Edge AI for Industry 4.0: An Internet of Things Approach

Athanasios Tziouvaras
attziouv@uth.gr
Electrical and Computer Engineering
University of Thessaly
Volos, Greece

Fotis Foukalas
f.foukalas@ucl.ac.uk
Brain Sciences
University College London
London, United Kingdom

ABSTRACT

In this paper, we study the edge artificial intelligence (AI) techniques for industry 4.0. More specifically, we assume fog computing takes place on the edge of Industrial Internet of Things (IIoT) networks. We provide details about the three main edge AI techniques that can contribute to the future industrial applications. In particular, we deal with the active learning (AL), transfer learning (TL) and federated learning (FL), where AL is used to deal with the problem of unlabeled data, the TL is used to start training with a pre-trained model and the FL is a distributed solution to provide privacy. Finally, their combination is developed too that we name it federated active transfer learning (FATL). Simulation results are carried out that reveal the gain of each solution and their FATL combination. The deployment of FATL in IIoT networking standards such as IEEE P2805 is described too that can be extended as our future work.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning algorithms**; Simulation evaluation; *Cooperation and coordination*; *Distributed algorithms*; • **Networks** → **Mobile networks**; • **Applied computing** → **Industry and manufacturing**.

KEYWORDS

Internet of things, Industrial applications, Edge AI, Industry 4.0

ACM Reference Format:
Athanasios Tziouvaras and Fotis Foukalas. 2020. Edge AI for Industry 4.0: An Internet of Things Approach. In *Proceedings of PCI 2020: Panhellenic Conference on Informatics (PCI '20).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/1122445.1122456

PCI '20, Greece,

ACM ISBN 978-1-4503-8897-9. . . $15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Industry 4.0 will change the way that manufacturing facilities will operate in the future. The solution is mainly provided by the deployment of a large amount of sensor and actuator devices forming an Industrial Internet of Things (IIoT) network. Such an industrial network will be able to collect data from all over the shop floor that can be aggregated to the edge of the network [1]. Obviously, edge computing is part of the industry 4.0 vision, where the edge will provide important computing solutions such as AI, security and privacy, data management and aggregation. Edge computing could be also considered fog computing that is closer to the devices comparing to the cloud computing [2]. Having such a progress in the industrial digital technologies, the edge AI for IIoT networks is considered an open challenge towards Industry 4.0 [3].

This work is dedicated to the edge AI for industry 4.0, where the data are collected and aggregated to the edge from an IIoT network. More specifically, we assume a fog node at the edge that is able to collect and aggregate the data from the IIoT network as in [4], which next are used to deploy different edge AI solutions. Edge AI solutions are considered deep neural networks (DNNs) deployment with particular strategies so that mitigating the problems coming from the large amount of devices. To this end, we consider active learning (AL) to address the problem of unlabeled data, transfer learning (TL) to provide a pre-trained model and federated learning (FL) build a global model with privacy provision. First, all three edge AI solutions are analyzed, simulated and compared. Next, their combination is considered that we call federated active transfer learning (FATL). Such a comprehensive solution could manage all issues related to the application of edge AI to the IIoT networks. The obtained simulation results reveal the gain of using the FATL into the future manufacturing facilities. A detailed discussion of how the FATL solution could be deployed in industrial automation standards such as IEEE P2805 is also provided.

To our knowledge, there is no such a study about the edge AI to the industry 4.0 that considers the AL, TL, FL and their combination. For example, authors in [5] deal with deployment of the AI into the IIoT networks in terms of latency, power consumption and reliability. However, they have not studied any edge AI solution to the edge using the collected data. In [6], the authors provide a cloud-assisted framework to deploy AI to smart factories; however, they

have not also deployed any edge AI solution. In [7], the authors deal with the TL deployment using edge computing and provide results too. However, they don't provide a complete edge AI paradigm that can tackle with unlabeled data and their privacy. In [8], the authors provide a comprehensive survey in the industrial edge computing and the embedded intelligence that could be provided including an overview on the IEEE standards. However, they don't provide any deployment example of edge AI as embedded industrial intelligence solution. In [9], the authors provide a low latency edge AI framework using DNNs, without though dealing with all edge AI solutions and their combination. Another framework for edge AI is provided in [10] without providing detailed solution though. Therefore, a combined solution that deals with the open challenges on edge AI deployment such as unlabeled data, pre-trained models and privacy has not been studied and proposed yet for Industry 4.

