Federated Region-Learning: An Edge Computing Based Framework for Urban Environment Sensing

Binxuan Hu, Yujia Gao, Liang Liu, and Huadong Ma
Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia
Beijing University of Posts and Telecommunications, Beijing 100876, China
{hbx200402, gaoyujia, liangliu, mhd}@bupt.edu.cn

***Abstract*—Sparse sensory data caused by insufficient monitoring sites and their incomplete records becomes the main challenge of fine-grained environment sensing. In this paper, we develop a novel inference framework, named Federated Region-Learning (FRL), for urban environment sensing. The proposed framework inherits the basic idea of federated learning, and also considers the regional characteristics during the distribution of training samples so as to improve the inference accuracy. Moreover, we exploit an edge computing architecture to implement the FRL for improving the computational efficiency. We also apply FRL to PM$_{2.5}$ monitoring in Beijing. The evaluation shows that our FRL improves computational efficiency nearly 3 times than centralized training mode and increases accuracy by more than 5% compared with normal distributed training.**

I. Introduction

In the last decades, environment pollution has grown up to be a major problem that influences people's health, especially those in metropolitan cities of the developing world. BBC News Network reported in 2016 that about 5.5 million people die of air pollution in the world every year. Fighting against the urban pollution will require an effort of comprehensive long-term environmental data collection and synthesis that mainly relies on monitoring sites. However, the monitoring sites are usually insufficient such that it is difficult to obtain fine-grained environment status over the whole city. Taking air monitoring in Beijing as an example, only 35 sites were deployed in the 16,000 km^2 area, i.e., 457 km^2 per site. Each monitoring site reports a piece of record per hour which includes 6 kinds of pollutants: PM$_{2.5}$, PM$_{10}$, NO$_2$, CO, O$_3$ and SO$_2$. But the reported records from monitoring sites are incomplete sometimes. According to statistics of one year's records from main monitoring sites in Beijing, there exists about 23% incomplete records in total. Therefore, *the sparse sensory data caused by insufficient sites and incomplete records becomes the main challenge of fine-grained environment sensing.*

Recently, more and more researchers start to exploit the big data technology to solve the above challenge [1], [2], [3], [4]. As shown in Fig. 1, amount of environmental data with different categories is collected, and some advances in machine learning (such as deep learning) are introduced to train inference models. These models, describing the relationship among different data categories, can be used to infer a certain kind of environmental data according to some other kinds of data. For example, the authors of [1] proposed a co-training-based semi-supervised learning approach to infer the fine-granularity air quality according to the AQIs reported by a few air quality monitor stations and four datasets (meteorological data, taxi trajectories, road networks, and POIs) observed in the city. However, in the existing work the models are trained as a centralized form, i.e., all data is centralized for building a uniform model. This centralized training mode entails substantial problems:

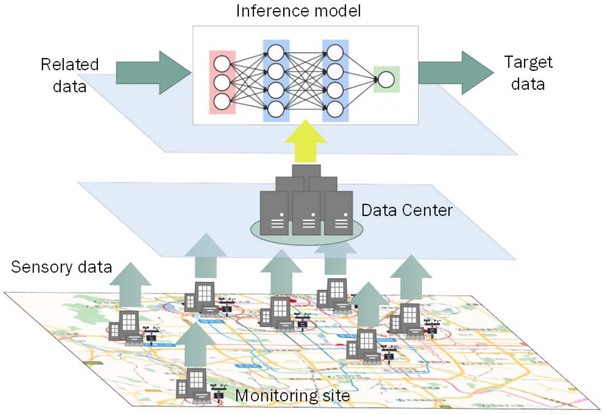

Fig. 1. Centralized training mode for environment monitoring.

- *Computational efficiency*. The commonly used deep learning models (such as CNN and LSTM) are complex and their performance relies on massive training samples. That means for the large urban area, the centralized training mode has low computational efficiency for generating a model over the whole city.
- *Model performance*. Existing research reveals that many kinds of environmental data have considerable spatial variations. For example, Paper [1] finds that the PM$_{2.5}$ reported by stations are quite different sometimes though they are geospatially close. This implies that for the whole city different regions may have different characteristics. A uniform model, which only abstracts the generalities but omits the regional characteristics, will face the performance bottleneck.

