RuralAI in Tomato Farming: Integrated Sensor System, Distributed Computing and Hierarchical Federated Learning for Crop Health Monitoring

Harish Devaraj[1], Shaleeza Sohail[2*], Melanie Ooi[1**], Boyang Li[3], Nathaniel Hudson[4,5*], Matt Baughman[4], Kyle Chard[4,5], Ryan Chard[5], Enrico Casella[6*], Ian Foster[4,5***], and Omer Rana[7**]

[1] *School of Engineering, University of Waikato, Hamilton 3210, New Zealand*
[2] *School of Information and Physical Sciences, University of Newcastle, NSW 2000, Australia*
[3] *School of Engineering, University of Newcastle, NSW 2308, Australia*
[4] *Department of Computer Science, University of Chicago, IL 60637, United States of America*
[5] *Data Science & Learning Division, Argonne National Laboratory, IL 60439, United States of America*
[6] *Department of Animal & Dairy Sciences, University of Wisconsin-Madison, WI 53715, United States of America*
[7] *School of Computer Science and Informatics, Cardiff University, Cardiff CF10 3AT, United Kingdom*
** Member, IEEE*
*** Senior Member, IEEE*
**** Fellow, IEEE*

Abstract—Precision horticulture is evolving due to scalable sensor deployment and machine learning integration. These advancements boost the operational efficiency of individual farms, balancing the benefits of analytics with autonomy requirements. However, given concerns that affect wide geographic regions (e.g., climate change), there is a need to apply models that span farms. Federated Learning (FL) has emerged as a potential solution. FL enables decentralized machine learning across different farms without sharing private data. Traditional FL assumes simple 2-tier network topologies and thus falls short of operating on more complex networks found in real-world agricultural scenarios. Networks vary across crops and farms, and encompass various sensor data modes, extending across jurisdictions. New hierarchical FL (HFL) approaches are needed for more efficient and context-sensitive model sharing, accommodating regulations across multiple jurisdictions. We present the RuralAI architecture deployment for tomato crop monitoring, featuring sensor field units for soil, crop, and weather data collection. HFL with personalization is used to offer localized and adaptive insights. Model management, aggregation, and transfer are facilitated via a flexible approach, enabling seamless communication between local devices, edge nodes, and the cloud.

Index Terms— Internet of Things (IoT), sensor systems, sensor applications, federated learning, precision horticulture.

1. I. INTRODUCTION

Tomato (*Solanum lycopersicum*) is a popular and versatile crop widely cultivated worldwide. Tomato crop health monitoring is an important task for farmers, as it can help them optimize their resource use, improve their crop quality and yield, and prevent or mitigate plant disease [1]. However, current methods of tomato crop health monitoring are often labor-intensive, time-consuming and inaccurate, as they rely on manual inspection, visual assessment, or chemical analysis. Moreover, these methods do not capture the spatial and temporal variations of the crop conditions nor do they provide timely and actionable feedback to the farmers. Although some researchers have proposed using machine learning techniques to support plant health monitoring, most current work utilizes images of leaves and fruits [2], [3], [4], which may not provide instant or accurate predictions.

We implement the RuralAI architecture [5]–a novel approach using IoT sensors, distributed computing and serverless federated learning–for tomato crop health monitoring. Our approach aims to address the following research question: *How can we design and deploy a sensor network using distributed computing and federated learning to achieve real-time tomato health monitoring for resource-constrained infrastructure?* To answer this question, we make the following novel contributions:

1) We design and configure an IoT sensor architecture that can collect data from various sources (e.g., soil, weather, and image data) and transmit these data to the edge and cloud server nodes for local and global model training and aggregation. We consider bandwidth, cost of transmission, and power consumption of the sensor network, and optimize data sampling and compression techniques accordingly. 2) We deploy a novel hierarchical federated learning framework built on a Function-as-a-Service (FaaS) platform that spans cloud, on-premise, and on-edge devices. In this manner, we benefit from the simple programming model of serverless computing and implement hierarchical federated learning to enable scalable, secure, and efficient distributed machine learning across heterogeneous devices and networks. 3) We develop a federated learning strategy that can preserve regional information. We use a hybrid approach of centralized and personalized hierarchical federated learning at local gateways using fuzzy logic to adapt the model to the local data distribution and preferences by customizing the fuzzy rule base and membership functions.

