Achieving Democracy in Edge Intelligence: A Fog-based Collaborative Learning Scheme

Tiehua Zhang, Zhishu Shen, Jiong Jin, Member, IEEE, Xi Zheng, Member, IEEE, Atsushi Tagami, Member, IEEE, Xianghui Cao, Senior Member, IEEE

Abstract—The emergence of fog computing has brought unprecedented opportunities to the Internet of Things (IoT) field, and it is now feasible to incorporate deep learning at the edge of the IoT network to provide a wide range of highly tailored services. In this paper, we present a fog-based democratically collaborative learning scheme in which fog nodes collaborate on the model training process even without the support of the cloud, contributing to the advances of IoT in terms of realizing a more intelligent edge. To achieve that, we design a voting strategy so that a fog node could be elected as the coordinator node based on both distance and computational power metrics to coordinate the training process. Also, a collaborative learning algorithm is proposed to generalize the training of different deep learning models in the fog-enabled IoT environment. We then implement two popular use cases, including a user trajectory prediction and a distributed image recognition, to demonstrate the feasibility, practicality and effectiveness of the scheme. More importantly, the experiments on both use cases are conducted through a real-world, in-door fog deployment. The result shows that the scheme can utilize fog to obtain a well-performing deep learning model in the cloudless IoT environment while mitigating the data locality issue for each fog node.

Index Terms—Fog Computing; Internet of Things (IoT); Collaborative Learning; Democratic Voting Strategy; Edge Intelligence.

I. INTRODUCTION

The proliferation of the Internet of Thing (IoT) technologies has gained tremendous prominence due to the rapid development of the smart home, smart factory and smart city, etc. The widely deployed sensing devices are generating the sheer amount of ambient-related data for different analytical purposes. Currently, the majority of these data are transferred to the cloud for storage and further analysis due to the resource constraint at the edge of IoT network. However, there is a rising concern over the data privacy towards storing the data at the cloud data center [1]. Recently, the introduction of fog computing has attracted increasing attentions from researchers in the IoT field, owing to its capability of providing various resources in the vicinity of end users/devices (things). In particular, these things are able to connect directly to the deployed fog nodes in the quest for different services, including both the sensitive data storing and processing service, rather than clinging to the remote cloud. The capability of bringing computation, networking and control functionalities closer to the things essentially empowers fog computing to serve as the placeholder of many real-time IoT applications and deliver the respective services back to the requester promptly.

It has been witnessed a great potential of applying deep learning in the IoT field over the last decade owing to its superior data processing and analytical capability in the big data era. In particular, many researchers and industrial practitioners prefer to utilize the deep learning model as the central processing unit for various IoT applications to enhance users’ overall service experience [2]. For instance, entity moving trajectory prediction, particularly human moving trajectory service [3], is deemed as one of the crucial research problems due to its practicality and usefulness in terms of realizing the smart living in the pervasive environment. However, the majority of deep learning-based applications are both trained and deployed at the cloud due to the considerable computational resources needed, and it is thus no longer suitable for tasks like the real-time route planning in the autonomous driving system craving for a stringent service transmission time.

A newly emerged research area in the IoT, namely edge intelligence, has provided an alternative to tackle this issue, in which both the training and inference phases in deep learning process happens in the vicinity of end requester without degrading the quality of service from the trained model [4]. Briefly speaking, edge intelligence could stimulate the continuous evolution of the IoT applications in a broad spectrum of domains, spanning from human activity recognition, computer vision tasks to big data analytics. The long-standing issues happening in the cloud-centric approach such as high end-to-end service transmission latency, data communication overhead and privacy could be largely mitigated. Even though the fog network architecture could provide reliable backbone support for the edge intelligence, the research on fog-empowered edge intelligence is still in its infancy stage.

Therefore, it is of significance to investigate using only the fog plane resources to empower the collaborative deep learning process without exposing the local sensitive data to others. By doing so, not only the service transmission latency issue would
be alleviated, but also more on-demand, context-aware pervasive services could be highly customized for each user under complex IoT environment. Fig. 1 demonstrates the architecture of the fog-empowered edge intelligence. The bottom layer consists of a range of end devices/users connected into the network through the fog, and these devices/users are intrigued by various services. The fog plane receives and processes the service request coming from mobile devices, self-driving cars or robotics, etc., and fog nodes in this plane work towards generating the deep learning model together (dashed line among fog nodes) to provide pervasive services like real-time data analytic service or robotic control service. The cloud layer, on the other hand, only copes with a group of time-tolerant services like large-scale data backup service. While many researchers focus on using the cloud-fog stereotype to complete the training (either training the model at the cloud or using the cloud as the parameter server [5]), there is little attention on confining the model training and inference process entirely at the fog plane without the reliance on the cloud. Notably, the cloudless edge intelligence could help unleash the full potential of fog computing and benefit a broad spectrum of cases where the connection to the cloud is unavailable, such as the fog-enabled smart agriculture in the remote area.

