AFRL-AFOSR-JP-TR-2024-0046

Distributed inferencing and federated learning, for distributed-edge-AI miniaturised satellite constellations

RUSSELL BOYCE
UNIVERSITY OF NEW SOUTH WALES
HIGH ST
KENSINGTON, , 2052
AUS

01/28/2024
Final Technical Report

Air Force Research Laboratory
Air Force Office of Scientific Research
Asian Office of Aerospace Research and Development
Unit 45002, APO AP 96338-5002

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
20240128	Final	START DATE 20210924	END DATE 20230923

4. TITLE AND SUBTITLE
Distributed inferencing and federated learning, for distributed-edge-AI miniaturised satellite constellations

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
	FA2386-21-1-4047	
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Russell Boyce

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
UNIVERSITY OF NEW SOUTH WALES
HIGH ST
KENSINGTON 2052
AUS

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AOARD UNIT 45002 APO AP 96338-5002	AFRL/AFOSR IOA	AFRL-AFOSR-JP-TR-2024-0046

12. DISTRIBUTION/AVAILABILITY STATEMENT
A Distribution Unlimited: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
In this research, we report the deployment of a networked space-edge-AI test bed that is analogous to the behaviour of a real satellite constellation and incorporates actual edge AI hardware. We demonstrate the implementation of federated learning on the AI test bed, outline the AI test bed system architecture, and conduct a preliminary study to assess the AI model's performances in performing image classification on the AI test bed. The result of this preliminary study serves as a proof-of-concept which demonstrates the feasibility of applying federated learning for a constellation of miniaturised satellites to improve satellite's AI capabilities.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U	SAR	8

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
GEOFFREY ANDERSEN	

Standard Form 298 (Rev.5/2020)
Prescribed by ANSI Std. Z39.18

Final Report for AOARD Grant FA2386-21-1-4047 and FA9550-19-S-0003

"Distributed inferencing and federated learning, for distributed-edge-AI miniaturised satellite constellations"

22 December 2023

Name of Principal Investigators (PI and Co-PIs): Dr Bassel Al Homssi

- E-mail address: b.al_homssi@adfa.edu.au
- Institution: UNSW Canberra at the Australian Defence Force Academy, Australia
- Mailing Address: Northcott Dr, Campbell ACT 2612

Period of Performance: 09/24/2021 – 12/31/2023

Abstract: In this research, we report the deployment of a networked space-edge-AI test bed that is analogous to the behaviour of a real satellite constellation and incorporates actual edge AI hardware. We demonstrate the implementation of federated learning on the AI test bed, outline the AI test bed system architecture, and conduct a preliminary study to assess the AI model's performances in performing image classification on the AI test bed. The result of this preliminary study serves as a proof-of-concept which demonstrates the feasibility of applying federated learning for a constellation of miniaturised satellites to improve satellite's AI capabilities.

Report:

I. Introduction

Miniaturized satellites are limited in computational power and on-board hardware when compared to large monolithic satellites, restricting their ability to process large volumes of data. Although miniaturized satellites are relatively less powerful compared to larger satellites, they are relatively cheap to produce in a large scale. The fact that they are cost-effective and easy to be replaced makes collaborative miniaturized satellite constellations feasible which opens the opportunity to undertake missions beyond the capabilities of a single expensive monolithic satellite. While the research community has been exploring edge-AI and distributed learning algorithms for satellite constellations, the current solutions do not address many of the challenges that arise in practical constellations. As a result, a more practical framework into edge-AI constellation approaches that includes distributed inferencing and federated learning, and the scheduling challenges that arise with such approaches, is required. However, due to launching costs, performing the research and development required to address this problem is quite challenging and is difficult to be achieved prior to testing and validating the approaches and their scalability. In this research, we report the deployment of a network space-edge-AI test bed that emulates the behaviour of a real satellite constellation and incorporates actual edge-AI hardware. Our proposed research has the following aims:

1. The development of a lab-based, modular and scalable, virtual and hardware-in-theloop (HIL) test bed for accelerating AI-enabled-miniaturised-satellite-constellation R&D.
2. Development and demonstration on the test bed, of methods for distributed inferencing

on AI-enabled-miniaturised-satellite-constellations.

