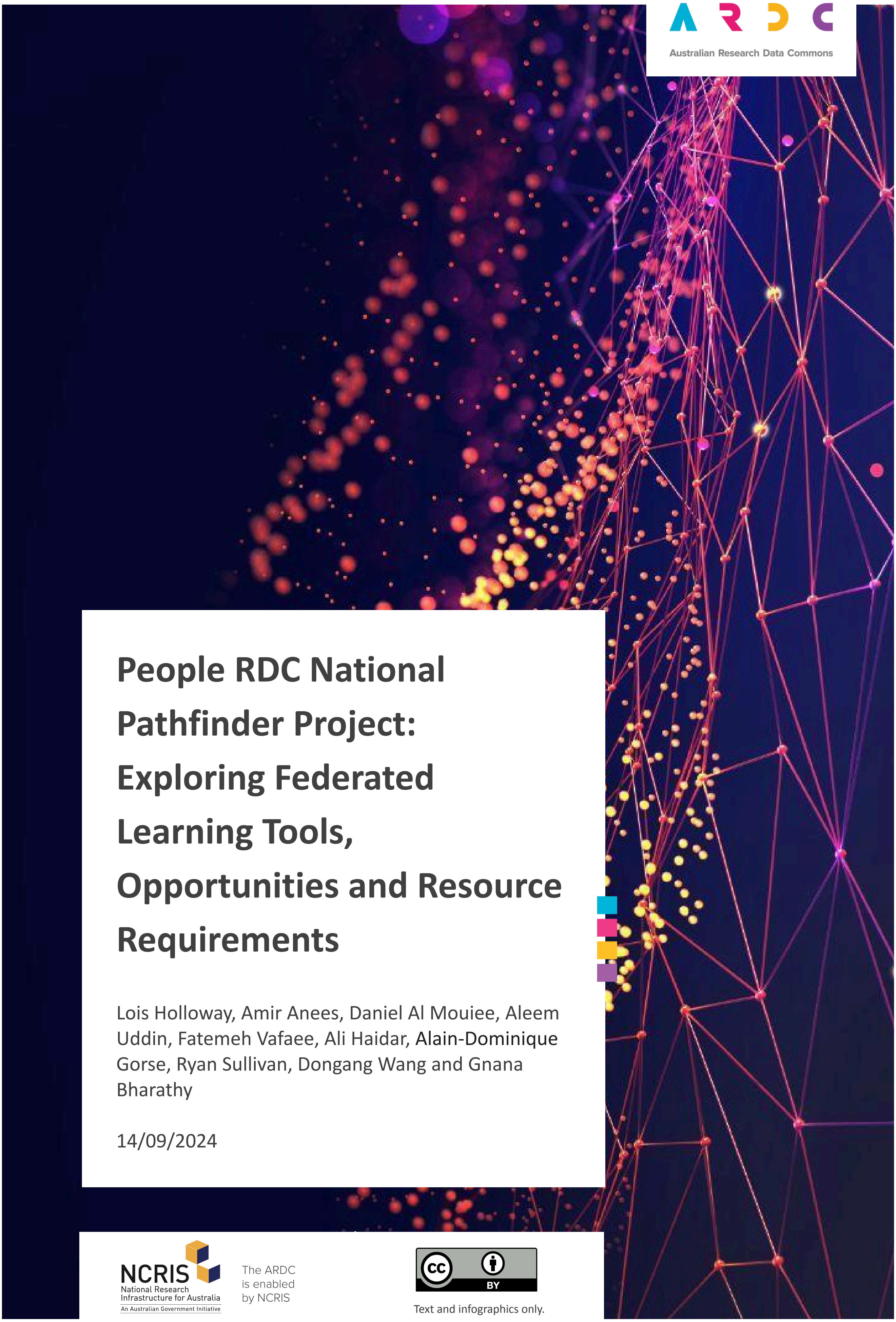
ARDC
Australian Research Data Commons



People RDC National Pathfinder Project: Exploring Federated Learning Tools, Opportunities and Resource Requirements

Lois Holloway, Amir Anees, Daniel Al Mouiee, Aleem Uddin, Fatemeh Vafaee, Ali Haidar, Alain-Dominique Gorse, Ryan Sullivan, Dongang Wang and Gnana Bharathy

14/09/2024

NCRIS
National Research Infrastructure for Australia
An Australian Government Initiative

The ARDC is enabled by NCRIS

Cite as:

Holloway L, Anees A, Al Mouiee D, Uddin A, Vafaee F, Haidar A, Gorse D, Sullivan R, Wang D, Bharathy G (2024). People RDC National Pathfinder Project: Exploring Federated Learning Tools, Opportunities and Resource Requirements. Report to ARDC, Australian Research Data Commons. DOI: 10.5281/zenodo.14038105

ARDC is enabled by NCRIS

Acknowledgement of Country

We acknowledge the traditional custodians throughout Australia and their continuing connection to, and deep knowledge of, the land and waters. We pay our respects to Elders both past and present.

Working Group

Authors

Name	Position and Affiliation	ORCID
Prof Lois Holloway	Principal Research Medical Physicist, South Western Sydney Local Health District, University of New South Wales and Ingham Institute	0000-0003-4337-2165
Dr Amir Anees	Postdoctoral Fellow, University of New South Wales, Ingham Institute and South Western Sydney Local Health District	0009-0001-2085-587X
Mr Daniel Al Mouiee	Software Engineer, University of New South Wales, Ingham Institute and South Western Sydney Local Health District	0000-0002-8766-7451
Dr Aleem Uddin	Research Infrastructure Specialist, ARDC	0000-0002-8519-5534
A/Prof Fatemeh Vafaee	Associate Professor, School of Biotechnology and Biomolecular Sciences, UNSW AI Institute, UNSW Data Science Hub, University of New South Wales Sydney	0000-0002-7521-2417
Dr Ali Haidar	Conjoint Fellow, University of New South Wales, Ingham Institute and South Western Sydney Local Health District	0000-0001-5092-949X
Dr Dominique (Dom) Gorse	Director, QCIF Bioinformatics and Data Science	0000-0003-1230-4194
Dr Ryan Sullivan	Head of Australian Imaging Service, Research Technology, School of Biomedical Engineering, The University of Sydney	0000-0001-5554-7378
Dr Dongang Wang	Postdoctoral Fellow, University of Sydney	0000-0001-5805-0244
Dr Gnana K Bharathy	Research Data Specialist (AI/ML & Architecture), Australian Research Data Commons (ARDC) and UTS	0000-0001-8384-9509

Acknowledgements and Thanks:

Ming Ding for many stimulating discussions, Adrian Burton, Roger Ward for review, and Carolyn Staines for copy-editing and Jason Yuen and Jo Savill for publishing assistance

Contents

NCRIS National Research Infrastructure for Australia An Australian Government Initiative ARDC is enabled by NCRIS

1. Background

1.1. Purpose and Context

1.1.1. Purpose

In the 4th Quarter of 2023, the Australian Research Data Commons (ARDC) reached out to the Australian Cancer Data Network (ACDN), who had previously collaborated on a federated learning project with ARDC, to jointly develop a pathfinder project.

- The study aims to explore the uses, needs, and challenges of federated learning in the context of sensitive health-related data, while ensuring the maintenance of privacy and confidentiality.
- Identify and establish a collaborative network among similar research groups.
- Develop suitable demonstrator artifacts to centre the dialogues around them.

This report presents the findings of this Pathfinder Project (see Section 1.2) for the analysis of sensitive health-related data while maintaining privacy and confidentiality. It focuses on requirements and current experiences with federated learning (Section 1.3).

1.1.2. Context

The Australian Research Data Commons (ARDC), through the People Research Data Commons (People RDC), is delivering national scale data infrastructure for health research and translation. In this context, the infrastructure is defined broadly as shared resource or coordinated activity and includes both hard and soft resources and assets such as:

- Underpinning hardware infrastructure: Compute support program (Nectar, MLeRP), graphics processing unit (GPU), storage
- National reference data assets: Data curation, vocabularies and analytic reference datasets, synthetic data, Research Data Australia, Research Vocabularies Australia, FAIR model for artificial intelligence (AI) reference data and machine learning (ML) models
- Tools & environment reference programs: Library of tools/collaborative and foundational infrastructure (models, analytics tools, hubs, virtual labs) etc.
- National-level cultural and coordination assets: Training and capacity development, culture and policy, communities of practice, guidelines

The People RDC engages with all parts of the health system to address four national-scale challenge areas, as shown in Figure 1.1:

1. Data Strategy and Discovery
2. Secure Data Access

3. Data Integration
4. Advanced Analytics

An important strategy for addressing the challenges associated with advanced analytics is the co-development of a national framework. This framework provides the specifications and reference architecture for future work. One of the known cardinal challenges of healthcare advanced analytics is managing the sensitivity in the data.

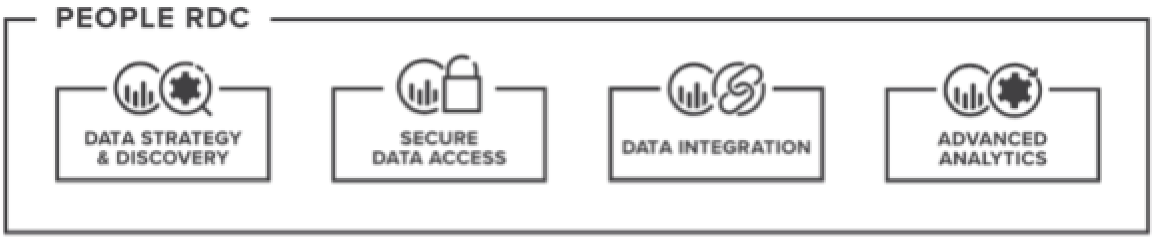


Figure 1.1. Four national-scale challenge areas, addressed by the People RDC by engaging with all parts of the health system.

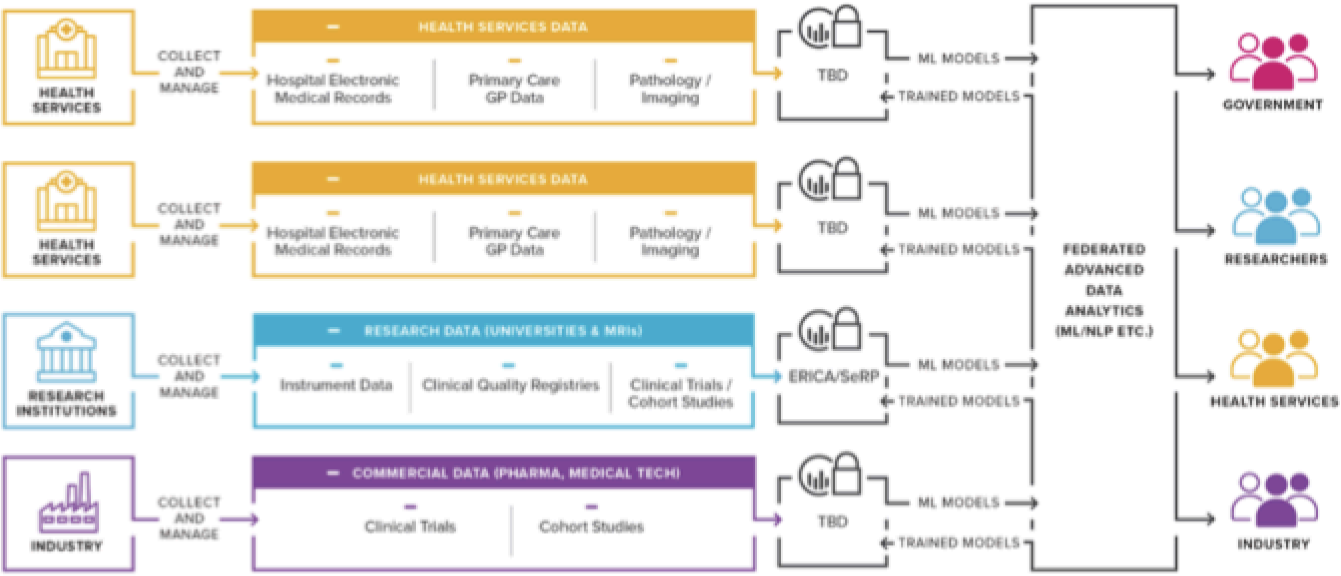


Figure 1.2. A top-level illustration of federated advanced data analytics.

Healthcare data, as a consequence of various protective regulations and concerns, is fragmented. To understand this key issue, People RDC investigated the landscape of federated learning and sought to develop a pathfinder to facilitate exploration of the approach.

As a companion to framework development, the overarching goal is to create a federated learning pathfinder for People RDC projects, which would provide insights for future ARDC partnership programs and foster a sense of community around the feasibility of constructing a federated learning infrastructure for healthcare data.

1.2. Background and Introduction to Federated Learning

High quality data analysis and model development requires access to large, diverse and granular datasets. Ideally this requires detailed (e.g. imaging and detailed treatment information) datasets to be available for learning from different geographical locations both across Australia and internationally. With regards to healthcare data, this is challenging due to ethics and privacy requirements that can limit data movement and restrict storage requirements.

Federated learning (Li, Fan, Tse, & Lin, 2020) is a decentralised approach to machine learning model training. It is gaining traction for its ability to preserve data privacy while allowing for collaborative learning across distributed sites. Instead of centralising data on a server, federated learning enables distributed sites to train models locally using their respective datasets and then share only model updates or gradients with a central server, as shown in Figure 1.3. This methodology not only ensures data privacy and security but also enables learning from diverse data sources without the need for centralised data aggregation.

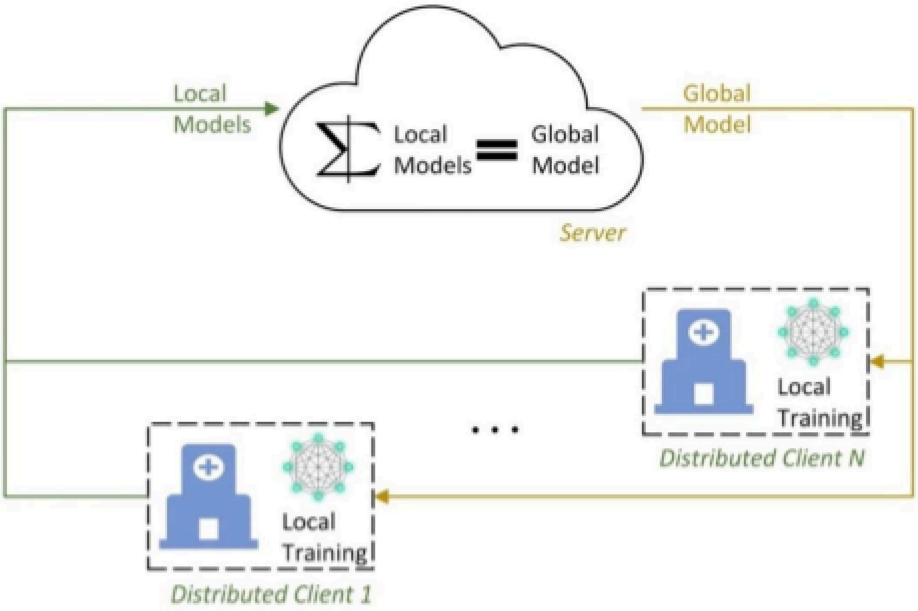
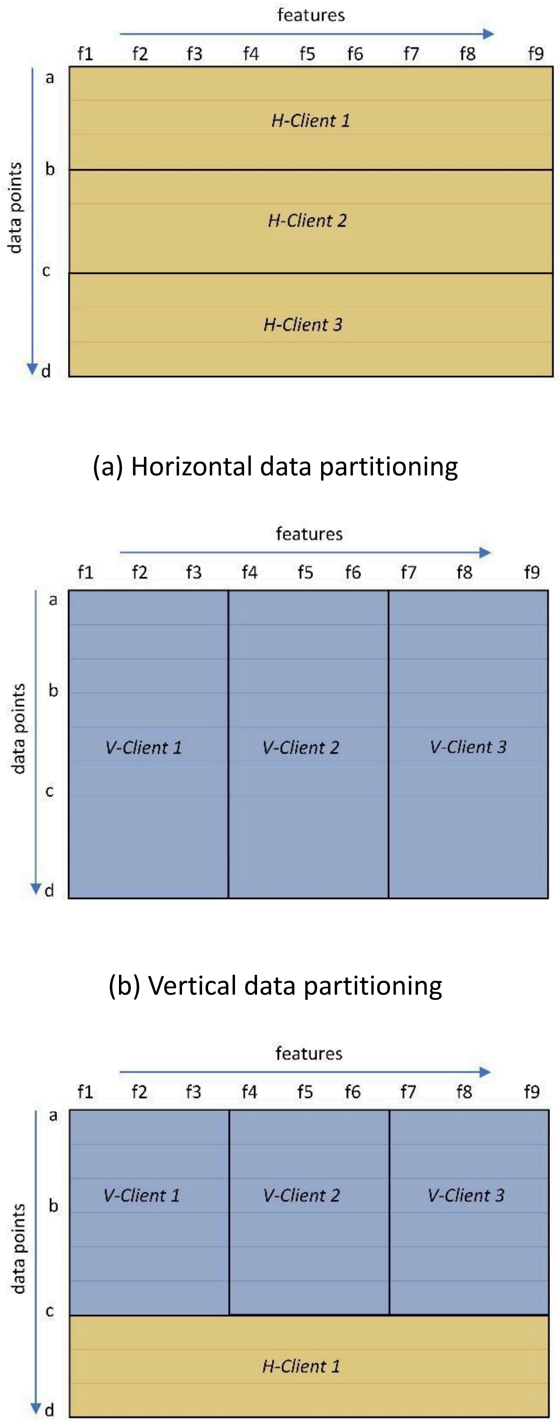


Figure 1.3. A top-level illustration of the federated learning process consists of a server and N participating clients. The server initializes a random global model and sends it to each participating distributed client. Upon receiving the model, each client trains it on their local dataset and sends the trained local model to the server. The server aggregates these local models to update the global model. This process continues for several rounds until the global model converges to the local minimum.



(a) Horizontal data partitioning

(b) Vertical data partitioning

(c) Combined data partitioning

Figure 1.4. Illustration of different data partitioning used in federated learning

Various types of federated learning approaches exist to accommodate different data partitioning scenarios (illustrated in Figure 1.4). Horizontal federated learning deals with situations where distributed sites have access to similar features but possess different data points. In contrast, vertical federated learning addresses cases where distributed sites hold different sets of features for the same data points. Further, data can be both horizontally and vertically partitioned between the sites.