The rest of this paper, it is organized as follows. Sec.2 provides an overview and details of all three edge AI solutions considered in our work such as AL, TL and FL. Sec.3 describes the deployment of all three solutions and their combination and Sec.4 provides comparative simulation results. Sec.5 concludes this work.

2 EDGE AI FOR INDUSTRY 4.0: AN IOT APPROACH

We now present the three different edge AI techniques that are most promising to deploy in IIoT networks namely federated learning (FL), active learning (AL) and transfer learning (TL) as follows:

- **Transfer learning** (TL) can be used to Industry 4.0 in order to provide lightweight intelligent manufacturing application given the use of pre-trained models that have been used already [11][12]. TL is a sort of meta learning growing models to become more commoditized, integrated and automated. The ability to use existing frameworks and models can be extended to new challenges and questions and in many cases these agents and models will train themselves and create efficiencies not even imagined in the recent past. These developments will help accelerate the potential of ML and AI moving forward and increase the movement away from code bases and programmers to deep neural networks (DNNs) and other frameworks that write their own code and manage their own behaviors. In our case, the fog nodes load pre-trained network from the cloud, and then customizes its predictive model by replacing the last layers and train with the target domain data. IIoT devices offload to the appropriate fog node after assessing the service accuracy following maximization offloading with latency constraint.
- **Federated learning** (FL) is part of the decentralized AI provision for future smart factories [13]. This vision is able to provide local training to IIoT devices so that not sending the data to the cloud in time critical situations. The intermediate fog-computing node will be able to deploy federated learning, which is the decentralized ML solution. Instead of uploading data to cloud for centralized training, the edge devices process their data locally and share model updates with the cloud server, which informs edge IoT regarding the features. The federated averaging algorithm is used on the server to combine client updates and produce a new global model. Federated cloud server learns from the data locally and the parameters of the model are sent back to the decentralized center. The server will get multi realization of this model, which have been trained to multi data sets and create a consensus out of it. This consensus is sent back to the local data sets, i.e. clients, where the clients send a feedback to the server as well. This process continues iteratively until the convergence is achieved with the required performance. A principal advantage of this approach is the decoupling of model training from the need for direct access to the raw training data.
- **Active Learning** (AL) is a machine learning (ML) approach that can achieve greater accuracy with fewer training labels if it is allowed to choose the data from which it learns [14]. There are cases in which unlabeled data is estimative and precise results are hard to calculate manually. In such a scenario, learning algorithms can actively query the user/teacher for labels. Since the learner chooses the examples, the number of examples to learn a concept can often be much lower than the number required in normal supervised learning. AL attempts to overcome the labeling bottleneck by asking queries in the form of unlabeled instances to be labeled by a human annotator (e.g.,operator or planner). For example, in production the work times are measured and validated periodically and due to this reason $15-20\%$ of work time is not considered by production planner. If the planned production time is higher than that spent by the operator it is not reported. In this case, AL will help to discover and update the work times for such operations automatically, which increase the production capacity.