3. Development and demonstration on the test bed, of approaches to federated learning on AI-enabled-miniaturised-satellite-constellations.

II. Research Progress

The original funded program of work sought (and successfully achieved) the development of a laboratory-based distributed-intelligence satellite constellation test range, and the demonstration on that range of federated learning. The AI test range applies the same flight hardware technology used for command and data handling in past Canberra Space missions. The hardware and software engineering products therefore provide this test range with a realistic foundation to simulate an actual on-air constellation.

The test range simulates an intelligent satellite constellation. Like a real on-orbit satellite constellation, each hardware-based node in this test range comprises a satellite flight computer that runs an onboard data handling subsystem that handles astrodynamics, controls one or more AI-capable payloads, and communications with the other nodes in the network. This constellation resides on its own network with an additional GPU-based head node that provides scheduling and additional network services when centralised federated learning is exercised. This network can be extended with remote nodes which, thanks to AOARD funding, our group was able to attract additional funds to add sophisticated digital-twinning capabilities.

We have developed a centralized federated learning approach that leverages the weighted sum technique to aggregate the different models trained by the different edge-nodes [7]. As expected, as the number of edge-nodes increased, the training performance became more robust and accurate as well as less time consuming. As a result, this proof-of-concept has shown that the test range is capable of performing distributed-AI tests and confirmed that federated learning for satellite constellations can enhance the performance training and learning in comparison to stand-alone edge-nodes.

#	Previous Aim	Current Progress
1	The development of a lab-based, modular, and scalable, virtual and hardware-in-the-loop (HIL) test range for accelerating AI-enabled-miniaturised-satellite-constellation R&D	UNSW Canberra Space has developed and assembled an in-house test range.
2	Development and demonstration on the test range, of methods for distributed inferencing on AI-enabled-miniaturised-satellite-constellations	UNSW Canberra Space has developed and demonstrated the inference of deep learning and federated learning onto the AI test range.
3	Development and demonstration on the test range, of approaches to federated learning on AI-enabled-miniaturised-satellite-constellations	UNSW Canberra Space has demonstrated distributed AI capabilities on the test range by employing centralized federated learning techniques.

Furthermore, we are currently in the process of submitting a peer-reviewed article on the key

findings and lessons learnt of this work to the provide an insight into the outcomes to the research community. On the other hand, these outcomes shed light on many challenges such as:

1. The determination of the values of the weighted sums at the aggregation for the different edge-nodes remains a challenge with many trade-offs.
2. Centralized federated learning heavily relies on the availability of the satellites in-orbit and the communication link health, federated learning relies on downlink to transmit the individual models and then the uplink to share the aggregated model.

The literature provides various centralized federated learning algorithms that rely on asynchronous model updating depending on the availability of the satellites, most of which assume that the ground terminal is able to communicate with the satellites. As a result, there is a prevalent need to explore other avenues such as various decentralized distributed learning techniques and the need for a more reliable/suitable weighting assignment.

III. Experiment: Federated Learning for Object Detection on AI testbed

In this section, we provide an overview of the federated learning design and system, the experiment setup of this study, as well as further details of the dataset and deep learning model used in this study.

A. System Overview

The AI test bed system we propose emulates federated learning, utilizing a group of miniaturized satellites as clients and the ground station as the server. For the server node, we use a Lambda Vector workstation equipped with 4 NVIDIA RTX A5000 GPUs, powerful for machine learning and deep learning model training and inference. As for the client nodes, we employ the NVIDIA Jetson Nano, an embedded system with GPUs designed for on-board processing and machine learning capabilities.

During this project period, we have made progress in setting up the AI testbed for object detection using federated learning to explore its viability when deployed for a constellation of miniaturized satellites. As an integral aspect of this setup, we have proposed a secure weight assessment for the satellites, based on their past performances using a probabilistic machine learning model.