The rest of the paper is organized as follows: Section II describes how we have implemented the RuralAI architecture for tomato crop health monitoring, covering the sensor design, federated learning, software, and model management. Section III discusses the experimental design and results of our prototype. Section IV presents future plans and deployment for the proposed prototype.

2. II. The RuralAI Architecture for Tomato Crop Health Monitoring

The RuralAI testbed is a three-tier system made up of a cloud layer, an edge layer, and a local layer. The local layer consists of multiple field sensor units deployed on crops located in New Zealand and Australia that share data exclusively with their farm's local gateway. Due to privacy restrictions, information sharing among the Farms is achieved by exchanging model parameters at the Edge FL server at the regional level. Both Edge FL servers and farm local gateways utilize lightweight models that are periodically updated from a globally aggregated model (Fig. 1).

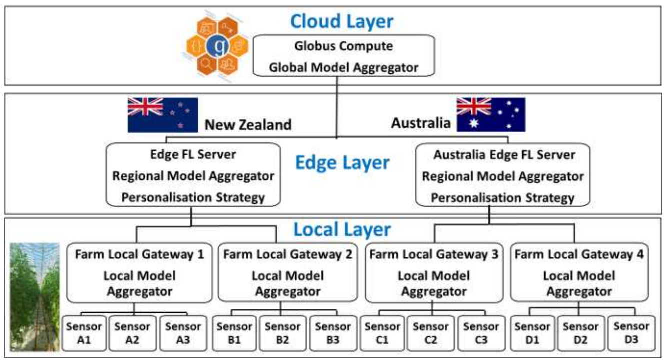


Fig. 1. Graphical representation of the sensors, local gateways, edge and cloud servers in the RuralAI tomato crop monitoring testbed

1. A. Sensor: Mechanical and Electronic Design

The RuralAI plant health monitoring sensor system (Fig. 2) forms the key interface between the physical environment and the digital world. Each sensor system can track vital plant health parameters by assessing soil components, plant well-being, and atmospheric conditions.

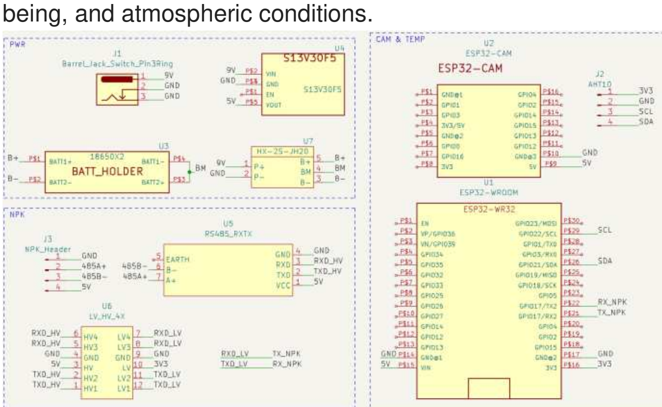


Fig. 2. The sensor system with multiple onboard sensors: soil NPK, air temperature and humidity, camera, and battery

The sensor system measures soil conditions integral for optimal plant health, including Nitrogen, Phosphorus, Potassium (commonly termed NPK), soil temperature, humidity, electrical conductivity (which indirectly measures soil salinity and pH levels). These sensors also measure air temperature and humidity, aiding in modeling current climate conditions to estimate precise plant care needs. For instance, if the prevailing weather suggests recent or imminent rainfall, the sensor disregards irrigation requirements indicated by soil moisture levels. This prevents waterlogging, safeguarding plant roots, and preserving yield. While soil and atmospheric parameters offer valuable insights into a plant's health, external elements like pests or inadequate lighting can also diminish crop yield. To monitor such external factors, we integrate a fixed field-of-view camera during deployment.

As optical sensors and image processing demand substantial computational resources, the camera node employs its onboard ESP32-S3 microcontroller. The soil and air monitoring sensors connect to a separate ESP32 microcontroller. Both microcontrollers transmit processed data to their local gateways. Sensor nodes are designed for deployment in remote, power-constrained fields, necessitating power efficiency to ensure extended operational periods between recharges. Each sensor node is powered by two 3500mAh Li-ion batteries. They interface various sensors and microcontrollers via a custom PCB, integrating data converters for compatibility and are encased for protection.