To overcome these problems, we develop a new framework named Federated Region-Learning (FRL) for urban environment sensing. Different from the centralized training mode, FRL inherits the basic idea of federated learning which can leave the training data distributed on the mobile devices and

978-1-5386-4727-1/18/$31.00 ©2018 IEEE

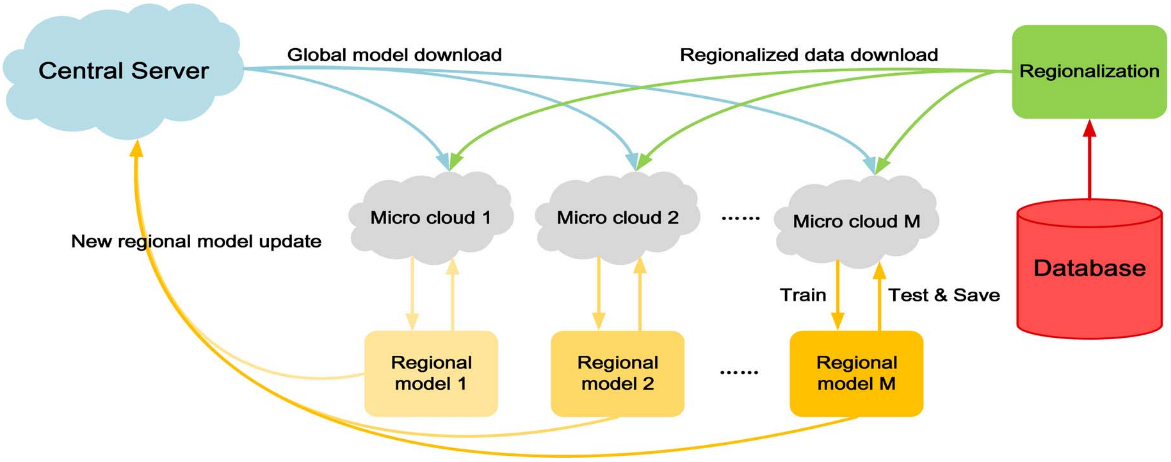

Fig. 2. Architecture of federated region-learning.

learn a shared model by aggregating locally-computed updates [5], [6], [7], [8]. But compared with the classic federated learning model, FRL considers the regional characteristics during the distribution of training samples so as to improve the inference accuracy. We first propose a regionalization method that divides the urban area into a set of regions. A regional model is trained by using samples generated in the corresponding region. By averaging the weights of all the regional models, a global model is generated. Then, the global model is sent back to regional models to continue the next round of training. After several iterations, the regional model with the highest test accuracy is selected as the inference model for the corresponding region. Moreover, from the perspective of model implementation, our FRL is build upon the edge computing architecture [9]. As the carrier of edge computing, a micro cloud is deployed for each region. The micro cloud is mainly responsible for: a) collecting data from each site within its region; b) downloading the global model from central server and training the regional model via local data; c) uploading regional models to the central server.

The contribution of this paper lies in the following aspects:

- Framework. We design an inference framework for environment sensing which combines the ideas of edge computing and distributed deep learning. It provides a complete solution to simultaneously solve the problems of computational efficiency and model performance.
- Model. We propose the concept and the model of federated region-learning which implements a distributed training mode. More importantly, FRL combines regional characteristics and structure generalities of learning networks, which achieves better performance than the global model and separate regional models.
- Application. We apply our proposed model and framework to $PM_{2.5}$ monitoring which is an important application for public health. Moreover, we also extend the two-layer structure of FRL to the multi-layer structure which can be used in broader application scenarios.

The remainder of the paper is organized as follows. Section II presents our method of division and the principle of federated region-learning. Experimental results for division and federated region-learning are reported in section III. A multi-layer structure for our framework is discussed in section IV. Finally, a brief conclusion and future work are provided in section V.

II. Proposed Framework

First, we give a overview of our framework with its work process. Next, we introduce how to divide the dataset from different sources into several regions for establishing regional models. Finally, we explain the working principle of federated region-learning and the method to train the regional models.

A. Overview

As shown in Fig. 2, our framework can be divided into two phases. In the first phase, we evaluate the relationship of each air quality monitoring site based on distance and pollution, and then divide the sites into regions by a suitable algorithm with the degree of association between sites. In the evaluation step, we analyze the change trend of $PM_{2.5}$ at each monitoring site as a reference standard for correlation between sites, and calculate the distance between sites according to their coordinates; in the division step, we use the Girvan-Newman(GN) [10] algorithm to make the strongest correlation between sites in each region.

