

A Democratically Collaborative Learning Scheme for Fog-enabled Pervasive Environments

Tiehua Zhang¹, Zhishu Shen², Jiong Jin¹, Xi Zheng³

¹School of Software and Electrical Engineering, Swinburne University of Technology, Australia

²KDDI Research, Inc., Japan

³Department of Computing, Macquarie University, Australia

Email: tiehuazhang@swin.edu.au, shen@kddi-research.jp, jiongjin@swin.edu.au, james.zheng@mq.edu.au

Abstract—The emergence of fog computing has brought unprecedented opportunities to many fields, and it is now feasible to incorporate deep learning at the edge to facilitate the development of pervasive systems (e.g., autonomous driving and smart grids). In this paper, we present our preliminary research on a democratic learning scheme so that fog nodes could collaborate on the model training process even without the support of the cloud, which is urgently needed in the pervasive computing context. The main objective of this work is to utilize the deployed fog nodes to train a well-performed deep learning model together, even with the limited local data from each participant. Instead of relying on the cloud by default, we design a voting strategy so that a fog node could be elected as the coordinator based on both distance and computational power metrics to help expedite the training process. We then experiment the effectiveness of the scheme through a real-world, in-door fog deployment and verify the performance of the trained model through a human moving trajectory tracking use case.

I. INTRODUCTION

The proliferation of pervasive computing has gained tremendous prominence due to the rapid development of smart home, smart factory and smart city, etc. The introduction of fog computing has attracted the increasing attention from researchers in the pervasive system community owing to its capability of providing various resources in the vicinity of end users/devices (*things*). These *things* are thus able to connect to the deployed fog nodes in the quest for different services rather than waiting for the responses from the remote cloud. Among these use cases mentioned above, entity moving trajectory prediction, particularly human moving trajectory, is deemed as one of the crucial research problems from the pervasive computing perspective due to its practicality and usefulness in terms of improving people’s life quality. For instance, in the smart home scenario, the moving information could be captured by the sensing equipment or wearable devices and sent to the processing units in the pervasive system for further analysis, then the location-based services could be provided accordingly.

Machine learning techniques, especially deep learning, have unleashed the great potential over the last decade and emerged as the key enabler to produce the superior results in entity’s moving trajectory prediction [1], [2]. While the majority of works focus on making good trajectory prediction through a centralized trained model, there is little effort on how to get multiple participants involved and train the corresponding

model collaboratively in the pervasive computing context. Even though the centralized training approach, for example using the cloud, has demonstrated the easy-to-deploy, economical-to-train advantages, it is still plagued by the issues like data communication overhead and data privacy due to the demand of uploading the data to the central server.

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. However, 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) field, 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.

In this preliminary study, we investigate the real-time user moving trajectory prediction problem in the fog-enabled pervasive environment, where the deployed fog nodes collaborate to train the model together using the data collected from their managed areas. In order to automate the training process and alleviate the dependence on the cloud, we also propose a voting strategy so that fog nodes could elect a candidate to coordinate updates of the local model on each fog node. Lastly, the effectiveness and performance of the proposed scheme is verified through a real-world, in-door fog deployment setting.

II. RELATED WORK

As the most promising paradigm that brings big data analytic capability from the remote cloud closer to the end users/things, fog computing has risen up to the spotlight. There are numerous studies focusing on the research of how to utilize machine learning in the fog context to achieve satisfactory data processing performance. Commonly, each fog node collects data from its managed end users/things, and then sends these data to the cloud for training the model separately to the respective fog nodes. For example, the authors in [3] proposed a recurrent neural network (RNN)-based algorithm to predict the handover point for cars inside vehicular networks. In their proposed algorithm, the cloud creates 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 is adopted in [4] for an RNN-based intrusion detection application in which data training and the detection function are performed locally on each fog node. Interestingly, the authors in [5] 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 is fed with the data from not only the target fog node itself but also its neighboring ones. In this manner, the detection accuracy is significantly improved.

It is well known that GPS data could be helpful for conventional user’s trajectory prediction. Still, the coarse granularity of the GPS data poses a challenge for in-door pervasive systems. Alternatively, the authors in [1] shifted their attention to image data. They 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. Taking into account all of these factors, we thus focus on using the more accessible RSSI signal data from reference wireless signal transmission devices (namely “fingerprinting” [6]) to benefit the trajectory prediction.

III. PROPOSED SCHEME

In this section, we introduce the democratic learning scheme, including both the voting strategy and a two-stage collaborative learning algorithm.

