Edge AI for Industrial IoT Applications

Abstract

In this paper, we study the edge artificial intelligence (AI) for industrial internet of things (IIoT) applications. We discuss about the edge AI technology that is considered the combination of AI with edge computing and provide an overview of edge AI applications for IIoT networks, where the following three challenges are important to address: a) personalization, b) responsiveness and c) privacy preserving. To this end, we propose a federated active tranfer learning (FATL) model, which through training and testing is able to address those open challenges. Details about the training and testing of the proposed FATL global model are given including the corresponding simulation setup. This work concludes with a discussion and comparison of the obtained simulation results with existing edge AI training solutions, which provide useful insights about the proposed FATL model. The simulation results highlight how the FATL global model can address efficiently the open challenges of edge AI for future IIoT applications.

Keywords

Edge Artificial Intelligence, Industrial Internet of Things, Federated Active Transfer Learning.

I. Introduction

Industry 4.0 will change the way that manufacturing facilities will operate in the future. The Industry 4.0 concept aims to the deployment of a large amount of sensor and actuator devices forming an Industrial Internet of Things (IIoT) network. Such an IIoT application will be able to collect data from all over the shop floor that can be aggregated to the edge of the network [1]. Computing to the edge is required given the huge amount of data produced by the IIoT devices. In this sense, edge computing will play an imporant role, where the edge should provide computing resources for edge intelligence with security and privacy, data management and aggregation provision. Edge computing can be realized through fog computing that is closer to the end devices comparing to the cloud computing [2]. Edge intelligence, i.e. artificial intelligence (AI) application with edge computing (edge AI) for training/testing or inference, is an imporant element for IIoT applications in order to build a model that can learn from the high amount of aggregated data [3].

This work aims to provide an edge AI training/testing solution for IIoT applications. More specifically, we assume fog computing to the edge that is able to collect and aggregate the data from the IIoT network, which are used to deploy different edge AI solutions. Edge AI solutions rely on deep neural networks (DNNs) with particular strategies so that mitigating the problems coming from the large amount of devices. However, the main focus of our solution is to address the main open challenges of edge AI for IIoT applications such as a) personalization, i.e. customized AI models at the edge tailored to individual device requirements, b) responsiveness, i.e. adaptive computing services to the new situations and c) privacy preserving for the data transmission to the remote cloud for privacy issues [4]. Towards this end, we propose a federated active transfer (FATL) model for training/testing, which provides the following advantages. In particular, the transfer learning (TL) can provide customized AI models developed at the edge servers, which are tailored to the new industrial system behaviors and requirements to deliver accurate results. The active learning (AL) can deal with the time-varying and unpredictable changes on the industrial data over the time, where the computing provided the necessary adaption in terms of training to the new collected data, i.e. situation. Finally, the federated learning (FL) provides the necessary privacy at the edge server for the processing information, which may not be willing to transmit to the remote cloud for privacy issues. This paper below provides details about training/testing of the proposed FATL global model including simulation results and some potential deployment considerations taking into account a new type of automation standard [5]. Simulation results highlight the strength of such an edge intelligence for training global model in case of IIoT applications.

To the best of our knowledge, there is no such a training/testing solution that has focused on the edge AI for IIoT applications addressing all major open challenges menioned above. For example, in [6] authors deals 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 for the specific industrial application. In [7], the authors provide a cloud-assisted framework to deploy AI to smart factories; however, they have not also deployed any edge AI solution. In [4], the authors proposed the use of TL at the edge. However, they don't address all three challenges of edge AI for IIoT applications. In [5], the authors provide a comprehensive survey in the industrial edge computing and the embedded intelligence provided by new IEEE standards. However, they don't provide any deployment example of edge AI as embedded industrial intelligence solution. In [8], the authors provide a low latency edge AI framework using DNNs, which is for edge intelligence inference and not training/testing. Another framework for edge AI is provided in [9] without providing a detailed solution though. Therefore, an edge intelligence training solution that deals with the three major challenges for IIoT applications such as unlabeled data, pre-trained models and privacy have not been found in the literature yet and this is the focus of this work.

The rest of this paper is organized as follows. Sec.II provides an overview of the edge AI for IIoT and the open challenges of such an application. Sec.III provides the proposed edge AI solution through FATL deployment. Sec.IV provides the simulation setup and results with a discusion to compare the proposed solution with similar approaches. Sec.V concludes this work.

II. Edge AI for Industrial Internet of Things

A. *Edge AI: AI with edge computing*

’Last Mile’ of AI with edge computing is considered the automated deployment of intelligent services and applications on edge nodes and devices. A massive number of edge devices (small, decentralized and low-power devices) have the capability to perform AI locally or in collaboration with other devices in a wide range of applications such as groups of IoT devices. The intelligence to such applications will be provided to the edge in order to deal with ultra reliable and low latency requirements of data transmission through networks (wireless or wired). Additionally, computational requirements are necessary in case of low power edge devices. Therefore, there is an increasing opportunity to deploy AI at the edge known as ’Edge AI’ (or edge intelligence). Edge AI is divided into edge intelligence model training and inference. Different types of enabling techniques have been proposed for traning and inference in order to tackle with the design requirements in the IoT domain [3]. Design requirements should address the high cost in terms of time, network bandwidth, storage capacity, loss of independence, security and privacy caused by centralized decision making [1].