Following this line of thinking, the primary aim of this work is to conduct research on the fog-enabled edge intelligence in the cloudless IoT environment, where the deployed fog nodes collaborate to train the deep learning model together using the data collected from each autonomous region. In order to automate the training process and alleviate the dependence on the cloud, we also propose a democratic voting strategy so that fog nodes could elect a coordinator to facilitate the model training on each fog node. Additionally, we attempt to gain insight and verify the effectiveness of the proposal through a real-world, in-door fog deployment setting. We summarize the main contributions of this work as follows:

1) To the best of our knowledge, we are the first only to use the fog plane to train a well-performed deep learning model collaboratively. The trained model at fog can provide the learning-related IoT services in the proximity of users, which points out a new direction of research on fog-enabled edge intelligence under the IoT context.

2) To reduce the reliance on the cloud, we propose a democratic voting strategy so that a coordinator node is elected based on the computational and distance matrices from the practical fog deployment to expedite the model training process.

3) An algorithm is designed to guide the fog through the training process, where the preliminary model training phase serves to alleviate the intractable non-IID data distribution problem at each participating fog as well as boost up the convergence of the training, whereas collaborative model training phase focuses on the details of the gradients update.

4) We verify the practicality and feasibility of the proposed scheme through the in-door moving trajectory prediction use case under the real-world fog deployment. Also, the distributed image recognition use case is applied to re-assure the robustness and effectiveness of the algorithm for different deep learning models.

The remainder of the paper is organized as follows. We conduct a comprehensive review of related works in fog computing and the state-of-the-art deep learning works for trajectory prediction and image recognition in Section II. In Section III, we introduce the proposed democratic learning scheme, including both fog voting strategy and fog collaborative learning algorithm. Afterwards, we walk through two different yet most commonly needed use cases, including the data collection process, in the edge intelligence context in Section IV, followed by the extensive experiment and analysis to demonstrate the practicality, effectiveness and robustness of the proposed scheme in Section V. Finally, We draw the conclusion and identify the future work in Section VI.

II. RELATED WORK

A. Data analysis in fog environment

As the most promising paradigm that brings big data analytics from the remote cloud closer to the end devices/users, fog computing has shed light on providing the time-stringent services at the edge of IoT network. By deploying applications in the fog node, the operational time on data analytics/communication can be significantly reduced compared with the conventional cloud-IoT approach [1], [6].

Table I summarizes the related works on data analysis in fog environment. When it comes to how to construct a fog platform for an efficient IoT service provisioning, two aspects are widely discussed: (a) reducing the service delay [7]–[9], and (b) reducing the energy consumption [10], [11], [16]. For example, Chen et al. [7] proposed a fog system that can adaptively be configured to host temporal and spatial services for sensor nodes. Kiani et al. [8] developed a hierarchical capacity provisioning scheme based on queuing analysis. Sciaronne et al. [10] proposed a probabilistic fingerprinting algorithm
TABLE I: Comparison of works on data analysis in fog environment.

<table>
<thead>
<tr>
<th>Research</th>
<th>Target</th>
<th>Method</th>
<th>Service</th>
<th>w/ fog-to-fog collaboration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chen et al. [7]</td>
<td>Reduce service delay</td>
<td>Online distributed algorithm</td>
<td>Industrial IoT services</td>
<td>No</td>
</tr>
<tr>
<td>Kiani et al. [8]</td>
<td>Reduce service delay</td>
<td>Hierarchical capacity provisioning scheme</td>
<td>Real-time IoT services</td>
<td>No</td>
</tr>
<tr>
<td>Fan et al. [9]</td>
<td>Minimize IoT dataloss latency</td>
<td>Probabilistic fingerprinting and offloading algorithm</td>
<td>Mobile cellular network</td>
<td>No</td>
</tr>
<tr>
<td>Sciarrone et al. [10]</td>
<td>Reduce energy consumption</td>
<td>Evaluation on energy-efficiency metrics</td>
<td>Indoor localization</td>
<td>No</td>
</tr>
<tr>
<td>Fiandrino et al. [11]</td>
<td>Reduce energy consumption</td>
<td>SLA configuration process</td>
<td>Datacenter communication</td>
<td>No</td>
</tr>
<tr>
<td>Bisio et al. [12]</td>
<td>Minimize energy consumption</td>
<td>Transparent context-aware cloud paradigm</td>
<td>D2D communication</td>
<td>No</td>
</tr>
<tr>
<td>Al-Ridha et al. [13]</td>
<td>Enhance QoS and load balancing, reduce energy consumption</td>
<td>Real-time IoT services</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Yousef et al. [14]</td>
<td>Reduce service delay</td>
<td>Delay-minimizing fog offloading policy</td>
<td>Real-time IoT services</td>
<td>Yes</td>
</tr>
<tr>
<td>Al-Khafajy et al. [15]</td>
<td>Minimize delay for IoT services</td>
<td>Fog resource management scheme</td>
<td>Real-time IoT services</td>
<td>Yes</td>
</tr>
</tbody>
</table>

