huge medical data. Further, federated learning framework also focuses on privacy preservation of sensitive medical information by leveraging decentralization to prevent security breaches from intrusive devices in the network.

B. Smart Industrial and Manufacturing Firms

Smart industries usually generate a massive amount of data mostly being produced from industrial automations and sensing activities. These industrial infrastructures are mostly dependent on centralized data centres for storage and processing of industrial data [18]. The decision making and autonomous capabilities in such environments may be restraint due to limited availability of computational resources, bandwidth issues, and other concerns such as security threats or system failures arising due to a centralized architecture. Thus, in accordance to this, the federated learning framework can prove to add more computational power for on-time processing of the industrial data generated from an extensive collection of industrial devices. It can also efficiently address challenges of the centralized architecture such as system failures and security issues through decentralization. The data collected from various devices can be trained at a modular level and can subsequently reduce the computational workloads associated for training conventional machine learning algorithms in a centralized system.

C. Service Recommendation Platforms

IoT-based systems tend to incorporate a multitude of services starting from smart healthcare services to smart industrial automations by enabling developers to build composite value-added web services by combining existing web APIs giving rise to IoT mashup applications [19]. In this context, service recommendation systems come into existence as they provide more accessibility in service discovery and integration. Since, IoT-based environments are capable of generating a large collection of data, it becomes essential to derive new solutions for extracting and integrating the required services [20]. Thus, the federated learning approach can address some of the non-trivial tasks associated with management and discovery of services in dynamic IoT-based environments. This involves recommendation of context-aware services for increasing data processing speed, adaptability, and confidentiality of information associated with such environments.

D. Real-time Object Tracking

With the increase in popularity of IoT-based smart environments, real-time object tracking has become an integral application for most smart environments like smart transportation, smart parking lots, smart cities, and so on [21], [22]. Real-time object tracking systems adopt several sensor-based devices, learning algorithms, Internet connectivity, and cloud computing platform to enable collection, processing, and detection of object information. However, these systems non-trivially require high computing and storage capabilities to ensure fast detection of objects in near real-time applications. Further, privacy of information in such environments is crucial as multiple remote users may have control of the information. With the aid of federated learning framework most of these non-trivial challenges can be addressed as it leverages functionalities of edge computing to perform processing of data acquired by object tracking systems on the edge. The processed data is locally trained which ensures enhancement in computational speed, communication bandwidth, and privacy of information.

V. Results and Discussion

For the practical realization of federated learning, a federated dataset which is generally of non-IID type is required. Thus, we have selected two federated datasets viz., MNIST and E-MNIST (Extended MNIST) collected from [23] respectively. Since both the datasets are used popularly for image classification problem, we have applied federated learning application programming interface i.e., "tff.learning" to study the performance parameters for each dataset.

Initially, the considered datasets i.e., non-IID MNIST and E-MNIST were loaded into the TensorFlow API by specifying six pre-fetched sampled client data epitomizing data heterogeneity in a federated environment. Further, to deal with the noise we perform preprocessing of the data. The client data after getting trained is ready to generate a global model by using any federated algorithm. We have selected the optimization functions to be "client_optimizer_fn" i.e., used for updating local model and "server_optimizer_fn" i.e., used for aggregated update of global model. The learning rates for client optimizer and server optimizer were selected to be $\eta = 0.02$ and $\eta = 1.0$ respectively. The algorithm selected to perform federated aggregation is "Federated Averaging". The performance parameters were determined by selecting the number of iterations to be ten and six optimizer states.

The respective loss values corresponding to MNIST dataset for federated learning approach is provided in Fig. 3 for six optimizer states. It is observed that the loss value substantially decreases with the increase in optimizer states indicating the learning proficiency of the federated averaging algorithm. In Fig. 4, the surface plot for prediction accuracy pertaining to six optimizer states performed iteratively for ten consecutive iterations of the MNIST dataset is presented.

In Fig. 5, loss values for the E-MNIST dataset obtained for six specific optimizer states are provided. The loss values were computed for ten iterations for all six optimizer states, and it was observed that the loss values for training the federated model over the considered dataset distinctively reduced for the sixth iteration. The prediction accuracy for E-MNIST data is provided in Fig. 6.

The minimum loss as well as peak accuracy obtained for MNIST dataset are 0.652 and 0.843 respectively. Similarly, the obtained minimum loss and peak accuracy for E-MNIST dataset are 0.646 and 0.853 respectively. The CPU time is an essential measure of performance for determining the efficiency of a learning algorithm. Therefore, the CPU time for i^{th} iterations can be expressed as, $^1\!/_n \sum_{i=1}^{n} T_i$, for some workload n. The CPU time for the proposed federated learning approach over the MNIST and E-MNIST datasets is presented in Fig. 7. The CPU time has been computed for six optimizer states.