To facilitate the implementation of federated learning, numerous open-source tools and frameworks have emerged. These tools provide developers and researchers with the necessary infrastructure and algorithms to experiment with federated learning setups efficiently. However, deploying federated learning in real-world scenarios presents a set of unique challenges.

Integrating federated learning tools into existing systems can be complex, requiring compatibility with diverse infrastructures and technologies. Participating sites may not have the required infrastructure or skills. There may be challenges around data governance, in the context of federated learning, which is a change from the well understood centralised data sharing approach. Ensuring the security and privacy of sensitive data during federated learning processes must also be carefully managed, particularly in applications where regulatory compliance is mandatory.

Achieving scalability and optimal performance while minimising communication overhead and resource consumption poses additional hurdles in real-world deployments. Addressing these challenges demands interdisciplinary collaboration among experts in machine learning, distributed systems, cybersecurity, and regulatory compliance as well as discipline specific data experts. Innovative solutions and robust methodologies are necessary to overcome the obstacles and unlock the full potential of federated learning in real-world applications.

1.3. Report Focus

This report provides an overview of requirements and current experiences with federated learning. It covers the following:

1. A comparison of key federated learning tools available and infrastructure requirements to support federated learning with the goal of establishing a suitable blueprint for a federated learning architecture that can be effectively implemented. The intention of this work is to identify and assess opportunities and requirements for these tools as part of a national infrastructure solution. This includes:
    - Building on work to date to review open-source software available for federated learning (horizontal and vertical); Section 2.
    - Providing overview of key differentiators of the different open-source software tools for federated learning (e.g. ease of use, communication requirements, ability to adapt software) for both horizontal and vertically distributed data; Section 2.

- Comparing practical implementation of a refined number of open-source software tools (up to 5) for federated learning in the simulation environment, considering both horizontal and vertically distributed datasets; Section 3.
- Consideration of the infrastructure, particularly data storage, compute, and communication pathways necessary to support implementation of federated learning generally but specifically in a health care environment; Section 4.

It should be noted that data standardisation is also a key requirement for effective federated learning. As work on data standardisation is being undertaken by the ARDC elsewhere (in the Integration Stream 3.* People RDC Projects) it has not been covered in this report. The Integration Stream of work covers areas such as Data Standards and Common Models.

2. Consideration of use cases that could become cardinal edge cases for the development of a national infrastructure, including discussion of case study of designs, deployments, that are available to or informing national infrastructure. The discussions include features, coordination and resources required, successes as well as lessons learnt (or pitfalls to be avoided); Section 5.

3. Conclusions and Recommendations to ARDC on infrastructure and other support required to enable and encourage use of federated learning by Australian research groups, particularly focused on health care (ARDC people). These recommendations were developed following a workshop on federated learning including research teams working with federated learning or related areas; Section 6.

2. Comparison of Open-Source Tools

2.1. Aim

A primary objective of this report is to provide a comprehensive comparison of open-source federated learning tools. Specifically, the aim is to identify tools that not only incorporate the federated learning paradigm but also exhibit robust security features while offering a flexible framework for the integration of additional features.

2.2. Background

A similar study on the comparison of different open-source federated learning tools was done in (Riedel, et al., 2024). Their evaluation began with a literature review, organised using a Latent Dirichlet Allocation model to identify key concepts. The frameworks were then assessed based on criteria categorized into Features, Interoperability, and User Friendliness, and a weighted scoring system was applied. Fifteen open-source FL frameworks were evaluated, with Flower achieving the highest total score of 84.75%. Other frameworks like FLARE, FederatedScope, PySyft, FedML, and OpenFL also performed well. FederatedScope excelled in Features, while PySyft, FedML, Flower, IBM FL, and FLARE topped Interoperability. EasyFL was the best in User Friendliness. On the other hand, FATE AI, PaddleFL, and FedLearner scored the lowest, mainly due to poor Interoperability and User Friendliness.

Our work differs from this study by focusing on additional criteria specific to practical and technical aspects relevant to the implementation and usability of federated learning frameworks, as described in the next section.

2.3. Selection Criteria

The federated learning tool assessment criteria is aimed to streamline the evaluation process for federated learning (Li, Fan, Tse, & Lin, 2020) tools. This criterion was determined in discussion with experts in the field, with seven criteria determined as described below:

2.3.1. Authentication and Security

Authentication is the process of verifying the identity of users or systems to ensure that only authorized entities can access sensitive information or perform specific actions. In the context of federated learning, authentication is crucial for securing communication between different nodes or devices participating in the learning process. A robust authentication system safeguards against unauthorized access and ensures the integrity of the federated learning environment. Security features encompass encryption and other measures to protect data during transmission, safeguarding against potential threats or breaches.

2.3.2 Node Setup and Ease of Use

Node setup refers to the process of configuring and connecting individual nodes or devices within a federated learning system. Ease of use evaluates how straightforward it is for users to set up and initiate the federated learning process. A tool with user-friendly node setup and interfaces streamlines the implementation process, reducing the complexity of integrating federated learning into existing systems. Tools that are easy to use are more likely to be adopted widely, especially by users with varying levels of technical expertise.

2.3.3 Programming Language Support

Programming language support assesses the ability of federated learning tools to work seamlessly with different programming languages. A tool that supports multiple languages provides users with flexibility, allowing them to integrate federated learning into projects developed in diverse programming environments. This criterion is essential for ensuring that the tool can be easily adapted to existing software ecosystems, promoting interoperability and versatility in application.

2.3.4. Learning Capabilities

Learning capabilities refer to a federated learning tool's capacity to perform different types of learning tasks. Horizontal learning involves collaborative learning on similar data across different nodes, while vertical learning entails learning from different but complementary data items across distributed datasets. Robust learning capabilities are essential for addressing a variety of scenarios and data distributions, ensuring the tool's applicability to a wide range of use cases.

2.3.5 Technical Expertise and Debugging

Technical expertise measures the level of proficiency required by users to implement and operate a federated learning tool. A tool that demands minimal technical expertise facilitates wider adoption and usability. Additionally, debugging tools are crucial for identifying and resolving issues during the development and deployment phases. Adequate debugging support simplifies troubleshooting, enabling users to address potential challenges efficiently.

2.3.6. Documentation and Testing

Documentation quality evaluates the clarity, completeness, and accessibility of instructional materials provided by a federated learning tool. High-quality documentation is vital for guiding users through the installation, configuration, and utilization processes. Testing suites refer to sets of pre-defined tests that verify the functionality and reliability of the tool. Well-documented tools with comprehensive testing suites enhance user confidence and contribute to the overall reliability and stability of the federated learning environment.

2.3.7. Cloud Native

People RDC is aiming to provide national research infrastructure at scale and in this setting 'cloud native' is a desirable criterion. The cloud native approach is about building applications which are scalable and can run in public or private cloud or hybrid cloud infrastructure (Amazon, 2024). The cloud native approach is being led by a global body called The Cloud Native Computing Foundation (CNCF). CNCF is described as "the open source, vendor-neutral hub of cloud native computing, hosting projects like Kubernetes and Kubeflow to make cloud native universal and sustainable". The driving factors behind the adoption of Kubernetes are hinged on technical advantages elaborated below:

- Microservices approach: Adopts the microservices based approach in building modular applications which are easy to manage. Each microservice can be realised in the form of a container (CNCF, 2024).
- Container orchestration: Kubernetes as a container orchestrator allows building an application with many containers working together. Allowing numerous features such as scalability, networking, storage and so on (CNCF, 2024).
- Scalability: Allows applications to scale up and down based on usage.
- Simplify infrastructure requirements: Ability to run Kubernetes on varied hardware and underlying software including cloud.
- Better resource utilisation, faster development, simplified cloud migration (Amazon, 2024).
- Off the shelf containerised software: Ever increasing number of containerised applications (Veritis, 2024), including machine learning platforms such as Kubeflow.

Given these advantages of cloud native approach, the general recommendation for selecting a FL framework would be to verify if the framework provides any of the following. Firstly, if the framework has a containerised implementation. Second, if the framework has an implementation ready to be deployed on Kubernetes in the form of a helm chart, Kubernetes operator or simply has a Kubernetes implementation.

2.4. Chosen Tools and Analyses

The federated learning tools to be further investigated were selected by initially looking for tools that had the presence of the term "federated learning" in the GitHub name or description. Final tools selected were then required to have an open-source codebase, support for encrypted communication through Secure Socket Layers, and the availability of actively maintained software documentation on GitHub.

The tools selected include *FEDn, IBMFL, OpenFL, PySyft, Flower, AusCAT, Vantage6*, and *Flare*. Each tool's strengths and weaknesses were examined across the criteria described above, offering insights into their suitability for diverse applications and providing potential users with a thorough understanding of the comparative advantages and limitations of each tool. For each criteria tools were categorised into one of three levels: *Satisfactory, Requires improvement*, or *Unsatisfactory*. Following are the analyses for each tool, which are summarised in Figure 2.1:

2.4.1. FEDn

FEDn (Ekmefjord, et al., 2022) exhibits a satisfactory level of node setup, allowing users to configure and connect nodes efficiently. QuickStart simplicity is another strength. However, challenges arise in authentication, implying potential vulnerabilities in securing communication between nodes. The tool demonstrates commendable capabilities in horizontal learning; however, it falls short in vertical learning, constraining its applicability to specific data partitioned scenarios. FEDn's technical expertise requirements need improvement, although its built-in debugging tools and software testing suites are satisfactory. While QuickStart simplicity meets the required standard, a more robust framework for advanced features and improved security would enhance its versatility. In terms of cloud native FEDn is containerised with a plan to move to Kubernetes.

2.4.2. IBMFL

IBMFL (Ludwig, et al., 2020) excels in node setup, providing users with a streamlined process for integration. Built-in debugging tools stand out as a strength, facilitating efficient issue resolution. However, language support limitations hinder its adaptability to diverse programming environments. Authentication and technical expertise require improvement, suggesting potential vulnerabilities and a steeper learning curve. While horizontal learning capabilities are satisfactory, vertical learning and software testing suites fall short. IBMFL does not meet any of the cloud native requirements. IBMFL's strengths lie in scenarios where seamless integration and efficient debugging are prioritized over certain advanced features.

2.4.3. OpenFL

OpenFL (Reina, et al., 2021) demonstrates satisfactory performance in node setup and horizontal learning. However, challenges in language support and vertical learning limit its adaptability to diverse scenarios. QuickStart simplicity, built-in debugging tools, and software testing suites require improvement, impacting user-friendliness and overall reliability. OpenFL is containerised but does not use kubernetes. Authentication and technical expertise also need enhancements. OpenFL's strengths lie in projects where a simplified federated learning setup is acceptable, and users prioritise basic functionalities over advanced features.

2.4.4. PySyft

PySyft (Ziller, et al., 2021) supports both horizontal and vertical learning capabilities, making it well-suited for different data partitioned scenarios. Technical expertise is a strength, offering users a sophisticated framework for complex machine learning models. However, language support, manual node setup, built-in debugging tools, and documentation quality require improvement. Software testing suites fall short, potentially impacting the overall reliability of the tool. PySyft meets all of the cloud native requirements. PySyft's emphasis on advanced learning capabilities positions it as a powerful choice for projects where users are willing to invest in technical expertise and complex machine learning models.

2.4.5. Flower

Flower (Beutel, et al., 2020) demonstrates proficiency in horizontal learning, establishing a robust foundation for collaborative learning across nodes. Documentation quality is a strength, ensuring users have comprehensive guidance. However, challenges in authentication, language support, vertical learning, and built-in debugging tools impact its overall usability. Manual node setup and software testing suites also require improvement. Flower is containerised with Kubernetes implementation planned. Flower's strengths lie in scenarios where a simplified federated learning setup is acceptable, and users prioritise a tool with comprehensive documentation and a straightforward learning curve.

2.4.6. AusCAT

Locally developed, AusCAT (Field , et al., 2022), is not currently a true open-source platform but components of AusCAT are open source with the goal for the platform to be more broadly open source. AusCAT demonstrates satisfactory performance in node setup and language support, providing users with a foundation for integration. However, challenges in authentication, vertical learning, technical expertise, QuickStart simplicity, and software testing suites impact its overall suitability for more complex projects. Satisfactory documentation provides users with guidance, but improvements in security features and advanced capabilities are crucial for broader applicability. AusCAT is containerised but does not use kubernetes. AusCAT's strengths lie in projects where simplicity and ease of understanding take precedence over advanced functionalities.

2.4.7. Vantage6

Vantage6 (Moncada-Torres , Martin, Sieswerda, Soest, & Geleijnse, 2021) showcases strengths in authentication, ensuring secure communication between nodes. Node setup, language support, horizontal learning, and documentation quality are also well implemented. It demonstrates a particularly strong performance in vertical learning. However, challenges in technical expertise, QuickStart simplicity, and software testing suites highlight areas for improvement. Vantage6 meets all the cloud native requirements. Vantage6's emphasis on security features and satisfactory documentation positions it as a potential choice for projects where robust security is paramount, and users prioritise comprehensive documentation for implementation.

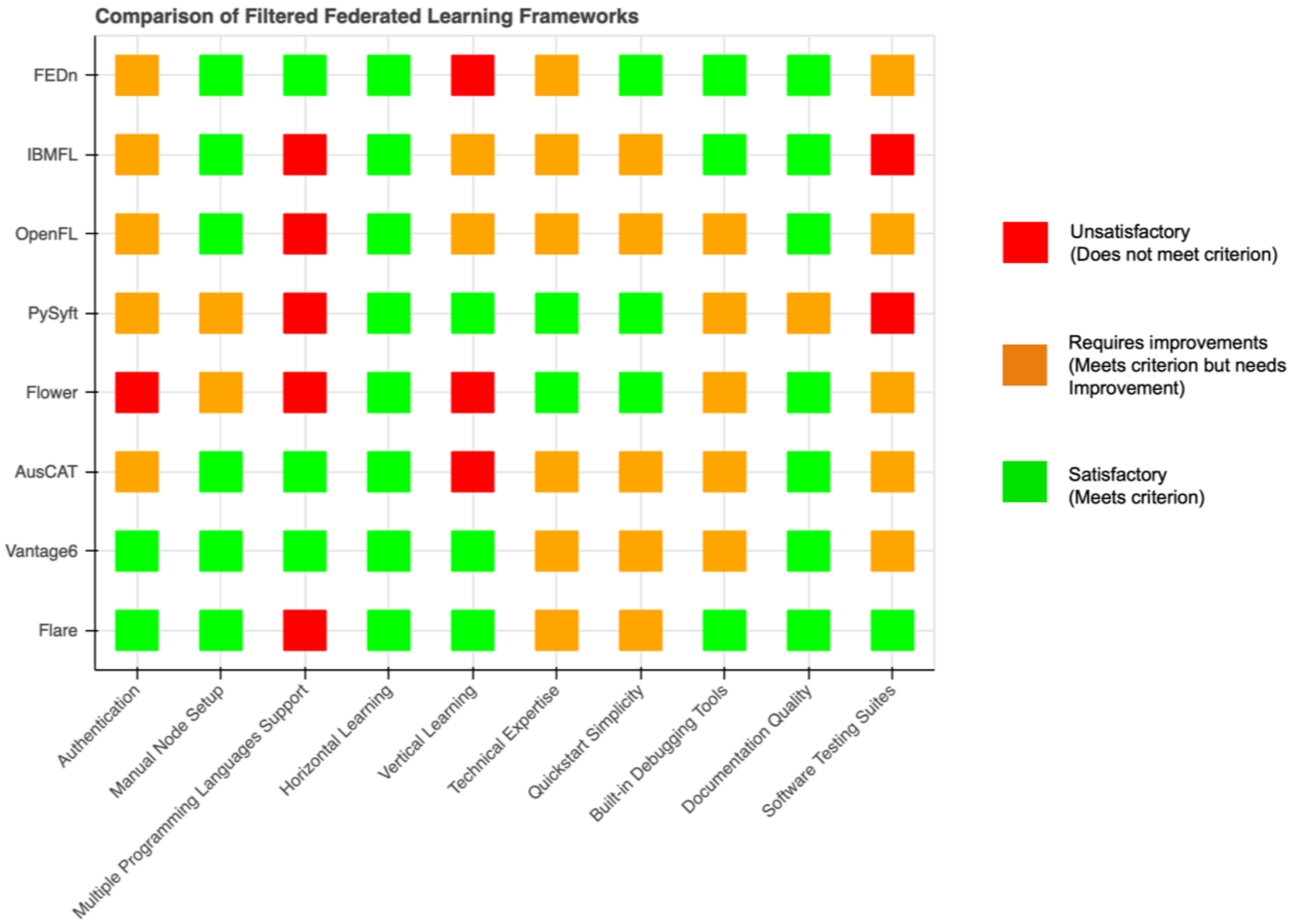


Figure 2.1 Comparison of FL frameworks against different criteria

2.4.8. Flare

Flare (Roth, et al., 2022) demonstrates satisfactory performance in authentication, node setup, horizontal learning, and documentation quality. However, language support, vertical learning, technical expertise, QuickStart simplicity, and software testing suites fall short. Despite these limitations, Flare's strengths in certain usability aspects make it suitable for projects where simplicity and horizontal learning are prioritized over advanced capabilities. Flare is containerised but does not use kubernetes. Users valuing a tool with a straightforward learning curve may find Flare to be a viable option, provided they can accommodate its limitations in other areas.

3. Open-Source Tool Deployment

Based on the initial overview presented in Section 2, three of these available tools were deployed to review the practicalities of deploying these tools using NECTAR. A summary of these deployment experiences is presented here.

3.1. Nvidia Flare

3.1.1. Quick start development

Flare is a tool backed by Nvidia with lots of effort poured into its development and maintenance.

With this, the QuickStart documentation is straight forward to follow, as a researcher or developer may setup a simulated federated learning environment very easily using Flare's Proof-of-concept (POC) command line interface. A dummy "workspace", Flare's concept of a directory for managing an entire federated learning, is setup and ready to perform an example task using a public dataset. This is both advantageous to federated learning researchers who wish to quickly experiment with ideas rapidly using the Flare tool without the need to create dummy virtual machines or perform tedious tasks for simulating a virtual federated learning network as are required with Flower's virtualised tooling for dummy federated learning networks. Additionally, POC environment allows developers to test new features that can be added to the Flare toolkit and streamline new features into the tool easily.