We propose employing a statistical-based computation to determine the weights assigned to each client. The clients' weights are calculated based on their past performances in accurately detecting objects in images. Assessing object detection models involves two crucial components: accuracy in the detected class and accuracy in the detected object's bounding boxes. The performance in class detection is measured through recall and precision. Precision measures the accuracy of positive predictions, while recall measures the ability of the model to capture all the relevant instances.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

Simultaneously, the accuracy in locating the object's bounding box is gauged by the intersection over union (IoU) threshold. This threshold calculates the ratio between the intersection of the ground truth box and the detected box and the union of both boxes.

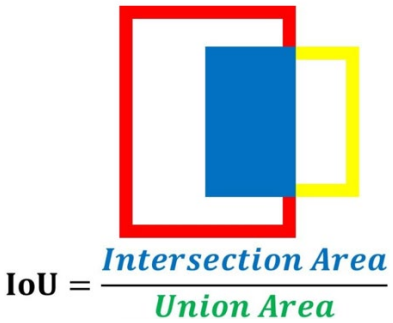

Figure 1 Intersection over Union (IoU) Threshold [1]

By combining both class detection metrics and the IoU threshold, the mean average precision (mAP) score is commonly used to evaluate the overall performance of object detection models by comparing the ground truth bounding box to the detected box. The mAP score computes the area under the recall-precision curve for a specific range of IoU thresholds. A higher mAP score indicates a greater likelihood that the object detection model can correctly predict the object class within the correct coordinates in the image.

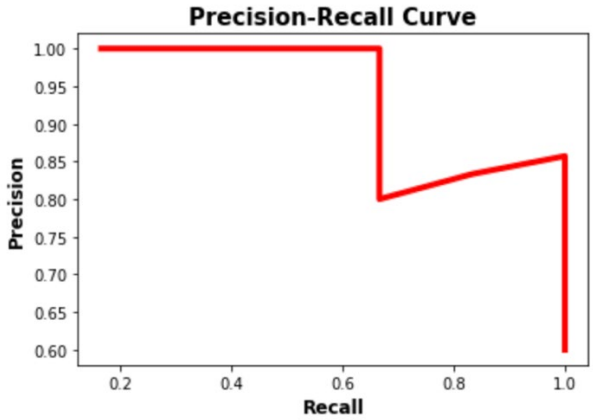

Figure 2 Precision Recall Curve [1]

To compute each client's weights, we use an incrementally trained probabilistic machine learning model, such as Naïve Bayes. To ensure the trustworthiness of newly joined clients, they are assigned a weight (or trustworthiness score) of a very low score, e.g. 0.1, each time they have just joined the federated learning system.

The following outlines the step-by-step process of how the federated learning process and client weights are updated using the statistical-based computation above:

(a) Before initiating the federated learning process, the server monitors incoming connection requests from clients. Upon receiving a connection request, the client is registered, and a connection is established between the server and client.

Subsequently, the server shares the parameters of the global object detection model.

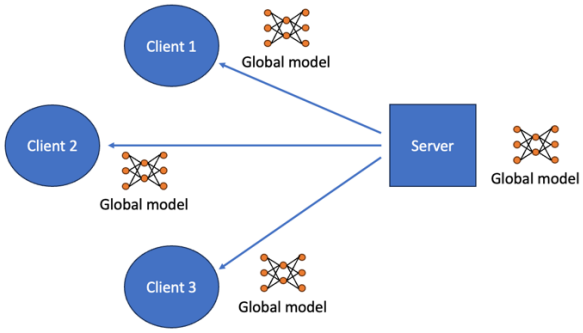

Figure 3 Server shares global object detection model

(b) Utilizing the list of registered clients on the server, while maintaining connectivity, each client independently trains a machine learning model based on its distinct data and observations. This results in the creation of a local object detection model for each client.

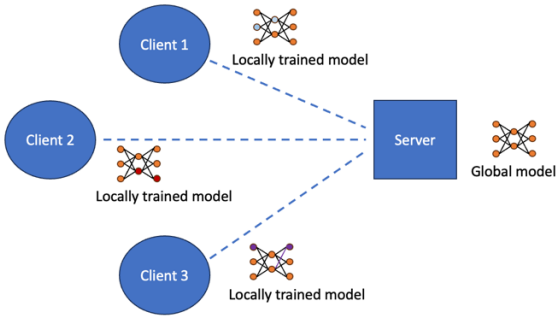

Figure 4 Clients train local model

(c) The weight for each client is determined based on the performance of its respective local object detection model.