Although all different approaches can benefit from the design of edge AI, it is possible to combine them to make an integrated end-to-end AI solution. For example, FL can be used in order to not transfer the complete dataset to the network for latency and privacy reasons. AL can be used so that a lightweight training to take place over the IoT devices taking care of their limited resources. TL can introduce a pre-trained model to the edge, coming from the cloud, that can reduce the pre-processing training time for the given domain. The combined federated, active, transfer learning (FATL) is depicted in Fig.1. The different type of edge AI approaches attempt to encompass the local task schedules bounded to edge nodes and provide a global schedule based shop floor requirements (e.g. Manufacturing Execution System). For example, AL is used to better control data management based on logging data about access patterns. Increase of

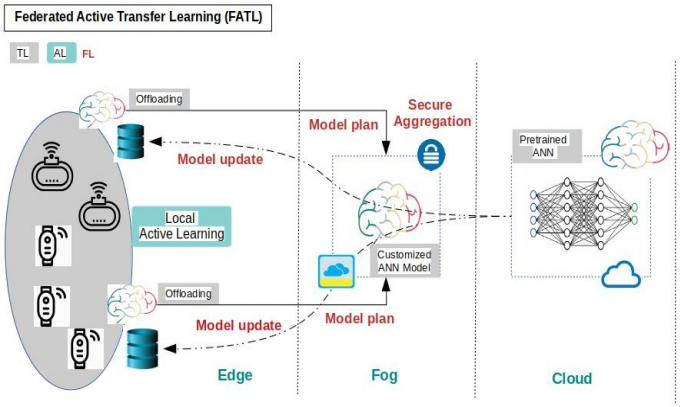

Figure 1: Federated Active Transfer Learning (FATL) as Edge AI solution to Industry 4.0.

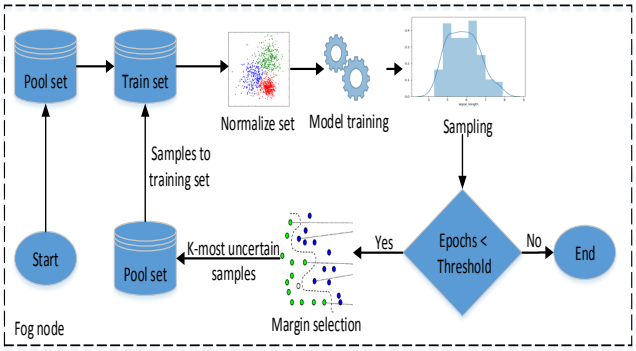

Figure 2: AL (pool-based) technique.

learning speed and reduction of labeled data during initial iterations of learning can be achieved with the help of AL too. Real human decisions may not be the most efficient, however they are decent for the model initialization. FL can improve the learning process, as it helps to obtain the learning meta-information applicable for sharing, in case of multiple scheduling agents are running in parallel. TL technique to transfer knowledge learned in one dataset and applies it to another dataset. AI is able to learn from past work orders and apply this knowledge to predict timings of scheduled work orders in production planning. This can be applied at different levels of production, for example, one product is produced collaboratively in several factories. This is a deployment example, where more details are given in the section below along with simulation results.

3 EDGE AI DEPLOYMENT

Fig.2 depicts the methodology employed for the implementation of AL. We opt for a pool-based sampling AL approach due to its wide applicability in edge computing applications as discussed in [14]. In pool-based sampling, the samples are selected from a pool of unlabeled data when some conditions are satisfied, as discussed below. In order to run simulations, we utilize the MNIST dataset [15], which contains a train set of 60K images of handwritten digits of 10 classes labeled '0' to '9' correspondingly and a test set of 10K images. For training and testing our approach we deploy Google's inception V3 [16] pre-trained deep neural network. We split the train set into a pool of unlabeled samples and a test set that contains labeled data. We arbitrarily select a number of samples from the pool set to form the initial train set while the rest of the samples form the validation set. In the sequel we normalize the newly formed sets and utilize the validation set to get the sample probabilities and the test set to acquire performance measurements. In order to re-select another amount of k samples for the pool set, we employ a margin sampling selection method. Margin selection selects the k-samples with the lowest difference between the two highest class probabilities. We incorporate such samples to the test set and thus, expanding it by k-samples. Afterwards, we proceed in executing training and validation steps. We repeat this process until the classifier converges and the test set is efficiently classified in 10 classes that represent the digits from '0' to '9'.