3.1.2. Real world deployment

As mentioned, this tool is backed by Nvidia and as such, a huge effort has gone into making the documentation clear for the tool for many things, including real world implementation of federated learning using Flare. A section dedicated to this can be found on their website.

We were successful in re-creating a federated learning system on the NECTAR cloud platform using Flare, by following their documentation step by step. They provide many tools to easily facilitate "production grade" setups that would otherwise require developer knowledge of handling this from an end-user point of view. These include Flare's provisioning module that handles the authentication and authorisation steps that are required to ensure that the identity of those contributing to the federated learning network is clear and transparent to the central server. Roles and different/custom levels of authorisation can be created using Flare directly allowing for a fine-grained control of participation. There is a technical overhead with understanding how to maintain these different options and interface with it, but it is appropriate for those with sufficient software engineering skills to understand and maintain.

As this is an open source product with the backing of a large company, this tool is very promising as many publications and events have been conducted on using Flare and its integration into the ecosystem of other AI tools such as Clara.

3.1.3. Unique features

- Dashboard for provisioning
- POC command line interface
- Backing and maintenance from Nvidia

3.2. Vantage 6

3.2.1. Quick start development

Vantage6 is a tool developed by The Netherlands Comprehensive Cancer Organisation (IKNL), who are interested in using federated learning to solve problems and conduct research questions into radiotherapy problems and challenges.

Comprehensive documentation exists for this tool, providing information on setting up the server and clients. This can be done using command line interface tools to setup a simulated server and clients to perform an example task. Similar to Flare, a POC tool can be used to quickly create a federated learning network. Docker is required to obtain base code for a deployable server and client (termed as “node” in Vantage6), unlike Flower and Flare where Docker images are not necessarily required for deployment.

This is advantageous to federated learning researchers who wish to experiment with ideas rapidly using the Vantage6 tool without the need to create dummy virtual machines or perform tedious tasks for simulating a virtual federated learning network. Additionally, this POC environment allows developers to test new features that can be added to the Vantage6 toolkit and streamline new features into the tool easily.

3.2.2. Real world deployment

Real world development of this tool is not as straight forward as Flare’s dedicated real-world deployment sections but can be achieved using the documentation throughout the tool’s website.

We were successful in re-creating a federated learning system on the NECTAR cloud platform using Vantage6. Tools are provided to easily facilitate “production grade” setups that would otherwise require developer knowledge of handling this from an end-user point of view. These include a dashboard for handling the setup of different components in their federated learning architecture which runs a docker container and can interface with the central server through its API, such as managing federated learning user authorisation, interfacing with encryption and API keys to easily manage this on a client level and monitoring learning tasks in the wider network. There is a technical overhead with understanding how to maintain these different options and interface with it, but this is at an appropriate level for those with software engineering skills to understand and maintain.

3.2.3. Unique features

- Dashboard for handling authorisation/authentication and monitoring the federated learning network.
- API endpoints that can be called using HTTP requests, not just Python clients.

3.3. Flower

3.3.1. Quick start development

Flower is a federated learning framework that supports large-cohort training and evaluation, both on real edge devices and on single-node or multi-node compute clusters. The quick start documentation is very easy to follow. It is designed with simplicity in mind, offering an intuitive and user-friendly interface for setting up and managing federated learning experiments. Its lightweight coordination server and straightforward API make it easy for developers to integrate Flower into their existing machine learning pipelines with minimal effort. It can be simulated on a single machine using Python files, without the need of any containerisation tool as mentioned at their website. Further, it abstracts away much of the complexity involved in building and deploying federated learning systems, allowing developers to focus on model design and optimisation rather than low-level implementation details. By providing high-level abstractions for tasks such as model aggregation, communication, and synchronisation, Flower simplifies the development process and accelerates the iteration cycle for federated learning experiments.

3.3.2 Real world deployment

We have successfully re-created a simulation environment on a single machine and on the NECTAR cloud platform as well. It offers built-in datasets for simulation purposes, alongside the flexibility to use custom datasets. Users can define various machine learning models such as logistic regression and neural networks for training, employing a client class to train the model on the training dataset and evaluate it on the testing dataset. For the server-side, users can choose the specific averaging techniques for the aggregation. Flower enables SSL for establishing secure connections between servers and clients, with comprehensive guidance on starting an SSL-enabled secure Flower server and connecting Flower clients securely, alongside a complete code example, although users are advised to consult the guide for in-depth SSL setup instructions. For Docker containerisation, it offers two images – a base image containing essential dependencies shared by both server and client, and a server image built upon the base image, which installs the Flower server via pip.

3.3.3. Unique features

- Ease of Use and Deployment
- Reduced Complexity in Development

3.4. PySyft

3.4.1. Quick start development

PySyft is an open-source federated learning library developed by OpenMined. It aims to make private machine learning accessible by enabling secure and privacy-preserving data analysis. PySyft extends popular machine learning frameworks such as PyTorch and TensorFlow. The documentation is comprehensive and user-friendly providing clear guidance on setting up and managing federated learning experiments.

PySyft facilitates secure multi-party computation (SMPC) and differential privacy ensuring an extra layer of privacy is maintained throughout the learning process. It provides high-level abstractions for tasks such as secure aggregation, encrypted communication, and differential privacy, simplifying the development process and allowing researchers to focus on model design and optimization.

3.4.2. Unique features

- Ease of Use and Deployment
- Support for Vertical data partitioning as well as horizontal

4. Data storage and communication

4.1. Data Storage and Computational Requirements

A diagram for a typical federated learning use case is provided in Figure 4.1. This highlights the resource components that are required for federated learning.

The key resource requirements for federated learning are data storage and computational capacity at each of the nodes, a server system with computational capacity and the ability to communicate between these components. Given the reason for using federated learning is often to ensure security and privacy of data, these requirements are also likely to impact resource requirements.

The magnitude of data storage requirements and computational capacity will vary from project to project depending on the algorithms and data that are being used and the models being developed. Different scenarios for the node/data storage situations are considered here:

4.1.1. Node/s positioned within an organisational IT environment

A common situation is that the data at each node would remain behind an organisational firewall (e.g. a hospital). In this situation the data storage and computational capacity must be provided as part of, or at least linked to, the infrastructure of the organisation. The requirements for the data storage and computational capacity must also meet the local organisational requirements as well as those for federated learning. This may make it challenging or impossible to be able to utilise broadly available research data infrastructure (e.g. nectar and MLeRP in its current form). To enable communication in this situation the ability to communicate outside the organisation through the firewall must be addressed. This would commonly require 'white-listing' of relevant sites.

4.1.2. Node/s positioned within a Trusted Research Environment (TRE)

In some instances, for example where registry linked data is involved, data is stored in a trusted research environment (TRE). A TRE can also be known as a secure research environment (SRE), data safe haven or secure data environment.

A TRE is controlled computing infrastructure designed to facilitate secure research practices while safeguarding sensitive data. It serves as a centralised platform where researchers can access and analyse sensitive information without compromising privacy or security. Key characteristics of TREs include robust data encryption, stringent access controls, comprehensive logging and monitoring systems to track user activities and detect any unauthorised access and curated gateways. These environments often comply with relevant regulations and standards, such as GDPR (Goddard, 2017) in the European Union or HIPAA (Chen & Benusa, 2017) in the United States, to ensure data protection and privacy

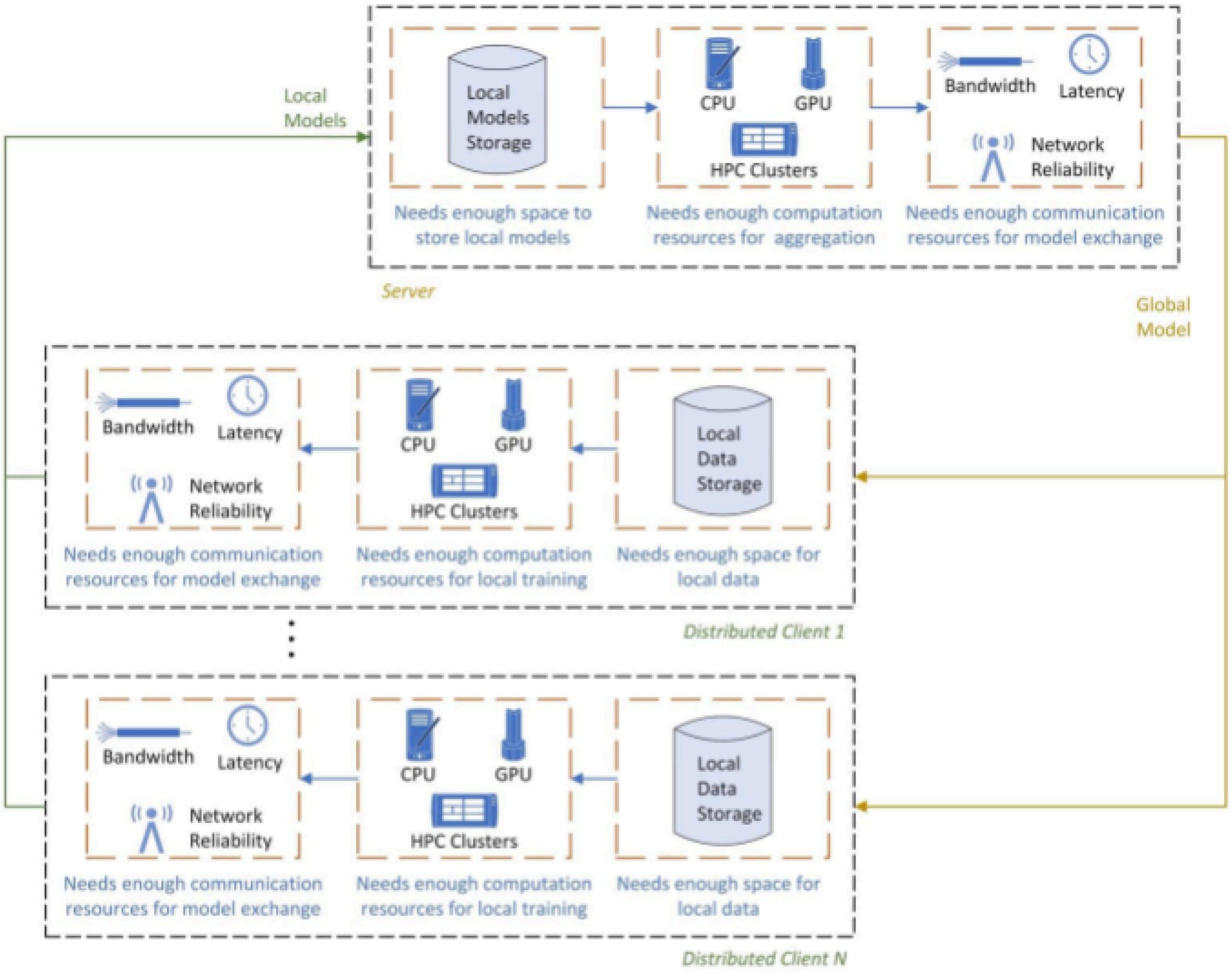

Local Models
Local Models Storage
CPU
GPU
HPC Clusters
Bandwidth
Latency
Network Reliability
Needs enough space to store local models
Needs enough computation resources for aggregation
Needs enough communication resources for model exchange
Server
Global Model
Bandwidth
Latency
Network Reliability
CPU
GPU
HPC Clusters
Local Data Storage
Needs enough communication resources for model exchange
Needs enough computation resources for local training
Needs enough space for local data
Distributed Client 1
Bandwidth
Latency
Network Reliability
CPU
GPU
HPC Clusters
Local Data Storage
Needs enough communication resources for model exchange
Needs enough computation resources for local training
Needs enough space for local data
Distributed Client N

Figure 4.1. Overview of resource requirements in terms of storage, computation and communication in a federated learning process.

compliance. Examples of TREs implemented in various countries include the UK Secure Research Service (SRS) (ONS, 2024), Secure eResearch Platform (SeRP). In Australia, TREs include Secure Unified Research Environment (SURE) (Moore, Guiver , Woollacott, Klerk, & Gidding, 2016), E-Research Institutional Cloud Architecture (ERICA) (ARDC, 2024), KeyPoint, or Monash SeRP. A common requirement for these secure environments is that there is manual inspection of data ingress and egress. This is a particular challenge if data must be stored in a TRE in a federated learning network.

There isn't a one-size-fits-all definition for what constitutes a TRE; rather, design decisions are tailored to meet the specific needs of each organisation. The Five Safes Framework has emerged as a cornerstone guiding principle within this realm. Ensuring safe projects underscores the ethical utilisation of data, necessitating projects with clearly defined purposes. Access to data is restricted to authorised and reliable individuals (safe people), who undergo rigorous checks, and receive training in data privacy. Data must be adequately safeguarded, including measures such as de-identification to prevent privacy breaches (safe data). Safe settings govern the data environment, demanding secure IT infrastructures and protocols (safe settings). Meanwhile, safe outputs guarantee that sensitive information remains undisclosed, aligning with standards set forth by regulatory bodies like the Australian Bureau of Statistics regarding data publication.

The Five Safes approach offers flexibility, empowering data custodians to evaluate the risks and potentials associated with data sharing and release. Typically, TRE administrators oversee safe settings, while stakeholders collectively share responsibility for ensuring the other four aspects (safe projects, safe people, etc.). However, governance within the medical domain poses unique challenges, particularly concerning the integration of health data. The intended flexibility of the Five Safes framework encounters constraints due to the stringent security protocols imposed by data providers. Consequently, many custodians err on the side of caution, implementing top-tier security measures across all dimensions, which may prove excessive for certain specific purposes.

There are requirements for particular datasets to be stored within a TRE. Utilisation of federated learning can provide an opportunity to learn from the datasets that must be stored in a TRE without requiring combining of the entire dataset which may not be possible. Using horizontal, vertical or a combination of horizontal and vertical (as described in Section 1) federated learning different datasets can be utilised. For instance, in healthcare research, a TRE/SRE might contain patient records from one geographical region, while other nodes hold data from other regions, ensuring data diversity without sharing sensitive patient information across nodes. in genomic research, a TRE/SRE might hold genetic sequences, while other nodes hold phenotypic data or clinical outcomes. In financial research, a TRE/SRE might contain transactional data while other nodes hold demographic or socio-economic information. This partitioning strategy allows for collaborative analysis without exposing individual-level data across nodes, thus maintaining privacy and security.

4.2. Integrating Federated Learning within TREs/SREs

The integration of federated learning within TREs or SREs poses a significant challenge due to the common requirement for manual inspection of data ingress and egress within these secure environments. Federated learning, being an iterative process that often spans multiple rounds of sharing the model parameters, necessitates seamless data flow between the participating devices or servers. However, the stringent security protocols of TREs/SREs mandate manual inspection of data ingress and egress for each round, which may not be feasible in certain applications of federated learning. Importantly this manual inspection process is set-up for reviewing data. In federated learning it is models and not data which are being transferred into and from TREs and the manual inspection process is rarely appropriate for assessing risks of model transfer. This manual inspection also introduces potential bottlenecks and delays, hindering the efficiency and scalability of the federated learning process within these secure environments. Moreover, the repetitive nature of manual inspection increases the risk of human error and may compromise the timeliness and accuracy of research outcomes. Therefore, there is a pressing need to explore alternative solutions or enhancements to streamline the integration of federated learning within TREs/SREs, ensuring both data security and research efficiency are effectively balanced.

4.3. Automatic Inspection of Data for TREs

The primary solution to address the challenge of manual inspection of data ingress and egress for federated learning within TREs/SREs is the implementation of automatic inspection systems. Automatic inspection refers to the process of using advanced technological systems and algorithms to monitor, analyse, and detect patterns or anomalies in data flows without the need for manual intervention. Automated approaches can also be more appropriate for review than manual review processes set-up for reviewing data and non-ideal for reviewing models. Within the context of TREs/SREs, automatic inspection systems could play a crucial role in ensuring the security, privacy, and compliance of research activities, particularly in scenarios such as federated learning where data ingress and egress occur iteratively over multiple rounds. By employing automated tools, organisations can streamline the inspection process, reduce the risk of human error, and enhance the efficiency of data monitoring and analysis. Moreover, automatic inspection systems enable real-time detection of suspicious activities or deviations from expected behaviour, allowing for prompt intervention and mitigation of security incidents.

Following are a few of the factors that need to be addressed in the implementation of automatic inspection:

4.3.1. Collaborative governance

Collaborative governance models involve establishing frameworks where stakeholders from various domains, including researchers, data custodians, and security experts, work together to govern and oversee the implementation of processes and policies within TREs/SREs. These models ensure that decisions regarding data access, security protocols, and compliance measures are made collectively, taking into account the perspectives and expertise of all involved parties.

By involving stakeholders in the governance process, transparency is fostered regarding the objectives, methodologies, and outcomes of automatic inspection systems. Transparency helps build trust among stakeholders and ensures that all parties understand the rationale behind the implementation of automated inspection processes.

Collaborative governance models establish clear lines of accountability, ensuring that responsibilities for overseeing and managing automatic inspection processes are clearly defined. This accountability helps mitigate risks and ensures that any issues or concerns related to the implementation of automated inspection systems are addressed promptly and effectively.

Involving stakeholders in the governance of automatic inspection processes enables efficient decision-making and implementation. By leveraging the collective expertise and insights of researchers, data custodians, and security experts, governance models can streamline workflows, expedite approval processes, and optimize resource allocation, leading to increased efficiency in implementing and managing automated inspection systems. Lastly, collaborative governance models facilitate proactive risk management by enabling stakeholders to collectively identify, assess, and mitigate risks associated with automatic inspection processes. By bringing together diverse perspectives and expertise, governance models help organizations anticipate potential challenges and develop comprehensive risk mitigation strategies to safeguard data integrity, confidentiality, and compliance within TREs/SREs.

4.3.2. Streamlined approval processes

Streamlined approval processes within TREs/SREs involve integrating efficient procedures for approving data ingress and egress requests with automated inspection systems, enhancing the efficiency and accuracy of data monitoring and analysis. By implementing pre-approved templates, checklists, or protocols, organizations can expedite the review of data flows while ensuring alignment with security and compliance requirements. This approach standardizes the evaluation criteria, facilitates expedited review, and enhances oversight and governance of data activities within TREs/SREs. Integrated with automated inspection systems, streamlined approval processes optimize workflow efficiency, ensure consistency in data analysis, and enable stakeholders to promptly identify and address any anomalies or deviations from expected behaviour, thereby strengthening the security and integrity of research activities conducted within secure research environments.

