Lite-Agro 2.0: Integrating Federated and TinyML in Pear Disease Classification IoAT-Edge AI

Catherine Dockendorf
Dept. of Computer Science and Engineering
University of North Texas, USA
Email: catherineaprilsarmiento@my.unt.edu

Alakananda Mitra
Institute of Agriculture and Natural Resource
University of Nebraska-Lincoln USA.
Email: amitra6@unl.edu

Saraju P. Mohanty
Dept. of Computer Science and Engineering
University of North Texas, USA.
Email: saraju.mohanty@unt.edu

Elias Kougianos
Department of Electrical Engineering
University of North Texas, USA.
Email: elias.kougianos@unt.edu

***Abstract*—The success of Tiny Machine Learning (TinyML) is dependent on the process of the model training, along with optimized algorithms and quality image dataset. In the process of Pear disease identification through tree leaf analysis, the need to protect the privacy of protected data has led to the topic of Federated Learning (FL). Prior Lite-Agro 1.0 study aims to apply Neural Network trained Deep Learning models on lightweight MCUs. With this second Lite-Agro 2.0 expansion, we explore different models, the integration of Federated Averaging (FedAvg) algorithm and explore decentralized training of models on multiple client devices with an emphasis on preserving data privacy. In federated learning, the data does not leave the client device and individual weights are passed on and aggregated onto a global server which then generates a global model. This federated learning research benchmarks frameworks such as Flower.Dev, model performance, and tabulates training values such as accuracy and loss given various FedAvg algorithms**.

***Index Terms*—Smart Agriculture, Agriculture Cyber-Physical System (A-CPS), Internet of Agro Things (IoAT), Smart Village, TinyML, Edge-AI, Federated Learning, Data Privacy and Security, Plant Health and Plant Disease**

I. Introduction

With the Earth's population projected to reach 8 billion, the next question that comes to mind is how society is to continue feeding this growing population. Researchers turn to Smart Agriculture [1] [2] in addressing this upcoming challenge. *Lite-Agro* 1.0 is a prior study that explores the potential deployment of Tiny Machine Learning (TinyML) [3] models on a low power microcontroller. Studies involving identification of diseases in pear leaves, is based off of earlier works [4]-[7]. The DiaMos public dataset is trained on an AMD Ryzen 9 5950X 16 Core CPU in *Lite-Agro 2.0*, while *Lite-Agro 1.0* [7], uses mentioned CPU/ROCM with Radeon Yeston RX 6800XT GPU. After exploring TinyML models, this paper looks at how deployment may be achieved. The objective of this research is to explore the differences between the performances of various models and the effects of FL integration.

II. Contributions of the Current paper

At times, specialized hardware is needed for training, equipment can translate to cost and insufficient processing power can translate to time delays. We turn towards FL in addressing these issues during model training. The problem derives from power source limitations of small to mid farms that are situated outside of major city limits. *Lite-Agro 2.0* strives to address the need for low cost devices in agriculture.

A. Research Problem

The research aims to discuss different model parameters and to determine a balance in designing a TinyML-FL solution to pear disease identification.

B. Proposed Solution

Given the data shown in this study, readers get an insight on how models perform with the current DiaMOS dataset [8]. We discuss hardware, firmware and software considerations in the design which will be detailed in Section IV. In summary, the proposed solution to address privacy concerns, is to train in a federated way and ensuring the images stay on the device.

C. Novelties of the Current paper

The novelty of *Lite-Agro 2.0*, besides hardware exploration of FL training in TinyML [9] embedded devices, lies in the export of a TinyML model that is geared towards low power image computations on a lightweight multi client device MCUs. To deploy the models on ESP32CAM MCU, we integrate Google's open source library namely Tensorflow, Keras, with TinyML device, EspressIF firmware [10] and Flower.dev framework [11] . The current paper looks at how FL libraries can assist in inter-device training between multiple TinyML embedded device clients through FL frameworks such as Flower and TensorFlow Federated.

1. The system detects severe diseases in pear leaves.
2. Dataset has 4 different classes; healthy, curl, spot and slug.

III. Related Works

Lite-Agro 1.0 and 2.0 are related to works such as [12] and [13]. In [12], the diseases identified in pear leaves were *Septoria piricola* [14] [15], *Alternaria alternate*, *Gymnasporangium haracannum* [16] whereas in [8], leaves were classified into much simpler classes of healthy, curl, spot and slug. In this second paper, the topic switches from TinyML models to federated ML.

