Blockchain-assisted Ensemble Federated Learning for Automatic Modulation Classification in Wireless Networks

Umer Majeed, Choong Seon Hong
Department of Computer Science and Engineering,
Kyung Hee University, South Korea
Email: {umermajeed,cshong}@khu.ac.kr

Abstract

Automatic modulation classification (AMC) is one of the critical tasks in Software-defined radio (SDR). Deep learning enabled automated cognitive radio processes have gained immense popularity due to high performance without the need for explicit feature search. This paper designed the ensemble federated learning scheme to train high quality deep learning model for AMC on decentralized data. A blockchain network has been used to train the base federated models while the ensemble learning is performed on a server using Tensorflow. Simulation results show that the ensemble federated model outperforms the base federated models.

I. Introduction

The technology of Software-defined radio (SDR) has brought a revolution in communication systems by enabling dependent channel adaption to utilize scarce resources efficiently. Automatic modulation classification (AMC) is an important task in cognitive radio to automatically recognize the modulation format of the unknown signal. Once the signal detection process is completed, AMC is performed before demodulation [1]. The researchers are exploiting the high performance of deep learning technology to enhance wireless communication systems. With deep learning techniques such as convolutional neural networks, cognitive radio tasks such as AMC can be done without the need for explicit feature search.

Federated Learning enables to train the high-performing shared machine learning model on the decentralized data. Ensemble Learning is an efficacious method to boost performance and stability in comparison to the stand-alone single classifier. In this paper, We designed an ensemble federated learning scheme for automatic modulation classification. The contribution of this study is summarized as follows:

- We trained five base federated learning models using the same local datasets on three different signal processing units for signal modulation classification.
- The trained federated learning models are later ensembled using simple majority voting rule to build ensemble federated model.
- The simulation results show that the ensemble federated model has improvised accuracy compared to base federated models.

The remaining of the paper is organized as follows: Section II illustrates the system model for signal modulation classification under ensemble federated settings. Section III formulate the ensemble federated learning problem. Section IV briefly describe the dataset used and implementation details in our study. Section V presents the simulation results and Section VI concludes our paper.

II. System Model

The system model consists of three signal processing units A, B, C having D_a, D_b, D_c as the datasets respectively. As per the EFLChain framework [2] specification, a blockchain network N_B provides federated learning model training service for multiple global models. A server S is available to ensemble the federated learning models extracted from the EFLchain framework.

III. Problem formulation

The signal proccessing units A, B, C train 5 Convolutional Neural Network (CNN) models $(M_1, M_2, M_3, M_4, M_5)$ through federated learning using datasets D_a, D_b, D_c respectively on the blockchain network N_B as specified in

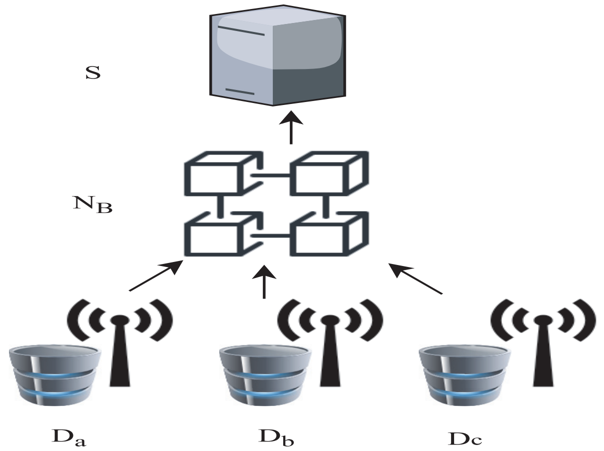

Fig. 1. System Model

EFLchain framework. The server S ensemble the trained federated models to make ensemble federated model M_E using simple majority voting rule.

