A Review on Edge Intelligence based Collaborative Learning Approaches

Lahiru Welagedara
Computing Department
Informatics Institute of Technology
Colombo, Sri Lanka
lahiru.welagedara98@gmail.com

Janani Harischandra
Computing Department
Informatics Institute of Technology
Colombo, Sri Lanka
janani.h@iit.ac.lk

Nuwan Jayawardene
Department of Computer Science & Engineering
University of Moratuwa
Sri Lanka
nsgaj12@gmail.com

2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC) | 978-1-6654-1490-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/CCWC51732.2021.9376119

***Abstract*—Edge Intelligence (EI) based collaborative learning approaches have been researched in the recent years by several researchers. As majority of existing collaborative learning approaches are designed in the context of servers, design and development of novel systems are required in order to apply collaborative learning in resource constrained devices. Researchers possess different interpretations of the existing collaborative learning approaches. This paper presents an analysis of the existing work conducted on collaborative learning approaches on the domain of EI. The strongholds and drawbacks of the existing work will be analyzed to identify an ideal collaborative learning approach to be applied on the Internet of things (IoT) edge. The analysis will further include an investigation into the future enhancements and research gaps. The partitioned model training approach has been identified as the most ideal approach for the IoT edge based on the conducted critical analysis process. The reduction of communication overhead in the partitioned model training approach and the application of the partitioned model training approach in other unexplored deep learning model architectures have been identified as future research directions. This paper is a work in progress section of an ongoing research and in the future the findings will be used to design a system to apply the collaborative learning approach in IoT Edge and bridge the existing research gaps.**

Keywords—edge intelligence, edge computing, AI, collaborative learning, IoT

I. Introduction

With the increasing numbers of mobile computing and IoT products, a huge amount of data is generated in the network edge which results in higher bandwidth requirements [1]. The emergence of advanced novel applications have demanded low network latency [2]. Edge Computing (EC) brings the computational resources from the cloud to edge of the network to perform data processing at the edge of the network which provides benefits such as faster response time and reduction of network latency [3]. The edge here is referred to as the computing or network resource that resides between the data source and the cloud data center. As per the wide range of benefits provided by EC in the computing industry, leading tech companies such as Google and Microsoft have utilized the technology to develop applications such as Google Cloud IoT Edge and Azure IoT Edge to deliver edge computing services to consumers.

The rapid growth in the AI deep learning domain has demanded to develop systems to run AI tasks seamlessly in a resource efficient manner. EI has been evolved by the integration of AI and EC [4]. In the year 2018, Gartner hype cycle identified EI as an emerging technology in the years to come. Furthermore, tech giants such as Microsoft, Intel and IBM have demonstrated advantages of the application of EI which has led to the developments of smart homes [5], live video analytics systems [6] and Industrial Internet of Things (IIoT) [7].

EI based collaborative learning approaches have been applied on deep learning models in the past few years which has proved to provide superior classification performance as a result of the expanded coverage on large volume of training data gathered at the edge of devices. The training process will be conducted in collaboration with multiple devices rather than using a single device. As researchers possess different interpretations of the collaborative learning approaches, the existing approaches have been mainly classified into three categories in this paper namely, distributed machine learning, partitioned model training and data obfuscation/encryption.

This paper is organized in a manner to provide an introduction to Edge Intelligence in Section I. The three types of collaborative learning approaches will be briefly described in Section II. The existing work associated with the collaborative learning approaches will be reviewed in Section III. It will be followed by the critical analysis of the existing approaches in Section IV. The findings from the analysis will be presented in Section V, followed by the conclusion in Section VI.