In the second phase, the micro cloud first downloads the global model from central server and collect the data within its own region, then ready to training regional model. The training process is illustrated in the Fig. 2. Each micro cloud uses the regional data to train a regional model based on the global model. After testing and saving the best model, the micro cloud uploads the best model to the central server to obtain a new global model. Finally, the global model is once again distributed to each micro cloud for the next iteration. our framework improve the computational efficiency of the model training process and provide a more accurate real-time air quality value.

B. Regionalization

In this part, we discuss a clustering algorithm for regionalization of air quality monitoring sites. In practice, the distribution of monitoring sites is scattered. This means that the air quality monitoring sites may be redundant in some regions, but lacking in other regions. In order to better extract the regional characteristics of data and manage data more efficiently, before training regional models, we first divide the sites reasonably to make our regional models work better. The geographical location and other relevant training data of different sites are considered as the basis for division. Our clustering algorithm can be divided into two phases: 1) construction of weighted network which is based on sites' location and their relevance; 2) division of weighted network. In the first step, we take λ sites as vertexes, and the ratio of correlation between changes in air quality to the distances between each site are regarded as the edge weight to construct weighted network. In the division step, we use the Girvan-Newman algorithm to divide the weighted network and obtain the regional division of sites.

First of all, We define a connected network $G = (V, E)$. In the network, the V represents the datasets of vertexes, and the E represents the Zero-one matrix of the network, if vertex i and vertex $j(0 < i, j \leq \lambda)$ are connected, $E_{i,j} = 1$, otherwise $E_{i,j} = 0$. We also define a weighted matrix A, which has the same size as E, to represent weights of the edge between vertex i and vertex j in the network. Normally, we use the Pearson correlation coefficients [11] to express the relevance between sites. The weight $A_{i,j}$ of the edge for vertex i and vertex j is the ratio of the Pearson correlation coefficients $r_{i,j}$ about the value of air quality data to the distance between two vertexes. Where $A_{i,j}$ and $r_{i,j}$ are defined below:

$$A_{i,j} = \frac{r_{i,j}}{d_{i,j}}, \tag{1}$$

$$r_{i,j} = \frac{Cov(V_i, V_j)}{\sqrt{Var[V_i]Var[V_j]}}, \tag{2}$$

where V_i denotes the daily average air quality data of each vertex, $d_{i,j}$ is the distance between vertex i and vertex j, $Cov(.)$ defines covariance of the two variables, and $Var(.)$ denotes the variance. In GN algorithm, there is a core concept of modular function Q ($Q \in [0, 1]$), which means that the network partition is the best when the modular Q function is maximum. In normal weighted networks, the Q function can be expressed as:

$$Q = \frac{1}{2M}\sum_{i,j}[(A_{i,j} - \frac{k_i k_j}{2M})\delta(\sigma_i, \sigma_j)], \tag{3}$$

where δ is a membership function. When the vertex i and vertex j belong to the same region, the membership function δ is 1, otherwise it is equal to 0. M is the sum of the weights of the edges in the network, which is defined as $M = \frac{1}{2}\sum A_{i,j}$. k_i is the degree of vertex i, which is calculated by summing the i-th row of the connectivity matrix.

Algorithm 1 Division of monitoring sites

```
Require: Daily average PM$_{2.5}$ data of each vertex $V_i$ Zero-one matrix $E_{i,j}$ and distance $d_{i,j}$
Ensure: Divided zero-one matrix $P_{i,j}$
Calculate $M = \frac{1}{2}\sum A_{i,j}$
Calculate $r_{i,j}$ from Eq. 3
Initialize $A_{i,j}$ using Eq. 2 and let $T = 0$
Back up $E_{i,j}$ into $P_{i,j}$
while exist any $E_{i,j} \neq 0$ do
    $T = T + 1$
    Initialize each $b_{i,j} = 0$ and $\varphi_{i,j} = 0$
    for each edge $E_{i,j} \neq 0$ do
        Calculate its edge betweenness $b_{i,j}$
        $\varphi_{i,j} = b_{i,j}/A_{i,j}$
    for each edge $E_{i,j}$ with the maximum ratio $\varphi_{i,j}$ do
        Record this edge $E_{i,j}$ in $R_T$
        Remove this edge $E_{i,j} = 0$
    Calculate the modular function $Q_T$ of the network from Eq. 4
for $\tau = 1$ to $T$ do
    Find maximum $Q\tau$ value and record this $\tau$
for $\eta = 1$ to $\tau$ do
    Remove the edges $R\eta$ in $P_{i,j}$
Return $P_{i,j}$
```