4.3.3. AIML enabled inspection of data flow

Machine learning models are computational algorithms trained to recognize patterns and make predictions based on data. Within TREs/SREs, these models play a crucial role in automating the inspection of data ingress and egress. By training on historical data within TREs/SREs, machine learning models (supervised and unsupervised) can predict and classify normal and abnormal patterns in data flows. They can detect anomalies or deviations from expected behaviour, automatically analysing data in real-time and flagging any deviations warranting further investigation. This automation enhances the efficiency and accuracy of data monitoring and analysis within TREs/SREs compared to manual methods, which are time-consuming and error prone. Additionally, machine learning models adapt and evolve over time, continuously improving their accuracy and effectiveness in detecting anomalies, thus ensuring TREs/SREs remain resilient against emerging security threats.

4.3.4. Privacy preservation

Privacy-preserving techniques such as differential privacy, secure multi-party computation, and homomorphic encryption provide an extra layer of security in the context of automatic inspection within TREs/SREs. While not directly assisting in the automation of inspection processes, these techniques ensure that sensitive data remains confidential and protected throughout automated analysis. These privacy-preserving techniques guarantee that data ingress and egress undergo inspection without compromising privacy, thus safeguarding sensitive information throughout the automated analysis process. Differential privacy (Zhang, Lu, & Liu, 2023) adds noise to data before analysis to prevent individual records from being identifiable. It ensures that the output of automated analysis does not compromise the confidentiality of underlying data, thereby enhancing overall security. Secure Multi-Party Computation (SMPC) (Mansouri, Önen, Jaballah, & Conti, 2023) (Fereidooni, et al., 2021) enables multiple parties to compute functions over their inputs while keeping those inputs private. SMPC allows for collaborative analysis across multiple nodes without exposing sensitive information, thereby bolstering security during automated inspection. Homomorphic encryption (Wibawa, Catak, Sarp, & Kuzlu, 2022) enables computations on encrypted data without decryption. It ensures that sensitive data remains confidential during automated analysis, adding an additional layer of security to the process. Figure 4.2 illustrates the incorporation of secure aggregation in federated learning via homomorphic encryption and differential privacy.

4.3.5. Conditions in favour of federated learning

We have seen that TREs, by design, include stringent security protocols, particularly concerning the manual mediation of data egress, which can inhibit their direct participation in federated learning. On the other hand, federated learning requires continuous interaction among nodes for model updates, which poses a challenge for TREs due to their reliance on human-mediated data flows. At every iterative step, every federated learning node sends not the data but the model parameters to the central server which determines resolution on the common model. This requires seamless information flow between devices or servers. However, the security protocols of TREs require manual inspection of data ingress and egress for each round, which can be impractical at the frequency of interaction required by federated learning in addition to increasing the risk of any potential human error. Besides, the current governance structure for TREs also would make it unsuitable for federated learning.

On the other hand, federated learning actually poses less risk than federated analytics, as the data is not exchanged, and the human element is removed. This needs to be recognised in the governance structure.

Currently, there are some promising developments in Federated Analytics in projects like FED-NET, which involves periodic and less frequent data exchanges where manual oversight is more feasible. However, this springboard holds the potential to evolve and accommodate federated learning, particularly with enhancements to support automated data mediation and real-time interaction.

The current capabilities of platforms like TelePort and TRE-FX illustrate potential pathways for TREs to support federated learning in the future. These platforms are designed with multi-toolkit frameworks that facilitate data governance and collaboration across different TREs without altering existing governance structures.

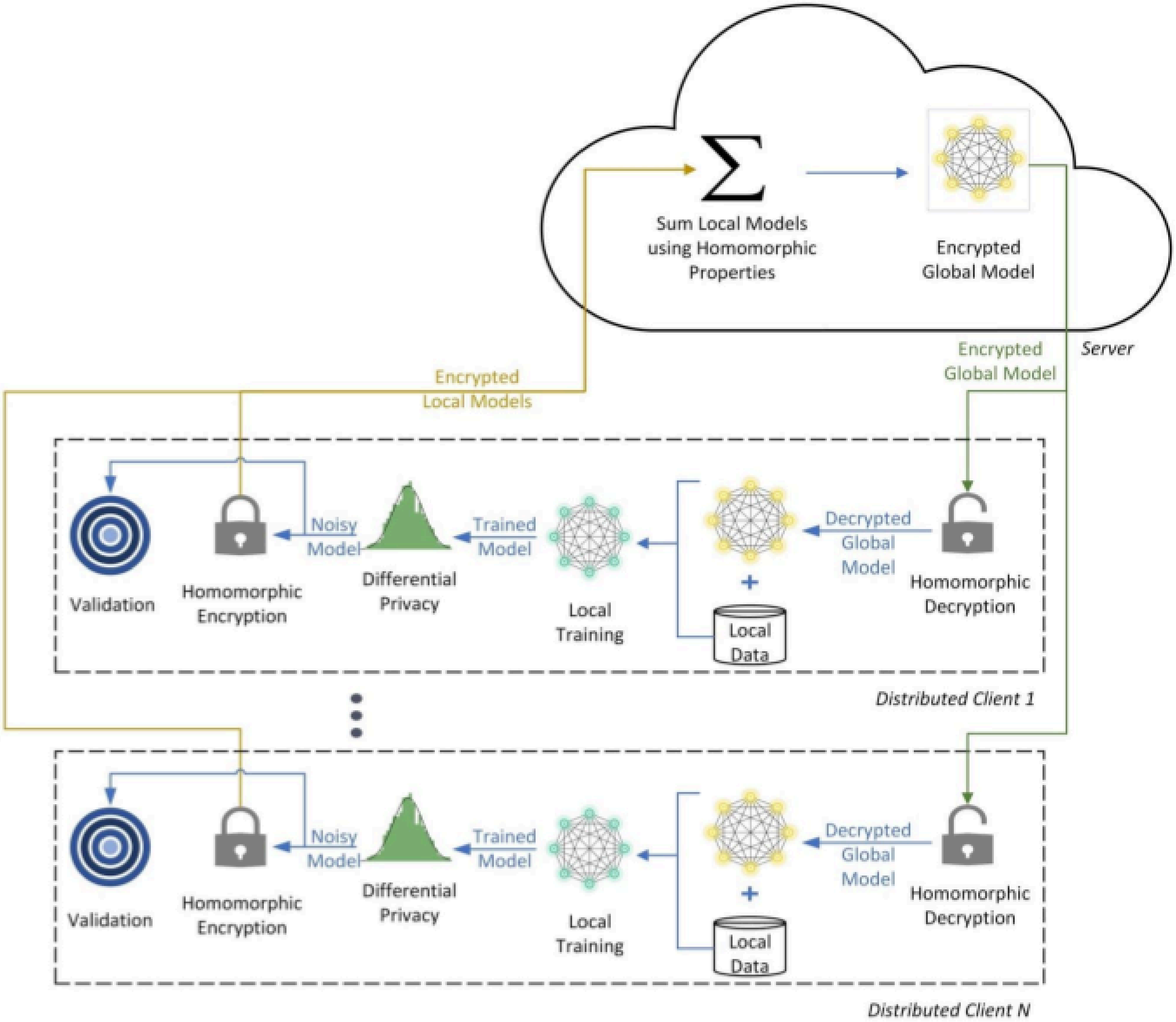


Figure 4.2. Overview of federated learning framework incorporating secure aggregation via homomorphic encryption and differential privacy.

TelePort, for instance, creates an ephemeral common space for data interaction among TREs, governed by existing egress rules. According to HDR UK, Trino abstracts the individual database layers and TELEPORT creates an interoperable “link” between each TRE, allowing each Trino instance to communicate seamlessly. This setup facilitates secure data sharing and collaborative analysis across different TREs, while TRE-FX enhances the governance framework by providing standardized egress processes and ensuring that data sovereignty and privacy requirements are consistently met across all nodes.

By abstracting the computation layer and providing a connected space for researchers to operate on and access data in different environments. While the current structure is available only for Federated Analytics, this setup may pave the way for collaborative federated learning projects without compromising data security within this space.

Given the significance and complexity of the area, a separate TRE project is being undertaken by ARDC. The project brings together a panel of TRE groups on a workshop to explore the key challenges and way forward and is also exploring the possibility of local TRE groups collaborating with overseas projects such as HDR UK. The interim report can be accessed at online [TRE Framework Report].

4.4 Secure Network of Servers as an Alternative to TREs

To achieve federated learning among secure nodes, it is possible to create a secure network of servers, each embedded in different data hosting locations. This network of servers would create a secure space where federated learning can be carried out without the data leaving the premises or jurisdiction.

Taking a leaf from TelePort and TRE-FX projects as well as upcoming Australian projects such as AIS-SHIELDS and FLERA (described later in this report), establishing such a secure network of servers involves creating an ephemeral space for secure data interaction. However, it has one less requirement: human intervention is needed only during the initial data loading and the final extraction of results (not the data). This reduction in human involvement during the machine learning process enhances security by minimizing the potential for human error and unauthorized access.

5. Use Cases for Federated Learning

5.1. Existing Implementation Case Study: Australian Cancer Data Network (ACDN)

5.1.1. Background

'Cancer is responsible for Australia's largest disease burden and is a leading cause of death (Australia, 2024)'. There are challenges in accessing and thus learning from Australian cancer data which is stored in detail at local institutions including hospitals and clinical trial organisations and in silos with state-based registries. Providing evidence to support decisions on the most effective form of treatment for individual patients can be challenging, particularly for patients who do not meet the eligibility criteria for randomised clinical trials that form the backbone of practice guidelines. The ability to harness Australia's cancer data, which includes both tabular items (e.g. age, disease stage), imaging (e.g. CT, MRI), omics and other specific data types (e.g. radiotherapy treatment dose distributions) has the potential to enable learning and generation of additional evidence for our patients and clinicians.

The Australian Cancer Data Network (ACDN) is a collaboration from three platforms, seen graphically in Figure 5.1. This includes 'AusCAT', a federated learning platform developed initially in collaboration with a team from MAASTRO clinic (Field M. , et al., 2021), The Netherlands and the Australian radiation oncology community; Cancer Alliance QLD (QLD, 2024), a collaborative organisation across health services, jurisdictions and organisations in QLD with the goal of supporting clinician-led service improvement, harnessing and making available cancer data; and CaVa, a research program working to make available clinical practice datasets in a researcher ready format to investigate variations in cancer treatment. The specific datasets include clinical practice data from treatment centres, registry data and clinical trial datasets. Together our collaboration is using federated learning to learn from large and diverse cancer datasets.

5.1.2. Governance

The governance of the three collaborating platforms in the Australian Cancer Data Network are all managed separately; however, the governance of the network as a whole is coordinated by a central executive committee with representatives from each of the platforms and supported by clinical, technical, data and translational expert panels.

Governance for the federated learning work relies on an overarching ethics protocol with approval to utilise data at each of the contributing nodes for the purpose of undertaking combined analysis and model development. Sub-projects asking particular research questions are included within the ethics protocol or may have other governance arrangements (e.g. legislative approval). Each of the institutions/nodes involved can choose which sub-projects they are or aren't involved in.

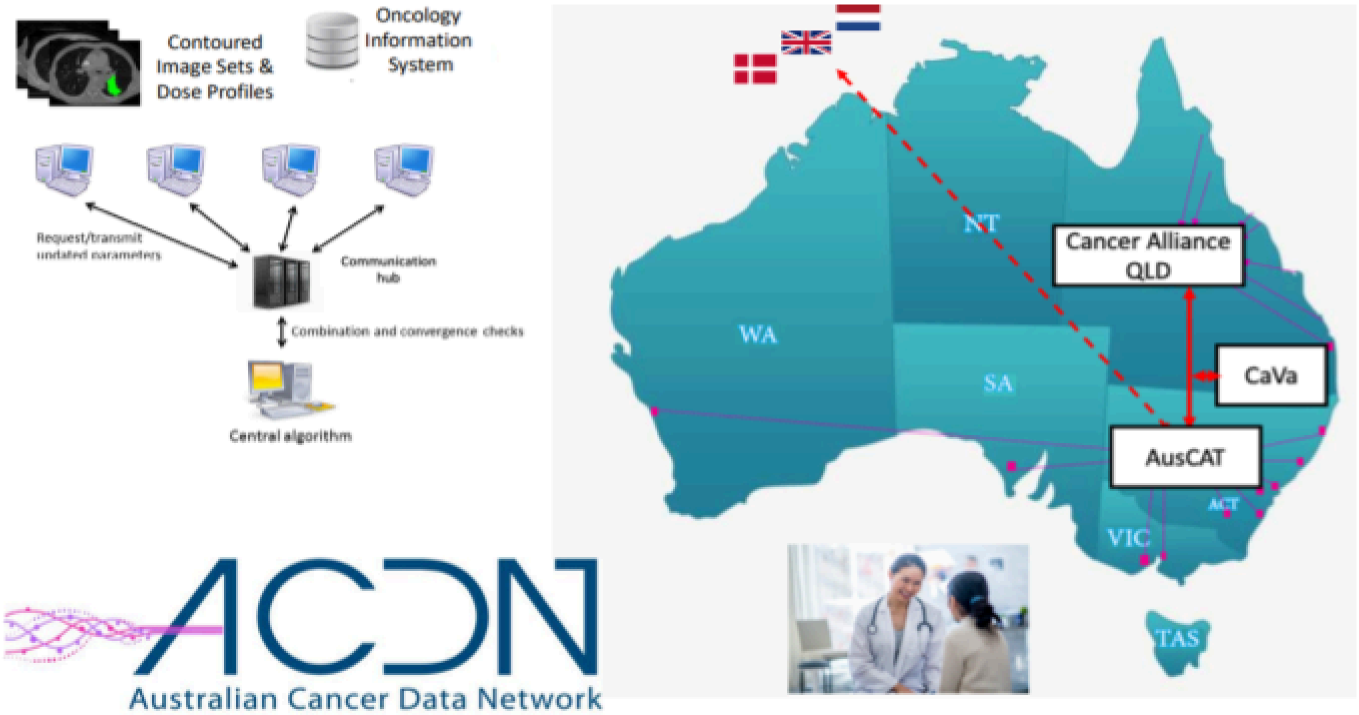

Contoured Image Sets & Dose Profiles
Oncology Information System
Request/transmit updated parameters
Communication hub
Combination and convergence checks
Central algorithm
NT
WA
SA
VIC
ACT
TAS
Cancer Alliance QLD
CaVa
AusCAT
ACDN
Australian Cancer Data Network

Figure 5.1 The Australian Cancer Data Network (ACDN) is a collaboration of three platforms, Australian Computer Assisted Theragnostics (AusCAT), CaVa (exploring unwarranted cancer variation) and Cancer Alliance QLD.

5.1.3. Federated learning infrastructure

A custom software platform for distributed learning was developed for the AusCAT network (Field , et al., 2022). The AusCAT node infrastructure includes two main parts:

Firstly, there are components for setting-up databases at the nodes, with a pipeline of data extraction to generate a de-identified dataset and a key database that contains the identifiers. The nodes are at hospitals around Australia and data storage and computational power is provided by the hospitals either hardware or in the cloud with the systems managed within hospital IT infrastructure (and appropriate firewalls). The project is working towards setting up nodes for registry datasets (which would be vertically partitioned in comparison to the hospital datasets which are horizontally partitioned). This requires addressing both governance and data storage requirements (the need for TRE/SREs).

Secondly, infrastructure enables federated learning. This uses Java web services to coordinate communication between clinic systems. Algorithms can be sent to each clinic, where they generate and share model parameters and statistics with the central server and then through iterative transfer of parameters across the clinics and the server, develop the final model. This has been used for horizontal federated learning. The project has demonstrated proof of principle with vertical and combined learning but have not yet implemented this on the ACDN network.

While the federated learning components of AusCAT have proven effective (Hansen, et al., 2022) (Field M. , et al., 2024) a number of open-source platforms as described above are now available and maintaining and expanding this proprietary federated learning platform requires significant resources. By adopting open-source tools, we can leverage existing technologies without the need for extensive in-house development, ensuring that we align with the broader research community. Open-source platforms like Flower offer robust, community-supported solutions that facilitate interoperability and collaboration. This move will ensure ACDN stays at the forefront of federated learning advancements, benefiting from shared innovations and maintaining compatibility with widely used frameworks.

5.1.4. Specific example cases (including how training, validation and testing is completed)

Non-small cell lung cancer survival following radiotherapy treatment. This investigation (Field M. , et al., 2024) developed a survival model for non-small cell lung cancer patients using federated learning across 6 centres in NSW. This was a linear regression model with data split based on time-period. Data from 2011-2016 was used for bootstrap training and internal validation and data from 2017-2019 was used for validation. This split in data was used to ensure that the model was validated on the most recent data as is most useful for considering the clinical applicability of the model. The data used was federated for both the training and the validation.

Cardiac toxicity model following radiotherapy treatment. A current project is working towards developing a cardiac-toxicity model following radiotherapy treatment. There is evidence that radiation dose to the heart increases the risk of cardiac toxicity (e.g. heart attacks) following treatment but there is limited evidence on how the distribution of dose affects this risk. In this project a developed cardiac segmentation algorithm is being used to determine the radiation dose to cardiac substructures using imaging and radiation dosimetry data available at individual centres. A combined model will then be developed using federated learning. Data will remain federated for both training and validation. A random split of data may be used to separate training and validation datasets or one or two centres may be separated as the validation cohorts.

Prognosis models for anal cancer. In this international study prognosis models are being developed for anal cancer (Theophanous, et al., 2022). Using federated learning enables access to a large dataset which would not otherwise be possible for anal cancer which is relatively uncommon. The data is remaining federated for training and validation. A separate external validation is also being undertaken with datasets from centres that were not involved in the original training and validation.