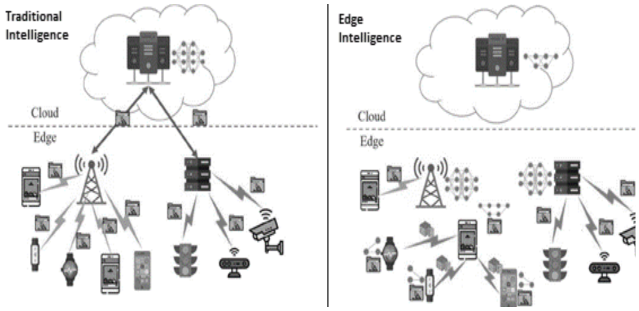

Fig. 1. Comparison of traditional intelligence and edge intelligence

978-0-7381-4394-1/21/$31.00 ©2021 IEEE

II. Methodology

Based on existing research, EI based collaborative learning approaches can be mainly divided into three categories namely, distributed machine learning, partitioned model training and data obfuscation/encryption. A brief introduction of these three learning approaches have been provided in this section.

1) **Distributed Machine Learning** - In the distributed machine learning approach which is also known as Federated Learning (FL), clients (edge devices) train a local deep learning model where training data is kept locally without transferring to cloud. Clients send updates of the trained model to a main coordinator (server) where the global model exists and the global model is updated with the values sent by individual clients. Finally, the updated version of the global model is distributed among local clients.

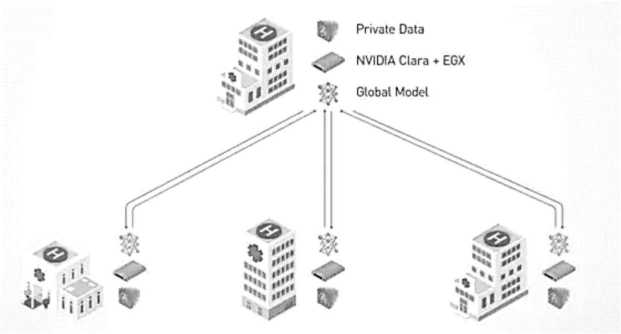

Fig. 2. Federated learning model [10]

2) **Partitioned Model Training** - In the partitioned model training approach which is also known as Deep Neural Network (DNN) partitioning or split learning, a neural network is first partitioned and a number of front layers are deployed at the edge device and rest of the layers of the neural network are deployed in the cloud data center. Forward propagation of the output of the front layers will continue in the cloud, loss is calculated and backpropagated through the network to update the weights of the model. The step is repeated until the model acquires optimal accuracy.

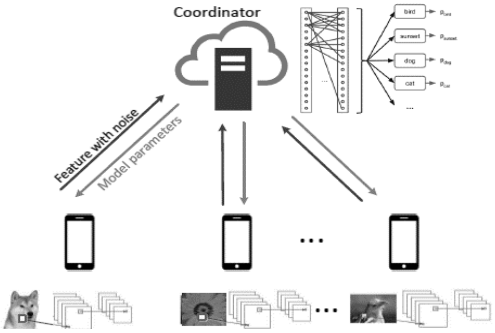

Fig. 3. DNN partitioning model [17]

3) **Data obfuscation/encryption** - In Data obfuscation/ encrypted approach, encrypted/obfuscated training data is transmitted to the coordinator to build deep learning models. Different approaches such as data encryption, noise addition and fake data sample augmentation have been utilized to encrypt or obfuscate training data. Sensitive data in individual or group samples will be protected and further safeguarded from untrusted third parties when this approach is followed. This approach is mainly utlized to preserve privacy of data in a collaborative learning approach.

III. Existing work

The existing systems that have been implemented on the three types of collaborative learning approaches have been reviewed in this section. In some of the existing research, researchers have brought forward their own interpretation of collaborative learning using different naming conventions. For example, distributed machine learning is known by the name federated learning and partitioned model training is known by the name DNN partitioning or split learning. The research that contained various interpretations have been grouped together and then divided into three main categories in order to conduct the review.

A. Distributed machine learning

Distributed machine learning was first introduced by researchers McMahan et al. as a decentralized approach where a shared global model was trained using locally collected data from a collection of distributed clients. The experiments conducted in the research have shown that high quality models could be trained in relatively few communication rounds. Federated-Averaging algorithm was introduced in the same research in order to update the global model by averaging the model updates sent by mobile clients. The evaluations have proved that the convolutional neural network (CNN) model has reached a target accuracy of 99% by decreasing the number of communication rounds by 35x and the 2NN model has reached a target accuracy of 97% by decreasing the number of communication rounds by 46x [8].