A. Voting Strategy

There are two essential questions worth considering when a candidate coordinator node is required to be voted: 1) is the potential candidate relatively close enough to other nodes? 2) is the potential candidate computationally powerful enough to enable the training process? With these in mind, we use 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 [7]. When it comes to deciding if a particular node is suitable to serve as the candidate node, it is clearly irrational only to consider the relative distance, as the node with the best relative distance may not be the one with enough computing resources.

Henceforth, there are two vectors used to represent the relative distance $\mathbf{d} \in \mathbb{R}^{n-1}$ and proportional computational power $\mathbf{c} \in \mathbb{R}^{n-1}$ in each fog node, where \mathcal{F} is the set of fog nodes deployed in the pervasive environment with the size of n . In addition, a system-level hyper-parameter λ is introduced in order to cater to different deployment

requirements of the system, 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 $\mathcal{F} = \{f_1, f_2 \dots f_n\}$ to define the set of fog nodes in the network and \mathbf{v}_{f_i} to represent the voting vector of any participating fog node $\{f_i | \forall f_i \in \mathcal{F}\}$, which can be formulated as:

$$\mathbf{v}_{f_i} = \lambda * \mathbf{d}_{f_i} + (1 - \lambda) * \mathbf{c}_{f_i}, \mathbf{v}_{f_i} \in \mathbb{R}^{n-1}$$

Specifically, each value in vector \mathbf{d} and \mathbf{c} is calculated as the following:

$$\begin{aligned} \mathbf{d}_{f_i} &= [d'_{f_1}, d'_{f_2}, \dots, d'_{f_n}]^T, d'_{f_i} \notin \mathbf{d}_{f_i} \\ d'_{f_n} &= 1 - \frac{s_{f_i, f_n}}{\sum_{\forall f_i \in \mathcal{F}} \sum_{\forall f_k \in \mathcal{F}, f_k \neq f_i} (s_{f_i, f_k})}, d'_{f_n} \in \mathbf{d}_{f_i} \\ \mathbf{c}_{f_i} &= [c'_{f_1}, c'_{f_2}, \dots, c'_{f_n}]^T, c'_{f_i} \notin \mathbf{c}_{f_i} \\ c'_{f_n} &= \frac{c'_{f_n}}{\sum_{\forall f_j \in \mathcal{F}} (c'_{f_j})}, c'_{f_n} \in \mathbf{c}_{f_i} \end{aligned}$$

where the value $d'_{f_n} \in \mathbf{d}_{f_i}$ 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} . When it comes to calculating the vector of computational power, from node f_i ’s perspective, the proportional computing power of node f_i is simply the ratio between its power and the reported total computing power of all fog nodes under this pervasive environment.

After calculating the respective voting vector in each fog node, these nodes then communicate with their peers to retrieve other voting information. We adopt the **Coombs Rule** [8] to find out the winning candidate by ruling out other competing nodes.

B. Collaborative Learning Algorithm

A two-stage collaborative learning algorithm is introduced here to allow the fog nodes to train the deep learning model together, consisting of the model initialization phase and the model training phase.

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 due to limited local data pool, the coordinator node firstly trains the preliminary model under a small number of epochs (around 50) using its local private data, then it iteratively sends the model to other fog nodes to ensure the preliminary model is initiated properly by drawing knowledge from multiple local data distributions managed by different fog nodes. This model initialization process resembles the well-known transfer learning in which a trained model from one task is used as the starting point of another task. Yet, our goal here is to ensure the preliminary model generalizes as much data information from different parties as possible to facilitate the collaborative training phase

Algorithm 1 Collaborative Learning Algorithm

Input: coordinator node f_i ($f_i \in \mathcal{F}$), a set of other fog nodes $\mathcal{S} = \{f_j \mid j \neq i, f_j \in \mathcal{F}\}$, small epoch number ξ and the positive training rate λ

Phase 1: *Preliminary Model Training*

Output: \mathcal{M} : preliminary model trained on each node by ξ epochs

- 1: $\mathcal{M} \leftarrow$ start at the coordinator node f_i to initiate the model by training ξ epochs
- 2: **for** each $f_j \in \mathcal{S}$ **do**
- 3: compose the local dataset $\mathcal{D}_j \leftarrow \{(x_j, y_j)\} \triangleright x_j$ and y_j represent the local training data and label, respectively.
- 4: $\mathcal{M}_j \leftarrow$ receive the \mathcal{M} from the coordinator f_i
- 5: train the \mathcal{M}_j for ξ epochs on \mathcal{D}_j
- 6: send the \mathcal{M}_j back to f_i and overwrite \mathcal{M}
- 7: **end for**