5.2. Existing Implementation Case Study: FLERA+

5.2.1 Background

Applied artificial intelligence (AI) research in health, and particularly in human imaging, is a transformative technology that will accelerate diagnosis, and facilitate precision management of a range of human diseases. Its success relies heavily on data availability during model development or clinical validation stages. Many roadblocks obstruct the integration of precision imaging into clinical decision-making. Technical, logistical and governance issues have prevented public and private health providers, often the custodians of real-world imaging datasets, from participating in cutting-edge applied AI research, which has remained largely within the domain of research institutes and technology companies.

In 2020, the MRFF-funded TRANSCEND (TRanslating AI Networks to Support Clinical Excellence in Neuro diseases). This project was established to overcome the bench-to-bedside roadblock by creating a permanent bi-directional interface between AI RRD and clinical practice. The TRANSCEND eco-system provides a rich federated learning environment for clinical applications and broad expertise to advance applied AI research, building upon the team's previous R&D work in the CRC-P project: "AI: new smarts for the medical imaging industry". FLERA (Federate Learning Ecosystem for Research in Australia) represents the natural evolution of TRANSCEND: the goal is to be the partner of choice for supporting the accelerated development and adoption of AI solutions in health that rely on federated learning for healthcare.

5.2.2 Outcome

FLERA comprises four critical capacities:

1. FLERA Experience: This encompasses the overall federated learning collaboration network and successful federated learning experiences of TRANSCEND, which can be referenced for new federated learning projects and facilitate multicentre AI collaborations within and outside the FLERA network.

2. FLERA Box: An end-to-end engineering solution designed for the rapid deployment of federated learning across stakeholders in health provider networks, ensuring operational efficiency and maximum performance. The engineering solution incorporates "requirements-design-evaluation" development cycle, which takes requirements from clients and provide support from aspects like performance target, hardware requirements, model design, federated training and evaluation. The FLERA Box has been tested with hospitals (including Royal Prince Alfred Hospital, St Vincent Hospital, Westmead Hospital, etc.) and data providers (including iMed Radiology, Synergy Radiology, Flinders University, etc.) on multiple applications.

3. FLERA AI Research: Focuses on themes that continually improve AI training efficiency, advancing the field of AI in health research. Previous research has covered multiple aspects in neuroimaging and neurological research and applications and to redesign the algorithms used in federated learning framework to enhance model performance. We've focused on real-world challenges and provided solutions when many data centres are involved, including labels with noise, lack of labels from participated centres, imaging inhomogeneity across data centres, and predicting performance requirements for given task.

4. FLERA Team: Led by the original PIs from TRANSCEND, this growing multidisciplinary team continues to excel in large-scale AI adoption in Australia. Currently led by Prof. Michael Barnett, Prof. Fernando Calamante, Dr. Chenyu Wang and Dr. Ryan Sullivan.

FLERA has translated and implemented AI technologies into health applications across multiple disciplines. In Multiple Sclerosis, we developed lesion models, LLM based prognosis models, and spinal cord assessment models, which have been made available to the MS research community through FLERA and MSBIR for improved disease progression monitoring. Additionally, we developed CT-based brain haemorrhage detection models for CT triage and brain tissue models for quantifying various brain diseases. Importantly, leveraging NVIDIA MONAI, we created a robust AI development pipeline that rapidly transforms imaging analysis tasks into AI-powered applications.

The research outcomes promote economic solutions for facilitating federated learning, preserving privacy in large-scale, multidisciplinary AI collaborations. We have 'packaged' all learnings from TRANSCEND project into its post MRFF funding cycle form: Federated Learning Ecosystem for Research In Australia, the FLERA program. The FLERA program comprises FLERA Teams, FLERA Box, FLERA Research, and FLERA Experience, offering a comprehensive ecosystem for AI innovation in health. This interdisciplinary collective includes all necessary expertise, AI models, tools, engineering solutions, governance, and most importantly, successful experience in large scale AI adoption in health.

5.3. Implementation Case Study: AIS-SHIELDS

5.3.1. Australian Imaging Service Background

The Australian Imaging Service (AIS) is a nationally federated platform for secure imaging data management and analysis, focusing on clinical and pre-clinical imaging modalities such as Magnetic Resonance Imaging (MRI), Computed Tomography (CT), Ultrasound, Positron Emission Tomography (PET), X-Ray, etc. AIS fully launched in 2022 and currently consists of 13 research institutions with funding from the Australian Research Data Commons (ARDC) and the National Imaging Facility (NIF) NCRIS capabilities. AIS integrates directly with clinical scanners for consenting patients, doing on-site de-identification of direct identifiers before uploading images to university nodes for long term curation, analysis, and collaboration. AIS's mission is to increase research reproducibility and drive the adoption of innovative but trusted analysis techniques.

Starting as an institutional initiative at the University of Sydney in 2017, the national Australian Imaging Service was created through the ARDC Platforms 2019 AIS Project with a network of central DVC-R and ICT teams across 7 Universities using the open source XNAT for imaging data management. AIS operates with core institutional support from the University of Sydney with a portfolio of research grants for feature enhancements. The original ARDC project focused on developing a standardized, secure, and scalable architecture built around XNAT and Kubernetes. AIS was subsequently extended in the ARDC Platforms 2020 AEDAPT Project adding secure virtual desktops built on Neurodesk (Renton, et al., 2024) and in the NIF 2021 AIS Pipelines Project building out the workflow engine built on ARCANA (Close, et al., 2020) and with a library of curated pipelines. The 2023 EU Horizon Infrastructure FoundingGIDE project is standardizing biological, preclinical, and clinical imaging ontologies used internationally while the MRFF NCRI AIS-SHIELDS project is adding NLP, AI Segmentation, and federated learning capabilities.

AIS uses a data centric computing model with all computational services tightly coupled with the data repository. This increases accessibility by allowing all tools to be accessed via a browser UI, reproducibility by using version-controlled software stacks so multi-site studies can use identical tools across the full duration of a study, and security by integrating computational data access and auditing managed by the data repository without data needing to leave AIS.

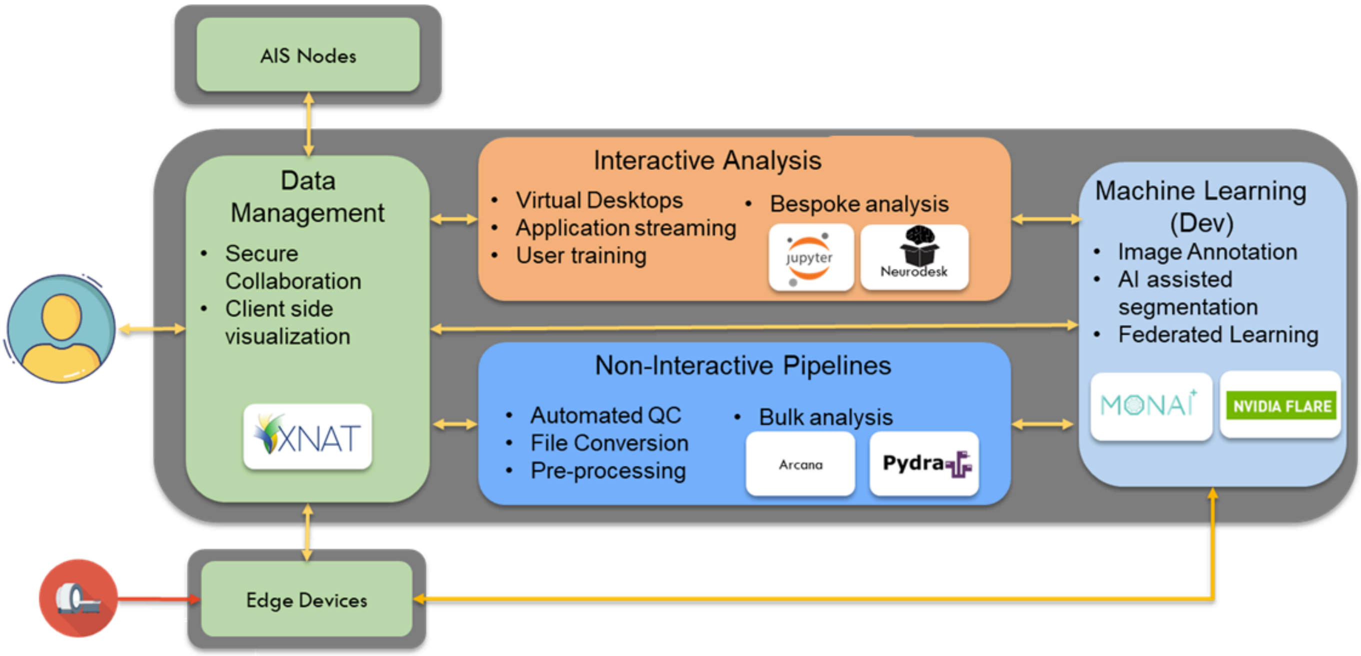


Figure 5.2. Overview of AIS

AIS currently consists of five key services, as shown in Figure 5.2.

1. **Data Movement:** Secure movement from image acquisition to repository, and between repositories, including de-identification, encryption, and routing

2. **Data Management:** Built around XNAT, this provides long term archival data management, with per project, per data type user access controls directly coupled with analysis platforms so data doesn't need to leave the platform

3. **Automated Pipelines:** Built around ARCANA/Pydra workflow engine and kubernetes schedulers, this provides the ability to run containerized workflows for bulk analysis, automated QC, file conversion, pre-processing, etc.

4. **Interactive Visualization and Analysis:** Built around JupyterHub and Neurodesk, this provides secure virtual desktops preloaded with reproducible imaging software.

5. **Machine Learning:** (Still in heavy development) Built around MONAI, this provides AI assisted image segmentation and classification by running PyTorch models directly integrated with image viewers.

5.3.2. ACRF Centre of Excellence in Melanoma Imaging and Diagnosis Background

The ACRF Centre of Excellence for Melanoma Imaging and Diagnosis (ACEMID) has been establishing a network of 16 Total Body Photography (TB-Photography) clinical scanners in urban and regional locations to create a national teledermatology network for detection, monitoring, and treatment of Melanoma and related diseases in partnership with QLD Health, NSW Health, VIC Health, and Melanoma Institute of Australia. TB-Photography offers an excellent and impactful imaging modality and will lead to major advances in the field of dermatology; however, it requires AIS's input and advanced capabilities as it produces images that are very sensitive and need extra protection to maintain patient's privacy. ACEMID has partnered with AIS to build the ACEMID Research Repository across AIS nodes at the University of Queensland, University of Sydney, and Monash University, complementing the national clinical teledermatology network.

At present, there is a significant two-fold gap in the maturity and progress of imaging and reporting standards in the field of dermatology compared to those found in radiology, especially related to diagnosis, monitoring, and treatment of melanoma and skin cancers. Firstly, individual imaging modalities are siloed, using non-standard formats and separate software platforms, precluding their combined linkage. Secondly, unlike traditional radiology imaging that focuses on the internal parts of the body, dermatology focuses on the visible parts of the body; therefore, images are inherently identifiable and sensitive (patients are nude or semi-nude), raising significant privacy concerns for patients, affecting their willingness to participate in screening programs. This has knock on affects for all melanoma and skin cancer patients who undergo 1.1 million Medicare treatment services in Australia every year.

5.3.3. Federated Learning Infrastructure

AIS-SHIELDS is a new MRFF National Critical Research Infrastructure project that converges the work on AIS, the ACEMID Research Repository, & FLERA to implement federated learning within the AIS context.

AIS operates as a federated network of institutional nodes deployed on kubernetes with each researcher only able to access the project(s) to which they have been granted access. Universities' have an AIS node with all 5 services mentioned above that acts as the decadal data store of the research data. Clinical sites where the data is acquired will have Edge Devices, which can perform transient processing such as de-identification, encryption, routing, real-tie analysis, or in this case federated learning, as depicted in Figure 5.3.

The data flow is usually Instrument<->Edge Device<->AIS Node, optionally between AIS Nodes as well. All software and containers are stored in the AIS Github Organization (https://github.com/Australian-Imaging-Service) which is used for CICD to deploy and update each node.

Both AIS Nodes and Edge Devices run on Kubernetes on top of a diverse set of underlying infrastructures, allowing the tooling to be standardized. The Kubernetes clusters for AIS Nodes tend to be larger, using services such as AWS Elastic Kubernetes Service (EKS), potentially with many dozens of worker nodes (Virtual Machines assigned to the cluster) with dynamic scaling. Kubernetes clusters for AIS Edge Devices are much smaller, often 1-3 individual Virtual Machines on which Microk8s has been implemented.

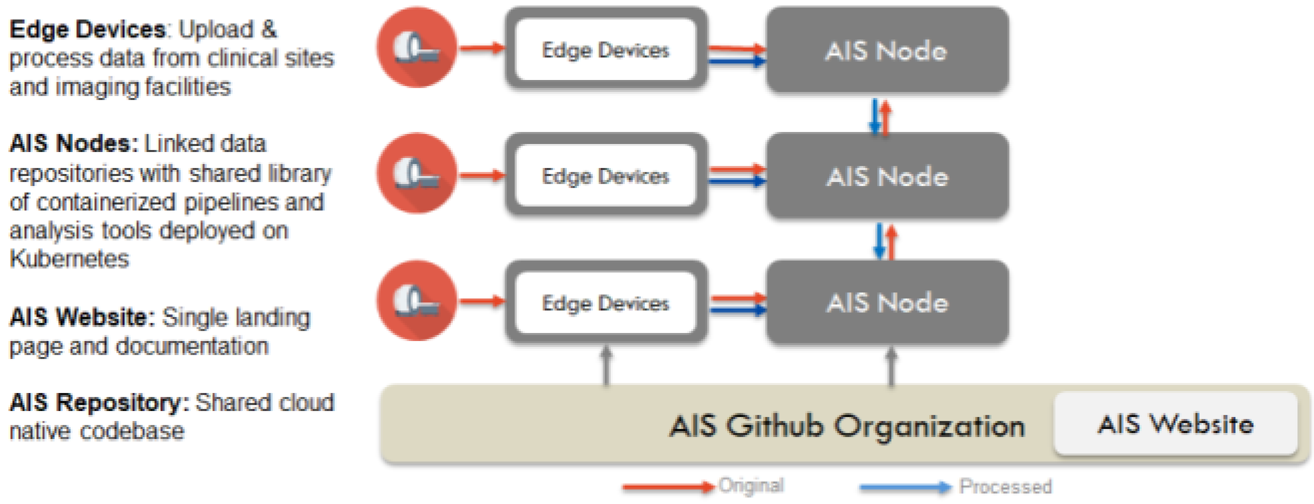

Edge Devices: Upload & process data from clinical sites and imaging facilities
AIS Nodes: Linked data repositories with shared library of containerized pipelines and analysis tools deployed on Kubernetes
AIS Website: Single landing page and documentation
AIS Repository: Shared cloud native codebase
Edge Devices
AIS Node
Edge Devices
AIS Node
Edge Devices
AIS Node
AIS Github Organization
AIS Website
Original
Processed

Fig. 5.3. Illustration of AIS Nodes and incorporation of Edge devices.

A challenge with deploying within clinical sites is the differences in technology. Research technology, particularly in the case of machine learning, is heavily Linux based with software containers. Hospital IT however tends to be Windows based with no containerization. AIS has had some success bridging the two by deploying Microk8s on NSWHealth Windows Machines. The University of Sydney central ICT team, which manages the AIS GitHub Organization, did a vendor assessment with Microk8s as the software application. From the NSWHealth point of view Microk8s is a Windows application, and they manage it as other applications, being responsible for the underlying VM and security of the OS image. From the AIS point of view, the research tools then see a Linux based Kubernetes cluster. Specific firewall whitelists are made to the AIS container registry to allow pre-approved containers to be pulled and updated to run on the edge cluster. A second set of firewall rules are made for any egress of data between the edge device and the AIS Node. This deployment approach for edge devices has to date focused on secure data egress where image data is captured from a scanner and needs to be de-identified, encrypted, and routed to the correct project in XNAT on an AIS node in a secure and audited manner. In AIS-SHIELDS, this is being expanded to add local computational capability. In principle, an entire AIS Node could be run on an edge device if there were sufficient storage and computational resources available.

The workflow for image labelling is:

1. Upload data to XNAT
2. XNAT automatically triggers n many pipelines to run on the images
3. From XNAT, open the data in an image viewer integrated with MONAI Label to add annotations to the dataset

For federated learning, AIS is working to add NVFlare as a service in the Kubernetes cluster that can be accessed via the XNAT UI like how ARCANA pipelines and Neurodesk virtual desktops, matching data access of the initiating user. This builds upon the previous FLERA work. AIS will manage the edge devices, allow researchers to access the pre-processing pipelines and federated learning clients to run on their datasets. The long-term intention is to apply this infrastructure to the ACEMID Total Body Photography scanners so that federated learning can be applied securely without the participant data leaving the clinical site to widen participation.

5.4. Existing Implementation Case Study: NINA

5.4.1 Background

The National Infrastructure for Federated Learning in Digital Health to Generate New Models of Care for Chronic Diseases (NINA) project seeks to answer the following question: Can we leverage disruptive, cutting-edge federated learning technology to overcome existing barriers in accessing health data for research, thereby facilitating research aimed at enhancing outcomes for chronic diseases?

Currently, Australian datasets are siloed, isolated both geographically (across different states) and across the care continuum (spanning primary and hospital care). NINA aims to establish a national capability and infrastructure network to enable federated digital learning in Australia. The overarching hypothesis of the project is that by establishing the necessary critical federated learning research infrastructure, we can create breakthrough research opportunities for improving outcomes in chronic diseases.

The main objectives are:

- Objective 1 - co-design new scalable ethics and governance pathways for federated learning in health, ensuring compliance with existing legislation.
- Objective 2 - establish the technology and demonstrate its potential for safely accelerating development of chronic disease research with the creation of national synthetic datasets (as required) to test federated learning approaches.

- Objective 3 - provide infrastructure that enables healthcare data to remain in situ and harmonised in separate databases with data and analytics capability brought to the datasets (through federated learning systems) while preserving privacy.
- Objective 4 - implement federated learning using infrastructure (from 3) to deliver innovative research to inform better outcomes for chronic disease exemplars (diabetes, rheumatoid arthritis, osteoarthritis and cancer).
- Objective 5 - ensure this infrastructure is transitioned to business as usual through implementation, evaluation and sustainment planning.