To cluster the sites, the edge betweenness $b_{i,j}$ corresponding to each connected edges $E_{i,j}$ in the Zero-one matrix of unweighted network should be calculated firstly. Next the edge betweenness $b_{i,j}$ is divided by the weight $A_{i,j}$ of the corresponding edge with the result $\varphi_{i,j}$. Then, we remove the edge $E_{i,j}(E_{i,j} = 0)$ with the maximum ratio $\varphi_{i,j}$, and calculate the modular function Q of the network. When there are multiple edges with the maximum ratio $\varphi_{i,j}$ in the calculation, these edges should be removed at the same time, and record Q value with the removed edges at T-th partition. The above process is reiterated until all the edges in the network have been removed. Finally, we find the serial number T with the maximum Q value, and then remove the edge from the original situation to T-th partition. The rest matrix is the final connected matrix with the divided regions. Complete pseudo-code is given by Algorithm 1.

C. Federated Region-Learning

Federated region-learning is based on distributed learning and edge computing, and its weight updating method is similar to the principle of federated learning. However, federated region-learning is focused on the lower layer model such as a regional model, not the central model on the top. We test and save the lower layer model on regional micro cloud before updating the model to center server, and the best model will be the regional model to infer the air quality data for the region. This means each region not use the same global model to obtain the air quality data, but uses its own extra trained model to infer the air quality data. Obviously, the regional model has more regional characteristics of the region, so that the accuracy of the model for the region will inevitably increase accordingly.

In order to implement this method, we construct a novel architecture as shown Fig. 2. In this architecture, the data of monitoring sites are converted into M regional micro clouds by GN algorithm. And every micro cloud $i (i \in \{1, 2, \ldots, M\})$ has its own regional model R^i with its partitioned dataset S_i. In the beginning of each communication round t, the central server distributes current global model weights W_t. Micro clouds train new models R^i_{t+1} using local datasets S_i based on the global layer model weights W_t. After Training, each cloud compare the accuracy of current regional model R^i with the accuracy of the new models R^i_{t+1}, and take the better model as the new regional model R^i. Then, the new regional model R^i is sent to the central server for next federated. While federating all regional model weights set R on communication t, the model weights can be updated by following formula:

$$W_{t+1} = \sum_{i=1}^{M} \frac{k_i}{n} R^i, \tag{4}$$

where $k_i = |S_i|$ and $n = \sum |S_i|$. New global model W_{t+1} is obtained by averaging entire regional model set R. Finally the new global model is immediately distributed to every micro cloud again and process iterates. In addition, there are three key hyper-parameters to control the amount of computation in federated region-learning: $C (C \in [0, 1])$, the fraction of clients which need to train the new regional models on each round; E, the number of local epoch each micro cloud trains its local dataset on a round; and B, the size of local minibatch used for the micro cloud updates. For federated region-learning, the meaning of the parameter E is special. The regional model is the best model which taking E epochs training on the micro cloud. Thus the size of E will determine the degree of influence of the global model and the regional characteristics. The larger E is, the more obvious the regional characteristics of the model are, and the weaker the connection with the global model is; the smaller E is, the more fuzzy the regional characteristics of the model are, and the stronger the connection with the global model is. Therefore, the adjustment of parameter E is particularly important.