In the literature, we can identify different types of edge AI solutions that can be found either to the ”cloud-to-edge” or ”edge-to-end-devices” and address different types of challenges. In [9], authors propose an edge-to-end-devices AI solution to address the requirements of different applications by choosing a matched algorithm for a specific edge configuration. In [10], the authors propose an energy-delay optimization framework for running AI on the cloud-edge while in [11] authors provide a edge-to-device task scheduling solution to reduce data size and operation count at the edge. In [12], the authors propose a deep learning application for data reduction in the edge-cloud of the IoT network infrastructure. In [13], the authors propose an edge-to-end-device solution which relies on a network embedding technique for effective resource discovery in edge environments. As mentioned above, we focus on the edge AI that takes place at the edge-to-end-devices. Thus, our focus is on how to leverage AI to the edge and IoT devices without the involvement of cloud in order to address major challenges of edge AI application to IIoT.

B. Edge AI for Industrial IoT

It is obvious from the above that edge AI aims to combine the potential advantages of edge computing (shorter latency times, reduced bandwidth, improved trustworthiness) with all the common benefits of AI. Security and privacy can also be enabled by deploying edge AI. Trustworthy user-centric AI which offers not only protection (security, privacy) but also transparency and verifiability to the user can be also provided, offering privacy and security by design, with a smaller attack surface. More specifically, the need for edge AI for IIoT is already pointed out with the following main design requirements: a) personalization, i.e. customized AI models at the edge tailored to individual device requirements, b) responsiveness, i.e. adaptive computing services to the new situations and c) privacy preserving for the data transmission to the remote cloud for privacy issues [4]. Existing solutions do not address all these three major challenges for *"edge AI for IIoT applications"* for model training as explained thoroughly below [1].

For example, the authors in [4] focus on the personalization challenge by deploying a TL based solution. The authors in [14] deal with the privacy preserving by focusing on cryptographic methods. The work in [15], which already mentioned above, provide an active federated learning framework. However, the authors do not address the challenge of personalization. On the other hand, we can find a few interesting works on edge AI in the literature, which consider model inference rather than model training or a non industrial application. For example, the authors in [16] present an interesting cooperative edge intelligence framework, which relies on model partition technique for model inference. In [8], the authors also provide a model partitioning

[1]Notably, our solution is considered for model training and not inference.

solution for edge inference too. Finally, the authors in [11] provide an edge training model that is able to manage the network traffic requirements in IoT applications. Nevertheless, this solution requires edge AI to the edge server with complete data transmission to the edge server.

Towards this end, we propose a federated active transfer (FATL) model for training/testing, which provides the following advantages. More specific, we consider active learning (AL) as an inexpensive labeling data technique that achieves higher performance with the same number of data samples [15]. AL is a useful tool in case of unlabeled data that can not be transmitted for labeling to centralized cloud repositories due to frequent time-varying and unpredictable changes, where the computing provide the necessary adaption in terms of training to the new collected data, i.e. situation. Moreover, federated learning (FL) technique allows the training and testing in a distributed fashion without the need for data transmission to the cloud for training purpose and thus, supporting the required privacy [3] [14]. The combination of AL with FL has highlighted in [15], where edge computing is combined with the AI efficiently for industrial applications. Another model training solution is considered the transfer learning (TL), which is a useful technique for exploiting pretrained models in order to avoid training models from scratch [3]. TL has been proved useful for IIoT applications, where the pretrained model from the cloud is downloaded to the edge [4]. The edge is then responsible to train the model using customized AI models, which are tailored to the new industrial system behaviors and requirements. Obviously, AL, TL and FL can be deployed in a ”cloud-to-edge-device” methodology in order to address challenges such as pretrained models, unlabeled data and privacy. Such a distributed edge AI model training is imporant for industrial applications as pointed out in the overview above. It is evident from the above that there is no any recent work that deals with all main three challenges of edge AI for model training. To this direction, we propose a model training that combines the FL, TL and AL features addressing personalization through AL, i.e. customized AI models at the edge, b) responsivenes through TL, i.e. adaptive computing services to the new situations and c) privacy preserving through FL, i.e. without data transmission to the remote edge.

III. Edge AI through Federated Active Transfer Learning

A. Federated Active Transfer Learning (FATL)

Active Learning (AL) can achieve greater accuracy with fewer training labels since it chooses the data from which it learns [15]. There are cases in which unlabeled data is estimative and precise results are hard to calculate by 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 an human annotator.

Transfer learning (TL) is employed to provide pre-trained models that have been used already [17]. 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. Edge nodes can load pretrained network from the cloud, and then customize its predictive model by replacing the last layers and train with the target domain data. IIoT devices offload to the fog node after assessing the service accuracy following maximization offloading with latency constraint.

Federated learning (FL) is able to provide local training to IIoT devices so that not sending the data to the cloud in time critical situations [14]. Instead of uploading data to cloud for centralized training, the edge devices process their data locally and share model updates with the cloud server. 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 again too. This process continues iteratively till the convergence is achieved with the required performance.