Fig.3 depicts the methodology we follow for the implementation of TL. TL methodologies are divided into three categories according to the domain and task characteristics of each case as discussed in [11]. In this sense, inductive TL, unsupervised TL and transductive TL can be employed to implement a TL model. In our case we use the same data set we use for AL and thus, we employ an inductive TL technique which is suitable for transferring knowledge between different tasks within similar domains. We define two similar domains, one containing the numerical digits from 0 to 4 and another containing the numerical digits from 5 to 9 while we specify the tasks as the classification process of each image to its corresponding class. Our model also utilizes a learning setting similar to multi-task learning as stated in [11] as the source task contains labeled data and thus, no self-learning procedure takes place. Despite the existing similarities with multi-task learning, our model does not try to learn the target and source task simultaneously; instead it transfers acquired knowledge from the source task to the target task sequentially.

We employ the same data set we employed for AL and we split the training set into two different test sets, one that contains digits from 0 to 4 and another that contains digits from 5 to 9. We train the model using the first set and we obtain the appropriate performance measurements. We opt for the Google's inception V3 pre-trained deep neural network for the training process. In the sequel we fine tune the deployed neural network in order to enable transferring knowledge of parameters as discussed in [12]. For transferring knowledge of parameters to work, the related tasks should share some parameters or distribution of hyperparameters.

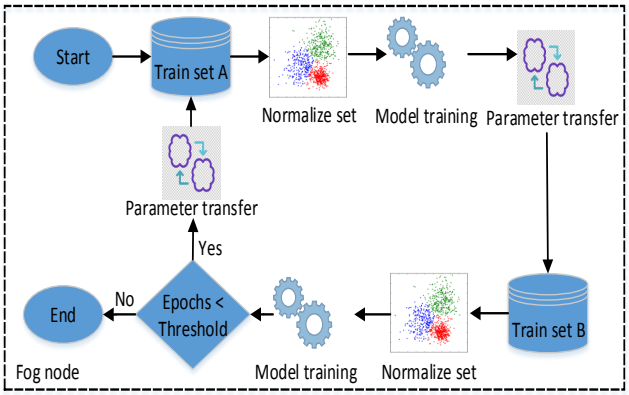

Figure 3: TL technique.

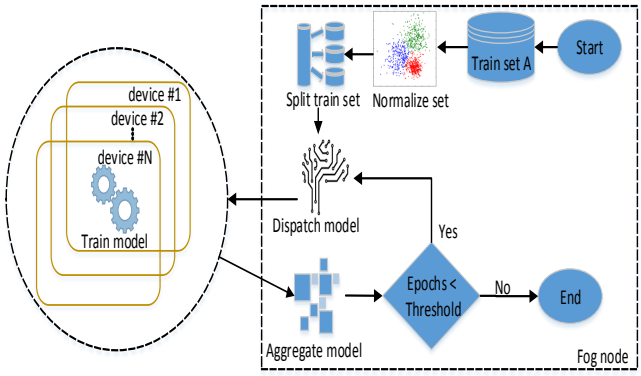

Figure 4: FL technique.

In our case input weights of the first DNN layers are tuned in order to enable certain features from the source domain to be utilized to help improve the performance in the target domain while the remaining hyperparameters of the DNN remain unchanged and thus, transferred to the new model. In order to avoid over fitting we also freeze the last-layers of the neural network where specific features are extracted as discussed in [17]. After we complete the fine tuning processing, we proceed to train the new model with the train set that belongs to the second domain. Finally we utilize the test set that contains all 10 classes from every domain to get the accuracy and loss of our model.

In order to implement the FL, we use the MNIST data set as described on previous AL and TL implementations along with the inception V3 DNN. Fig.4 depicts the process under which the federated learning methodology is deployed. We adopt a horizontal federated learning model as discussed in [18] as devices share the same feature space, but they differ in sample space. We use random selection to split the initial train set into a number of smaller train sets equal to the amount of participating devices. Afterwards, the server distributes the initial model and the train sets to the connected devices. Then the model is trained on each device locally using the train set that corresponds to the device ID. During this process devices do not communicate with each other and thus, they do not share any data or any model hyperparameters. After a federated round passes, the devices send the locally trained models to the server, where they are aggregated into a global model. We define a federated learning round as the minimum amount of time that is required for the clients to locally train their models and to dispatch the results to the main server as stated in [19]. The server aggregation process fine-tunes the DNN by changing its hyperparameters according to the validation data acquired by the clients. Then the server distributes the model back to the clients which proceed to the next federated round of training. This process is repeated until the global model converges to a predefined threshold.