TABLE I
Related Federated Averaging Works on TinyML Platforms

Author	Paper Name	Novel Contribution
Kopparapu and Lin, et.al. [17]	TinyFedTL: Federated Transfer Learning on a microcontroller	FedAvg framework for TinyML CIFAR10 Benchmark
Idoje, Dagiuklas Mudessar et.al. [17]	FL: Crop Classification In a Smart Farm Decentralized Network	Decentralized Federated Averaging Model using SGD and Adam Optimizer in Chickpea Prediction

A summary of the literature survey among the papers mentioned can be described as an exploration on the applications of Federated Averaging algorithms and the variations and optimizations explored by the researchers. Kopparapu and Lin et al [18] developed a FedAvg framework on the publicly available CIFAR10 dataset under Google Research. Idoje, Dagiuklas, Mudessar et al. [17], uses an Adam Optimizer in contrast to the RMSProp optimizer used in this paper. Shown below is the mathematical explanation of FL algorithm:

Algorithm 1: *Federated Averaging Algorithm*; N: No. of clients, R: No. of Rounds; E: No. of local epochs

SERVER executes random weight initialization w_0.
Average weight = Random weights
for *each comm round (epoch) round=1, 2,...* **do**
Sample m clients **for** *each client* $k \in S_t$ ***in parallel*** **do**
$=\omega_{t+1}^{k} \leftarrow ClientUpdate(k, \omega_t) = \omega_{t+1} \leftarrow \sum_{k=1}^{K} \omega_{t+1}^{k}$
ClientUpdate(k,w$_t$) $: Split P_k$ *into Batches of size* β
for *each local epoch i from 1 to E* **do**
for *batch* $b \in \beta$ **do**
$=\omega_{t+1}^{k} \leftarrow ClientUpdate(k, \omega_t) = \omega \leftarrow \omega - nReturn\omega\ toserver$

A widely accepted averaging algorithm that the decentralized data and server utilizes is called the Federated Averaging algorithm [19] which has been worked on by Google researchers namely McMahan et al.. In this algorithm, clients train their copy of their global model using a Stochastic Gradient Descent calculation. The server then calculates an aggregated sum and looks at how we can take the model generated from the TensorFlow training and pass it on to a common server device.

IV. System Architecture of Lite-Agro 2.0

A. Overview

Lite-Agro 2.0 studies TensorFlow Federated [19] and integrates Flower [11] on TinyML [20] raspberry pi server connected - ESP32CAM [21] microcontroller. An FL environment involves working with multiple clients communicating with a server. Privacy is a core concept of FL thus, the dataset images stay on the local devices. A global model resides on the main server which is passed onto the clients where it trains statically. An aggregated sum is then, passed back to the server to calculate a new global model which is shown on the overall system diagram below. Due to ethernet being the protocol preference over USB or wireless bluetooth, the hardware selection is narrowed down.

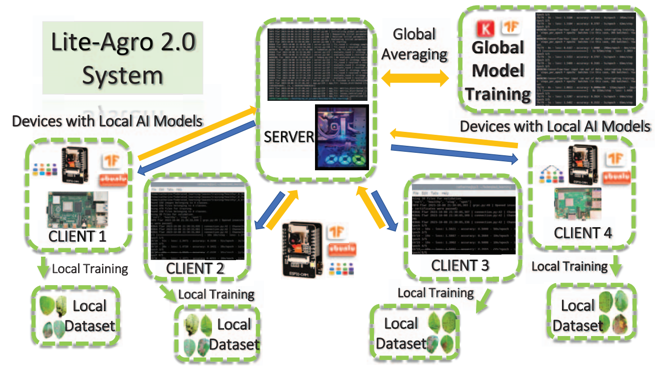

Fig. 1. System Overview of Lite-Agro 2.0

1) Training Protocol: To attain consistent comparison during training, parameters were initialized and 4 classes were declared. A 20% validation split and arguments 32 batch size, softmax classifier activation, categorical cross entropy loss function, rmsprop optimizer and arbitrary epoch selection of 5, 30, 80 and 100 were repeated throughout experiments. Learning rates were selected based on what has performed well in an earlier study.