It is obvious from the above that all three AI techniques can address the challenges of edge AI for IIoT applications. More specifically, AL can address the personalization challenge by deploying customized AI models at the edge given the individual device requirements. TL can address the responsiveness challenge by applying adaptive computing to the new environment. Finally, FL can address the privacy challenge for the data transmission. To this end, we propose the **federated, active, transfer learning (FATL)** model in Fig.1. Starting from the left side (i.e. end-devices), Fig.1 depicts the local AL carried out on the IIoT devices in order to increase the learning speed and reduce the labeled data during initial iterations of learning. On the right side of this figure, a pretrained artificial neural network (ANN) is deployed through TL technique, which is able to transfer knowledge learned in one dataset and applies it to another dataset. In the middle, the secure aggregation and model plan are provided through FL, which can improve the learning process. Both TL and FL are deployed on the edge part of the architecture, which helps to obtain the learning meta-information applicable for sharing in case of multiple scheduling IIoT devices running in parallel while AL is deployed on the IIoT end-devices. Further, the proposed FATL model can accommodate several industrial use cases such as: a) big data analysis for device preventive maintenance of a smart factory, b) autonomous manufacturing process such as quality control and optimization for process control, c) automatic fault detection for device predictive maintenance, d) production reconfiguration processes such as product quality tracing and optimization and e) virtualized factory environments [5] [18] [19].

B. Edge AI through FATL

We would like now to implement the FATL model's training process in order to provide edge AI for IIoT applications as depicted in Fig.2. The proposed implementation assumes a number of ECN devices, where each of them is using its own dataset to train the model locally.In order to commence the FATL model training process we initially formulate the required datasets described below:

- **Pool set A** contains unlabeled data for the learning task A.
- **Pool set B** contains unlabeled data for the learning task B.
- **Train set A** contains training data for the learning task A.
- **Train set B** contains training data for the learning task B.

These datasets are created by normalizing the initial dataset and splitting it accordingly into the four groups described above. The training process begins with the FL, where the global learning model is dispatched to the participating ECN devices. The local training process occurs simultaneously on all ECN devices without using device-to-device communication or sharing any model hyperparameters. Each ECN device uses the pool-based sampling AL approach, where samples are selected from a pool of unlabeled data (pool set) and combined with the existing labeled data in order to create the train set. To this end, the model is trained for the learning task A by employing the margin selection method to select k-samples with the lowest difference between the two highest class probabilities from the pool set A. Such samples are incorporated to the train set A and thus, expanding it by k-samples.We find the optimal k value by running multiple simulations of the AL technique and selecting the k value, which results in the best accuracy.

The next step of the training process requires the model to be tuned in order to enable the transferring knowledge of parameters according to the TL paradigm [17]. For transferring knowledge of parameters to work, the related tasks should share some parameters or distribution of hyperparameters. In this case, input weights of the first DNN layers are tuned in order to enable certain features from the source domain to be utilized to improve the performance on the target domain while the remaining hyperpameters of the DNN remain unchanged and thus, transferred to the new model. After the model tuning completes, the training continues with the train set that belongs to the second domain ,i.e. the train set B. For this purpose the margin selection technique is invoked; this time targeting the train set B similarly to the process described above. TL methodologies are divided into three categories according to the domain and task characteristics of each case as discussed in [20]. As such inductive TL, unsupervised TL and transductive TL can be employed to implement a TL technique. Our framework uses the same data domain for all samples and thus, an inductive technique is employed which is suitable for transferring knowledge between different tasks within similar domains. Our model also utilizes a learning setting similar to multi-task learning as stated in [20] 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 learn the target and source task simultaneously; instead it transfers acquired knowledge from the source task to the target task sequentially. After the local training completes the local models are dispatched from the ECN devices to the Fog where the FL aggregation takes place. The aggregation process fine-tunes the DNN by changing its hyperparameters according to the validation data acquired by the clients. Then, the fog re-distributes the model back to the ECN devices which proceed to the next federated round of training. 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 fog as stated in [21]. This process is repeated until the global model converges to a predefined threshold.

We also assume that the FATL model could be deployed using industrial edge computing standards. For example, the IEEE P2805 could be considered, which specify the self-management protocols, data acquisition, filtering and buffering protocols as well as cloud-edge collaboration protocols [5]. The main entity of the IEEE P2805 is the edge computing node (ECN) that is distributed among the end-devices, edge and the cloud. Assuming ECN entities, the IEEE P2805.1 specifies the self-management protocols, the IEEE P2805.2 specifies the data acquisition, filtering and buffering protocols and the IEEE P2805.3 specifies the cloud-edge collaboration protocols [5]. Thus, Fig. 2 depicts also the FATL model by considering those ECN entities, which are available computing resources that contribute to the deployed edge AI model. Specifically, the FATL model is extended and deployed from the ”ECN fog” to the ”ECN end-devices”, where each of those entities implement the distributed FATL model. More specific, the IEEE P2805.1 includes identification and data sharing functionalities, which are employed by the ECN devices, and the ECN fog before the model transmission procedure takes place. The IEEE P2805.2 includes data acquisition, filtering and buffering protocols, which are used by the ECN fog to receive the local models from the ECN devices. Our future work could be a detailed design, development and performance evaluation of the FATL model using the IEEE P2805 protocols.

IV. Simulation Setup and Results

A. Simulation Setup

To benchmark the proposed FATL model, we utilize the MNIST dataset [22] and Googles Inception V3 DNN [23]. We use the MNIST dataset since it is useful in order to evaluate the performance of computational intelligence solutions like the FATL edge intelligence model as also considered in [14] [15]. MNIST dataset 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. 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. In order to avoid over fitting we also freeze the last-layers of the Inception V3 neural network, where specific features are extracted as discussed in [24].