For simulation purposes we adopt the Tensor flow framework in which we can instantiate a numbers of devices and a main server capable of aggregating the user data.

Now,we would like to deploy the active transfer learning while it has been trained using a federated environment according to FL framework above (Fig.4). Thus, our model will follow a Federated Active Transfer learning (FATL) paradigm as presented in Fig.1. However, given the actual implementation details of all different edge AI techniques described above, we identify the following deployment steps:

- Distributing the global model that is located at the edge to the participating users.
- Splitting the database of each user into two smaller databases. Each user will conduct inductive transfer learning between those two databases.
- The database of each user also contains unlabeled data and thus, a proper AL sampling selection mechanism is selected in order to include such data. In our case we employ a margin sampling technique.
- Each user trains its local model using the aforementioned active transfer learning technique.
- After a federated round passes all users upload their local models on the fog node.
- The fog node performs an aggregation of the collected local user models and produces a new global model.
- The fog node distributes the new model to the users.
- The users perform active transfer learning on their new local model.
- This process is repeated until the global model converges.

We would like now to present the deployment of FATL using a specific Industry 4.0 standard. In [8], the authors mention the IEEE P2805 that is being developed for industrial edge computing deployment. There are three different types of standars, which deal with self-management protocols, data acquisition, filtering and buffering protocols as well as cloud-edge collaboration protocols. The main element of the IEEE

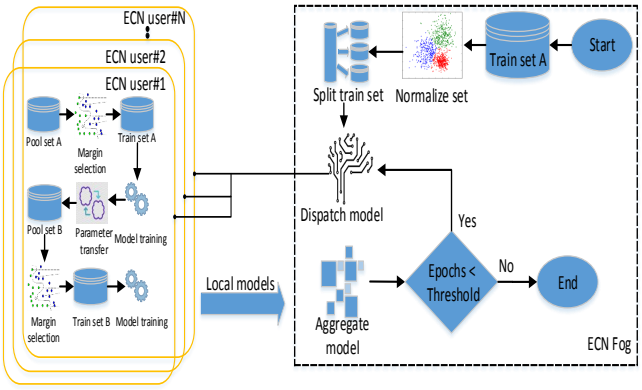

Figure 5: FATL edge AI paradigm.

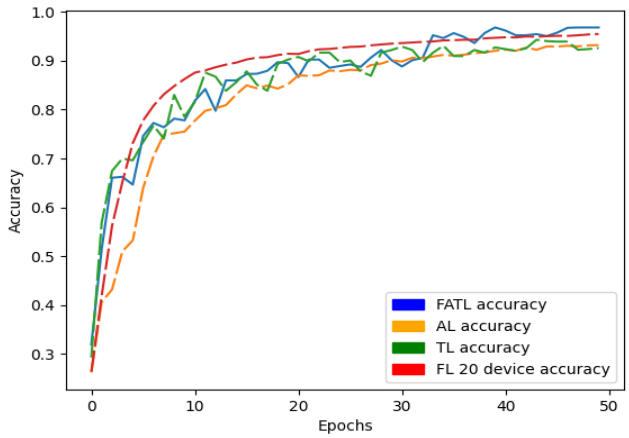

Figure 6: Training accuracy over epochs per learning technique and the proposed FATL solution.