VI. Concluding Remarks and Future Scope

This paper epitomized the proficiency of federated learning technique in training decentralized data. The datasets we considered were of non-IID type which makes the data compatible with federated learning. The federated averaging algorithm

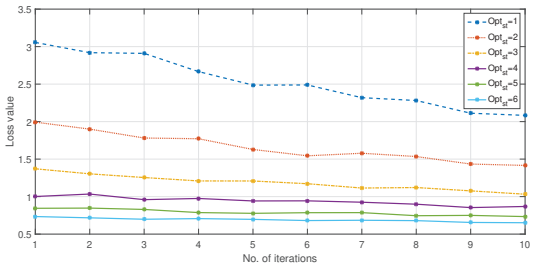

Fig. 3. Loss values corresponding to MNIST dataset for ten iterations.

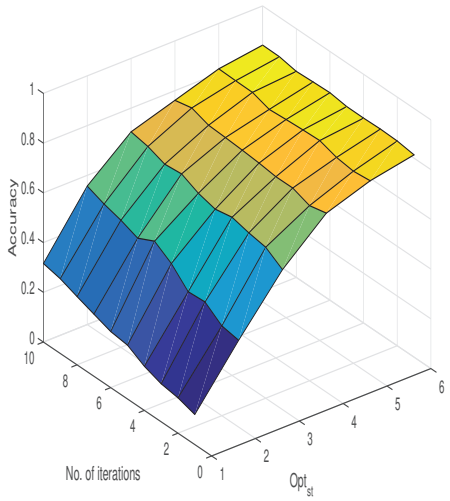

Fig. 4. Accuracy of MNIST dataset with ten iterations and $Opt_{st} = 1$ to 6.

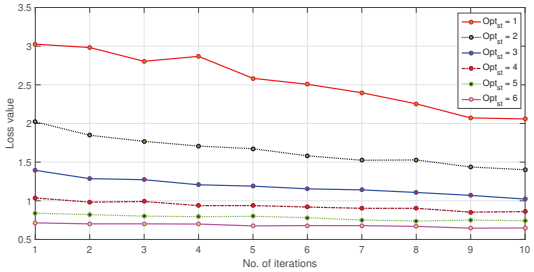

Fig. 5. Loss values corresponding to E-MNIST dataset for ten iterations.

was used to aggregate the locally trained models into an updated global model. We achieved significantly convincing accuracies after a few iterations which signifies the increment in performance parameters due to multiple times locally training the data and constantly updating the global model. The execution time for implementing federated averaging algorithm and obtaining performance parameters were shown for the considered datasets. Federated learning is an iterative process and the prime benefit is the privacy-preserving nature of the technology. For preservation of privacy blockchain

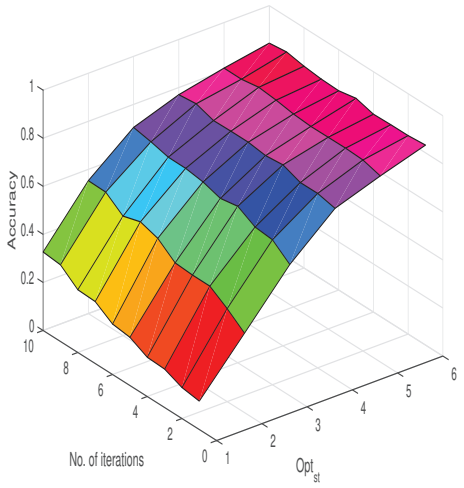

Fig. 6. Accuracy of E-MNIST dataset with ten iterations and $Opt_{st} = 1$ to 6.

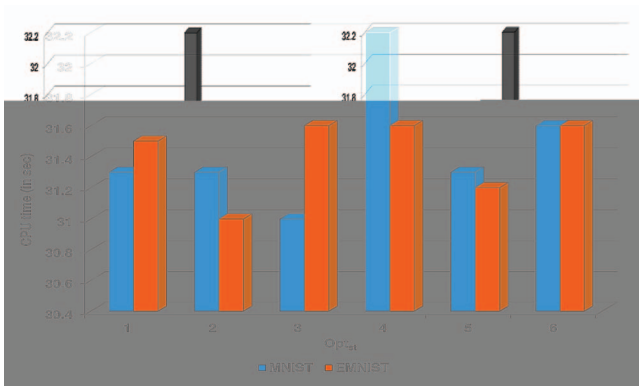

Fig. 7. CPU time (in sec) for six specific optimizer states corresponding to MNIST and E-MNIST datasets.

technology, secure-aggregation protocol and many such can act as a future research direction in this domain. Further, the aggregation of locally trained models into an updated global model is confined to two algorithms presently viz., federated averaging and federated stochastic gradient descent (Fed. SGD). Designing novel algorithms with better optimization strategies, lesser execution time and better performance parameters can also be a future scope in this direction.