for indoor localization in which the experimental results indicated that energy saving can be achieved by offloading data over to the fog. Meanwhile, fog-to-fog horizontal data communication can further enhance the distributed processing capability at fog nodes while reducing dependency on the cloud [17]. For this reason, the impact of utilizing fog-to-fog collaboration is also widely discussed with the overarching goal of either service delay reduction [14], [15] or energy consumption reduction [13]. However, the fog collaboration in the deep learning context, which is also a significant criterion to measure the development of fog platform in the big data era [1], has been neglected in prior studies like [18].

There are numerous research focusing on how to utilize deep learning in the fog context to achieve satisfactory data processing performance. Commonly, each fog node collects data from its managed end devices/users, and then sends these data to the cloud for training the model separately to the respective fog nodes. For example, Memon et al. [19] proposed a recurrent neural network (RNN)-based algorithm to predict the handover point for cars inside vehicular networks. In their proposed algorithm, the cloud delicately created the prediction model for each fog node through learning the features composed of the device’s location and service time information from respective fog nodes. Similar approach was adopted in [20] for an RNN-based IoT device’s intrusion detection application in which data training and detection function are performed locally on each fog node. Interestingly, Shen et al. [17] designed an in-network self-learning algorithm for anomaly detection for building energy management system (BEMS) so that fog node could communicate with each other horizontally via the information-centric networking. In their work, the detection function was fed with the data from not only the target fog node itself but also its neighboring ones, leading to a significant improvement of detection accuracy. However, while these previous works either rely on the cloud to train the model or expose the locally collected data to serve the in-fog training in an isolated manner, none of them investigates on how to refine the collaborative training process only at the fog layer without exposing the local data to others.

B. Deep learning-based applications in edge intelligence

Undoubtedly, deep learning has played a pivotal role in edge intelligence when it comes to providing superior service experience in many fields. Among all, moving trajectory prediction and image recognition are generally considered as the essential tasks that need to be tackled if one wants to establish an intelligent living environment. Herein, we follow up on the progress of these two widely adopted use cases.

1) Trajectory prediction: Currently, many moving trajectory prediction algorithms, regardless of the traditional or state-of-the-art deep learning techniques, rely on information propagated by either global positioning system (GPS) or various entity moving images and videos to achieve a satisfactory result, yet it is considered inapplicable in many in-door trajectory predicting scenarios due to the coarse granularity of such information. Alternatively, the use of signal strength indicator (RSSI) has been studied extensively in the wireless sensor network (WSN) context, where the communicating signals from each sensor could be further utilized in many cases without the quest for additional hardware installation other than the sensors. Undoubtedly, one of the biggest beneficiaries by incorporating deep learning at the fog layer is the user trajectory prediction. Instead of using the GPS data, the alternative of user’s trajectory prediction is to exploit the image or video data. For example, Alahi et al. [3] proposed an RNN-based learning approach for trajectory prediction using aerial image information, where a series of motions are handled by long short-term memory (LSTM) cells, and information between LSTM is shared through a social pooling layer to enhance the trajectory prediction performance. However, the image data could not be obtained easily and overused due to the concerns over user privacy, not to mention the limited number of available devices that could be used for data collection. Other than that, Gupta et al. [21] developed a Social-LSTM based on a LSTM-GAN structure, which achieves better prediction accuracy and efficiency for the same task. Deo et al. [22] utilized Social-LSTM-like architecture to predict the trajectory of a car in an interactive driving scenario involving multiple cars. Different from the above approaches, Luo et al. [23] took advantage of the 3D spatial-temporal data and then use a small-scale CNN to forecast car trajectories. Lee et al. [24] designed an RNN-VAE based encoder-decoder architecture for trajectory prediction and achieved a low error rate on KITTI dataset [25].