In essence, the NINA project aims to:

- Integrate and harmonise data: NINA seeks to integrate and harmonise data at each site according to globally accepted standards.
- Pioneer AI/ML federated learning: NINA aims to pioneer the use of iterative AI/ML federated learning, bringing computing and AI/ML capabilities directly to the data.
- Establish a Digital-Health Accelerator: NINA plans to create a Digital-Health Accelerator for both industry and research. This includes an incubator phase that allows research organisations and industries to utilise synthetic datasets. These datasets contain equivalent data to that which will be used to train AI/ML at local sites.
- Develop Best Practices and Educational Programs: NINA will develop standard operating procedures and educational programs to expedite the transformation of research data analysis using federated learning.
- Showcase the Impact of federated learning: To demonstrate the effectiveness of this federated learning model, NINA will focus on applying federated learning to data related to three prevalent chronic diseases in Australia: diabetes, rheumatoid arthritis, osteoarthritis and cancer.
- Ensure Long-Term Impact: NINA is committed to ensuring the translation and long-term impact of the project by collaborating with industry, health and government departments, universities, and peak bodies.

5.4.2 Project governance

NINA is a five-year program funded by the MRFF National Critical Research Infrastructure scheme with additional cash and in-kind contributions from UQ, Monash and Macquarie universities, the Queensland Cyber Infrastructure Foundation (QCIF), Styker, Ansen Innovation, Athritis Research Canada, ARDC, ARMHUB, BioGrid, CSIRO, Microba, Medical Software Industry Association, QLD Health, A3BC Cancer Alliance QLD, the Department of Environmental and Health and Victorian Institute of Forensic Medicine (VIFM). Led by CIA Professor Clair Sullivan, University of Queensland, over 20 organisations are participating in NINA (Figure 5.4.).

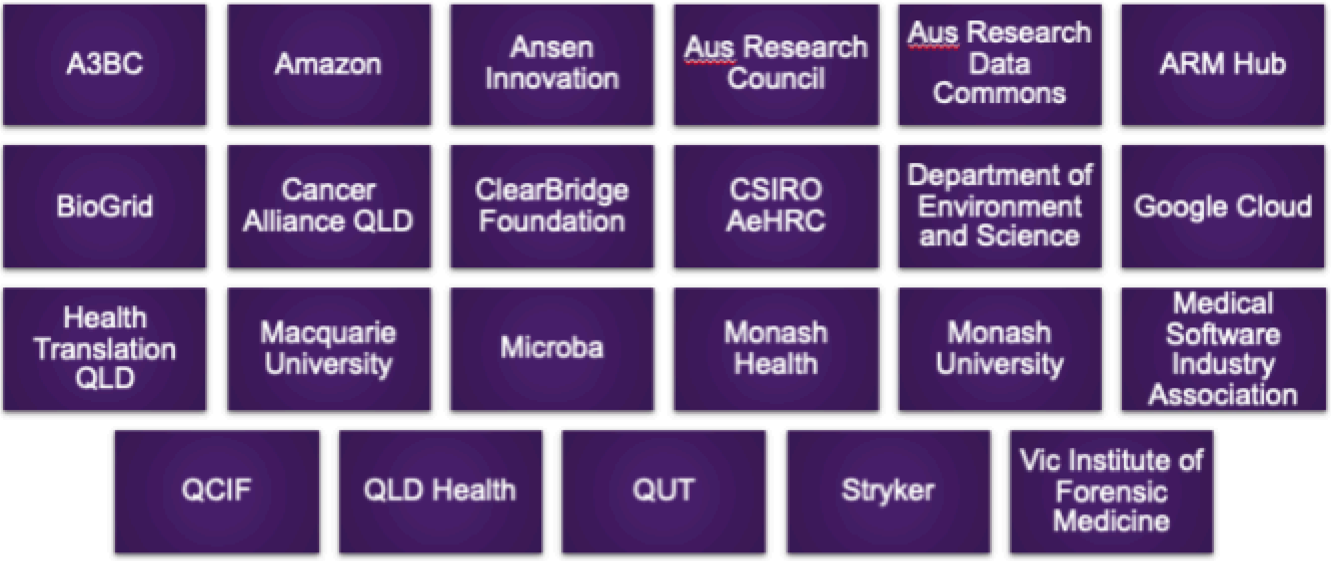
The NINA Steering Committee consists of all Chief Investigator team members, partner and consumer representatives across the four use cases, and has overall responsibility for delivering the project, including monitoring identified risks and managing project risks as they arise. It meets on a monthly basis and is chaired by Prof. Sullivan.

The National advisory group is comprised of eminent experts in digital health and the clinical domains of the use cases. It includes representatives from ADRC, Medical Software Industry Australia (MSIA), Australian Alliance for Artificial Intelligence in Healthcare (AAAiH) and Google Health. This committee provides invaluable strategic advice and monitor the project for compliance. Any issues will be raised directly with Prof. Sullivan, who will be responsible for implementing changes to the project to address the issues raised.

Importantly, the NINA project will ensure the voice of consumers is heard by including consumers in the design and evaluation of potential digital health solutions.

5.4.3 Federated Learning Infrastructure

NINA is dedicated to the practical application of federated learning in a variety of real-world settings. The project engages a diverse range of participating sites, such as health services, pathology services, industry partners, and registries. Each of these sites necessitates a specialised infrastructure, possesses varying degrees of IT and data science expertise, and adheres to unique data governance protocols and procedures. Through the deployment of tailored infrastructure at each location, NINA aims to evaluate how federated learning can enhance and expedite data accessibility. The project will explore whether federated learning mitigates existing data access challenges, introduces new concerns, or encounters distinct obstacles and roadblocks.



A3BC
Amazon
Ansen Innovation
Aus Research Council
Aus Research Data Commons
ARM Hub
BioGrid
Cancer Alliance QLD
ClearBridge Foundation
CSIRO AeHRC
Department of Environment and Science
Google Cloud
Health Translation QLD
Macquarie University
Microba
Monash Health
Monash University
Medical Software Industry Association
QCIF
QLD Health
QUT
Stryker
Vic Institute of Forensic Medicine

Figure 5.4. NINA Partners

To guide those real-world deployment, a test environment has been deployed on the Nectar cloud for three federated learning frameworks allowing to:

- Establish the infrastructure requirements for a participating site
- Conduct performance testing
- Simplify, fine-tune and document the deployment at a participating site
- Allow researchers and sites to experiment with the technology
- Assess security
- Provide a training ground for researchers and other stakeholders

6. Enabling Implementation of Federated Learning - Recommendations to ARDC

In the final period of writing this report, a workshop was held including research groups working with federated learning in Australia to exchange ideas on experiences and suggestions for supporting federated learning in the Australian research community in the future. This section presents a series of recommendations to the ARDC generated from writing this report and during the workshop.

As detailed in this report and references within the report enabling federated learning:

- Overcomes many barriers that exist with centralised learning. Data can be used for research projects while it remains at a local institution overcoming the challenge of moving data between jurisdictions to one central location. Some risks associated with linking data can also be overcome with datasets being able to be learnt from jointly but without linking the data. With data remaining at local institutions, it can also be updated in a timely manner overcoming the challenge of how up to date a dataset is once it has been collected and is available for the research.
- Is in the national interest facilitating learning from data across jurisdictions, supporting research work across Australia but as importantly supporting work between Australia and the rest of the world. This can be very challenging as not only Australian data requirements need to be met but also those from other countries. Federated learning is also being supported by many other countries and it is important that Australian researchers are able to be involved in these international efforts.
- Supports leading edge research. Many impactful data research projects require access to large, detailed datasets (e.g. imaging data) and to reduce bias in any data project diverse data is required. Federated learning enables access to these large datasets and by supporting access to diverse data can enable cutting edge research to be undertaken in the most appropriate manner.

To ensure these opportunities are effectively harnessed it is recommended to the ARDC that federated learning be supported as a mainstream approach. The following recommendations are made to the ARDC to enable this:

6.1. Support for Australian Federated Learning collaboration

There are a number of Australian research teams using federated learning enabling large scale, internationally linked, cutting-edge research to be undertaken. Although there is significant enthusiasm for this to occur, these research teams have not generally been working together and there is minimal support for other research teams who wish to consider using federated learning, limiting the impact use of federated learning may have for Australian researchers.

6.1.1. Current recommendations to ARDC

It is recommended that ARDC support collaboration between researchers undertaking federated learning across Australia. Following discussions at the federated learning workshop held in June 2024, the recommendation is that this could occur with the establishment of a working/interest group on federated learning within the machine learning community of practice (ML4AU CoP) perhaps in collaboration with the Australian Research Containers Orchestration Service (ARCOS) and the Australian Sensitive Data Interest Group (AuSDIG).

6.1.2. Collaborative activities to strengthen federated learning across research teams

- Establishing a communication channel (or links to existing communication channels) for Australian researchers working in or exploring the potential of federated learning. A suggestion during the workshop is that this could be set-up on Zulip.
- Using the communication channel and interest/working group to propose collaborative projects that would be of benefit to all federated learning researchers
- Using the communication channel and interest/working group to share expertise and experiences.

6.2. Support for Federated Learning Software Tools

There is a need to provide the necessary software tools for federated learning as described in detail in the above sections. To support this, it is important that there is ongoing software understanding and development knowledge.

6.2.1. Current recommendations to ARDC

- That ARDC endorse the review criteria recommended in the federated learning report as an initial criterion for assessing federated learning tools (noting the suggestion for federated learning groups to work together to expand this criteria)
- That ARDC endorse recommendations for the FLOWER, Vantage6, Pysyft and NVIDIA FLARE platforms to be used by Australian research groups
- That ARDC enable software engineering and machine learning expertise to be developed in these open-source tools to support international efforts and ensure local knowledge.
- That ARDC provide or support expertise & training (software engineering, federated learning and machine learning) for these recommended platforms that Australian research groups can utilise, developing and conducting technical tutorials and workshops on how to use key federated learning frameworks such as Flare, Vantage6, Pysyft and Flower, highlighting their suitability, and pros and cons, for a range of federated learning scenarios.

- That ARDC provide or support demonstrations of these platforms set-up on nectar (only publicly available or simulated data) for research groups to test and learn on. This could include setting up dedicated compute resources (such as GPU VMs or deployed Jupyter server notebooks) on the NECTAR cloud platform to provide a training ground for researchers and other stakeholders who are interested in evaluating federated learning frameworks using a production grade federated learning systems
- That ARDC provide or support approaches achieving implementation consistency of these federated learning platforms (to enable consistency and support review and implementation for data custodians and institutions)
- That ARDC provide or support approaches ensuring that these platforms can be rolled out robustly across different institutions and local set-ups.

6.2.2. Collaborative activities to strengthen federated learning across research teams

- Collaborative review of and further development of the federated learning platform assessment criteria to provide a more detailed assessment criteria that can be tailored for individual research project assessment of the federated learning platforms.
- Using the revised criteria independent assessment of the different platforms by different researchers and research teams to provide an uncertainty analysis of these assessments
- Consider standard interfaces/approaches to support implementation consistency (considering data custodians and institutions)

6.3. Supporting Data Storage and Computational Power

As described above there is a need for data storage and computational power requirements at the nodes and at the server as well as communication channels.

This could potentially be established on nectar and on MLeRP with individual institutions looking after their own data storage on these platforms. However, it is unlikely that accessing and storing data on nectar and MLeRP will meet the requirements of the health care institutions and particularly the registries where the datasets are.

An additional challenge for federated learning is where datasets (commonly linked registry data) must be stored in a trusted research environment as described in Section 4. In this environment aggregate analysis using federated data is still achievable (as there is only one or perhaps two exports required) however federated learning, particularly with advanced modelling requires multiple iterations and is unfeasible with manual review for export from such secure access platforms. This requires either a different approach to how the data is stored e.g. an alternative to the current TREs or an alternative federated learning architecture with involvement from organisations holding the data. For inclusion of registry data in a federated learning network it would be possible to use either a vertical or more likely a combined horizontal and vertical federated learning approach with a node set-up with registry data and someone at the registry supporting this to ensure separation of data handling, as necessary to manage best practice of managing linked data.

6.3.2. Current recommendations to ARDC

- That ARDC consider the options for cloud computing resources that could be used for federated learning where these resources need to be accessed from within IT infrastructure at organisations where the data is held (primarily health but also registries and other organisations). E.g. Funding for access to the currently approved health network cloud computing resources (or some available resources that meet the approved cloud computing requirements) to enable node set-up and federated learning could be considered.
- Related to the previous point, it was noted that if NCRIS resources are to be used with health data, the requirements for this need to be extended. ISO certification is one of these factors. As raised during the workshop this is something that the ARDC is currently discussing and that they are committed to progressing. This is a longer-term goal.
- Consideration of use of nectar or similar as a federated learning server location in the first instance (this is also related to the need for clear security and privacy documentation)
- That ARDC provide support for increasing robustness in the Kupernedes layer to increase confidence for organisations IT departments
- That ARDC provide support for discussions across jurisdictions (particularly across states but also organisations within each state) regarding accessing data storage and computational resources.
- That ARDC provide support for sharing of and co-developing documentation (that is maintained as systems, technologies and approaches are updated) to provide to organisations. This could include documentation and pathways for data node set-up including storage and federated learning set-up that have prospectively been reviewed and approved by jurisdiction IT teams (e.g. NSWHealth). It is likely that there would still be processes and approvals needed at a local level (e.g. in NSW within Local Health Districts) but this would be much smoother if there was central IT knowledge and support for such a platform.
- That ARDC provide support to explore alternate options to TRE/SREs for use of datasets that must currently be stored in such environments in a federated learning network.
    - Support for discussions and where necessary changes in current approvals/practices for registries to support a federated learning model, enabling a node to be set-up and supported by the relevant registry. In an ideal setting this framework could be used for multiple federated learning projects/platforms (e.g. a cancer network as well as a neurology and a cardiology network)

  - Support to work with secure access environment platform providers (e.g. the UNSW developed ERICA platform) to provide a secure access environment where there is an automated review of data extracted from the secure access environment.
  - Support to work with those establishing policies over how registry data is stored to develop adaptions where necessary that meet requirements and what is technically feasible.

6.3.3. Collaborative activities to strengthen federated learning across research teams

- Sharing of experiences regarding establishing datasets for federated learning within the various health and registry organisations. Looking at building on success of initial projects to streamline this for future projects.
- Collaborative effort with research groups and ARDC to approach organisations (e.g. registries and health departments)
- Sharing experiences in use of cloud resources within health departments as availability and costing of these services develop over time.

6.4. Security, data privacy and data equity

Security, data privacy and data equity are key areas that cut across choice and appropriateness of almost all areas of federated learning set-up including data storage and computational requirements, software choice, governance and the practicalities of implementation. As such during the workshop it was decided that this topic should be addressed as a key theme.

6.4.1. Current recommendations to ARDC

- That ARDC provide a service to demonstrate maintained security testing and documentation for recommended federated learning platforms that can be consistently shared with institutions so that Australian research groups using federate learning are consistent in their messages to institutions.
- Can ARDC provide a service to demonstrate these security aspects, so we have a shared set of information provided to organisations (especially state health organisations).
- Cloud Native Environments are recommended as an option for federated learning infrastructures due to their security, privacy, and flexibility. They offer features like identity and access management, encryption, and security monitoring, while also supporting machine learning frameworks and facilitating horizontal scaling. These environments also enable deployment in distinct Secure Networks, ensuring reproducibility and robustness of federated learning infrastructure.

6.4.2. Collaborative activities to strengthen federated learning across research teams

- Determining collaboratively how we most effectively demonstrate privacy and risks/benefits for our research projects using federated learning. In doing this it is important to clarify the difference in federated learning on device (e.g. google) vs federated learning on health sites (with benefits back to patients). It would also be useful to consider risk tiers/levels and risk of re-identification.
- Sharing of security testing and documentation on the tools that are being used across the research groups.
- Work together to formulate appropriate and realistic threat models (e.g., re-identification attacks, record linkage attacks, data reconstruction attacks, etc.).
- Work together to determine appropriate privacy protection metrics (e.g., differential privacy value adopted by the US census 2020, successful attack rate, etc.).
- Support each other to evaluate model fairness in federated learning (e.g. assessing accuracy across different population groups) and to implement federated learning approaches that support the development of non-biased models.
- Work together to assess and demonstrate the pros and cons of privacy vs model utility in a federated learning setting (e.g. adding noise will reduce quality of final model).
- Undertake a comparison with risks/benefits of federated learning compared to other approaches esp. Centralised. Consider a framework that can be used for new projects.
- For vertical learning (and linking of data for an individual patient), consider the risk of linking data
- For noting there is a research team at Macquarie University who are looking at risk profiles of identification to the individual and to the sites might have good input on risk (Mark Dras & Annabelle McIver).

6.5. Data standardisation

Although not addressed in this report as this is being considered by other ARDC initiatives the need for data standardisation for federated learning is key and recommendations on this are provided here.

6.5.1. Current recommendations to ARDC

- That the federated learning research groups are kept in the loop regarding other ARDC activity on data standardisation.
- That ARDC support implementation of data standardisation using a common and well recognised framework for federated learning projects (e.g. OMOP). Of note there is a current ARDC project

looking at translating electronic medical records to OMOP and there will be a ARDC framework document on common data models progressing soon.

6.5.2. Collaborative activities to strengthen federated learning across research teams

- Review of approaches to data standardisation across the different research groups and a broad goal to try and work towards consistency with the potential of linking across the federated projects in the future when appropriate.

6.6. Governance to Support Federated Learning

Similarly to security, governance is a key overarching area for federated learning and recommendations to support this have been separated out from the core infrastructure requirements.

6.6.1. Current recommendations to ARDC

- That ARDC generate or support generating agreed and consistent documentation regarding risks and benefits and IT implementation that can be provided to institutions by research teams wishing to undertake federated learning.
- That ARDC support discussions with overarching organisations such health services and registry data holders to ensure understanding of federated learning and requests for changes to process and/or resources as may be necessary. (Noted ARDC would be interested in doing this for the PBS or a similar dataset)
- That ARDC undertake or support approaching the NHMRC to consider providing guidance to ethics committees regarding federated learning (and perhaps machine learning in general).
- That ARDC undertakes work or supports work to consider different approaches to SRE/TREs for federated learning. This would consider how automation could be used appropriately (how do the 5 safes change if there is no human in the loop?) and could review international approaches (e.g. UK federation of TRE providers where queries can be shared, noting this is aggregate analysis rather than true federated learning).
- Some of these activities could be incorporated into ARDC plans to look at path finder projects working across organisations.