For the FATL model's simulation we adopt the following procedure. For the AL part, we mark some of the training images as unlabeled and thus, we group them into the pool sets A and B. We run our simulations with different amounts of unlabeled samples in order to obtain detailed information on the accuracy of our technique. For the TL part we define two similar domains, one containing the numerical digits from 0 to 4 (set A) and another containing the numerical digits from 5 to 9 (set B) while we specify the corresponding tasks as the classification of each image to the class it belongs. Regarding the FL, we adopt a horizontal FL model as discussed in [25] as devices share the same feature space but, they differ in sample space. We also use random selection to split the initial train set into a number of smaller train sets equal to the amount of participating devices. Finally, we use the PyTorch and Tensor flow frameworks, where we instantiate a number of devices and one server capable of aggregating the user data into a global model. Notably, for the FL, AL and TL model simulation, we adopt a similar procedure with FATL as we isolate the corresponding learning technique from the FATL simulation methodology. Under this premise, we use unlabeled and labeled image samples for the AL simulation, we change the target classification task for the TL and we deploy a number of devices to train the model in a distributed way for the FL simulation.

B. Simulation Results and Discussion

Fig. 3 depicts the results of FATL model compared with the AL, TL, FL individually. We use dash lines to depict the AL,TL and FL accuracy and continuous line to highlight the FATL accuracy. We observe that TL achieves better training accuracy through the training process when compared to AL, but AL marginally outperforms TL after 50 training epochs. Specifically, AL achieves 92% while TL at 90% training accuracy. On the other hand, FL converges faster than AL and TL while also achieving higher accuracy (almost 94%) over epochs when compared to AL or TL. It is also evident that the number of end devices in FL plays a major role in the models accuracy, i.e. results obtained for 20 end devices demonstrate higher accuracy compared with the results obtained with 10 end devices. 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 97.8%.

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 end devices separately. Fig.4 depicts the obtained results over 50 training epochs. Further analysis of the results indicates that both FL and FATL models converge faster when a large amount of devices are involved in the training process. Thus, training with 20 end devices display better convergence rates when compared to training with 10 end devices. The achievable training accuracy is also correlated with the amount of devices. Results demonstrate a small (1%) 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 I depicts the accuracy of each model on MNIST test dataset. We measure the test accuracy by using the corresponding trained model weights to classify the images from the MNIST test set. The test accuracy value reflects on the amount of correct classification outcomes as percentage of the total size of the test set.The results we obtain depict that the FATL model performs better compared to the other AI paradigms achieving 97.8% while TL achieves the lowest performance with 92% accuracy.

Moreover, we would like to compare our FATL model with previous related works in order to provide a clear discussion about the achievable performance of the FATL model. Notably, all the existing works considered below for comparison purposes use dataset similar to MNIST in terms of computational intelligence requirements. We can not find existing works in the literature, which use industrial datasets, except the one in [6], which does not focus on computational intelligence. As mentioned above, MNIST is a well known and approved dataset to evaluate the performance of computational intelligence solutions. To this end, Fig. 5 provides results of the FATL compared with the solution proposed in [15]. In [15], authors utilizes the MNIST dataset to obtain their results but they use a different sample selection k value. To this end, we adapt the k hyperparameter of our model to match the one selected by the aforementioned work. The new k value is set to 20 and determines the amount of unlabeled samples that are selected on each iteration to be integrated into the labeled data set as the AL training methodology dictates. We train the FATL model using 4 IoT devices and we illustrate the results we obtain in fig. 5. Each device utilizes the aggregated global model in order to test its accuracy. As such the same global model is used for all 4 devices but with each device using different test set. Results indicate that FATL solution achieves higher accuracy on every IoT device when compared to the technique proposed in [15]. This behavior highlights the efficiency of the margin selection methodology employed for AL technique.

Fig.6 provides the comparison of FATL model with previous work in [12] within the context of input data reduction. In [12]. authors employ an encoding methodology that reduces the amount of labeled data of the training set and thus, reducing the number of features. As a result the training set is reduced in size and the model may achieve high accuracy with lower amount of input data. Similarly, our AL technique utilizes labeled and unlabeled samples in order to build the training set. We configure the AL to utilize different portions of labeled and unlabeled data and we train our model with the resulting training sets. In this sense, we also reduce the amount of labeled data of the model input and we present the accuracy per data reduction we obtain in fig.6. We observe that with 54% and 77% data reduction FATL achieves better accuracy. This result is depicted due to the FL part that makes up for the lost accuracy of the AL method and thus, the FATL model achieves an overall higher accuracy even with large rates of data set reduction. On the other hand, when the training set is reduced by 87%, the AL achieves low accuracy too and the FL cannot compensate for such an accuracy loss resulting in lower overall model accuracy when compared with previous work in [12].

Fig. 7 provides the comparison of FATL model with the solution proposed in [26] in terms of the amount of time required for the training process to complete. In order to provide this comparison, we tuned the size of our dataset by either removing images or by adding additional rotated and scaled images in order to increase the training set size up to $1TB$. We depict the processing time of system tasks, i.e. the training process for different input data sizes in Fig.7. Results indicate that FATL requires lower training time as the knowledge transferring procedure of the TL saves the model from the re-training process of the newly acquired knowledge on each training iteration. Notably, the required FATL training time is higher than the TL, FL and AL training time separately, where among those three techniques, the TL requires the highest and the AL the lowest training time as confirmed from individual simulations.

Fig. 8 provides the comparison of FATL with the solution proposed in [4] within the context of accuracy over training set size. We evaluate the performance of the FATL model using the ILSVRC [25] dataset as considered in [4]. We also set the hyperparameters of our FATL model to match those used by the model to compare. We observe that training our model with low data sizes results in lower testing accuracy compared to (Fig.7, [4]). This is due to the fact that the FL technique requires a relatively high amount of samples to efficiently train the model as the dataset is divided among the participating IoT devices and thus, small size datasets result in significantly lower amounts of training samples per device. On the contrary, when the dataset size grows up to a 5.800 images the IoT devices use adequate samples to train the model. As a result the FATL distributed training process achieves better testing accuracy values with bigger data sets.