Therefore, we focus on using the more accessible RSSI signal data from reference wireless signal transmission devices (namely “fingerprinting” [26]) to benefit the trajectory prediction, and the work in this paper is the first of its kind.
to use fog to predict the user’s trajectory.

2) Image recognition: Deep learning is also widely adopted in the field of image recognition owing to its remarkable performance [27]–[29]. For example, He et al. [28] applied a deep residual learning framework based on the deep neural networks, which can attain a high accuracy with different image datasets. Also, how to better utilize fog to make the image recognition under the complex IoT environment has been studied as well. For instance, Hu et al. [30] proposed a fog-based face identification system, where local binary pattern (LBP) is used to extract face features. By adopting task partitioning, this fog platform can offload computing overhead from the cloud by which the processing burden and network communication volume is reduced. Similarly, in [31], the proposed fog-based image recognition application was proven efficient to reduce the latency and cloud usage by 5 times. Patman et al. [32] proposed an image processing method in which the machine learning model is utilized to predict task partitioning, this fog platform can offload computing potential candidate computationally powerful enough to enable the training process. Therefore, we use the RSSI signal as the indicator of the distance from one node to the others. In a nutshell, RSSI value expresses the inverse proportion to the square of distance between two communicating nodes, and this feature is hence widely applied in many distance measurement cases in the IoT environment [33]. When it comes to deciding if a particular node is suitable to serve as the candidate node, it is clearly irrational to only consider the relative distance as the node with the best relative distance may not be the one with enough computational resources. Meanwhile, we assume that each fog node provides multiple resources, including CPU, memory and bandwidth (BW) [34]. The computational power is thus related to the overall ratio of these three resources available combined.

Henceforth, there are two vectors used to represent the relative distance $d \in \mathbb{R}^{n-1}$ and proportional computational power $c \in \mathbb{R}^{n-1}$ in each fog node. In addition, a system-level hyper-parameter $\lambda$ is introduced in order to cater to different deployment requirements, the value of which could be chosen within a range of $[0, 1]$ and adapted to balance off the imperative needs between distance and resources. Specifically, we use $\mathcal{F} = \{f_1, f_2 \ldots f_n\}$ to denote the set of fog nodes in the network and $v_f$, to represent the voting vector of any participating fog node $f_i \in \mathcal{F}$, which can be formulated as:

$$v_f = \lambda \ast d_f + (1 - \lambda) \ast c_f, \quad v_f \in \mathbb{R}^{n-1}$$

(1)

Specifically, each value in vector $d$ and $c$ is calculated as the following:

$$d_f = [d_{f_1}^1, d_{f_2}^2, \ldots d_{f_n}^n]^\top, \quad d_{f_i} \notin d_f$$

(2)

$$d'_{f_n} = 1 - \frac{\sum_{f_i \in \mathcal{F}, f_i \notin f_n} (s_{f_i, f_n})}{\sum_{f_i \in \mathcal{F}, f_i \notin f_n} (s_{f_i, f_j})}$$

(3)

where the values $d'_{f_n}$ in $d_f$ essentially represents the distance score between node $f_i$ and $f_n$, the value of which will increase as the increase of the RSSI signal strength $s_{f_i, f_n}$. It is worth mentioning that in reality, the detected signal strength of $s_{f_i, f_n}$ is not always equal to that of $s_{f_n, f_i}$. Specifically, each value in vector $c$ and $\lambda$ is calculated as the following:

$$c_f = [c_{f_1}^1, c_{f_2}^2, \ldots c_{f_n}^n]^\top, \quad c_{f_i} \notin c_f$$

(4)

$$c'_{f_n} = \frac{c_{f_n}}{\sum_{f_j \in \mathcal{F}} (c_{f_j})}$$

(5)

$$c_{f_j} = \frac{\text{ratio}(CPU_f) \times \text{ratio}(MEM_f) \times \text{ratio}(BW_f)}{\text{total}(Resource_f)}$$

(6)

When it comes to calculating the vector of computational power, from node $f_i$’s perspective, the proportional computational power of node $f_i$ is simply the ratio between its power and the reported total computational power in this fog network. From fog node $f_i$’s point of view, the computational power $c'_{f_i}$ is calculated by multiplying the unused resource percentage of the three (CPU, memory and BW) computational resources. The larger the $c'_{f_i}$, the more resources are available to the respective fog node.