FL was utilized to predict the next-word on virtual keyboard of smart phones with the use of recurrent neural networks (RNN) in the research and was considered as one of the earliest applications of FL in a commercial setting. The research showcases the benefits of using FL to execute language models on device which does not require to transmit user's sensitive and private data to servers for processing. The model for the prediction task in this research utilizes a variant of long short term memory (LSTM) which is known as coupled input and forget gate (CIFG). The model has been evaluated on the number of correct predictions to the total number of tokens which is known as a recall metric. It was proven that the CIFG model based on FL outperforms the CIFG model trained on a normal server. Research has also proven that federated CIFG improves top-1 recall by 5% compared with the server trained CIFG on client cache data. When evaluated on server hosted logs data both federated CIFG and server trained CIFG have similar comparable values [9].

Nvidia Clara was an application developed by Nvidia using the FL approach to create a collaborative environment of devices to preserve patient data privacy in medical institutions. The server and clients are provisioned by Nvidia EGX platform. Nvidia EGX platform gathers local data from hospitals and trains the global model by transferring the model parameters through a secure link. Nvidia Clara has been used in several healthcare companies in the world. American College of Radiology uses Clara in its AI lab for medical imaging and allows its users to design, build and share AI models. UCLA Radiology has also harnessed the power of Nvidia Clara to bring AI for its radiology tasks [10].

Researchers Asad et al. have conducted a detailed review of FL algorithms focusing on communication efficiency and has proposed a novel FL strategy. The collaborative learning technique has been divided into three steps by the authors namely, task initialization, local model training and global model aggregation. Some of the challenges that were identified in the FL approach were the non-IID (non-independent-and-identically-distributed) data, number of clients, parameter server and limitations in battery and memory. The authors have evaluated five algorithms based on federated learning namely, Federated-Averaging (FedAvg), Sparse Ternary Compression (STC), Communication-Mitigating Federated Learning (CMFL), Federated Maximum and Mean Discrepancy (FedMMD) and Federated Dropout (Fed-Dropout). All the five algorithms have been tested on CNN models with CIFAR-10 and MNIST datasets. CMFL required fewer iterations to achieve an accuracy of 80% when compared with FedAvg algorithm and achieved highest accuracy in MNIST dataset. In non-IID environments, FedMMD reached target accuracy with 20% fewer communication rounds when compared to FedAvg. The evaluation results in the research has proven that FL preserves data privacy while reducing communication cost [11].

An algorithm was proposed to determine the ideal trade-off between averaging of global parameters and local updates by analyzing the convergence bound of gradient descent based on FL which was also known as aggregation frequency control. A novel convergence bound has been obtained that consists of non IID data distributed in edge devices and local updates from two global aggregations. Afterwards the authors have proposed a control algorithm which learns the model characteristics and data distribution system dynamics and adapt the global aggregation frequency to minimize learning loss. CNN model, K-means algorithm, squared-SVM algorithm and linear regression algorithm have been used for the evaluation along with MNIST, fashion-MNIST, CIFAR-10, user knowledge modelling and energy datasets. The algorithm evaluation has confirmed that the proposed approach has obtained near optimal performance [12].

An existing research proposed edge Stochastic Gradient Descent (eSGD) which was based on gradient compression in order to be applied in FL, to scale up edge training of CNN's and improve the high network communication cost on gradient aggregation. In the first phase, it identifies the important gradient coordinates to be transmitted and synchronized. In the second phase, it tracks and discard old residual gradient coordinates to avoid low convergence rate. The approach was validated on CNN model using the MNIST dataset and the experiments has proven that at the highest gradient drop ratio of 87.5% the accuracy has reached up to 81.5%. eSGD has facilitated to reduce bandwidth consumption and improve scalability [13].