6.5.2. Collaborative activities to strengthen federated learning across research teams

- Work together to determine common requirements for organisational governance and IT approvals to support work with ARDC to provide documentation for these requirements
- Sharing of governance documents and experiences, providing the opportunity to build on successes and learn from challenges.

- Work together to determine common dataset of interest to federated learning projects (e.g. the PBS dataset) and a prioritisation of these datasets to support work with ARDC to support access to these datasets using federated learning frameworks
- Where there is a ARDC work with the research teams and the relevant organisation to support discussions around how this could be achieved. (Noted ARDC would be interested in doing this for the PBS or a similar dataset)
- To be forward-looking and ensure the developed federated learning systems are compliant with the upcoming Australian regulations on AI, which goes beyond the Australian Privacy Act 1988. (See DISR's recent response on AI regulations: https://www.industry.gov.au/news/australian-governments-interim-response-safe-and-responsible-ai-consultation).
- Consider a consistent vocabulary around federated learning
- Of note the NINA project are working on a publication on governance for federated learning.

Appendix

A.1. Flower Implementation Guide

This section presents in detailed implementation of Flower framework for horizontal data partitioning. The dataset considered is tabular, however, imaging data can easily be incorporated. The code is available at: https://github.com/AustralianCancerDataNetwork/FlowerSimulations

A.1.1. Horizontal Partitioning

In horizontal data partitioning, all participating clients have the same features (input items) including the output item (labels), however, the data points are different as shown in Figure A1.

A.1.1.1. Server-Client Architecture

The horizontal federated learning framework utilizes a server-client architecture. In this setup, there is a central server responsible for coordinating the federated learning process, and three clients that contribute their local model updates to the server. This architecture enables collaborative model training across decentralized data sources while maintaining data privacy.

A.1.1.2. Components: Server and Clients

Server:

The server acts as the central coordinator in the federated learning process. Its primary responsibilities include:

- Orchestrating communication with clients.
- Aggregating model updates from multiple clients.
- Distributing the global model parameters to clients for further training. Managing the overall training process, including the number of rounds and convergence criteria.

Clients:

Clients represent individual devices or entities with local data that participate in the federated learning process. Each client:

- Trains a local model on its own data without sharing the raw data with the server or other clients.
- Computes model updates based on its local data and sends these updates to the server.
- Receives global model updates from the server and incorporates them into its local model for further training.

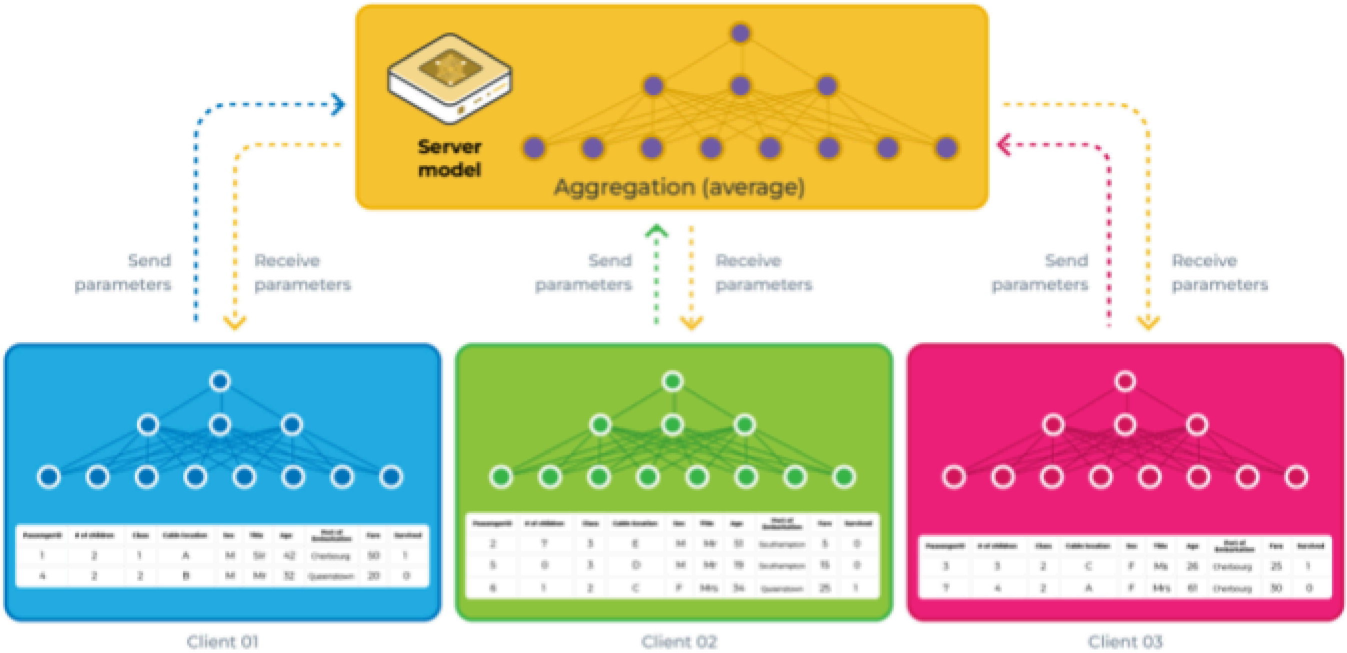


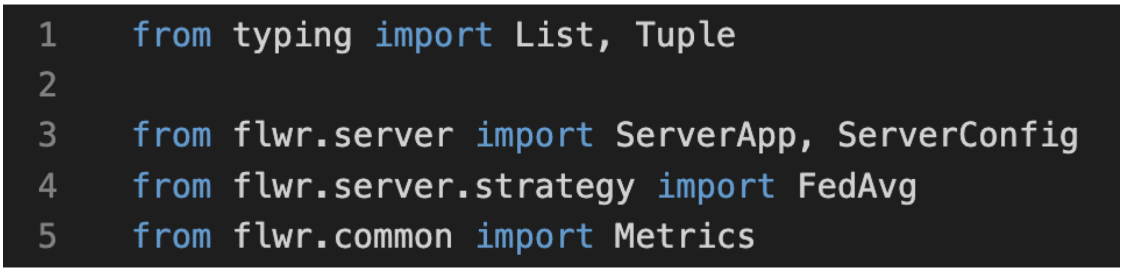
Figure A1. An illustration of horizontal federated learning setup CITATION Beu20 \l 3081 (Beutel, et al., 2020).

A.1.1.3. Server Code

The Server code consists of number of different components, described below:

Import Files

The code begins with necessary imports from the Flower framework. It imports classes and functions required for setting up the server, defining the federated averaging strategy, and handling common components such as metrics.



```
from typing import List, Tuple

from flwr.server import ServerApp, ServerConfig
from flwr.server.strategy import FedAvg
from flwr.common import Metrics
```

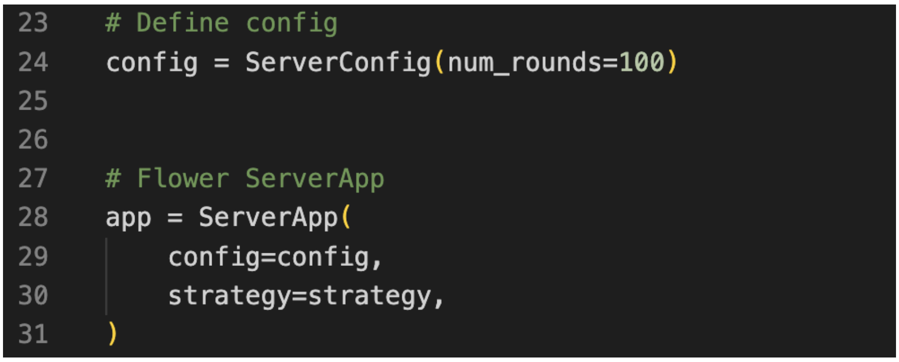
FedAvg Strategy

The federated averaging strategy, often abbreviated as FedAvg, is a key component of the framework's model aggregation process. FedAvg operates as follows:

- Upon receiving model updates from participating clients, the server aggregates these updates to compute a global model update.
- FedAvg typically employs a weighted average scheme (also employed in this example), where the contribution of each client's update is weighted by the size of its local dataset or another relevant metric.
- This weighted average helps mitigate the impact of imbalanced or varying dataset sizes across clients, ensuring fair representation in the global model.

Server Configuration

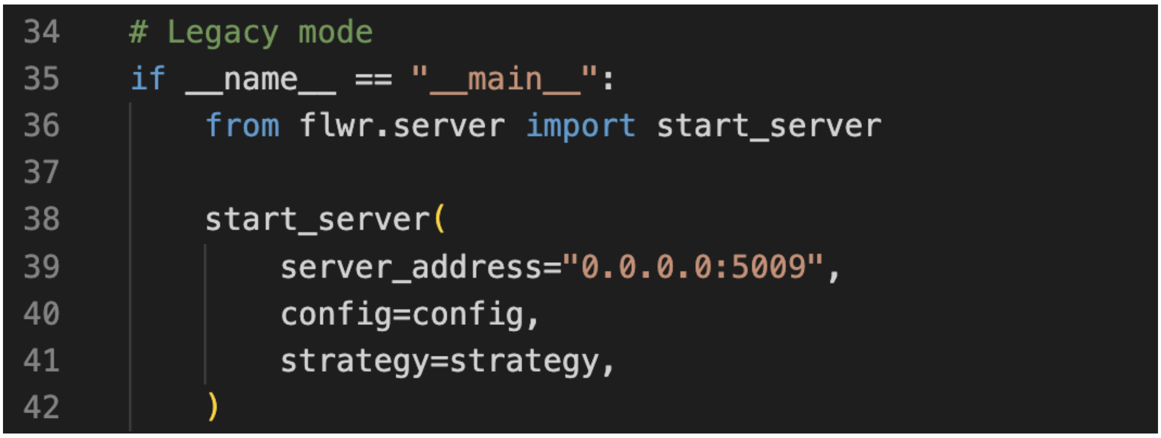
A “ServerConfig” object is created, specifying the number of training rounds (“num_rounds”, which is set to 100) for the federated learning process. Finally, the “ServerApp” is initialized with the specified configuration (“config”) and strategy (“strategy”). This sets up the server application ready to start.



```python
# Define config
config = ServerConfig(num_rounds=100)


# Flower ServerApp
app = ServerApp(
    config=config,
    strategy=strategy,
)
```

Legacy Mode:

This part of the code ensures that the server can be started directly when the script is executed as the main program. It uses the “start_server” function to start the server with the specified address (IP and Port, in this example the IP address is of its own machine, implying that the simulations for the clients and the server are done on the same machine, to employ on a different machine, specify the IP address and Port number of that specific machine), configuration, and strategy.



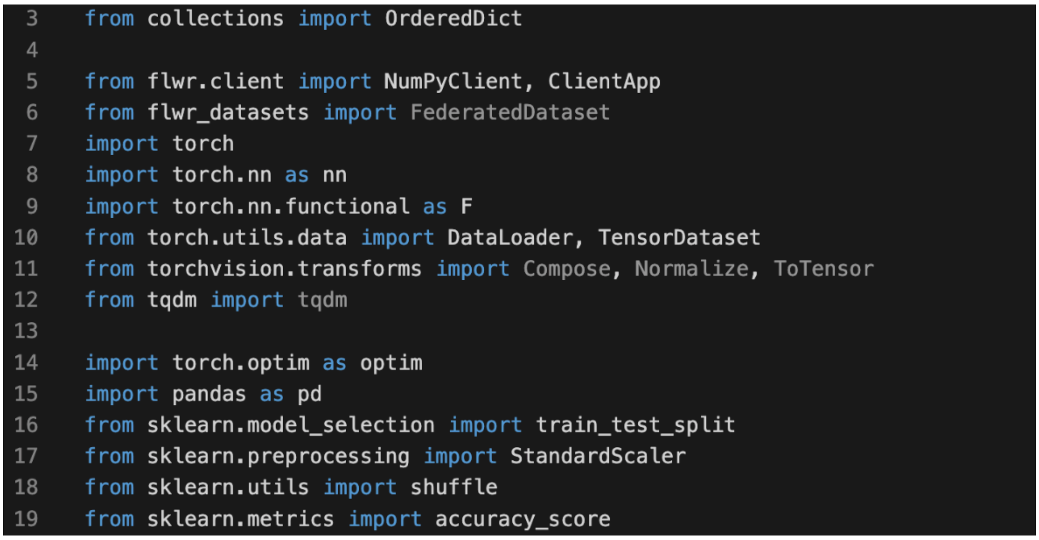
```
# Legacy mode
if __name__ == "__main__":
    from flwr.server import start_server

    start_server(
        server_address="0.0.0.0:5009",
        config=config,
        strategy=strategy,
    )
```

A.1.1.4. Client Code

The Server code consists of number of different components, described below:

Import Files

The code begins with necessary imports including libraries for data pre-processing (“pandas, sklearn”), neural network modeling (“torch, torch.nn”), Flower client setup (“flwr.client”), and other utility functions.

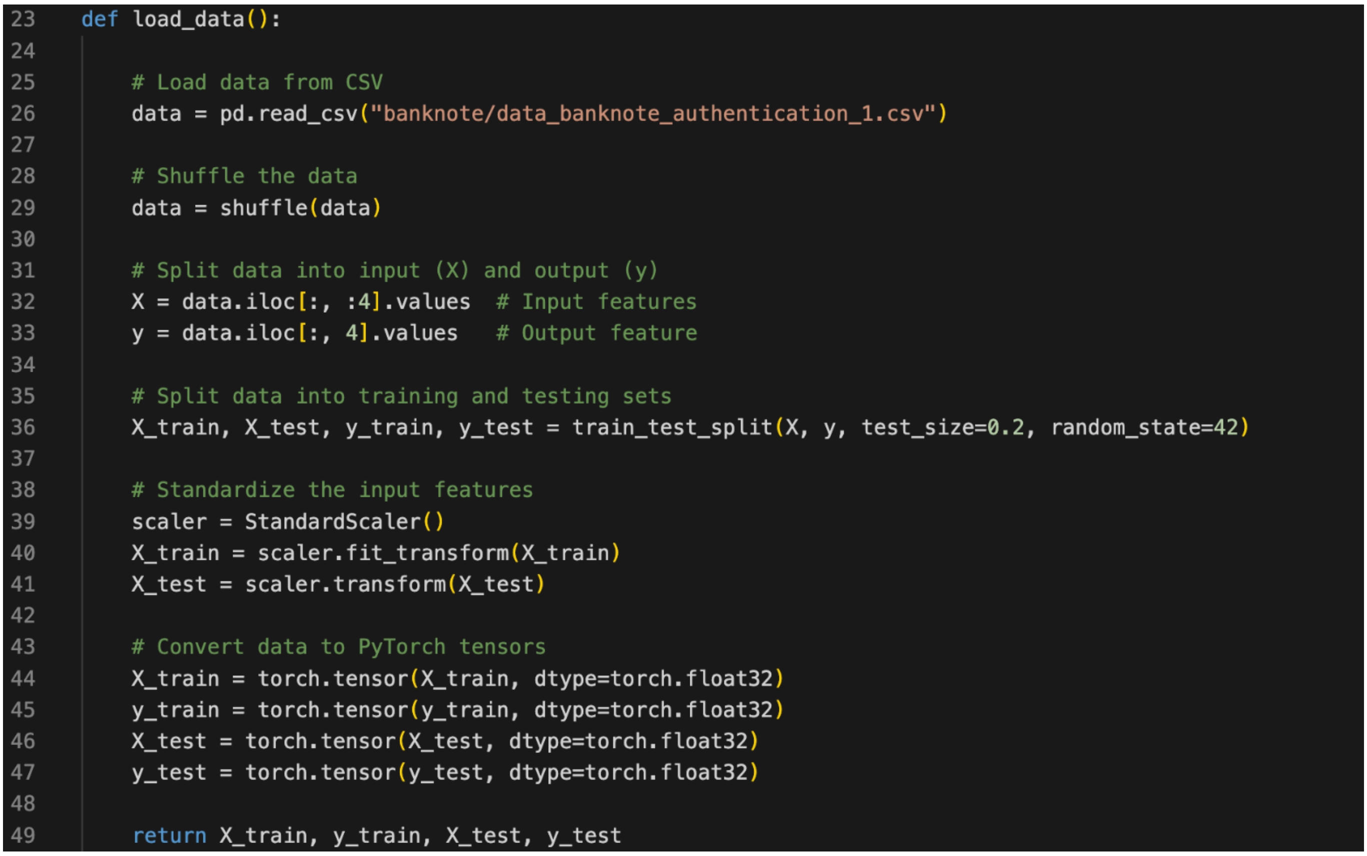


```
from collections import OrderedDict

from flwr.client import NumPyClient, ClientApp
from flwr_datasets import FederatedDataset
import torch
import torch.nn as nn
import torch.nn.functional as F
from torch.utils.data import DataLoader, TensorDataset
from torchvision.transforms import Compose, Normalize, ToTensor
from tqdm import tqdm

import torch.optim as optim
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.utils import shuffle
from sklearn.metrics import accuracy_score
```

Data Loading

The “load_data” function is responsible for loading and preprocessing the dataset. It reads the data from a CSV file (it can be any dataset, one can replace this with their own custom dataset, however, make sure that the features are in column form and the datapoints are in row form), shuffles it, splits it into input features (“X”) and labels (“y”), performs standardization, and converts the data into PyTorch tensors.