After calculating the respective voting vector at each fog node, these nodes then communicate with their peers to retrieve other voting information. We elaborate on the voting...
Algorithm 1 Coordinator Voting Procedures

Input: the set of deployed fog nodes $F$, system-level hyperparameter $\lambda$

1: procedure COMPUTATIONAL POWER CALCULATION
2: for each $f_i \in F$ do
3: use (7) to get ratio of each resource
4: calculate the $\delta f_i' \text{ based on equation (6)}$
5: send $\delta f_i'$ to other nodes in $S = \{f_j \parallel j \neq i, f_j \in F\}$
6: end for
7: end procedure

8: procedure VOTING VECTOR GENERATING
9: for each $f_i \in F$ do
10: measure the current $s_{f_i, f_j}$, where $f_j \in S$
11: calculate the $\delta f_i'$ based on equation (3)
12: compose $d_{f_i}$ and $c_{f_j}$
13: get $v_{f_i}$ based on equation (1) and $\lambda$
14: send $v_{f_i}$ to other nodes in $S$
15: end for
16: end procedure

17: procedure BALLOT COUNT
18: for each $f_i \in F$ do
19: $f_i^{\text{dict}} \leftarrow \emptyset$ \text{ } \triangleright \text{ initialize an empty dictionary}
20: $f_i^{\ast} \leftarrow \arg\max f_i^{\text{dict}}(v_{f_i})$
21: $f_i^{\text{dict}}(f_i^{\ast}) \leftarrow f_i^{\text{dict}}(f_i^{\ast}) + 1$
22: for each received $v_{f_i}$ do
23: $f_i' \leftarrow \arg\max f_i^{\text{dict}}(v_{f_i})$
24: $f_i^{\text{dict}}(f_i') \leftarrow f_i^{\text{dict}}(f_i') + 1$
25: end for
26: end for
27: $f^c \leftarrow \arg\max f_i^{\text{dict}}(f_i') \triangleright \text{ get the coordinator node}$
28: end procedure

process in Algorithm 1, consisting of three different procedures. The first procedure (lines 2-6) focuses on the calculation of $\delta f_i'$ at each fog node using both equation (6) and (7), and the local computational power is sent to other fog nodes. After each fog receives the $\delta f_i'$ from others, the node kicks off the second procedure by recording the current RSSI readings from its peers, which is then used to calculate the corresponding distance score (lines 10-11). Following that, the local voting vector could be assembled and sent out (lines 12-14). The last procedure is related to the ballot count, where each node uses the dictionary to accumulate the ballots based on the received voting vector (lines 19-21). The dictionary uses the fog identifier as the key and the integer as the value, and whichever fog node receives the highest voting score will get its value added by 1 (lines 22-26). Line 27 indicates that the fog with the highest ballot count will be elected as the coordinator.

B. Collaborative Learning Algorithm

We introduce a two-stage collaborative learning algorithm in here, consisting of the model initialization phase and the model training phase.

Algorithm 2 Collaborative Learning Algorithm

Input: coordinator node $f^c (f^c \in F)$, a set of other fog nodes $S = \{f_i \parallel i \neq c, f_i \in F\}$, small epoch number $\xi$, the positive training rate $\gamma$ and collaborative training epochs $e$

Phase 1: Preliminary Model Training

Output: $M$: preliminary model trained on each node by $\xi$ epochs

1: $M \leftarrow \text{start at the coordinator node } f^c \text{ to initiate the model by training } \xi \text{ epochs}$
2: for each $f_i \in S$ do
3: compose the local dataset $D_i \leftarrow \{(x_i, y_i)\}$
4: $M_{f_i} \leftarrow \text{receive the } M \text{ from the coordinator } f^c$
5: train the $M_{f_i}$ for $\xi$ epochs on $D_i$
6: send the $M_{f_i}$ back to $f^c$ and overwrite $M$
7: end for

Phase 2: Collaborative Model Training

Output: $M'$: generalized model trained collaboratively by all fog nodes

1: repeat
2: for each mini-batch $n$ do
3: for each $f_i \in F$ do
4: $M_{f_i}$ sends model $M$ to each node $f_i$
5: $y_i^{\text{batch}}$ sends model $M$ to each node $f_i$
6: $y_i^{\text{batch}} \leftarrow \frac{\partial}{\partial w_i} \log(y_i^{\text{batch}})$
7: $L_{f_i} \leftarrow \frac{1}{n} \sum_{i=1}^{n} y_i^{\text{batch}} \cdot \log(y_i^{\text{batch}})$ \triangleright mini-batch cross entropy loss
8: $\Delta w_i^{f_c} \leftarrow \frac{\Delta L_{f_i}}{\sum_{i \in F} \Delta w_i^{f_i}}$
9: upload $\Delta w_i^{f_c}$ to $f^c$
10: end for
11: end for
12: $w_e \leftarrow w_e - \frac{\gamma}{\sum_{i \in F} (\Delta w_i^{f_i})}$
13: distribute the $M'$ to each participant node to overwrite the local model
14: until (collaborative training epochs $e$)