B. Partitioned model training

In order to collaboratively train deep learning models in the health domain without sharing raw data, configurations of a distributed deep learning methodology was proposed which was named as splitNN. Partitioned model training approach was first introduced through this research. The configurations involve the splitting of the neural network layers among the clients and server where output of the cut layer is sent to the server and the server completes the forward propagation from received output layer from client. The gradients will be then backpropagated through the network to update the weights of the model. In the research, VGG and Resnet-50 models were tested on the CIFAR-10 and CIFAR-100 datasets using the partitioned learning approach and has been compared with FL. Split learning outperformed FL in terms of accuracy requiring very low computing resources as per the evaluation. The research has proven that when the number of clients participating in the distributed network is high, low computation bandwidth was required per client when the partitioned model training approach was utilized. Compared with FL, a drawback of splitNN was the requirement of more communication bandwidth when the number of clients participating were relatively less [14].

ARDEN was proposed to partition DNN's across mobile devices and cloud data center. Differentially private data transformation was conducted on the mobile devices and the complex training and inference was conducted on the cloud data center. A privacy mechanism was implemented using noise addition and data nullification. Three types of CNN's namely Conv-Small, Conv-Middle and Conv-Large have been used as DNN's and MNIST, SVHN datasets was used for the experiment. First three layers of the pre-trained Conv-Small network on CIFAR-10 dataset resides in the local neural network and Conv-Middle resides in the cloud datacenter. It was observed that the perturbance of local transformation affected the inference performance at the cloud end. Based on further experiments, ARDEN has proved to preserve data privacy and reduce the resource consumption by over 60% [15].

PipeDream was a research conducted on parallel DNN training based on the partitioned model training approach where graphics processing units (GPU) were considered as the workers. The system parallelizes the computation by using a pipeline across multiple machines and partitions the deep neural network in an optimal manner to minimize the overall training time and assigns the layers to each worker. PipeDream's optimizer partitions the layers and assigns them to the workers when the DNN model, dataset and machine configuration is given. The VGG16, Inception-v3 and S2VT deep learning models and ILSVRC12, MSVD datasets have been used for evaluation. PipeDream has proved that the combination of model parallelism, data parallelism and pipelining outperforms data parallel training. PipeDream has reduced communication up to 95% and is up to 5x faster compared with data parallel training [16].

Researchers Jiang and Lou came up with a privacy preserving collaborative learning technique where the edge devices and cloud collaboratively train a deep neural network. Convolutional layers of the neural network are deployed in the edge devices and the dense layers are deployed in the cloud. To preserve privacy, differential privacy (DP) technique was used in the edge devices. The CNN model was trained on the MNIST handwritten digit recognition dataset where each edge device participated in the training process. Privacy loss level and impact of batch size on the accuracy was observed. The evaluation results had proven that even though noise was added to the data it resulted in only 3% accuracy loss. Furthermore, when the batch size is smaller it resulted in lower noise levels which led to high accuracy [17].

Collaborative learning approach was utilized to partition the CNN based VGG-Face network between client and edge server with added DP mechanism to observe the partitioning effect on the accuracy of the network. In this approach, the client sends the activations of the convolutional layers to the server with added artificial noise instead of sending raw images. Separate evaluations were conducted on the time cost, memory cost, energy cost and network transmission cost of mobile clients. Multiple training has been conducted on different partition schemes in order to investigate the effect of partitioned position. The research has identified that training using small batch sizes is an efficient mechanism when it comes to resource constrained devices and application of DP mechanism ensure the tasks are suitable to be outsourced for untrusted servers [18].

C. *Data obfuscation/encryption*

A novel obfuscate function was implemented to preserve the privacy of training data by adding random noise to sample datasets or by augmenting the training dataset with a new dataset. Sensitive data in individual and group samples are protected using this approach. The approach has guaranteed to preserve privacy of data while achieving high accuracy. The evaluations had proven that the approach was successfully protected from model memorization attack, membership inference attack, model inversion attack and model classification attack [19].

A methodology was proposed to generate artificial data to preserve the privacy of original data by training a generative adversarial network (GAN). In this methodology, artificial data will be generated using GAN for training rather than exposing the original data. A differential privacy (DP) layer was introduced as a regularizer which results in stable training and improvements of image quality. The evaluation was carried out using MNIST, SVHN, and CelebA datasets. The models trained on artificial data has proven to reach high accuracy on SVHN (87.7%) and MNIST (98.3%) datasets. For the demonstration of data protection, a model inversion attack was launched and it was proven that information leakage has been reduced when trained using GAN [20].