B. Methodology

1) Dataset: Four classifications of leaves; healthy, spot, curl, and slug. DiaMOS plant consists of 3505 images of publicly available pear fruits and leaves. After image augmentation, classes were made to balance and generation of 3949 samples attained in the process. Preprocessing augmentation of images were automated through scripting techniques with imagemagick and AI, those that couldn't, manually edited with GIMP editor. The prospect of automated predictive capability and it's application of automation are endless. A variety of factors are considered in determining which FL framework is preferred such as the ability to support ML libraries; TensorFlow or PyTorch, then the availability of FedAvg algorithms. There are a number of frameworks being developed and after much research, the selection narrows down to framework Flower.dev. The table below shows a comparative analysis between existing works and a summary of FL strategies in contrast to the current paper(*Lite-Agro 2.0*).

TABLE II
COMPARISON OF TENSORFLOW FEDERATED PLATFORMS

Framework	Developer	Library	Strategy
Flower.dev	Univ of Cambridge	TensorFlow Keras PyTorch	FedAvg Aggregate QFedAvg DPFedAvg FedAdam FedProx FedOpt FedYogi SCAFFOLD QFedAvg FedMedian FedYogi
TensorFlow Federated	Google Research	Keras TensorFlow PyTorch	FedAvg Q-Fair FedAvg FedProx Async FedAvg
PySyft	OpenMined	PyTorch TensorFlow	FedAvg DP FedAvg Homomorphic Encryption

V. EXPERIMENTAL RESULTS

A. Hardware for Prototyping

After exploring various hardware experimental design, instead of an initial experimentation method where a USB un/mounting mechanism is used to copy models between different client devices, an Ethernet implementation is chosen as the best set-up. By using a generic offline network bridge adapter implementation values can be passed through cables. The set-up addresses concerns in cybersecurity of malicious eavesdropping of local weights that assists attackers in reconstructing training images. The set-up uses 3 Raspberry Pi 4.0s and one 3B model, mounted on a Pi Cluster rack. Tensorboard graphs and log values were captured during each run. Much of the process is automatable. As there is a bottleneck on the number of output monitors available, an HDMI switcher is used to switch between different HDMI output. Shown below is the set-up of the Lite-Agro 2.0 study:

Fig. 2. FL Experimental Implementation-Pear Disease Detection Set-up

B. Experiments

The summary is to run multiple training sessions on TinyML [20] embedded [3] devices. Values which would then be passed on as a generated aggregated sum by the central server.A variety of advanced FedAvg parallelization algorithm strategies may still be applied to improve the concept. Deep Learning models explored included EfficientNetB0, InceptionV3 [22], MobileNetV2 [23], and Ensemble of Convolutional Neural Networks such as efficientnetv2-b0 + InceptionV3 [22] *(Google)*, efficientnetv2-b0+ MobileNetV2 [23], and InceptionV3 [22] + MobileNetV2 [23]. Respective reasons for selection are explained in the Training Protocol section. Shown below are the comparison graphs between MobileNetV2: FedAvg and FedMedian Accuracy and Loss values. The report currently generates an error when parsing the precision and recall values and this is something that is actively being worked on. The best values were captured at epoch=30 while running multiple rounds to verify. The strategies are set to only run when 4 minimum available clients connect to train.

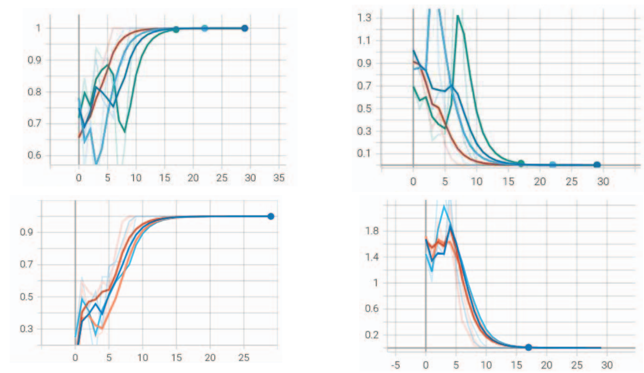

Fig. 3. FedAvg; Accuracy and Loss vs FedMedian; Accuracy and Loss

Table IV shows an initial experiment that was conducted with MobileNetV2 using the FedMedian strategy. The accuracy values performed worse at lower epochs however in later experiments conducted, captured 25% at epoch=1, then 78.12% at epoch=10 before it capped at 100%. This has to be verified whether it is overfitting or performing well. The loss value however continues to go down starting at 2.5978 at epoch=1 and goes down to 6.6735e-04 at epoch=30 for example. Flower.dev goes on to generate a zipped weights array after the training and those values are then converted into a .cc format that TensorFlow Lite accepts.There is a need for tinier models that so that it can be ported onto the TFLite and a potential for deployment. This is an area that will be explored next. To give a brief overview of current sizes of models generated by DL, Table III is shown in the next section.