Phase 2: *Collaborative Model Training*

Output: \mathcal{M}' : generalized model trained collaboratively by all fog nodes

- 1: **repeat**
 - 2: **for** for each $f_j \in \mathcal{F}$ **do**
 - 3: $\mathcal{M}_{f_j} \leftarrow f_i$ sends model \mathcal{M} to each node f_j
 - 4: $\Delta w_{f_j} \leftarrow$ calculate the local gradients based on local dataset \mathcal{D}_j
 - 5: upload the Δw_{f_j} to f_i
 - 6: **end for**
 - 7: update the \mathcal{M}' at $f_i \leftarrow \text{FedSGD}(\{\Delta w_{f_j}\}, \lambda)$
 - 8: distribute the \mathcal{M}' to each participant node to overwrite the local model
 - 9: **until**
-

while alleviating the adverse influence caused by the non-iid data distribution.

After obtaining the preliminary model, the coordinator node distributes the model to all other nodes to kick off the training phase. Every participant node takes the received model as the starting point and calculates the gradients of trainable variables based on the local data. These gradients are then sent back to the coordinator node to be aggregated as the final update of the model through the federated stochastic gradient descent (FedSGD) [9].

Algorithm 1 demonstrates the details of these two phases.

IV. CASE STUDY & DATA COLLECTION

We adopt the user's trajectory prediction in an exhibition hall as a use case pervasive system to demonstrate the merit of our proposed scheme. Specifically, the pervasive system could provision the prediction service on the user's trajectory to alleviate the overly-crowded flow during the exhibition and shows the current status of usage in each space, such as a booth, to the attendees. To realize the trajectory prediction, the attendee 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.

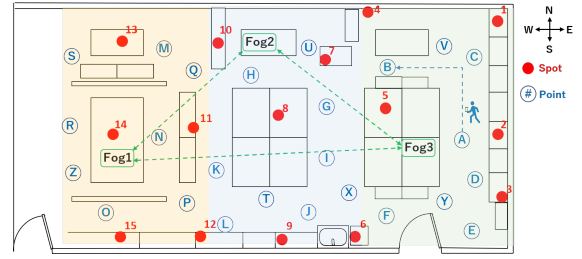


Fig. 1. Deployment layout of fog and sensors

TABLE I
RSSI VALUE RECEIVED BY EACH FOG NODE.

src. \ dst.	fog 1	fog 2	fog 3
fog 1	-	-82.34	-75.04
fog 2	-80.06	-	-77.18
fog 3	-79.51	-77.19	-

To evaluate the performance of our proposed scheme, we setup a wireless sensor network inside an office building as shown in Fig. 1. In this network, 15 sensors (Feasycom FSC-BP103, ID from 1 to 15) work as reference devices that send wireless signals to points (A, B, ..., Z) that user may walk through. At each point, fog node collects RSSI data from all managed sensors for a period of time. There are 3 fog nodes deployed in total as shown in Fig. 1, where each fog node keeps connected with other peers (green dotted arrows) and also manages the sensors placed 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.

V. PRELIMINARY RESULTS

A. Voting Result

Table I demonstrates the received signal value of different fog nodes. Based on the signal value, the distance score, as mentioned in Section III, could be calculated correspondingly. As the deployed fog nodes have the same hardware configurations in this initial work, the computational resources are thus the same, and each of which has the value of 0.3333. Additionally, the voting parameter λ is set to be 0.5, indicating that the distance and computing power are treated equally important at this stage. After the calculation, the fog 3 is elected as the coordinator node according to the voting strategy described in Section III-A.

B. Collaborative Training Result

We conduct the experiment and present the preliminary result in this part. As mentioned above, the algorithm focuses on enabling fog nodes to train the model collaboratively even without the support from the cloud and exposing the local data to others. For the experiment at this stage, we re-use the three fog nodes that are deployed and mentioned at Section IV

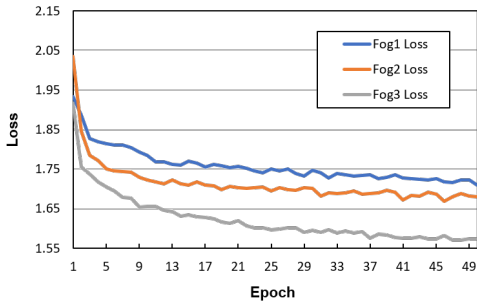


Fig. 2. Training loss for different fog nodes

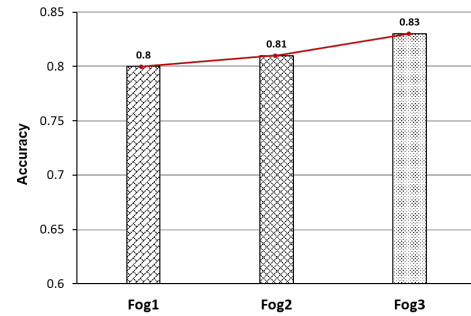


Fig. 3. Accuracy at different fog nodes

to observe the model training process. When it comes to the model structure, we simply adopt the deep neural network with three hidden layers including 64, 128, 32 hidden neurons at each layer, respectively.