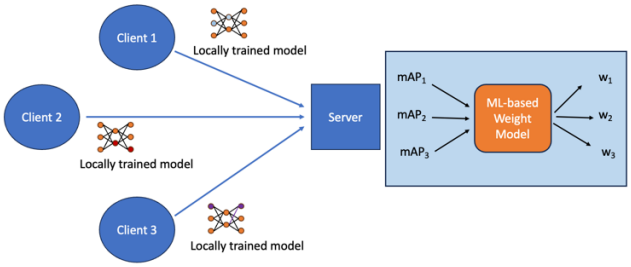

Figure 5 Clients weight calculation

(d) Following the completion of the model training process on the client's end, the model parameters are transmitted to the server for aggregation.

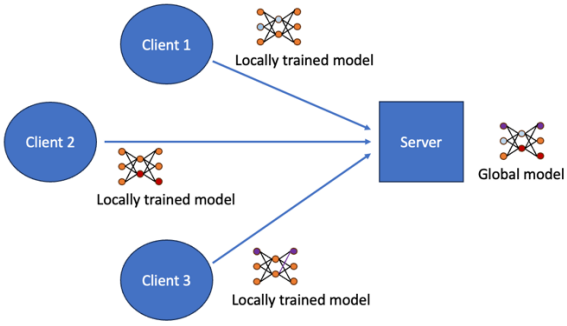

Figure 6 Model aggregation

(e) Once evaluated, the server-side machine learning model parameters are disseminated to clients to update the parameters of each local machine learning model. This marks the conclusion of the initial round in the federated learning process. Afterwards, the federated learning process recommences at step (b) for the second round. In this round, each client refines its local machine learning model based on its unique data and observations, followed by steps (c), (d), and the cycle continues.

B. Experimental Setup

In this section, we provide an overview of the AI test bed system architecture, the experiment setup of this study, as well as further details of the dataset and deep learning model used in this experiment.

1) Dataset

In this experiment, we utilized the xView dataset [2], which comprises 847 images at a resolution of 3094×2932 pixels, featuring 60 classes. The dataset includes a total of 600,345 objects across all images. These satellite images are sourced from DigitalGlobe's WorldView 3 (WV3) satellites, covering an approximate area of 1,400 square kilometers at a resolution of 30 cm. For our experiment, we partitioned the images into smaller patches of size 128x128 pixels to create a more manageable dataset for training and testing, resulting in approximately 150,000 images. To achieve this, we developed a script that automates the process of slicing the images and converting the labels to the corresponding patches. The script can be found at [3].

2) Deep Learning Model

In this study, we use YOLOv8 [4], a state-of-the-art object detection model known for its accuracy and efficiency. We employ the pretrained YOLOv8 weights provided by Ultralytics [5] and fine-tune them on the xView dataset. For fine-tuning and model training, we utilize PyTorch as the deep learning framework and Ultralytics, which provides a wrapper to train and validate the YOLOv8 model. Several modifications were made to the Ultralytics library to adapt it to our specific experimental setup and requirements, available at [6]. Meanwhile, we also employ Flower as the federated learning framework,

handling communication between the server and clients. The codebase containing the client and server federated learning implementations can be found at [7].

IV. Progress Update and Future Milestones

At this stage, we have successfully set up the environment to conduct experiments using the selected hardware devices. The dataset has been sliced, and labels have been converted using an automated script. Furthermore, the server and client nodes have been configured with the necessary hardware and software components, and they were able to communicate with each other through the Flower federated learning framework. We have executed the federated learning experiment on the AI testbed with four clients for 30 rounds. However, we encountered some compatibility issues with the Ultralytics package, affecting model aggregation and hindering performance improvement across the rounds.