III. Application: PM2.5 Inference Using FRL

In order to enable experiments to reflect the quality of regional models, we build a system which can infer the real-time $\text{PM}_{2.5}$ categories of its region. The system is constructed on a 2-layer Long Short-Term Memory Network (LSTM) [12] with 128 hidden cells for each layer. The LSTM network trains model for inferring the current $\text{PM}_{2.5}$ categories based on the previous 48 hours weather and air-pollution data (W&A data in short). In the LSTM network, the loss function is categorical cross-entropy, and the optimizer is Adam [13] whose initial learning rate is set to $l = 0.001$. To better compare performance between different methods, we also establish a centralized training mode system and a standard federated learning system as the baseline models, which have the LSTM network with the same structure and hyper-parameters. As for the dataset for experiments, we retrieve more than 100,000 W&A data with $\text{PM}_{2.5}$ categories on the web. These W&A data is generated by weather/air-quality monitoring sites and released by the official websites per hour. The W&A dataset contains 12 categories related data of $\text{PM}_{2.5}$, such as temperature, humidity, NO_2, and SO_2 etc. During the training process, we choose the data for the first 25 days of each month as the training set and the data for the last 5 days as the test set.

```
Algorithm 2 Federated region-learning
Require: Regional datesets $S_i$
Ensure: Best regional model $R^i_{t+1}$
function ServerRun( )
    Initialize $W_0$
    for each round $t = 0, 1, 2, \cdots$ do
        $F_t \leftarrow$ random set of $M \bullet C$ clients
        for each regional server $i \in F_t$ in parallel do
            $R^i_{t+1} \leftarrow$RegionalServerUpdate$(i, W_t)$
        $W_{t+1} \leftarrow \frac{k_i}{n} R^i_{t+1}$

function RegionalServerUpdate$(i, R_t)$
    $D \leftarrow$ split $S_i$ into batches of size $B$
    for each local epoch from 1 to $E$ do
        for batch $b \in D$ do
            $R_t \leftarrow R_t - l\nabla(w; b)$ //$l$ is leaning rate
    Test and save $R^i_{t+1}$in $S_i$
    Return $R^i_{t+1}$
```

In the experiments, our first objective is to evaluate the effect of the dataset division and choose the best division result to determine the deployment and data of the micro cloud. Moreover, our more important objective is to use the W&A dataset to verify whether the FRL can improve the efficiency of training model and the inference accuracy of $\text{PM}_{2.5}$ categories. In order to obtain better models, We also investigate different hyper-parameters combination of federated learning and FRL. The evaluating indicators are as follows: (a) the highest average accuracy of $\text{PM}_{2.5}$ categories classification for each regions; (b) communication rounds (each batch update requires a communication rounds) of training the best model. At the end of the experiment, we also discuss the multi-layer structure and proved its feasibility.

A. Regionalization Evaluation

At the beginning of this experiment, the W&A dataset is distributed on a total of 13 sample monitoring sites in Beijing. The daily average $\text{PM}_{2.5}$ of 15 days before and after sampling are selected as the principal data of this site, and the distances between each site are calculated based on their coordinates. Then we set up the network with the monitoring sites as the vertexes and the ratio of the $\text{PM}_{2.5}$ correlation coefficient

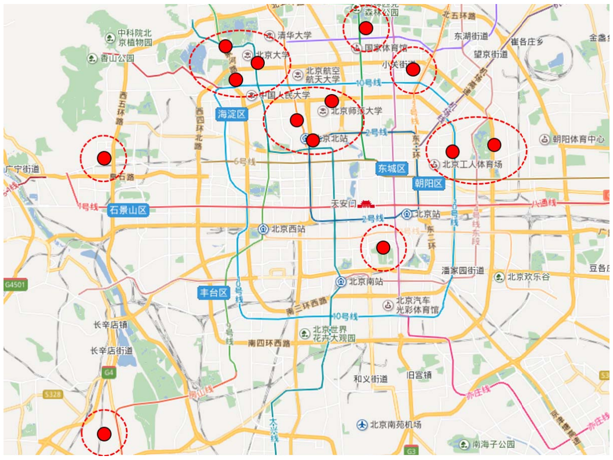

Fig. 3. Result of division.

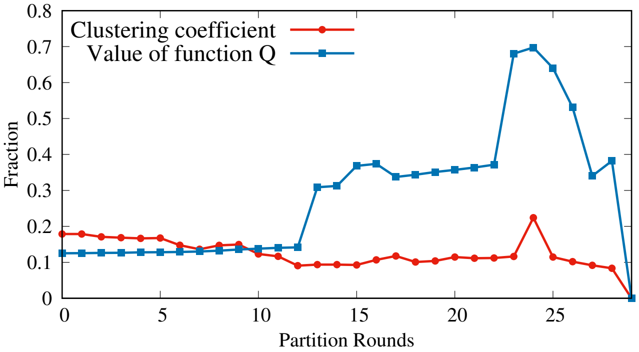

Fig. 4. Curve of the average clustering coefficient and the Q value.