REFERENCES

[1] Brendan McMahan and Daniel Ramage. Federated learning: Collaborative machine learning without centralized training data, Apr 2017.

[2] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

[3] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[4] Yongfeng Qian, Long Hu, Jing Chen, Xin Guan, Mohammad Mehedi Hassan, and Abdulhameed Alelaiwi. Privacy-aware service placement for mobile edge computing via federated learning. *Information Sciences*, 505:562–570, 2019.

[5] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 2020.

[6] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for federated learning on user-held data. *arXiv preprint arXiv:1611.04482*, 2016.

[7] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):12, 2019.

[8] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.

[9] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[10] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

[11] Jiawen Kang, Zehui Xiong, Dusit Niyato, Shengli Xie, and Junshan Zhang. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. *IEEE Internet of Things Journal*, 6(6):10700–10714, 2019.

[12] Youyang Qu, Shiva Raj Pokhrel, Sahil Garg, Longxiang Gao, and Yong Xiang. A blockchained federated learning framework for cognitive computing in industry 4.0 networks. *IEEE Transactions on Industrial Informatics*, 2020.

[13] Lei Zhou, Sandy El Helou, Laurent Moccozet, Laurent Opprecht, Omar Benkacem, Christophe Salzmann, and Denis Gillet. A federated recommender system for online learning environments. In *International Conference on Web-Based Learning*, pages 89–98. Springer, 2012.

[14] Li Huang, Yifeng Yin, Zeng Fu, Shifa Zhang, Hao Deng, and Dianbo Liu. Loadaboost: Loss-based adaboost federated machine learning with reduced computational complexity on iid and non-iid intensive care data. *Plos one*, 15(4):e0230706, 2020.

[15] Nuria Rodríguez-Barroso, Goran Stipcich, Daniel Jiménez-López, José Antonio Ruiz-Millán, Eugenio Martínez-Cámara, Gerardo González-Seco, M Victoria Luzón, Miguel Angel Veganzones, and Francisco Herrera. Federated learning and differential privacy: Software tools analysis, the sherpa. ai fl framework and methodological guidelines for preserving data privacy. *Information Fusion*, 64:270–292, 2020.

[16] Maria Ijaz Baig, Liyana Shuib, and Elaheh Yadegaridehkordi. Big data adoption: State of the art and research challenges. *Information Processing & Management*, 56(6):102095, 2019.

[17] W Nicholson Price and I Glenn Cohen. Privacy in the age of medical big data. *Nature medicine*, 25(1):37–43, 2019.

[18] Linghe Kong, Xiao-Yang Liu, Hao Sheng, Peng Zeng, and Guihai Chen. Federated tensor mining for secure industrial internet of things. *IEEE Transactions on Industrial Informatics*, 16(3):2144–2153, 2019.

[19] Buqing Cao, Jianxun Liu, Yiping Wen, Hongtao Li, Qiaoxiang Xiao, and Jinjun Chen. Qos-aware service recommendation based on relational topic model and factorization machines for iot mashup applications. *Journal of Parallel and Distributed Computing*, 132:177–189, 2019.

[20] Marwa Boulakbech, Nizar Messai, Yacine Sam, Thomas Devogele, and Mohammad Hammoudeh. Iot mashups: From iot big data to iot big service. In *Proceedings of the International Conference on Future Networks and Distributed Systems*, 2017.

[21] Sujit Bebortta, Nikhil Kumar Rajput, Bibudhendu Pati, and Dilip Senapati. A real-time smart waste management based on cognitive iot framework. In *Advances in Electrical and Computer Technologies*, pages 407–414. Springer, 2020.

[22] Sujit Bebortta, Dilip Senapati, Nikhil Kumar Rajput, Amit Kumar Singh, Vipin Kumar Rathi, Hari Mohan Pandey, Amit Kumar Jaiswal, Jia Qian, and Prayag Tiwari. Evidence of power-law behavior in cognitive iot applications. *Neural Computing and Applications*, pages 1–13, 2020.

[23] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.