V. Conclusion

In this work, we propose an edge AI model for IIoT applications namely federated active transfer learning (FATL). We first provide an overview of the edge AI (intelligence) that is the combination of AI with edge computing, where focus is given on the model training for IIoT applications. Next, the proposed FATL model is presented, which addresses the following three edge AI challenges: a) personalization, b) responsiveness and c) privacy preserving by employing active learning (AL), transfer learning (TL) and federated learning (FL) respectively. In particular, AL personalizes the AI model by changing the amount of labeled samples in respect to the task requirements, TL increases responsiveness due to the ability to quickly adapt the model to new learning tasks and FL provides privacy due to the distributed training process in which the devices do not share any data. Finally, implementation of the FATL model is provided in detail including simulation setup and results. Simulation results are discussed with regards to the performance evaluation compared with the single FL, AL and TL solutions. FATL shows the highest accuracy with a slower convergence though, where the number of devices increase the accuracy as well. Further, we compare our model with recent related works in order to highlight the performance increase while using the FATL model for training/testing at the edge of the network. Simulation results indicate that FATL achieves higher accuracy with a low amount of federated devices and the accuracy remains at highest levels even when the amount of training samples are significantly reduced. The training process of the FATL model also requires significantly less time while compared with other state of the art solutions.

References

[1] M. Wollschlaeger, T. Sauter and J. Jasperneite, The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0, IEEE Industrial Electronics Magazine, vol. 11, no. 1, pp. 17-27, Mar. 2017.

[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] 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.

[4] 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.

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

[6] 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.

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

[8] 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.

[9] X. Zhang, Y. Wang, S. Lu, L. Liu, L. Xu and W. Shi, OpenEI: An Open Framework for Edge Intelligence, https://arxiv.org/abs/1906.01864, Jun. 2019.

[10] W. Zhang, Z. Zhang, S. Zeadally, H-C. Chao and V. C. M. Leung, MASM: A Multiple-Algorithm Service Model for Energy-Delay Optimization in Edge Artificial Intelligence, IEEE Trans. On Industrial Informatics, vol. 15, no. 7, pp. 4216-4224, Jul. 2019.

[11] H. Li, K. Ota and M. Dong, Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing, IEEE Network, vol. 32, no. 1, pp. 96-101, Jan./Feb. 2018.

[12] A. M. Ghosh and K. Grolinger, Deep Learning: Edge-Cloud Data Analytics for IoT, 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), May 2019.

[13] B. Yuan, J. Panneerselvam, L. Liu, N. Antonopoulos and Y. Lu, An Inductive Content-Augmented Network Embedding Model for Edge Artificial Intelligence, IEEE Trans. on Ind. Informatics, vol. 15, no. 7, pp. 4295-4305, Jul. 19.

[14] 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.

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

[16] L. Zeng, E. Li, Z. Zhou and X. Chen, Boomerang: On-Demand Cooperative Deep Neural Network Inference for Edge Intelligence on the Industrial Internet of Things, IEEE Network, vol. 33, no. 5, pp. 96-103, Sep./Oct. 2019.

[17] 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.

[18] H. Xu, W. Yu, D. Griffith and N. Golmie, ”A Survey on Industrial Internet of Things: A Cyber-Physical Systems Perspective,” in IEEE Access, vol. 6, pp. 78238-78259, 2018.

[19] K. Tange, M. De Donno, X. Fafoutis and N. Dragoni, ”A Systematic Survey of Industrial Internet of Things Security: Requirements and Fog Computing Opportunities,” in IEEE Communications Surveys & Tutorials, DOI : 10.1109$/COMST.$2020.3011208, Jul. 2020.

[20] 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.

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

[22] 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.

[23] 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.

[24] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In Proceedings of the 27th NIPS14 - Volume 2 . MIT Press, Cambridge, MA, USA, 33203328, 2014.

[25] 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.

[26] Q. Chen, Z. Zheng, C. Hu, D. Wang and F. Liu, "Data-driven Task Allocation for Multi-task Transfer Learning on the Edge," 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 2019, pp. 1040-1050.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A.a Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. IJCV, 2015.