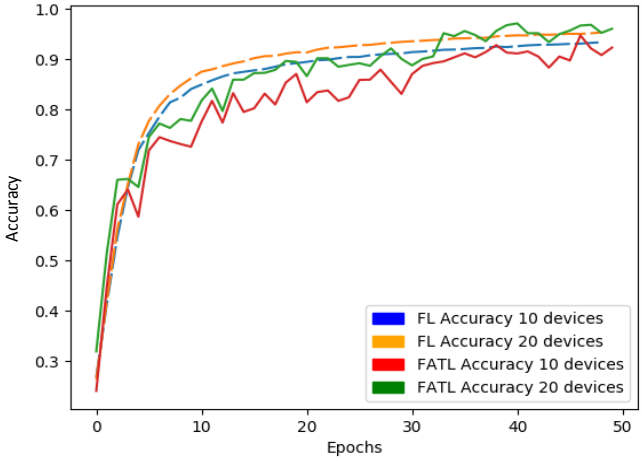

Figure 7: Training accuracy over epochs for different amount of devices.

P2805 is the edge computing node (ECN) that is distribtued from the IoT devices, to fog and finally to the cloud. Since the FATL is deployed on the fog to the IoT part of the architecture and also FATL is about the edge AI and not the actual protocol transmision information, we provide Fig. 5 below that depicts the FATL deployed using potentially the IEEE P2805. In this scenario, the ECNs are available computing resources that contribute to the deployed ML algorithm. Specifically the ECN fog is responsible for the model aggregation procedure while the ECN users conduct the de-centralized training process. Details on the protocols are considered another work that might be our future work, although we don't have full information about the standard yet.

4 SIMULATION RESULTS

To deploy and test our models, we utilize Google's Inception V3 DNN. Our train size is composed of 60K images while the test size contains 10K images. We set a batch size of 50 samples and thus, each training epoch requires 1200 training steps. We run the training process for 50 epochs and present the training accuracy over the training epochs we obtain below. Training accuracy is the amount of correct image classification versus the amount of classifications conducted on the training set during the training steps of the model.

Fig. 6 depicts the results of AL, TL, FL and their combination known as FATL. We use dash lines to depict the AL,TL and FL accuracy and continuous line to highlight the FATL accuracy. We observe that TL converges faster than AL but AL tends to achieve slightly better accuracy under given enough training epochs. Specifically, AL looks to converge at 92% while TL at 90%. FL on the other hand looks to converge faster than AL and TL while also achieving higher accuracy (almost 94%) over epochs when compared to AL or TL. It is also clear that the number of participating users in FL plays a major role in the model's accuracy as results obtained with 20 users display higher accuracy when compared to the results obtained with 10 users. The proposed FATL reveals the slowest convergence speed when compared to the other three AI techniques. However, it tends to achieve the highest accuracy after undergoing training for some epochs. Therefore, FATL manages to achieve an accuracy rate of 95%.

In order to compare the convergence rate of FL and FATL models in conjunction with the amount of participating IoT devices, we also run the experiments with 10 and 20 devices separately. Fig.7 depicts the results we obtain over 50 training epochs. Further analysis of the results indicates that both FL and FATL models converge faster when a larger amount of devices are involved in the training process. As

Table 1: Test accuracy of the ML models.

Model	Test accuracy of the ML models
AL	93%
TL	92%
FL	94%
FATL	97.8%

such training with 20 devices display better convergence rates when compared to training with 10 devices. The achievable training accuracy is also correlated with the amount of devices. Results demonstrate a small (2-3%) but clear accuracy improvement with 20 devices in contrast with the accuracy obtained with 10 devices.

The observations made above are verified when testing our models on the test datasets. Table 1 depicts the training accuracy of each model on MNIST test dataset. We measure testing accuracy the same way we measure training accuracy but instead of using the training set, we use the model's weights to classify the images of the test set. Testing accuracy shows that the FATL model performs better in terms of accuracy over the other AI paradigms achieving 97.8% while TL achieves the lowest performance with 92% accuracy.