2. B. Machine Learning: Hierarchical Federated Learning with Personalization

Federated Learning (FL) is a decentralized method for training a machine learning model. Individual nodes train on their local data, local models are then shared and aggregated to create a global model. Personalized Federated Learning (PFL) extends FL by taking the extra step of adapting the global model to nodes' local data. Hierarchical Federated Learning (HFL) is a generalization of FL that considers more complex networks wherein many aggregation nodes are organized in a hierarchy. These intermediate aggregation nodes can reduce the load at the FL cloud server. Additionally, personalization performs better in HFL when localization of the models is undertaken for a group of nodes associated with a local aggregation node (i.e., a personalized model is not generated for every node, but for a group of nodes [6]).

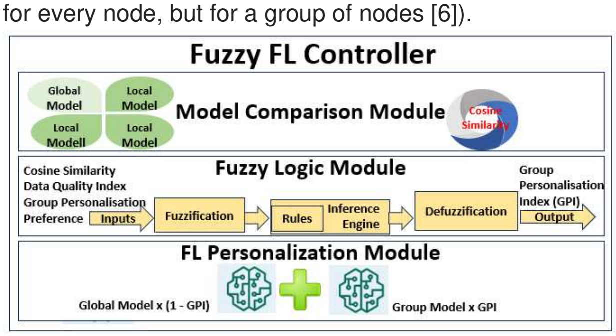


Fig. 3. Fuzzy FL Controller

A fuzzy FL controller (Fig. 3) is introduced at Edge FL servers to personalize regional group models based on the difference between the parameters of the region and the global model using cosine similarity. The controller considers this similarity measure, a data quality metric and personalization preference as input to calculate the group personalization index for every group considering fuzzy rules. This index is used to aggregate global and group models to get group-based personalized models at each local gateway.

3. *C. Software: Federated learning using Function-as-a-Service*

Our FL approach makes use of *Function-as-a-Service* (FaaS), a serverless computing paradigm that allows the execution of functions using a cloud-hosted platform. In this paradigm, pre-programmed functions are registered with a cloud registry and are then submitted as single tasks to remote computational resources (endpoints) for execution. FaaS requires that users pay only for the amount of computing resources used and significantly reduces the system configuration required for each user.

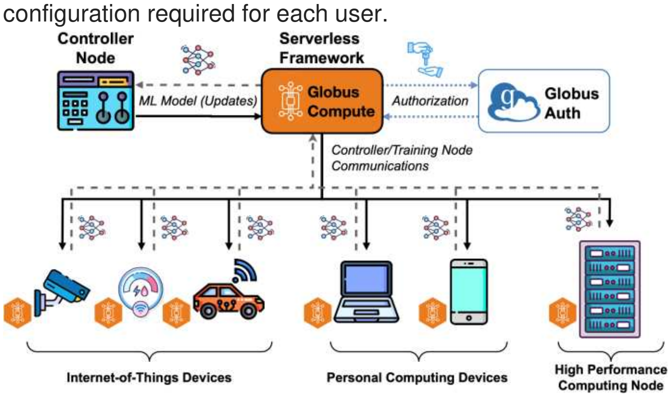


Fig. 4. Graphical representation of distributed coordination for FL using Globus Compute.

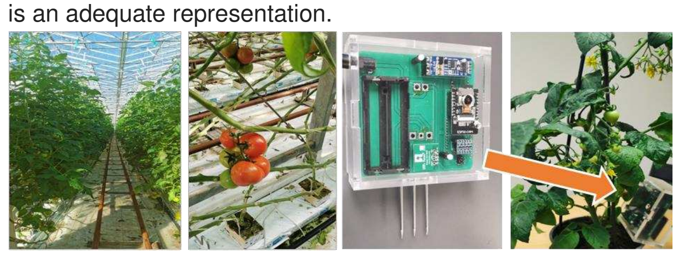
In prior work, we have developed FLoX, a framework for managing the FL process on remote FaaS endpoints [7]. FLoX is built on top of the Globus Compute, a federated FaaS platform that enables the execution of functions across an ecosystem of distributed endpoints. In Globus Compute, endpoints must first be deployed and configured, specifying local resources. A user launching an FL process from a controller node can then simply specify the list of endpoint UUIDs required to perform local training. Globus Compute is designed specifically to work with diverse computational system resources, from IoT nodes to high-performance computing nodes. A visualization of the FLoX system is shown in Fig. 4.