Let C be the set of classes labels, and $\hat{y}_1, \hat{y}_2, \hat{y}_3, \hat{y}_4, \hat{y}_5 \in C$ be the predicted output of M_1, M_2, M_3, M_4, M_5 of data point x in feature space $\mathcal{X}$ respectively, then the output of M_E is

$$J = \text{argmax}_{j \in \{1,2,\ldots,C\}} \sum_{t=1}^{5} d_{t,j}, \quad J \in C \tag{1}$$

Where

$$d_{t,j} = \begin{cases} 1 & \text{if } \hat{y}_t = j \\ 0 & \text{if } \hat{y}_t \neq j \end{cases}. \tag{2}$$

IV. Dataset

A. Dataset Details

We used GNU Radio (RML2016.10a) dataset [3] for signal modulation classification. The dataset has labeled raw IQ data for 11 different modulation techniques such as 8PSK, AM-DSB, AM-SSB, BPSK, CPFSK, GFSK, PAM4, QAM16, QAM64, QPSK, and WBFM.

B. Splitting

We take 25% of the dataset as a validation dataset D_V. While the rest of the 75% dataset is split equally between each of 3 signal processing units as (D_a, D_b, D_c). D_V is assumed to be publicly available without privacy instigation so that all the designed models can be validated on it.

V. Simulation Results

We used the EFLchain framework for the training of each global base model. EFLchain assigns a separate channel for the training of each global base model. Each block in a channel bundles the local model updates for a particular global iteration. Where the local model updates are aggregated using Federated averaging (FedAvg) [4] in the global model state trie at each channel after each global iteration [5]. The Hyperledger Fabric [6] is used as the underlying blockchain platform. The FedAvg process is implemented using the chain code. Whereas, The global model state trie is implemented in the LevelDB database. The detailed implementation of the EFLchain is out of the scope of this paper.

Table. I, Table. II and Table. III shows the models' architecture. Where, ZP2D is ZeroPadding2D layer.

TABLE I
Model Architecture for M_1

M_1		
Layer	**Value**	**Activation**
Input	(2,128)	-
Reshape	(2, 128, 1)	-
ZP2D	(0,2)	-
Conv2D	128 (1,3)	Relu
ZP2D	(0,2)	-
Conv2D	32 (2,3)	Relu
Flatten	-	-
Dense	64	Relu
Dense	11	Softmax

The base models are later ensembled using tensorflow [9] with keras library [10] on Google Colaboratory [11] platform.

The training accuracy of the ensemble federated model M_E can not be measured because of its ensembling nature and non-availability of training data at the server S. However, the validation accuracy for all federated models and ensemble federated model is shown in Fig. 2.

The performance metrics of all the models are depicted in

TABLE II
Model Architecture for M_2 and M_3

M_2			M_3 (VT-CNN2) [7], [8]		
Layer	**Value**	**Activation**	**Layer**	**Value**	**Activation**
Input	(2,128)	-	Input	(2,128)	-
Reshape	(2, 128, 1)	-	Reshape	(2, 128, 1)	-
ZP2D	(0,2)	-	ZP2D	(0,2)	-
Conv2D	192 (1,3)	Relu	Conv2D	256 (1,3)	Relu
ZP2D	(0,2)	-	ZP2D	(0,2)	-
Conv2D	64 (2,3)	Relu	Conv2D	80 (2,3)	Relu
Flatten	-	-	Flatten	-	-
Dense	128	Relu	Dense	256	Relu
Dense	11	Softmax	Dense	11	Softmax

TABLE III
MODEL ARCHITECTURE FOR M_4 AND M_5

M_4			M_5		
Layer	**Value**	**Activation**	**Layer**	**Value**	**Activation**
Input	(2,128)	-	Input	(2,128)	-
Reshape	(2, 128, 1)	-	Reshape	(2, 128, 1)	-
ZP2D	(0,2)	-	ZP2D	(0,2)	-
Conv2D	64 (1,3)	Relu	Conv2D	128 (1,3)	Relu
Dropout	0.2	-	Dropout	0.2	-
ZP2D	(0,2)	-	ZP2D	(0,2)	-
Conv2D	64 (2,3)	Relu	Conv2D	128 (2,3)	Relu
Dropout	0.2	-	Dropout	0.2	-
ZP2D	(0,2)	-	ZP2D	(0,2)	-
Conv2D	64 (1,3)	Relu	Conv2D	64 (1,3)	Relu
Dropout	0.2	-	Dropout	0.2	-
Flatten	-	-	Flatten	-	-
Dense	64	Relu	Dense	128	Relu
Dense	11	Softmax	Dense	11	Softmax