5 CONCLUSION

In this work, we deal with the deployment of distributed intelligence into the industrial domain as a solution towards edge AI for industry 4.0. The considered system is formulated by IIoT devices and a fog computing node. Both the fog node and the IIoT devices can run AI locally or in a distributed manner depending on the deployed edge AI solution. Thus, we discuss, develop and compare three popular edge AI solutions such as active, transfer and federated learning. Finally, their combination is proposed namely FATL and studied that shows a gain over the individual solutions. The biggest benefit is the fact that our proposed solution can deal with all major challenges appeared in case of deploying edge AI. Future work could be the deployment of such a distributed solution to specific IEEE standards, where we also give an example assuming the IEEE P2805 standard that is now under development.

REFERENCES

[1] P. Patel, M. I. Ali and A. Sheth, From Raw Data to Smart Manufacturing: AI and Semantic Web of Things for Industry 4.0, IEEE Intelligent Systems, vol. 33, no. 4, pp. 79-86, Jul/Aug. 2018.

[2] M. Aazam, S. Zeadally and K.A. Harras, Deploying Fog Computing in Industrial Internet of Things and Industry 4.0, IEEE Trans. on Ind. Informatics, vol. 14, no. 10, pp. 4674-4782, Oct. 2018.

[3] Y-L. Lee, P-K. Tsung and M. Wu, Techology Trend of Edge AI, IEEE Int. Symp. On VLSI, 2018

[4] 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, Proceedings of the IEEE, vol. 107, no. 8, pp. 1738-1762, Aug. 2019.

[5] A. H. Sodhro, S. Pirbhulal and V. H. C. de Albuquerque, Artificial Intelligence-Driven Mechanism for Edge Computing-Based Industrial Applications, IEEE Trans. On Industrial Informatics, vol. 15, no. 7, pp. 4216-4224, pp. 4235-4244, Jul. 2019.

[6] J. Wan, J. Yang, Z. Wang and Q. Hua, Artificial Intelligence for Cloud-Assisted Smart Factory, IEEE Access, Oct. 2018.

[7] W. Sun, J. Liu and Y. Yue,. AI-Enhanced Offloading in Edge Computing: When Machine Learning Meets Industrial IoT, IEEE Network, vol. 33, no. 5, pp. 68-74, Sep/Oct. 2019.

[8] W. Dai, H. Nishi, V. Vyatkin, V. Huang, Y. Shi and X. Guan, Industrial Edge Computing, IEEE Industrial Electronics Magazine, Dec. 2019.

[9] E. Li, L. Zeng, Z. Zhou and X. Chen, Edge AI: On-Demand Accelerating Deep Neural Network Inference via Edge Computing, IEEE Trans. On Wirel. Commun., vol. 19, no. 1, Jan. 2020.

[10] X. Zhang, Y. Wang, S. Lu, L. Liu, L. Xu and W. Shi, OpenEI: An Open Framework for Edge Intelligence, https://arxiv.org/pdf/1906.01864.pdf.

[11] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," in IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345-1359, Oct. 2010

[12] L. Shao, F. Zhu and X. Li, "Transfer Learning for Visual Categorization: A Survey," in IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 5, pp. 1019-1034, May 2015.

[13] M. Hao, H. Li, X. Lao, G. Xu, H. Yang and S. Liu, Efficient and Privacy-enhanced Federated Learning for Industrial Artificial Intelligence, IEEE Trans. On Industr. Informatics, Oct. 2019.

[14] Qian, S. Sengupta and L. K. Hansen, Active Learning Solution on Distributed Edge Computing, https://arxiv.org/abs/1906.10718, Jun. 2019.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, 86(11):2278-2324, November 1998.

[16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 2818-2826.

[17] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. "How transferable are features in deep neural networks?" In Proceedings of the 27th NIPS'14 - Volume 2 . MIT Press, Cambridge, MA, USA, 3320–3328, 2014.

[18] Q.Yang, Y. Liu, T. Chen, and Y. Tong. 2019. Federated Machine Learning: Concept and Applications. ACM Trans. Intell. Syst. Technol. 10, 2, Article 12 (January 2019), 19 pages.

[19] J. Goetz, K. Malik, D. Bui, S. Moon, H. Liu and A. Kumar. "Active Federated Learning" ArXiv abs/1909.12641, 2019.