4. *D. Coordination: Model Management*

HFL requires coordination between the various endpoints. One advantage of HFL is that the aggregation and communication of the models across the network can be sequestered into localized clusters to better fit the topology of the data communication network. However, despite this advantage, there are several core challenges. One of these central challenges is the communication in HFL using a cloud-hosted FaaS platform. While conventional FaaS relies on the cloud service and its communication with endpoints to transfer data, ML and FL present new challenges. Specifically, in HFL, it is more intuitive to communicate data using the natural topology of the network rather than have all data transmission done through a central cloud service which may require excessive network hops. To improve data transmission in FaaS, we can use direct endpoint-to-endpoint communication, such as via our recent work using object proxies [8]. Proxies serve as a lightweight reference to remote data and can be used in FaaS as a “pass-by-reference” mechanism wherein the function is submitted to endpoints along with a reference to some remote data. Therefore, only this lightweight proxy is communicated to the cloud and large data transfer is performed endpoint-to-endpoint.

3. III. Experimental Setup, Results and Discussion

Our testbed (Fig. 1) consists of two sites: New Zealand and Australia. Within each physical site, we emulate 2 Farms, whereby Farms 1 and 2 are situated in New Zealand and Farms 3 and 4 are in Australia. Farms 1 and 3 cultivate *normal* tomatoes, whereas Farms 2 and 4 focus on growing cherry tomatoes. This testbed includes only three crops per Farm. Each crop represents a sample from an ‘irrigation row’ within a production glasshouse. Given that the plants within an irrigation row share the same irrigation conditions, one sample per row is an adequate representation.



(a) (b) (c) (d)

Fig. 5. Example of (a) a row of crops in a production glasshouse whereby (b) data is sampled from selected plants using (c) the sensor field units, which are (d) inserted into the planter box

The probes within the sensor units are inserted into the soil of the planter box, while the camera is oriented towards the plant stem. Fig. 5 depicts the arrangement of the sensor field units and their placement. Images collected can provide insights into the current growth stage of the plant. At the initial stage, local models are exclusively trained using data from the crops within their respective Farms at the local gateway, following a centralized learning approach. Subsequently, model parameters undergo sharing and aggregation at both the edge and cloud layers through HFL, via FaaS. This distributed computing architecture allows for continuous model updates, promoting knowledge-sharing for enhanced performance and accuracy while ensuring the confidentiality of farm-specific data.

Due to the large amount of data collected, we present a truncated example of air and soil humidity sensor data from New Zealand in Fig. 6. In our testbed, we intentionally disrupt the irrigation of one representative crop each in Farms 1 and 2. From Fig. 6, this can quite easily be observed in Sensors A1 and B1. Data collection is ongoing until Feb 2024 and will serve as the foundation for forthcoming research. We aim to explore novel personalization strategies and identify model distribution variances using this platform.