FIGURES

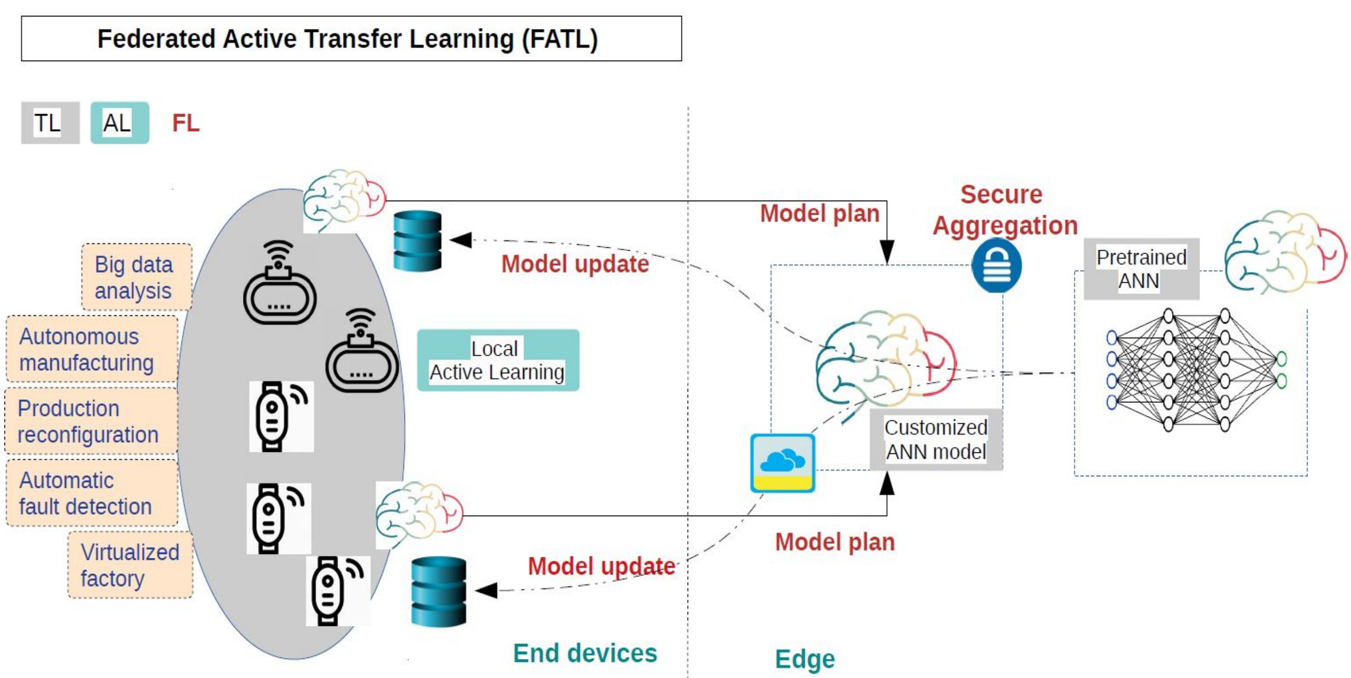

Fig. 1: Federated Active Transfer Learning (FATL) model.

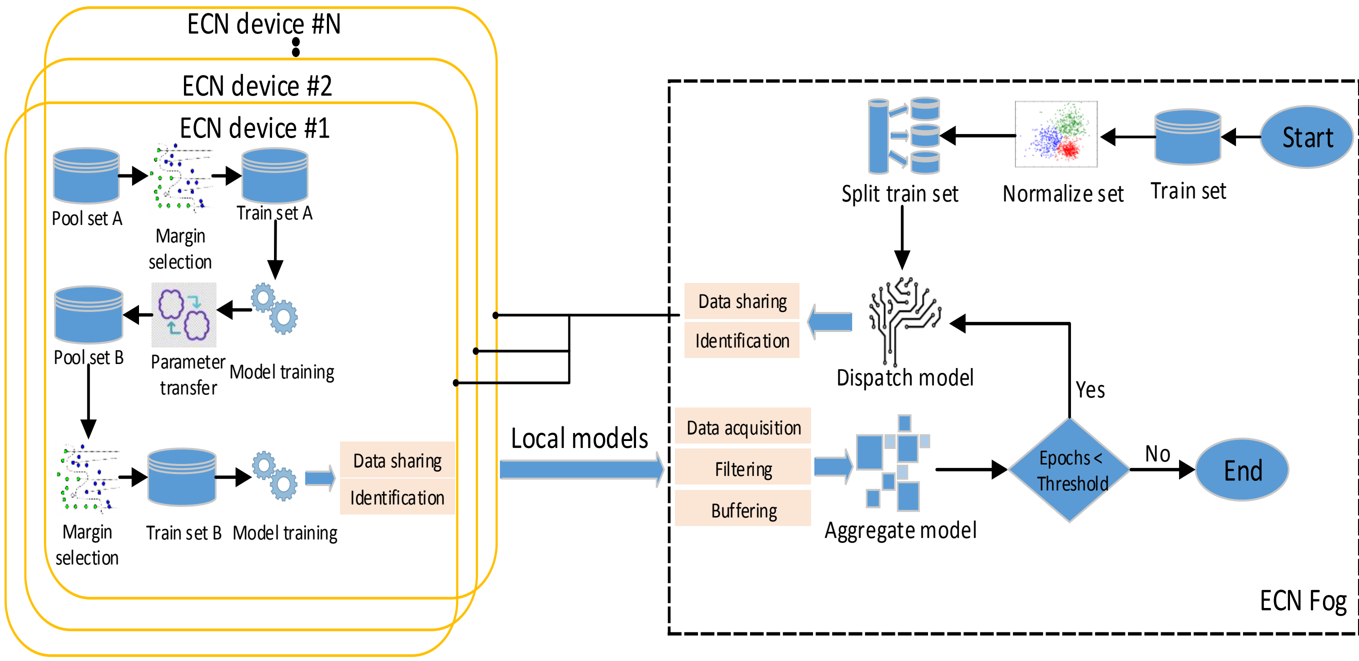

Fig. 2: Edge Artificial Intelligence (AI) through FATL.

TABLE I: Test accuracy of the ML models.