The homomorphic encryption (HE) scheme was used to retain privacy of the data that has been used for processing. The scenario consisted of data owners, providers and cloud service providers. Data owners will generate a private or public key to be shared with service providers. Content providers will be responsible for the encryption of data that needs to be transferred to the cloud for processing and building up the machine learning model. Finally, the cloud service providers conducts the processing, building of the model and saves an encrypted version of the model. Data owners will then be able to request for the encrypted model [21].

IV. Analysis of existing work

Table I presents an overall analysis of the existing work reviewed in section III. The strongholds and drawbacks of the three main types of collaborative learning approaches have been analyzed in this section.

TABLE I. ANALYSIS OF THE COLLABORATIVE LEARNING APPROACHES

Approach	**Analysis of Collaborative learning approaches**	
	Advantages	***Disadvantages***
Federated learning	Low communication cost observed when the number of clients participating were less Preservation of data privacy High quality deep learning models can be trained as data is gathered from a distributed environment High model accuracies have been observed Proved to be efficient when the training dataset is larger Proved to be efficient when no. of clients participating in the training process is less	High memory and energy consumption of edge devices as the whole model is trained locally on each device Bottleneck in the aggregation of gradients (gradient averaging) Proved to be inefficient when the model size is large Vulnerable to security threats Application of security mechanisms (DP) cause slight accuracy drops
DNN partitioning	Efficient utilization of edge devices Low computational resources are required from the edge devices (resource efficient) Preservation of data privacy as model parameters are transmitted instead of raw data High model accuracies have been observed High quality deep learning models can be trained as data is gathered from a distributed environment Low memory and energy consumption in edge devices Performs efficiently when trained using small batch sizes of training data Low communication cost observed when the number of clients participating were high Bottleneck in aggregation is mitigated	Inefficient when dataset size is large compared to federated learning Application of security mechanisms (DP) cause slight accuracy drops Risk of vulnerability to security threats unless privacy mechanisms are implemented

Approach	**Analysis of Collaborative learning approaches**	
	Advantages	***Disadvantages***
	Fast convergence rate was observed	
Data obfuscation/ encryption	Preservation of data privacy Satisfactory model accuracies has been observed Memory and energy consumption of edge devices will be slightly less	Approach is mainly targeted to improve data privacy High memory consumption and resource utilization in cloud data centers High communication cost Variations in model accuracies have been observed when different encryption method were used Inefficient when the training dataset is large

V. Findings from existing work

Based on the critical analysis conducted on existing works in section IV, the most ideal collaborative learning approach to be applied on resource constrained IoT edge devices was identified as the partitioned model training (DNN partitioning) approach. The DNN partitioning approach has been selected as the ideal approach as per the below factors that were identified through the analysis.

Computational power and Resource utilization - As IoT devices will be limited in computational power and resources, a resource efficient collaborative learning approach to train deep learning models is required. The low computational power requirement and the efficient utilization of edge computing resources have been identified as the main advantages of the DNN partitioning approach when compared with FL and data encryption approach. Partitioned model training approach has been proven to reduce the memory and energy consumption of edge devices as shown in the works of [14][15]. Application of FL in resource constrained edge devices demands for higher memory and energy consumption as the whole deep learning model will be trained locally on each edge device.

Efficiency in communication cost - Partitioned model training approach has been proven to be more communication efficient with increasing number of clients (edge devices). Even though both DNN partitioning and FL approaches have low communication cost, FL functions poorly with high communication cost when compared with partitioned approach, when the number of devices in the distributed environment is increased as shown in the work of [14]. Training using small batch sizes has been proven to perform well in DNN partitioning approach [17][18]. Data obfuscation/encryption will not be able to compete with the other two approaches, as it mainly focuses on data privacy and not communication cost which leads to a very high communication overhead because the entire set of encrypted training dataset will be transmitted to cloud data centers for processing.