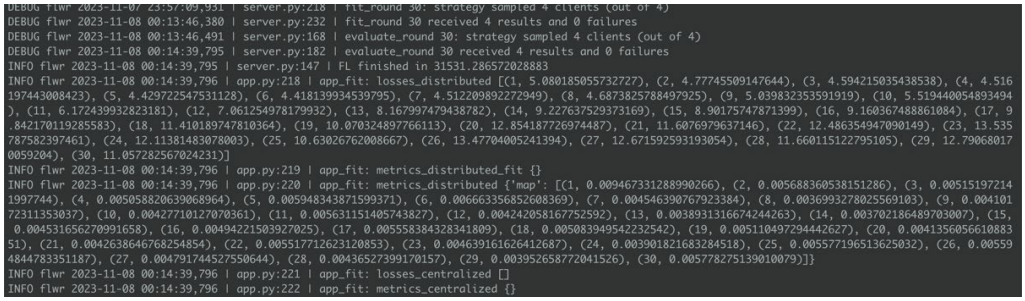

Figure 7 Loss and mAP scores per round

During the past months, we encountered several obstacles during setup and experiment, as follows:

a) Ultralytics compatibility

There were compatibility issues between the Ultralytics library and the required variables for the federated learning process on our experimental setup. To address these obstacles, we modified the Ultralytics library to cater to our specific requirements, particularly to obtain the classification loss values related to model training and validation at each round.

b) Model training synchronization

Another challenge involved coordinating and synchronizing the training process between the server and client nodes, leading to timeout errors. We are currently working on resolving these issues and exploring alternative approaches, such as utilizing other federated learning frameworks or implementing custom synchronization mechanisms manually, to avoid these timeout errors and achieve smooth coordination between the server and client nodes during training.

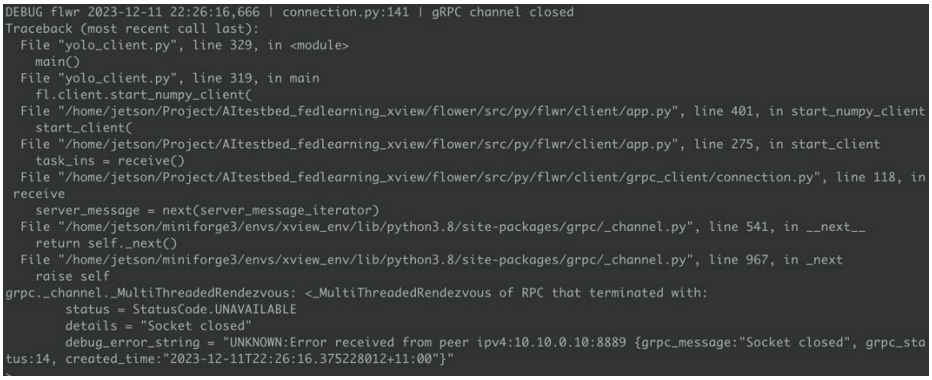

Figure 8 Timeout error

In summary, our current progress includes setting up the environment, slicing the dataset, configuring the server and client nodes, and establishing communication through the Flower federated learning framework. In the next stage, we plan to implement the customized client weight assessment and conduct experiments to evaluate the performance of the federated learning system in our setup.

V. References

[1] "Mean Average Precision (mAP) Explained," Paperspace Blog. Accessed: Dec. 18, 2023. [Online]. Available: https://blog.paperspace.com/mean-average-precision/

[2] D. Lam *et al.*, "xView: Objects in Context in Overhead Imagery." arXiv, Feb. 21, 2018. doi: 10.48550/arXiv.1802.07856.

[3] "widyarini/yolo-tiling," GitHub. Accessed: Dec. 18, 2023. [Online]. Available: https://github.com/widyarini/yolo-tiling

[4] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8." 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[5] "ultralytics/ultralytics: NEW - YOLOv8 in PyTorch > ONNX > OpenVINO > CoreML > TFLite." Accessed: Dec. 18, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[6] "widyarini/ultralytics at xview_modification." Accessed: Dec. 18, 2023. [Online]. Available: https://github.com/widyarini/ultralytics/tree/xview_modification

[7] "widyarini/fed_learning_aitb," GitHub. Accessed: Dec. 18, 2023. [Online]. Available: https://github.com/widyarini/fed_learning_aitb