to distance between each point as the weights of the edges. Because the monitoring sites are in the same city, their PM$_{2.5}$ values are very similar and the distances from site to site are also very short. Hence we only connect two vertexes(i.e. $E_{i,j} = 1$) of correlation coefficient $r_{i,j} > 0.95$ and distance $d_{i,j} < 10$ kilometers while building the weighted network, otherwise $E_{i,j} = 0$. During the process of dividing the sites, our program calculate the maximum value of function Q was 0.68, meanwhile 13 monitoring sites were divided into 8 regions. Fig. 3 summarises our regionalization specific results, and it is easy to see the result of the division is very regional. In the actual network, value of function Q is usually between 0.3-0.7, and the probability of the value of function Q greater than 0.7 is very small. Thus, the effect of our regionalization results is desirable as the value of function Q was very close to 0.7.

In addition, we introduce the clustering coefficient L_R proposed by [14] to evaluate our segmentation results. According to the definition of clustering coefficient, it can measure the degree of contact in the local region. It is defined by the formula:

$$L_R = \frac{1}{n}\sum_{i=1}^{n}\frac{\sum_{jk} A_{ij}A_{jk}A_{ki}}{\max_j\{A_{ij}\}\sum_{jk} A_{ij}A_{ki}}, \tag{5}$$

where n is the number of monitoring sites in the region, and $\max_j A_{ij}$ means the maximum weights of the adjacent edges of vertex i. The clustering coefficient of the region is positively correlated with the connectivity of the region. Fig. 4 shows the curve of the average clustering coefficient of each regions and the value of function changing with the dividing times. It should be noted that the reason why the clustering coefficient is relatively low is because we calculate the average value for each region. In some regions, there is just one vertex, and the clustering coefficient is zero, which reduces the average value of the clustering coefficient. In Fig. 4, we observes that the clustering coefficient and the value of function Q reachs the maximum value at the 13-th division. This indicates that the degree of clustering of each region is the largest and the division effect is best, when the Q value reache the maximum value.

TABLE I
BEST AVERAGE ACCURACY OF DIFFERENT PARAMETERS COMBINATION.

		Federated Learning				Federated Region-Learning			
B		96		196		96		196	
C	E	ACC	CR	ACC	CR	ACC	CR	ACC	CR
1	1	72.29	656	72.27	336	78.92	582	77.82	494
1	3	73.32	205	72.53	252	79.89	640	80.36	252
1	5	73.43	205	72.31	105	80.18	549	79.59	374
0.3	1	74.71	574	75.64	861	80.47	1777	79.01	1078
0.3	3	75.60	1066	74.62	735	81.06	806	80.56	651
0.3	5	**75.64**	**492**	**76.25**	**756**	**81.09**	**820**	80.25	385
0.5	1	75.29	1107	73.89	903	79.54	1000	77.25	722
0.5	3	74.12	738	76.18	693	79.81	615	**81.05**	**533**
0.5	5	75.04	738	75.69	378	80.24	607	80.18	395

TABLE II
COMPARISON BETWEEN THE CENTRALIZED TRAINING, FEDERATED LEARNING AND FEDERATED REGION-LEARNING.

	Centralized training		Federated Learning		Federated Region-Learning	
B	ACC	CR	ACC	CR	ACC	CR
96	80.27	2560	75.64	492	**81.09**	820
196	79.64	1440	76.25	756	**81.05**	533.4

B. Federated Region-Learning Evaluation

In this part, we evaluate the efficiency and effectiveness of federated region-learning by comparing the centralized training mode, standard federated learning and federated region-learning. The result of regionalization is used to converge the W&A data as the dataset of each region. Then we test the data separately for each region, and pick up the average value as the basis for evaluation. After eliminating invalid and default data, each district has about 4000 data for training and 1000 data for testing approximately. Moreover, In order to ensure fairness, we train a fixed initial model with 35.34% accuracy for each training method at the beginning.