Model quality and Model accuracy - All the three approaches have been proven to obtain good model accuracies. In DNN partitioning approach, it has been found that the convergence is much faster than FL, due to the bottleneck in aggregation or the gradient averaging mechanism in FL as mentioned in [13]. Data encryption approach will maintain good accuracy, but the accuracy varies on the encryption/obfuscation method used. An advantage of both DNN partitioning and FL approaches are that it is proven to build high quality models as data is gathered from a distributed environment of edge devices as shown in the work of [11].

Data privacy - Data privacy will be preserved by all the three approaches. In the DNN partitioning approach data privacy is preserved by only transmitting the parameters to the cloud data center instead of raw data [14][18]. In the FL approach data privacy is preserved by transmitting the gradients to the cloud data center instead of raw data [8]. In order to mitigate the risk of vulnerability to security threats, both DNN partitioning and FL approach can implement privacy mechanisms such as differential privacy. Data obfuscation/encryption preserves data privacy by encryption or obfuscation of the data which is its main advantage as shown in the work of [19]. A slight drawback in both DNN partitioning and FL approach is that it can lead to a slight accuracy loss when privacy mechanisms like differential privacy (DP) are applied [17].

Research gaps and Future enhancements -

As collaborative learning approaches such as FL has been designed with the main focus on context of servers or resources with relatively high computational power such as GPU's and high-end edge servers, the traditional FL scheme cannot be directly applied on resource constrained IoT devices because of the resource and communication overhead caused. Therefore, it is necessary to improve the existing technology in order to support efficient collaborative learning in resource constrained IoT edge devices.

A drawback found in the DNN partitioning approach was the communication overhead that arose when a large dataset is used for training. To reduce the communication overhead, an iterative model training approach based on DNN partitioning to enable the IoT edge devices to train deep learning models with new training data has been identified as a future enhancement.

The existing systems on DNN partitioning approach have been evaluated only on convolutional neural networks and recurrent neural networks. In order to observe the capabilities of the partitioning approach, application of the approach to a more complex deep learning model with better performance has been identified as a future research direction. Through the analysis of existing research, it was observed that application of other deep learning models such as hybrid deep learning models in the DNN partitioning approach are yet to be explored as future research.

Furthermore, exploration of a lightweight virtualization techniques to facilitate efficient Edge Intelligence placement on resource constrained IoT devices has been identified as a research direction of EI through existing research.

VI. Conclusion

The research area of edge intelligence based collaborative learning emerged very recently and some research gaps are yet to be filled. Even though several systems are developed on collaborative learning approaches, conventional schemes such as federated learning are designed in the context of servers and the same traditional scheme cannot be applied on resource constrained IoT edge devices. As per the findings, partitioned model training approach has been identified as the most ideal approach to be utilized in the context of resource constrained IoT edge devices. Reduction of the communication overhead arising due to large training datasets and the application of partitioned model training approach in other deep learning model architectures have been identified as future research directions. With this knowledge about collaborative learning approaches, the authors hope to design and implement an efficient collaborative learning approach for the resource constrained IoT edge.

References

[1] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing," Proc. IEEE, vol. 107, no. 8, pp. 1738–1762, Aug. 2019, doi: 10.1109/JPROC.2019.2918951.

[2] X. Zhang, Y. Wang, S. Lu, L. Liu, L. xu, and W. Shi, "OpenEI: An Open Framework for Edge Intelligence," in 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, Jul. 2019, pp. 1840–1851, doi: 10.1109/ICDCS.2019.00182.

[3] M. Al-Rakhami, M. Alsahli, M. M. Hassan, A. Alamri, A. Guerrieri, and G. Fortino, "Cost Efficient Edge Intelligence Framework Using Docker Containers," in 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, pp.800–807, doi: 10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00138.

[4] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence," IEEE Internet Things J., vol. 7, no. 8, pp. 7457–7469, Aug. 2020, doi: 10.1109/JIOT.2020.2984887.

[5] J. Cao, L. Xu, R. Abdallah, and W. Shi, "EdgeOS_H: A Home Operating System for Internet of Everything," in 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, Jun. 2017, pp. 1756–1764, doi: 10.1109/ICDCS.2017.325.