VI. CONCLUSION AND FUTURE WORK

The emerging topic of Tiny Federated ML [20] [18] brings up possibility for compact ML inference to be easier to implement. This will allow us to maximize the available compute power of local devices. Privacy is kept since the images stay on each device and only weights are shared to the global server. After exploring TensorFlow, Keras, Flower Federated Learning frameworks, we test out various models and Federated Averaging strategies on MCUs to compare which best works on the DiaMOS plant dataset. The core of this system design has been completed but due to time constraints of this study, there could still be improvements made with a dynamically training program. In the future, this study can be expanded to take Federated Averaging strategies other than the FedAvg and FedMedian mentioned. A power

TABLE III
DEEP LEARNING INITIAL TRAINING RESULTS AT EPOCH =5

CRITERIA	MOBILENETV2	EFFICIENTNETB0	INCEPTIONV3	XCEPTIONV3	VGG19
Image Size	224 X 224	224 X 224	299 X 299	299 X 299	224 X 244
H5 Size	17.8 MiB	45.9 MiB	167.4 MiB	159.5 MiB	1.0 GiB
TFLITE Size	8.5 MiB	22.3 MiB	83.1 MiB	79.4 MiB	532.5 MiB
CC Size	52.2 MiB	137.8 MiB	512.6 MiB	489.3 MiB	3.2 GiB

TABLE IV
FEDERATED LEARNING FEDMEDIAN TRAINING RESULTS AT EPOCH = 5 ROUND = 3

MODEL	**MOBILENETV2 CLIENT 1**					**MOBILENETV2 CLIENT 2**					**MOBILENETV2 CLIENT 3**				
Epoch	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Loss	1.67	1.51	1.23	1.49	0.53	1.95	1.67	1.55	1.43	1.51	1.56	1.57	1.39	1.55	1.22
Accuracy	0.58	0.45	0.55	0.42	0.88	0.32	0.34	0.32	0.42	0.50	0.50	0.37	0.50	0.39	0.50
S/Epoch	46	23	21	20	5	51	20	20	20	5	50	19	19	19	0.5
s/Step	2	1	1	1	.261	3	1	1	1	.260	1	1	1	1	.253
Loss	1.41	1.29	1.36	1.09	1.49	1.74	1.22	1.49	1.48	1.26	1.42	1.42	1.23	1.35	0.65
Accuracy	0.42	0.47	0.55	0.58	0.63	0.50	0.24	0.55	0.32	0.29	0.47	0.39	0.55	0.42	0.75
S/Epoch	19	19	19	19	19	20	20	20	20	4	20	20	20	20	4
s/Step	1	1	1	1	.223	1	1	1	1	.226	1	1	1	1	.231
MODEL	**MOBILENETV2 SERVER ROUND 1**					**MOBILENETV2 SERVER ROUND 2**					**MOBILENETV2 SERVER ROUND 3**				
Epoch	1	2	3	summary		1	2	3	summary		1	2	3	summary	
Loss	1.5180	1.5184	0.4167	1.3853		1.5156	1.5489	2.0653	1.4456		1.5207	1.5482	1.1552	1.551	
Accuracy	0.3544	0.3797	1	0.0312		0.3797	0.3165	0	0.0312		0.3924	0.2532	0	0.0312	
s/Epoch	8	5	296 ms	0		5	5	131 ms	0		5	5	133 ms	0	
s/Step	105 ms	62 ms	4 ms	525 ms		64 ms	65 ms	131 ms	151 ms		64 ms	64 ms	2 ms	153ms	

reading was taken and the client devices ran from 3.0W to 6.0W during active FL training. Further research expansion ideas are a comparison of the performance of models and other strategies, power reading observations, and running the training at longer epochs.

REFERENCES

[1] A. Mitra, S. L. T. Vangipuram, A. K. Bapatla, V. K. V. V. Bathalapalli, S. P. Mohanty, E. Kougianos, and C. Ray, "Everything You Wanted To Know About Smart Agriculture," *CoRR*, vol. abs/2201.04754, 2022.[Online]. Available: https://arxiv.org/abs/2201.04754

[2] A. Degada, H. Thapliyal,and S. P. Mohanty, "Smart village: An IoT based digital transformation," Proc. *IEEE WF-IoT*, 2021. pp. 459-463.