In federated learning and federated region-learning, we try various combinations of hyper-parameters by using $E \in \{1,3,5\}$, $C \in \{0.3, 0.5, 1\}$ and $B \in \{96, 196\}$. Table 1 shows the accuracy and communication rounds of each case within $t < 50$. Because the size of B can directly influence

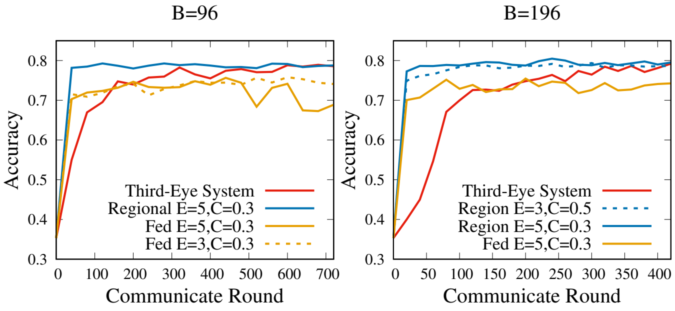

Fig. 5. Test set accuracy VS. communication rounds.

the calculation of communication rounds, we put $B = 96$ and $B = 196$ into two cases to discuss when compare their communication costs. As a result, federated region-learning achieves highest accuracy 81.09% within 820 communication rounds when $E = 5$, $C = 0.3$ and $B = 96$, and the best performance setting of federated learning is that $E = 5$, $C = 0.3$ and $B = 196$. Certainly, we cannot ignore that standard federated learning only used 492 communication rounds to achieve 75.64% accuracy when $E = 5$, $C = 0.3$ and $B = 96$. Similarly, we also take the batch size equal to 96 or 196 in centralized training system.

Table 2 lists the results for the highest accuracy models with $B \in \{96, 196\}$. The results show that the regional models achieved the higher accuracy than the centralized training system model and federated learning model in less communication rounds. In the case of $B = 96$ and $B = 196$, federated region-learning needs 3.12 times and 2.7 times less communication rounds than the centralized training model to obtain the better accuracy models, and obtains 5.45% and 4.8% higher accuracy models than federated learning model within small difference communication rounds. To have a better understanding of prediction performance of all the models w.r.t communication rounds, we further plot the curves of the accuracy of each best model with the change of communication round. Fig. 5 demonstrates that the federated region-learning performs better than other methods. The federated region-learning can produce a dramatic decrease in communication costs with a higher best accuracy. These suggest that our idea of the federated region-learning is more efficient and effective than the centralized training system and standard federated learning in training the regionalization model by LSTM. And federated region-learning can well extract the regional characteristics and obtain a higher accuracy.

IV. Discussion: Multi-Layer Structure

To develop our method of federated region-learning to adapt to more situations which need to train the models that contain a wider area. However, wider areas mean more data, more models, and larger parameters. These will increase the cost of training and reduce the efficiency of transmission. We discuss constructing a multi-layer structure to obtain larger area models, not just the district models, but also the city models or even country models etc. These kinds of models can be trained in multiple low layers with a acceptable communication efficiency and model accuracy. Each low layer contains several micro clouds with several regional models. For example, we can build a 3-layers structure which have two low layers to extract the features of each district and community. Each district and community has its own micro cloud to train or federate the regional model. We believe this method can effectively avoid problems such as excessive transmission of model data and inefficient training.

In a N-layers structure, we can consider that every weight update of models is similar to the process of averaging the regional model weights for updating global model in federated region-learning. Thus, to average all the weights of a N-layer structure, we have:

$$\begin{aligned} W^j_{t+1,2} &= \sum_{v=1}^{V_{1,j}} \frac{k^1_{j,v}}{k^1_j} W^v_{t+1,1}, \\ W^j_{t+1,3} &= \sum_{v=1}^{V_{2,j}} \frac{k^2_{j,v}}{k^2_j} W^v_{t+1,2}, \\ &\vdots \\ W_{t+1,N} &= \sum_{v=1}^{V_{N,j}} \frac{k^N_{j,v}}{k^N_j} W^v_{t+1,N-1}. \end{aligned} \tag{6}$$

Note that the N-th layer is for central server, and $W^j_{t+1,i}$ is the weight of j-th model on $i (i \in \{2, \ldots\})$-th layer. Where $V_{i,j}$ is the number of sub-clients or sub-servers which belong to the j-th lower model, and $\frac{k^i_{j,v}}{k^i_j}$ is the radio of each sub-dataset size to the size of all lowery layer datasets. Since $k^i_{J,v} W^v_{t+1,i} = \sum_{j=1}^{J} k^{i-1}_j W^j_{t+1,i-1}$ where J is the number of middle model in i-th layer, the above formula can be put into the follow. To integrate above formulas, we reduces Eq. (6) to:

$$\begin{aligned} W_{t+1,N} &= \frac{\sum_{j=1}^{J} \sum_{v=1}^{V_{1,j}} k^1_{v,j} W^v_{t+1,1}}{\sum k^1_j} \\ &= \sum_{m=1}^{M} \frac{n_m}{n} W^m_{t+1}. \end{aligned} \tag{7}$$