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

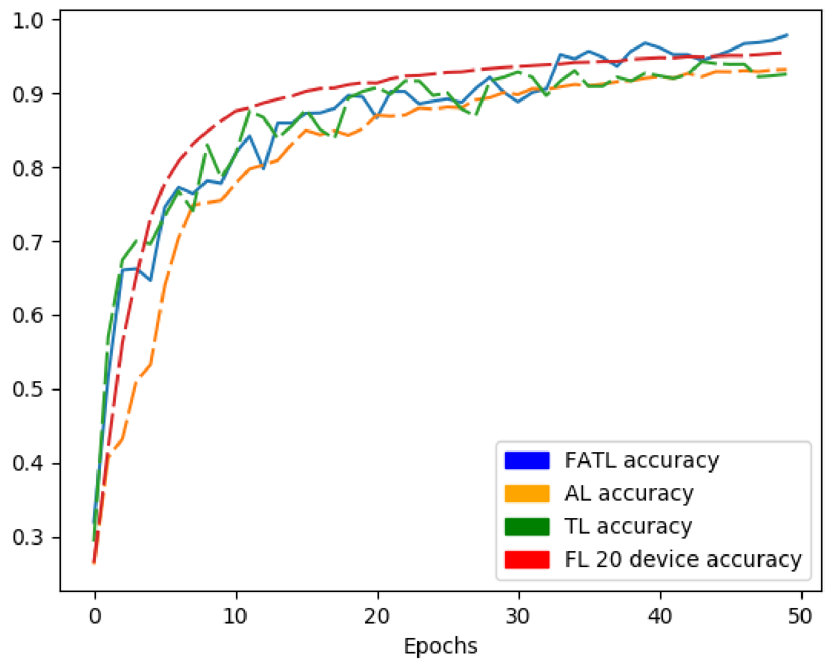

Fig. 3: Training accuracy over epochs per learning technique and the proposed FATL solution.

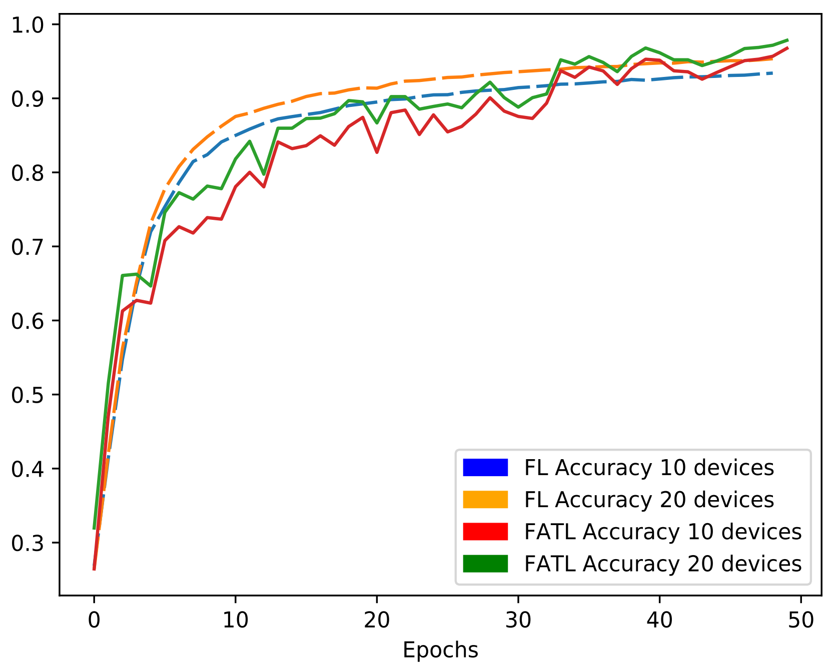

Fig. 4: Training accuracy over epochs for different amount of devices.

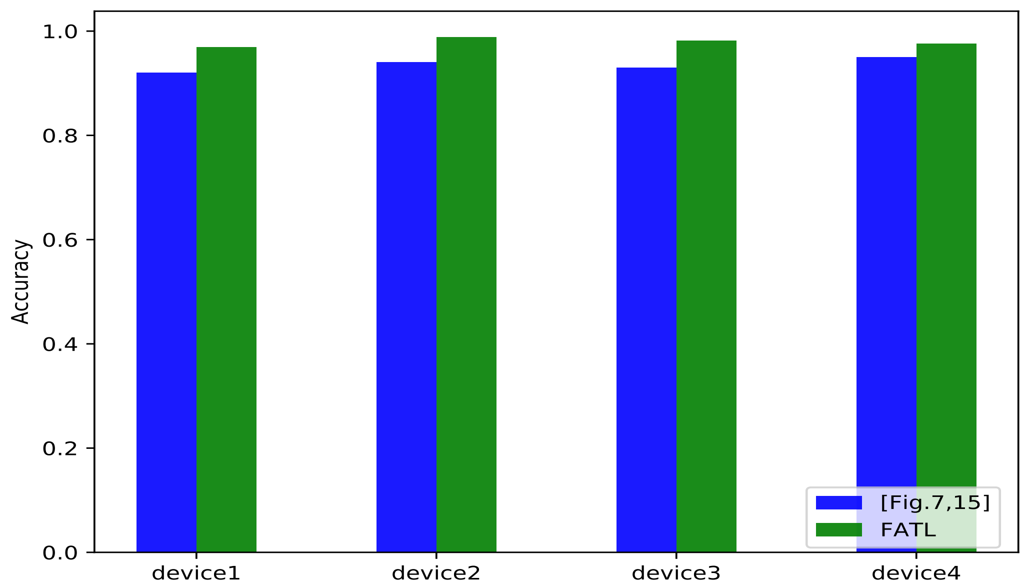

Fig. 5: Test accuracy in case of 4 IoT devices.