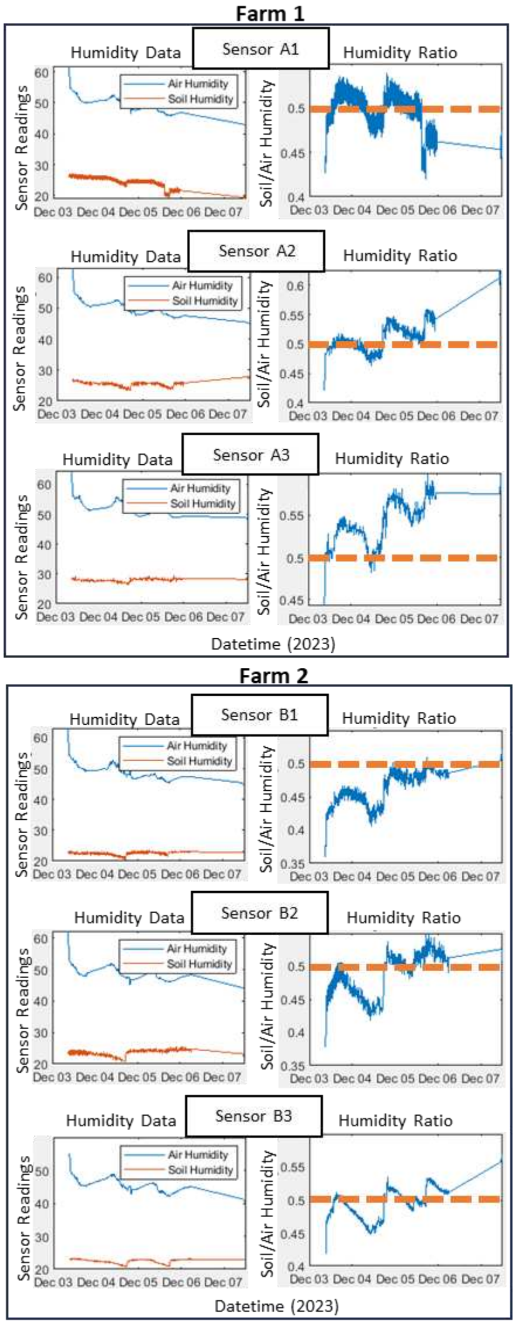


Fig. 6. Example of soil and air humidity data, where a ratio of soil to air humidity below 0.5 indicates possible water supply issues

4. CONCLUSION

This article presents a functional tomato crop management system using the RuralAI architecture. Real-time data from field sensor units forms the basis for initial health models for the tomato crop. Each farm constructs a local model deployed on our edge computing node, using FLoX. These models are aggregated into a global model, facilitating sharing among farms. This approach enables the development of locally reflective models while allowing sharing among farms to identify common trends. It ensures data remains decentralized, addressing privacy concerns associated with centralized cloud servers.

We believe federated learning with on-farm processing and data capture is pivotal in precision horticulture. Empowering farm owners and operators to manage and process their data is a critical step towards meeting the demands for economic and operational autonomy within this community.

ACKNOWLEDGMENT

This work was supported in part by the University of Waikato-University of Newcastle Partnership Seed Fund, the New Zealand Royal Society under the Rutherford Discovery Fellowship program, and the US Department of Energy (DOE) under Contract DE-AC02-06CH11357

.REFERENCES

[1] V. Jindal, V. Kukreja, S. Mehta, M. Manwal and K. Joshi, “Emerging Paradigms: A Federated Learning CNN Approach for Tomato Leaf Disease Detection,” 2023 3rd Asian Conference on Innovation in Technology (ASIANCON), Ravet IN, India, 2023.

[2] Y. Liu et al., “High-Precision Tomato Disease Detection Using NanoSegmenter Based on Transformer and Lightweighting,” Plants, vol. 12, no. 13, p. 2559, Jul. 2023.

[3] S. Mehta, V. Kukreja and S. Vats, “Bean Leaf Disease Diagnosis in the Age of Federated Learning and CNN: A Severity Analysis Approach,” 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), Trichy, India, 2023.

[4] U Shruthi, V. Nagaveni, C. Arvind, and G. Sunil, “Tomato Plant Disease Classification Using Deep Learning Architectures: A Review,” *Algorithms for intelligent systems*, pp. 153–169, Jan. 2022, doi: https://doi.org/10.1007/978-981-16-7389-4_15.

[5] P. Patros, M. Ooi, V. Huang, M. Mayo, C. Anderson, S. Burroughs, M. Baughman, O. Almurshed, O. Rana, R. Chard, K. Chard, “Rural AI: Serverless-powered Federated Learning for Remote Applications,” IEEE Internet Computing, vol. 27, no. 2, pp. 28-34, Sept 2022.

[6] Y. He, D. Yan and F. Chen, “Hierarchical federated learning with local model embedding”, Engineering Applications of Artificial Intelligence, vol 123, p. 106148, 2023.

[7] N. Kotsehub, M. Baughman, R. Chard, N. Hudson, P. Patros, O. Rana, I. Foster, and K. Chard, “FLoX: Federated Learning with FaaS at the Edge,” in the Proceedings of the IEEE International Conference on e-Science, Oct 2022.

[8] J. G. Pauloski, V. Hayot-Sasson, L. Ward, N. Hudson, C. Sabino, M. Baughman, K. Chard, I. Foster, “Accelerating Communications in Federated Applications with Transparent Object Proxies,” in the Proceedings of the ACM/IEEE International Conference on High Performance Computing, Network, Storage, and Analytics (Supercomputing), Nov 2023.