Algorithm 2 demonstrates the details of these two phases, and we refer to the elected fog node as the coordinator node to facilitate these two phases. Specifically, in order to solve the intractable non-IID data distribution problem caused by the limited local data pool, the coordinator node first trains the preliminary model under a small number of epochs $\xi$ (around 20) using its locally private data (line 1 in Phase 1). It then iteratively sends the model to other fog nodes to ensure that the preliminary model is initiated properly by drawing knowledge from multiple local data distributions managed by different fog nodes (lines 3-6). This model initialization process resembles the well-known knowledge transfer learning in which a trained model from one task is used as the starting point of another task. In other word, the designated Phase 1 enables the preliminary model to generalize as much data information from different parties as possible to tackle with the non-IID data distribution. Note that the selection of the hyper-parameter $\xi$ leads to the different loss convergence of the collaborative model training in Phase 2, which will be examined extensively in the experiment section.
In the second phase, the coordinator node first distributes the obtained model to other nodes to initiate the collaborative training process. Every participant node takes the received model as the starting point while preparing the local batch features and labels at the same time (lines 2-5). Each node then feeds the batch features into the local model to get the predicted labels, which is used, along with the true batch labels, to calculate the mini-batch cross entropy loss (lines 6-7). Subsequently, a Jacobian matrix composed of the gradients of the loss with respect to each trainable weight in the local model is calculated and sent to the coordinator (lines 8-9). These gradients are then aggregated as the final update of the model $M'$, which is then dispensed to each participant to overwrite their local model as an update of this round (lines 12-13). The collaborative training process will repeat until reaching the training epochs $e$ (line 14).

IV. CASE STUDY

We adopt the indoor user’s trajectory prediction as the primary use case in the fog-enabled IoT environment to present the merit of our work. Say in the context of an exhibition event, an accurate indoor prediction can be of importance when it comes to alleviating the overly-crowded flow and showing the usage status of a particular space, such as a booth, to the attendees. To realize the trajectory prediction, the participant can bring a communicating device, such as a mobile phone, to gather the signals from reference devices, e.g., sensors or WiFi access points, inside the building. Also, the image recognition task is often considered critical so as to better realize the smart environment, including smart factory, building and city. To verify the robustness and effectiveness of the proposed algorithm in different model structures, a convolutional neural network (CNN), aiming to facilitate the distributed image recognition service under fog, is trained from scratch.

1) In-door trajectory data collection: To replicate the accurate RSSI signal waves as a user moves towards different spots, we set up an indoor IoT network composed of fog nodes and wireless sensors, as shown in Fig. 2. In this network, 20 sensors (Feasycm FSC-BP103, ID from 1 to 20) work as reference devices that send wireless signals to points (A, B, …, Z) that user may walk through. At each point, the regional fog node collects RSSI data from all managed sensors for a period of time. There are four fog nodes deployed in field, each of which gets connected with its peers (green dashed arrows) and also manages the sensors residing in that area (with different colored overlays). When a user walks between two points inside the building, the signal receiving device (smartphone) held by the user collects RSSI data from sensors placed in the same area and then sends these data to the respective fog node for further service operations. Fig. 3 interprets the changes of the received RSSI signal of a wearable device carried by a user from different sensor-placed spots 1 to 5 while a user walks from point A to D in the yellow-coloured fog region. Evidently, the received signal strength from a fixed sensor spot varies along with the change of distance with the wearable device. The result also demonstrates that fluctuation naturally exists in this time-series RSSI data due to the unexpected interferences in the real-world IoT environment that are intractable to mitigate. However, the collaboratively trained deep learning model is expected to comprehend such situation and explore the underlying correlation among RSSI signals to achieve a satisfactory performance for the given learning-related applications regardless.

2) Image recognition: When it comes to the data used for distributed image recognition, we use the Fashion-MNIST dataset, which comprises the images of 70,000 fashion items from 10 categories, with 7,000 images per category, as the training and testing data [35]. Fig. 4 illustrates some sample
product images that are randomly chosen from the dataset. In reality, the distributed image recognition service, benefiting from the collaboratively trained CNN model, is essential to solving many real-world problems. One example is product identification in the smart shopping mall or automatic distribution of items to the warehouse in the smart factory.