The formula for weight updating equates to Eq. (4). In conclusion, for the i-th layer, the weight of j-th lower layer model is averaged according to the follows :

$$W^j_{t+1,i} = \sum_{v=1}^{V_{i,j}} \frac{k^i_{j,v}}{k^i_j} W^v_{t+1,i-1}. \tag{8}$$

The weight updating formula of multi-layer structure is deduced. It does not affect the result of 2-layer federated region-learning while training the low layer models. If we train the low layer models by using a multi-layer structure, we can compute the weights of each models by Eq. (8). Also, it is important to note the setting of hyper-parameters$E$, B, and C. The parameter E and B is just valid only when the micro clouds of lowest layer are training the model, and is meaningless in other layer. As for parameter C, we need to adjust it at every layer.

V. Conclusion and future work

In this paper, we proposed the concept and the model of federated region-learning which combines regional characteristics and structure generalities of learning networks. Based on that, we exploited an edge computing architecture to develop an inference framework for urban environment sensing which can simultaneously solve the problems of computational efficiency and model performance. We also extended the two-layer structure of FRL to the multi-layer structure for broader application scenarios. In the future, we will complete the multi-layer structures of FRL and use it to solve the distributed learning problem in more application fields.

VI. Acknowlegements

This work is supported by National Key R&D Program of China No. 2017YFB1003000, the National Natural Science Foundation of China under No. 61332005 and No. 616320081, the Funds for International Cooperation and Exchange of NSFC under Grant No. 61720106007, the Fundamental Research Funds for the Central Universities (No. 2017RC38), and the 111 Project (B18008).

References

[1] Y. Zheng, F. Liu, and H. Hsieh, "U-Air: When Urban Air Quality Inference Meets Big Data," in *Proc. of KDD2013*, August 11-14, 2013, Chicago, Illinois, USA.

[2] L. Liu, W. Liu, Y. Zheng, H. Ma, and C. Zhang, "Third-Eye: A Mobilephone-Enabled Crowdsensing System for Air Quality Monitoring," Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 2, no. 1, pp. 1C26, Mar. 2018.

[3] Zhang, Zheng, et al. "Outdoor Air Quality Level Inference via Surveillance Cameras." *Mobile Information Systems*,2016,(2016-6-1) 2016(2016):1-10.

[4] Zhengxiang Pan, Han Yu, Chunyan Miao, and Cyril Leung. "Crowdsensing Air Quality with Camera-Enabled Mobile Devices." in *Proc. of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, California, USA. 2017 (pp. 4728-4733).

[5] McMahan, H. Brendan, et al. "Communication-efficient learning of deep networks from decentralized data." *arXiv preprint arXiv:1602.05629* (2016)..

[6] Koneny, Jakub, et al. "Federated optimization: Distributed machine learning for on-device intelligence." *arXiv preprint arXiv:1610.02527* (2016).

[7] Suresh, Ananda Theertha, et al. "Distributed mean estimation with limited communication." *arXiv preprint arXiv:1611.00429* (2016).

[8] Arjevani, Yossi, and Ohad Shamir. "Communication complexity of distributed convex learning and optimization." *Advances in neural information processing systems.* 2015.

[9] Shi, Weisong, et al. "Edge computing: Vision and challenges." *IEEE Internet of Things Journal* 3.5 (2016): 637-646.

[10] Girvan, Michelle, and Mark EJ Newman. "Community structure in social and biological networks." in *Proc. of the national academy of sciences* 99.12 (2002): 7821-7826.

[11] Benesty, Jacob, et al. "Pearson correlation coefficient." *Noise reduction in speech processing.* Springer Berlin Heidelberg, 2009. 1-4.

[12] Hochreiter, Sepp, and Jrgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.

[13] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980* (2014).

[14] Holme, Petter, et al. "Korean university life in a network perspective: Dynamics of a large affiliation network." *Physica A: Statistical Mechanics and its Applications* 373 (2007): 821-830.