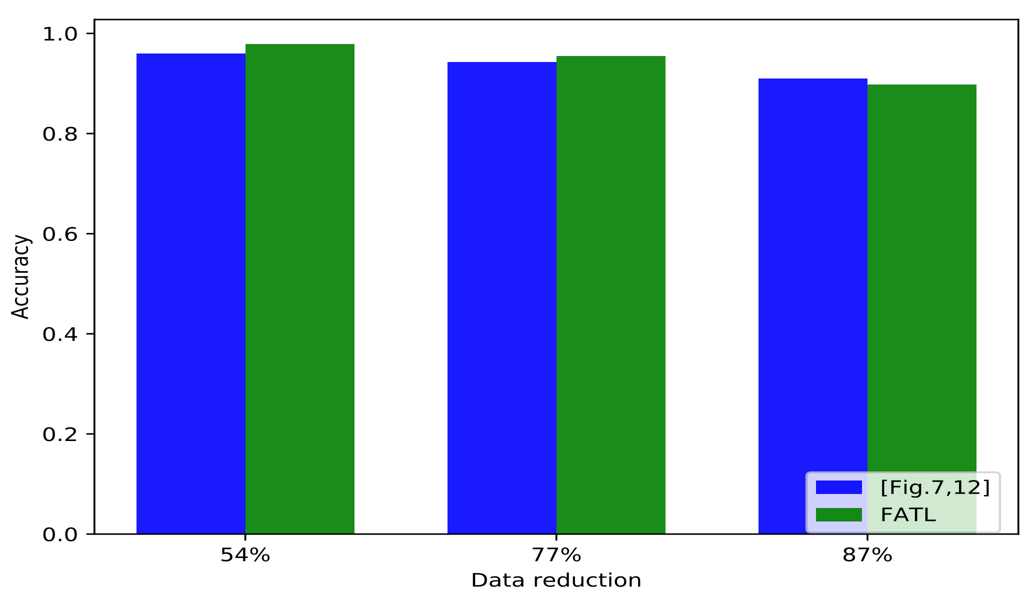

Fig. 6: Test accuracy per data reduction.

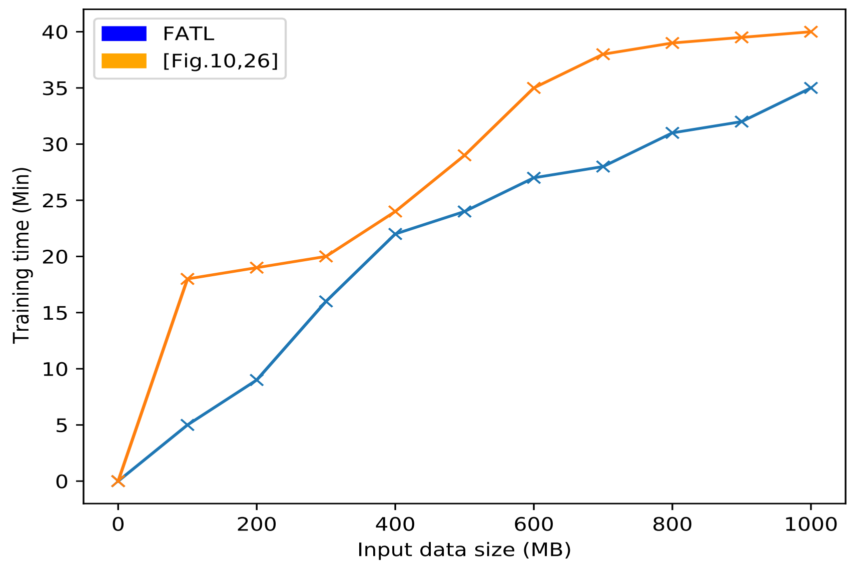

Fig. 7: The processing time of system tasks with different data input sizes.

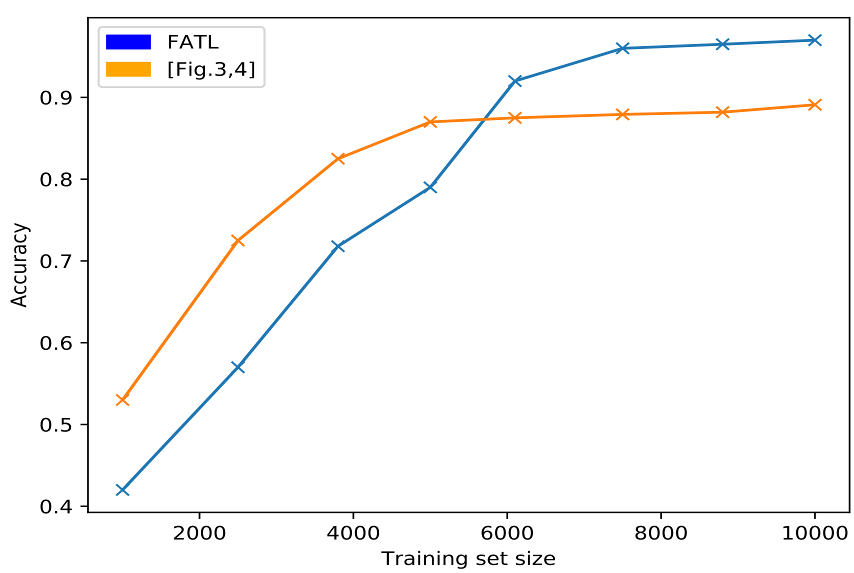

Fig. 8: Test accuracy over training size.