V. EXPERIMENTAL RESULTS

A. Voting Result

Table III demonstrates the snapshot of the received signal value of different fog nodes. Based on the signal value, the distance score (elucidated in Section III) could be calculated accordingly. As all deployed fog nodes have the same hardware configurations in this work, the computational resources at each node are thus the same, each of which has the value of 0.25. It is worth pointing out that we use four power-plugged laptops with the same configuration to serve as the fog role, the wireless communication among which is enabled. Explicitly, each fog node is equipped with a four core Intel i5-6200U 2.3GHz CPU and 4GB memory. From the performance perspective, this specification is tested powerful enough to enable the training process even for a centralized data processing where all data are utilized for model training.

Following that, Table IV shows the elected coordinator node based on different \( \lambda \) values from 0 to 1. Note that the larger the \( \lambda \) value is, the more impact the fog node’s placement has to the final voting result. In the real-world fog deployment, fog 3, which roughly locates in the centre area of the deployed network, is more likely to be elected. Since each fog node has the same computational power, the final voting result entirely relies on the in-door geographic location of fog node. In general, the selection of the coordinator fog node also depends on the practical requirement from the network admin staff who needs to consider the trade-off between current fog node’s location and its computational power. This indicates that the value of \( \lambda \) is flexibly adjustable for various fog deployment scenarios, where the remote fog-enabled smart agriculture may prefer the computational power towards the distance, but cases like smart building tend to choose the opposite.

B. Collaborative Training Result

We conduct comprehensive experiments on both in-door trajectory prediction and distributed image recognition in this part. As mentioned above, the algorithm focuses on enabling the fog nodes to train the model collaboratively even without the support from the cloud and exposing the local data to others, which is inspired by many IoT service providers that wish to keep users’ data private. The pre-deployed fog nodes, as shown in Fig. 2, all participate in the collaborative model training process. Regarding the model structure, we implement the base of both deep neural network and convolutional neural network for trajectory prediction and image recognition tasks through PyTorch\(^1\), respectively, the details of which can be found in Table V.

1) Use Case 1 - Trajectory prediction: As the primary use case, we first present the collaborative training loss at each fog node in a 100 epochs period, as shown in Fig. 5. The y-axis, indicating the loss value, is scaled down through logarithm for better demonstrating purpose. To be more specific, each sub-figure demonstrates the descending degree of loss from four different settings of \( \xi \) as revealed in Algorithm 2. It can be observed that fog 4, as the last node conducting the training on preliminary model \( M \), has a notable drop of loss at the early stage of the training compared to what happened in other nodes, especially when \( \xi \) is larger than 60. This result can be interpreted as whichever node that trains the \( M \) at last benefits from the generalization of the data distribution that is more desirable to its local dataset. In other words, the model \( M \) trained at Phase 1 of the Algorithm 2 is in favor of the last node on which it is trained, and it becomes more favorable towards the local dataset at the last node along with the increase of \( \xi \) (a rapid drop of loss for both \( \xi = 60 \) and 80). From \( \xi \)'s perspective, the unconstrained increase of \( \xi \) could impede the convergence speed of loss. Specifically, the loss generally reaches the lowest with a faster pace when the \( \xi = 20 \), even though the loss with different \( \xi \) at different fog node would converge to the same magnitude through the collaborative training process.

Based on the observation from Fig. 5, we adopt model \( M' \) trained with \( \xi = 20 \) as the final inference model at each fog node. The performance of the model is entirely determined

---

1 https://pytorch.org
by the prediction accuracy on the single point user position localization as well as 2 points, 3 points and 4 points-based user trajectory prediction, respectively. Fig. 6 illustrates the user trajectory accuracy of the collaboratively trained model based on the local test dataset at each fog node. Note that the model will only be considered making an accurate trajectory prediction in a 2 points-based user trajectory context if a user moves from point $A$ at timestamp $t_i$ to point $B$ at timestamp $t_{i+1}$ both being predicted correctly, and this rule is applicable for both 3 and 4 points-based user trajectory as well. In practical, we can say that a user’s trajectory is essentially composed of a discrete number of points in the real-world scenarios. The test dataset of each n points-based user moving trajectory on each fog node is not generated irrationally, it is carefully collected by us through mimicking the daily walking trajectory in the fog deployed field. In a nutshell, the result evidently verifies that our proposed collaborative learning scheme empowers fog to estimate the trajectory of a moving object merely based on the variance of the RSSI signal. There has been an expected gradual drop of accuracy starting from 1 point location prediction up to 4 points-based trajectory prediction at each fog node, yet the overall performance of the collaboratively trained model is still considered exceptional. From fog node’s perspective, the accuracy across different points-based trajectory in Fig. 6 essentially reflects the result from Fig. 5, where fog 4 achieves the highest accuracy among all (0.986, 0.966, 0.953 and 0.929, respectively), and the shared model still performs well on other fog nodes with locally collected data. We want to point out that the quality of RSSI signal data used for training at each fog node differs due to the real-world physical settings of the IoT environment in which sensors are deployed. For instance, there are more obstacles like office cabinets (drawn in Fig. 2) in fog 1 managed area and thus inevitably causing more interference,
which could deteriorate the quality of collected data in that area as well as the accuracy.