Fig. 2 demonstrates the training loss for each fog node in a 50 epochs iteration, and the y axis is scaled down through logarithm for better presentation purpose. It can be clearly observed that fog 3, as the coordinator node, is able to decrease the loss to the lowest compared to what happened in fog 1 and 2. The fog 2, on the other hand, starts with a higher loss value yet achieves a better performance than fog 1. Since the RSSI signal data are only collected in a several weeks time period, all training loss from different fog nodes are able to converge within a short amount of time.

Fig. 3 further illustrates the user trajectory accuracy of the collaboratively trained model based on the local test dataset at each fog node. It is worth pointing out that the model will only be considered making an accurate trajectory prediction if a user moves from point **A** at timestamp t_i to point **B** at timestamp t_{i+1} being predicted correctly at both spots. It also verifies that fog is able to estimate the trajectory of a moving object merely based on the variance of the RSSI signal, which essentially contributes to the continuous advancement of localisation field. Regarding the value shown in Fig. 3, it also reflects the result from Fig. 2 that even though fog 3 achieves the highest accuracy among all, the shared model still performs well on other fog nodes with locally collected data. In addition, the quality of RSSI signal data used for training at each fog node differs due to the real-world physical environment where sensors are deployed. For instance, there are more obstacles like office cabinets (drawn in Fig. 1) in fog 1 managed area and thus causing more interference, which could deteriorate the quality of collected data in that area as well as the accuracy.

VI. CONCLUSION AND FUTURE WORK

In this paper, we present the initial work on using a democratic approach to enable collaborative learning in the fog-enabled pervasive environment. The real-world, in-door fog and sensor deployment are used to verify the effectiveness of our proposed scheme at this stage. It is assured by the preliminary experimental results that an appropriate coordina-

tor node could be elected owing to the practically designed voting strategy, which empowers the training process without the reliance on the cloud server. Additionally, the two-stage collaborative training algorithm could alleviate the impact on no-i.i.d training data issue on each fog node while not exposing the local data to others, and a satisfactory trajectory prediction could be achieved for each participant regardless.

For future work, we will continue to investigate how emerging collaborative deep learning in fog environment could facilitate the development of the pervasive system. For instance, we are currently looking into refining the deep learning model structure and collaborative learning process to make its performance competitive with the centralized model trained at the cloud. 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 different pervasive systems like smart agriculture systems.

REFERENCES

- [1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 961–971.
- [2] D. Varshneya and G. Srinivasaraghavan, "Human trajectory prediction using spatially aware deep attention models," *arXiv preprint arXiv:1705.09436*, 2017.
- [3] S. Memon and M. Maheswaran, "Using machine learning for handover optimization in vehicular fog computing," in *Proceedings of the ACM/SIGAPP Symposium on Applied Computing*, 2019, pp. 182–190.
- [4] A. Diro and N. Chilamkurti, "Leveraging LSTM networks for attack detection in fog-to-things communications," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 124–130, Sep. 2018.
- [5] Z. Shen, T. Zhang, J. Jin, K. Yokota, A. Tagami, and T. Higashino, "ICCF: An information-centric collaborative fog platform for building energy management systems," *IEEE Access*, vol. 7, pp. 40402–40415, 2019.
- [6] R. Faragher and R. Harle, "Location fingerprinting with bluetooth low energy beacons," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2418–2428, Nov 2015.
- [7] P. Davidson and R. Piché, "A survey of selected indoor positioning methods for smartphones," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 1347–1370, Secondquarter 2017.
- [8] A.-H. Mogos, B. Mogos, and A. M. Florea, "A voting approach for comparing several swarm intelligence algorithms," in *Proceedings of the IEEE International Conference on Control Systems and Computer Science*, 2015, pp. 287–293.
- [9] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *Proceedings of Neural Information Processing Systems Workshop on Private Multi-Party Machine Learning*, 2016, pp. 1–5.