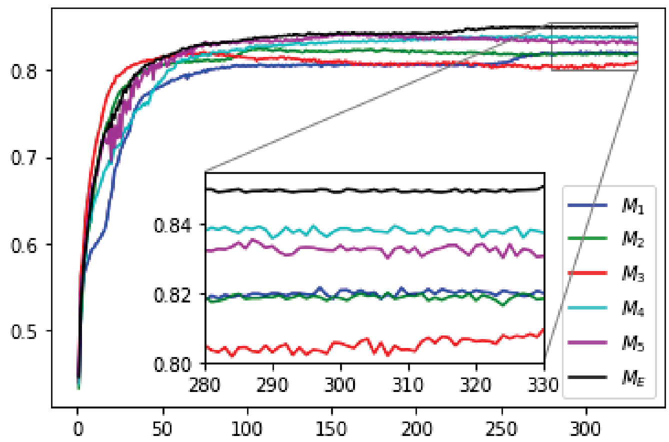

Fig. 2. Validation Accuracy

Table. IV. The ensemble federated model M_E has improved performance relative to base federated models.

TABLE IV
PERFORMANCE EVALUATION ON VALIDATION DATASET D_V

	Precision	Recall	F-1	Auccuracy
M_1	0.93	0.82	0.83	0.82
M_2	0.89	0.82	0.83	0.82
M_3	0.86	0.81	0.82	0.81
M_4	0.96	0.84	0.85	0.84
M_5	0.97	0.83	0.85	0.83
M_E	**0.97**	**0.84**	**0.85**	**0.85**

VI. CONCLUSION

Deep learning-based Automatic modulation classification (AMC) has reaped extensive adoration in automated cognitive radio operation. In this paper, we designed and implemented the ensemble federated learning scheme for AMC. A blockchain network is used to train the base federated models while the ensemble learning is performed on a server. Simulation results show that the ensemble federated model transcends the base federated models.

ACKNOWLEDGMENT

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-01287, Evolvable Deep Learning Model Generation Platform for Edge Computing) *Dr. CS Hong is the corresponding author.

REFERENCES

[1] J. Jagannath, N. Polosky, D. O'Connor, L. N. Theagarajan, B. Sheaffer, S. Foulke, and P. K. Varshney, "Artificial Neural Network Based Automatic Modulation Classification over a Software Defined Radio Testbed," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.

[2] U. Majeed and C. S. Hong, "EFLChain: Ensemble Learning via Federated Learning over Blockchain network: a framework," *Proc. of the Korean Information Science Society*, pp. 845–847, 2019.

[3] "Deepsig Dataset: RADIOML 2016.10a." [Online]. Available: https://www.deepsig.io/datasets

[4] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," 2016.

[5] U. Majeed and C. S. Hong, "FLchain: Federated Learning via MEC-enabled Blockchain Network," in *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2019, pp. 1–4.

[6] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*. ACM, 2018, p. 30.

[7] T. J. O'shea and N. West, "Radio Machine Learning Dataset Generation with GNU Radio," in *Proceedings of the GNU Radio Conference*, vol. 1, no. 1, 2016.

[8] M. Sadeghi and E. G. Larsson, "Adversarial Attacks on Deep-Learning Based Radio Signal Classification," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 213–216, 2019.

[9] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[10] "Keras." [Online]. Available: https://www.tensorflow.org/guide/keras

[11] "Google colaboratory." [Online]. Available: https://colab.research.google.com/