It is worth mentioning that we also trained the model in a centralized manner and use it as the benchmark to further compare the model performance with each fog node. To simulate the cloud training process, all independent dataset at each fog nodes are sent to one place to generate the global training dataset. After the model finishes the training using this dataset, it is also evaluated with n-points trajectory prediction separately. The results from the centralized trained model (0.965, 0.934, 0.908 and 0.852) indicate that the proposed scheme could produce a comparable result from the model trained by all participating fog nodes collaboratively even without the support from the cloud and exposing the local data to others.

2) Use Case 2 - Image recognition: To verify the robustness of the proposed collaborative learning algorithm, it is also applied to train a convolutional neural network from scratch to fulfill the recognition-related IoT services with the help of fog.

As explained in the Section IV, the dataset used for both training and testing comes from the widely used Fashion-MNIST dataset, which includes a total of 70,000 fashion products from 10 categories with 60,000 training images and 10,000 testing ones. Each fog node is randomly assigned with a distinct number of image categories, say fog 1 contains two categories of images while fog 2 has three, whereas the test set, including all ten categories of items, is the same across each fog node. To observe the variance of the prediction, each fog node randomly selects 5000 images from the test set at each test iteration and run the process 10 times, which eventually draws the accuracy results with mean and variance as shown in Fig. 8. We apply the benchmark here based on results from the centralized model, which is trained using the same training approach as described in User Case 1. Specifically, the centralized model takes advantages of the complete training dataset, including all ten categories of items, as opposed to only two or three classes of images in each fog node. Also, Fig. 7 presents the detailed confusion matrix snapshot of one test iteration through a format of heatmap.

It is witnessed in the Fig. 7 that some items could be easily mis-classified by fog due to the ambiguous difference of images in the training dataset. For instance, the shirt, pullover, t-shirt, dress and coat are a group of items that could mislead the trained CNN model as the lines, edges and shape of the item resemble one another from the model’s point of view. However, the final image recognition accuracy, as demonstrated in Fig. 8, suggests that the collaboratively trained CNN model is still able to obtain a similar performance compared with the centralized trained model (0.914 benchmark line). The results also re-emphasize that non-IID data distribution issue is mitigated for image recognition case as well.

The experimental results from these two practical use cases prove that the fog-based collaborative learning scheme could benefit a spectrum of deep learning-empowered IoT applications by enabling the training of the model in the fog plane effectively. Additionally, the scheme also demonstrates the capability of generalizing the training of models with different underlying structures (both DNN and CNN in here).

VI. CONCLUSION AND FUTURE WORK

In this paper, we present the research work on using a democratic approach to enable collaborative learning in the fog-enabled IoT environment. The real-world, in-door fog and sensor deployment are used to verify the effectiveness of our proposed scheme, in which each fog node manages a specific area and utilize the received the RSSI signal data from users
to make the corresponding trajectory prediction. Apart from that, we also look into the possibility of allowing distributed image recognition through CNN under the same experimental setup. It is assured by the experimental results from both use cases that our scheme could empower the training process even under a cloudless environment. Furthermore, the two-stage collaborative training algorithm could alleviate the impact on non-IID training data issue on each fog node without exposing the local data to others, while achieving a satisfactory model performance for both use cases regardless. Henceforth, we believe that our proposed approach could benefit the arising edge intelligence through refining the learning process in the fog-enabled IoT environment.

For future work, we will continue to investigate the possibility of using multiple data sources to improve the quality of the collected data instead of only using RSSI signals. In the meantime, we are expanding the scale of deployment for both fog nodes and sensors in other environment settings to explore the feasibility of the scheme in other use cases like smart agriculture. We also plan to implement an extensive comparison between the fog only strategy and the cloud-reliant one.

REFERENCES