[6] G. Ananthanarayanan et al., "Real-Time Video Analytics: The Killer App for Edge Computing," Computer, vol. 50, no. 10, pp. 58–67, 2017, doi: 10.1109/MC.2017.3641638.

[7] L. Li, K. Ota, and M. Dong, "Deep Learning for Smart Industry: Efficient Manufacture Inspection System With Fog Computing," IEEE Trans. Ind. Inf., vol. 14, no. 10, pp. 4665–4673, Oct. 2018, doi: 10.1109/TII.2018.2842821.

[8] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Federated Learning of Deep Networks using Model Averaging," Feb. 2016, [Online]. Available: https://www.researchgate.net/publication/301847059_Federated_Learning_of_Deep_Networks_using_Model_Averaging.

[9] A. Hard et al., "Federated Learning for Mobile Keyboard Prediction," arXiv:1811.03604 [cs], Feb. 2019, [Online]. Available: http://arxiv.org/abs/1811.03604.

[10] N. Nvidia, "NVIDIA Clara Federated Learning to Deliver AI to Hospitals While Protecting Patient Data," NVIDIA Clara Federated Learning to Deliver AI to Hospitals While Protecting Patient Data, 2019. https://blogs.nvidia.com/blog/2019/12/01/clara-federated-learning/.

[11] M. Asad, A. Moustafa, T. Ito, and M. Aslam, "Evaluating the Communication Efficiency in Federated Learning Algorithms," arXiv:2004.02738 [cs, eess], Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.02738.

[12] S. Wang et al., "Adaptive Federated Learning in Resource Constrained Edge Computing Systems," IEEE J. Select. Areas Commun., vol. 37, no. 6, pp. 1205–1221, Jun. 2019, doi: 10.1109/JSAC.2019.2904348.

[13] Z. Tao and Q. Li, "eSGD: Commutation Efficient Distributed Deep Learning on the Edge," HotEdge, p. 6, 2018.

[14] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split learning for health: Distributed deep learning without sharing raw patient data," arXiv:1812.00564 [cs, stat], Dec. 2018, [Online]. Available: http://arxiv.org/abs/1812.00564.

[15] J. Wang, J. Zhang, W. Bao, X. Zhu, B. Cao, and P. S. Yu, "Not Just Privacy: Improving Performance of Private Deep Learning in Mobile Cloud," arXiv:1809.03428 [cs, stat], Jan. 2019, [Online]. Available: http://arxiv.org/abs/1809.03428.

[16] A. Harlap et al., "PipeDream: Fast and Efficient Pipeline Parallel DNN Training," arXiv:1806.03377 [cs], Jun. 2018, [Online]. Available: http://arxiv.org/abs/1806.03377.

[17] L. Jiang and X. Lou, "Differentially Private Collaborative Learning for the IoT Edge," International Workshop on Crowd Intelligence for Smart Cities: Technology and Applications (CISC), p. 7, Sep. 2019.

[18] Y. Mao, S. Yi, Q. Li, J. Feng, F. Xu, and S. Zhong, "Learning from Differentially Private Neural Activations with Edge Computing," in 2018 IEEE/ACM Symposium on Edge Computing (SEC), Seattle, WA, USA, Oct. 2018, pp. 90–102, doi: 10.1109/SEC.2018.00014.

[19] T. Zhang, Z. He, and R. B. Lee, "Privacy-preserving Machine Learning through Data Obfuscation," arXiv:1807.01860 [cs], Jul. 2018, [Online]. Available: http://arxiv.org/abs/1807.01860.

[20] A. Triastcyn and B. Faltings, "Generating Artificial Data for Private Deep Learning," arXiv:1803.03148 [cs, stat], Apr. 2019, [Online]. Available: http://arxiv.org/abs/1803.03148.

[21] T. Graepel, K. Lauter, and M. Naehrig, "ML Confidential: Machine Learning on Encrypted Data," in Information Security and Cryptology – ICISC 2012, vol. 7839, T. Kwon, M.-K. Lee, and D. Kwon, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 1–21.