[3] N. Schizas, A. Karras, C. Karras, and S. Sioutas, "Tinyml for ultra-low power ai and large scale iot deployments: A systematic review," *Future Internet*, vol. 14, no. 12, 2022. [Online]. Available: https://www.mdpi.com/1999-5903/14/12/363

[4] A. Mitra, S. P. Mohanty, and E. Kougianos, "aGROdet: A Novel Framework for Plant Disease Detection and Leaf Damage Estimation," in *Proceedings of the 5th IFIP International Internet of Things Conference (IFIP-IoT)*, 2022, pp. 3–22.

[5] A. Mitra, A. Singhal, S. P. Mohanty, E. Kougianos, and C. Ray, "ecrop: A novel framework for automatic crop damage estimation in smart agriculture," *SN Computer Science*, vol. 3, 2022.

[6] A. Mitra, S. P. Mohanty, and E. Kougianos, "A smart agriculture framework to automatically track the spread of plant diseases using mask region-based convolutional neural network," in *Proc. of IFIP-IoT*, 2022, pp. 68–85, 2022.

[7] C. Dockendorf, A. Mitra, S. P. Mohanty, and E. Kougianos, "Lite-Agro: Exploring Light-Duty Computing Platforms for IoAT-Edge AI in Plant Disease Identification," in *Internet of Things - Advances in Information and Communication Technology*, 2024, pp. 371–380.

[8] G. Fenu and F. M. Malloci, "Classification of pear leaf diseases based on ensemble convolutional neural networks," *AgriEngineering*, vol. 5, no. 1, pp. 141–152, 2023. [Online]. Available: https://www.mdpi.com/2624-7402/5/1/9

[9] H. Han and J. Siebert, "Tinyml: A systematic review and synthesis of existing research," in P*roc. International Conf. AI in Information and Communication*, 2022, pp. 269-274.

[10] Espressif Systems, "Espressidf: IoT development framework," https://www.espressif.com/en/products/sdks/esp-idf, Last Accessed: May 08, 2023.

[11] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P. P. B. de Gusmão, and N. D. Lane, "Flower: A friendly federated learning research framework," *arXiv:2007.14390*, 5 Mar 2022.

[12] F. Yang, F. Li, K. Zhang, W. Zhang, and S. Li, "Influencing factors analysis in pear disease recognition using deep learning," *Peer-to-Peer Networking and Applications*, vol. 14, pp. 1–13, 05 2021.

[13] G. Fenu and F. M. Malloci, "Diamos plant: A dataset for diagnosis and monitoring plant disease," *Agronomy*, vol. 11, no. 11, 2021. [Online]. Available: https://www.mdpi.com/2073-4395/11/11/2107

[14] M. Chatzidimopoulos and A. Pappas, "Epidemiology and control of septoria pyricola in pear leaf and fruit," *Journal of Plant Pathology*, vol. 98, pp. 447–452, 11 2016.

[15] T. Thomidis and S. Katerinis, "Occurrence of a fruit spot disease of pear caused by septoria pyricola in tyrnavos larissa, northern greece," *Plant Disease*, vol. 98, pp. 845–845, 2014.

[16] B. Lāce, "Gymnosporangium species – an important issue of plant protection," *Proceedings of the Latvian Academy of Sciences. Section B. Natural, Exact, and Applied Sciences.*, vol. 71, pp. 95–102, 2017.

[17] G. Idoje, T. Dagiuklas, and M. Iqbal, "Federated learning: Crop classification in a smart farm decentralised network," *Smart Agricultural Technology*, vol. 5, p. 100277, 2023.

[18] K. Kopparapu and E. Lin, "Tinyfedtl: Federated transfer learning on tiny devices," *arXiv:2110.01107*, 3 Oct 2021.

[19] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," *arXiv:1602.05629*, 26 Jan 2023.

[20] P. Warden and D. Situnayake, *TinyML: Machine Learning with TensorflowLite on Arduino and Ultra Low Power Microcontrollers*. O'Reilly Media, Inc., 2019.

[21] M. Ziaul Haque Zim, "Tinyml: Analysis of xtensa lx6 microprocessor for neural network applications by esp32 soc," *arXiv e-prints*, p. arXiv:2106.10652, Jun. 2021. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2021arXiv210610652Z

[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. CVPR*, 2016, pp. 2818-2826.

[23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. CVPR*, 2018, pp. 4510-4520.