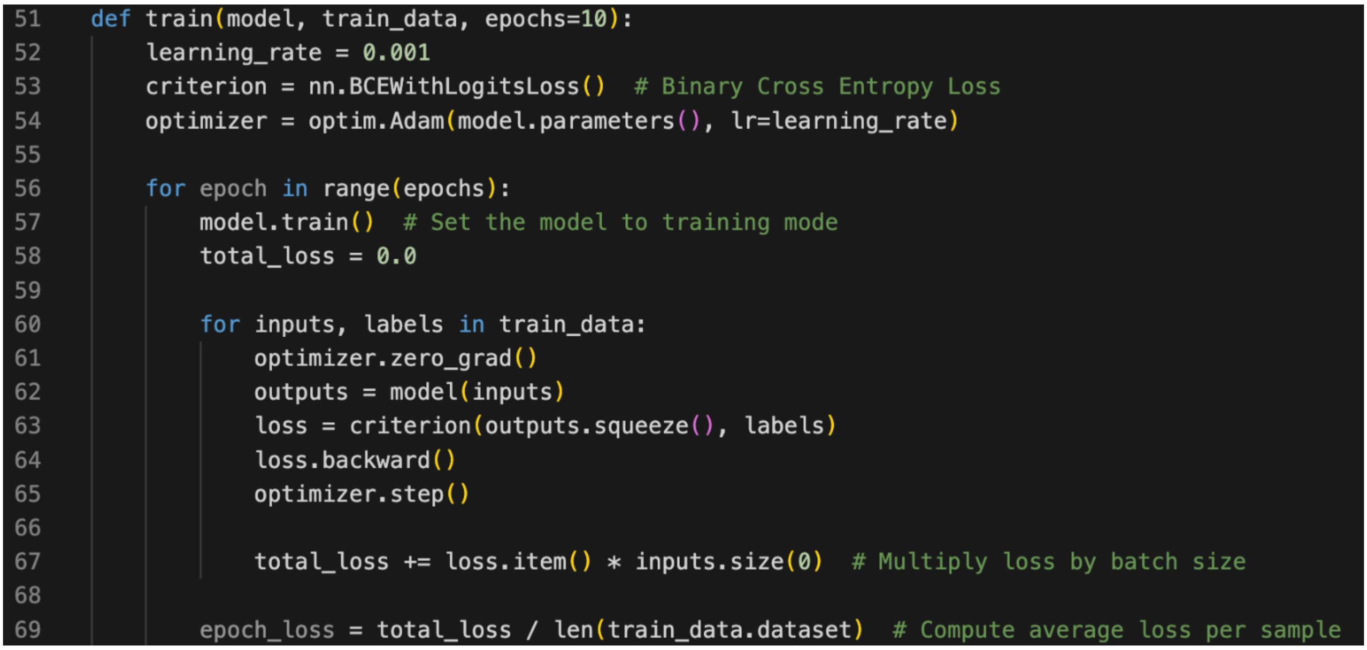
Further, it also split the data into training and testing datasets as well.



```python
def load_data():

    # Load data from CSV
    data = pd.read_csv("banknote/data_banknote_authentication_1.csv")

    # Shuffle the data
    data = shuffle(data)

    # Split data into input (X) and output (y)
    X = data.iloc[:, :4].values  # Input features
    y = data.iloc[:, 4].values   # Output feature

    # Split data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # Standardize the input features
    scaler = StandardScaler()
    X_train = scaler.fit_transform(X_train)
    X_test = scaler.transform(X_test)

    # Convert data to PyTorch tensors
    X_train = torch.tensor(X_train, dtype=torch.float32)
    y_train = torch.tensor(y_train, dtype=torch.float32)
    X_test = torch.tensor(X_test, dtype=torch.float32)
    y_test = torch.tensor(y_test, dtype=torch.float32)

    return X_train, y_train, X_test, y_test
```

Train Function

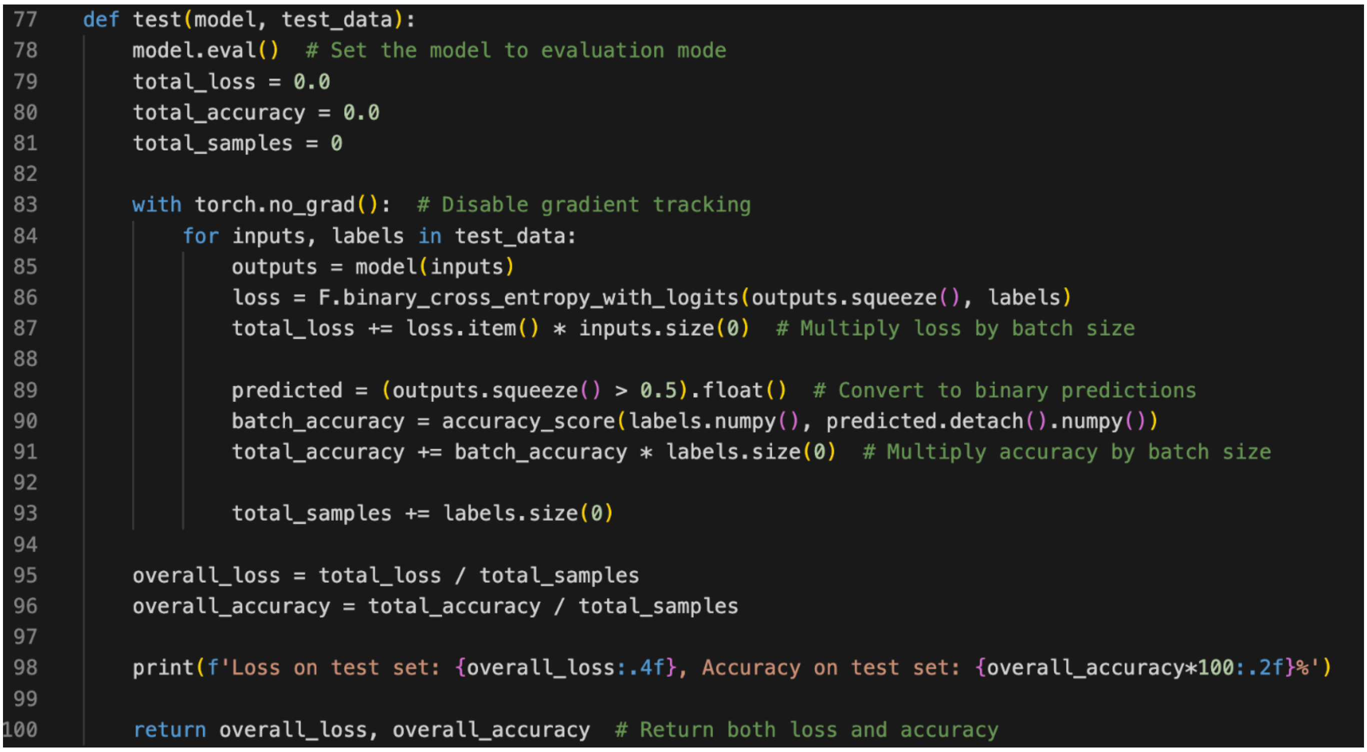
This function is responsible for training the neural network model (“model”) using the provided training data (“train_data”). It takes parameters such as the model, training data, and number of epochs. It is the same training function as can be used in a centralised manner.



```
def train(model, train_data, epochs=10):
    learning_rate = 0.001
    criterion = nn.BCEWithLogitsLoss()  # Binary Cross Entropy Loss
    optimizer = optim.Adam(model.parameters(), lr=learning_rate)

    for epoch in range(epochs):
        model.train()  # Set the model to training mode
        total_loss = 0.0

        for inputs, labels in train_data:
            optimizer.zero_grad()
            outputs = model(inputs)
            loss = criterion(outputs.squeeze(), labels)
            loss.backward()
            optimizer.step()

            total_loss += loss.item() * inputs.size(0)  # Multiply loss by batch size

        epoch_loss = total_loss / len(train_data.dataset)  # Compute average loss per sample
```

Test Function

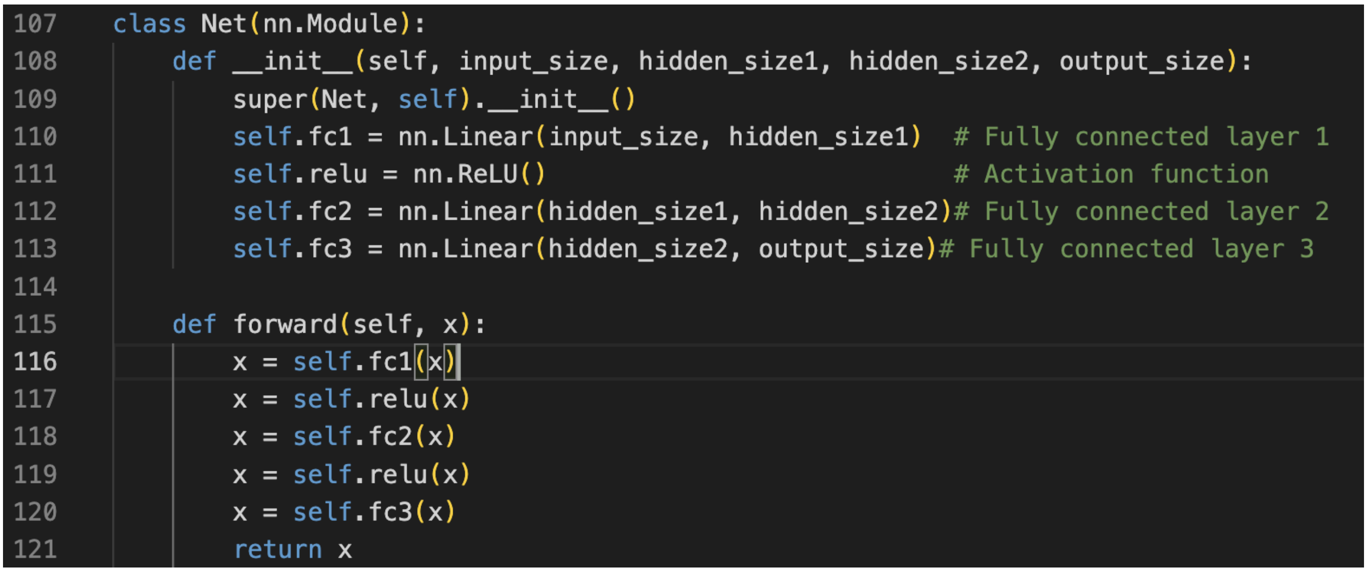
This function evaluates the performance of the trained model on the provided test data (“test_data”). It takes parameters such as the model and test data. It is the same testing function as can be used in a centralised manner.



```
def test(model, test_data):
    model.eval()  # Set the model to evaluation mode
    total_loss = 0.0
    total_accuracy = 0.0
    total_samples = 0

    with torch.no_grad():  # Disable gradient tracking
        for inputs, labels in test_data:
            outputs = model(inputs)
            loss = F.binary_cross_entropy_with_logits(outputs.squeeze(), labels)
            total_loss += loss.item() * inputs.size(0)  # Multiply loss by batch size

            predicted = (outputs.squeeze() > 0.5).float()  # Convert to binary predictions
            batch_accuracy = accuracy_score(labels.numpy(), predicted.detach().numpy())
            total_accuracy += batch_accuracy * labels.size(0)  # Multiply accuracy by batch size

            total_samples += labels.size(0)

    overall_loss = total_loss / total_samples
    overall_accuracy = total_accuracy / total_samples

    print(f'Loss on test set: {overall_loss:.4f}, Accuracy on test set: {overall_accuracy*100:.2f}%')

    return overall_loss, overall_accuracy  # Return both loss and accuracy
```

Neural Network Model

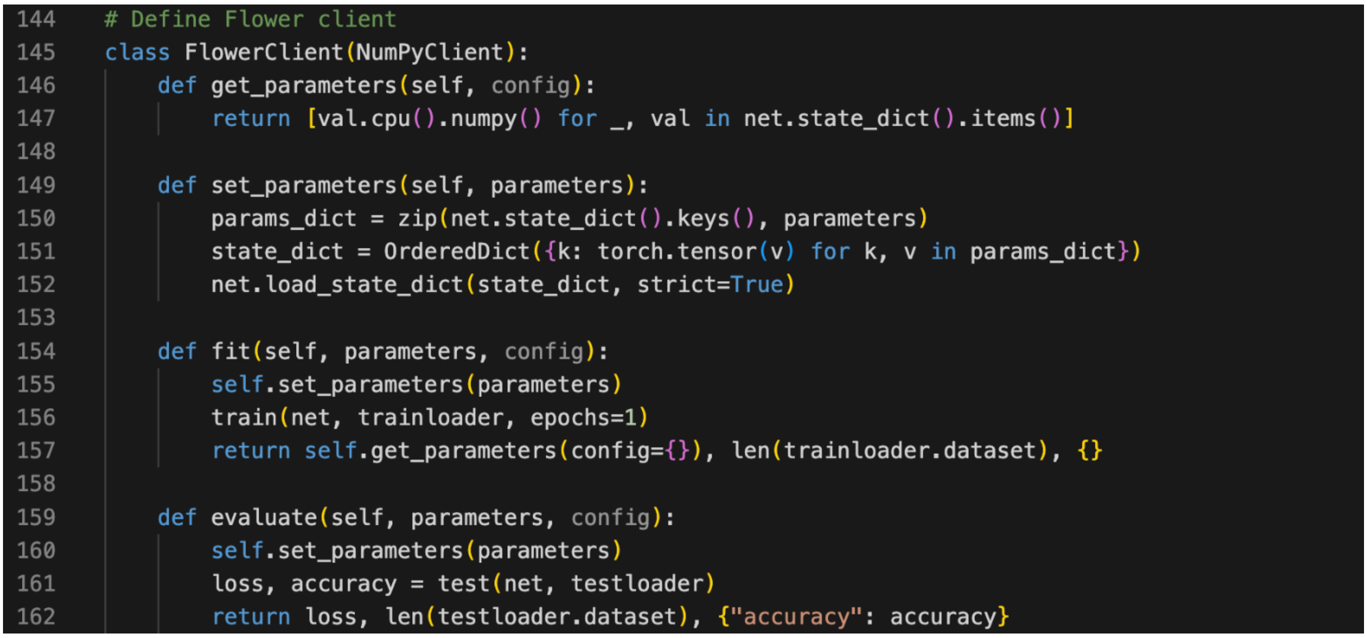
The Net class defines the architecture of the neural network. It specifies the layers, activation functions, and input/output sizes of the network.



```python
class Net(nn.Module):
    def __init__(self, input_size, hidden_size1, hidden_size2, output_size):
        super(Net, self).__init__()
        self.fc1 = nn.Linear(input_size, hidden_size1)  # Fully connected layer 1
        self.relu = nn.ReLU()                            # Activation function
        self.fc2 = nn.Linear(hidden_size1, hidden_size2)# Fully connected layer 2
        self.fc3 = nn.Linear(hidden_size2, output_size)# Fully connected layer 3

    def forward(self, x):
        x = self.fc1(x)
        x = self.relu(x)
        x = self.fc2(x)
        x = self.relu(x)
        x = self.fc3(x)
        return x
```

Flower Client

This is the meat of the client code. The “FlowerClient” class extends the “NumPyClient” class provided by Flower. It overrides methods such as “get_parameters”, “set_parameters”, “fit”, and “evaluate” to define the behaviour of the client during the federated learning and communication process.

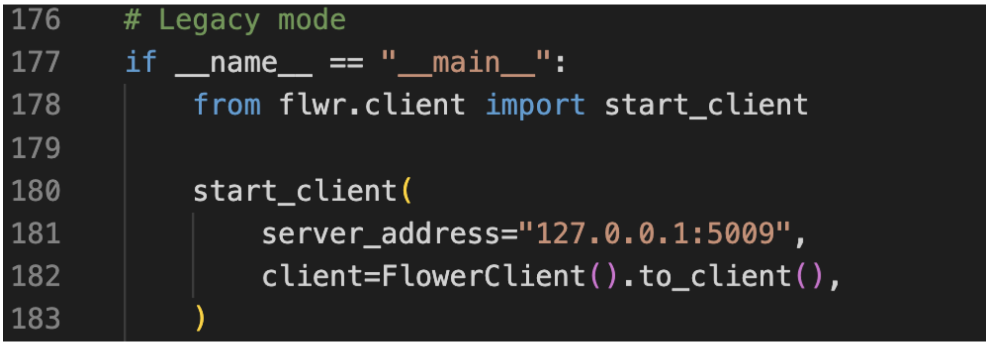


```python
# Define Flower client
class FlowerClient(NumPyClient):
    def get_parameters(self, config):
        return [val.cpu().numpy() for _, val in net.state_dict().items()]

    def set_parameters(self, parameters):
        params_dict = zip(net.state_dict().keys(), parameters)
        state_dict = OrderedDict({k: torch.tensor(v) for k, v in params_dict})
        net.load_state_dict(state_dict, strict=True)

    def fit(self, parameters, config):
        self.set_parameters(parameters)
        train(net, trainloader, epochs=1)
        return self.get_parameters(config={}), len(trainloader.dataset), {}

    def evaluate(self, parameters, config):
        self.set_parameters(parameters)
        loss, accuracy = test(net, testloader)
        return loss, len(testloader.dataset), {"accuracy": accuracy}
```

Starting the Client

If the script is run directly, it imports the “start_client” function from the Flower client module (flwr.client). The “start_client” function is then called with the following arguments:

- “server_address”: The address of the federated learning server to connect to. In this case, it's "127.0.0.1:5009", indicating that the server is running on the local machine (localhost) and listening on port 5009.
- client: An instance of the “FlowerClient” class converted to a Flower client using the “to_client()” method. This represents the client that will participate in the federated learning process.



```
# Legacy mode
if __name__ == "__main__":
    from flwr.client import start_client

    start_client(
        server_address="127.0.0.1:5009",
        client=FlowerClient().to_client(),
    )
```

A.1.1.5. Running the Example

We can simply start the server in a terminal as follows:

“python3 server.py”

Now we are ready to start the Flower clients which will participate in the learning. To do so simply open three more terminal windows and run the following commands.

Start client 1 in the first terminal:

“python3 client_1.py”

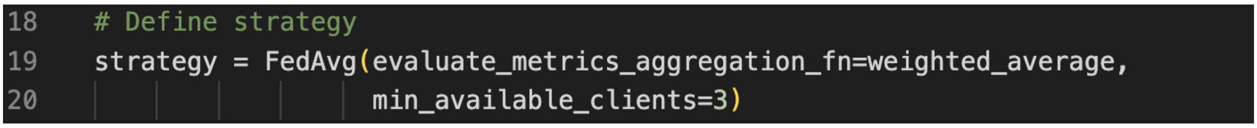
Start client 2 in the second terminal:

“python3 client_2.py”

Start client 3 in the second terminal:

“python3 client_3.py”

The above is for three clients, if there are more clients we need to run those as well. The number of participating clients can be specified by the server in “FedAvg” function.



```
# Define strategy
strategy = FedAvg(evaluate_metrics_aggregation_fn=weighted_average,
                  min_available_clients=3)
```

A.2. Implementation on Nectar Cloud

This section presents the implementation of the Horizontal Federated Learning setup using Flower tool. The underlying python files and programming environment remains the same as described in section A.1.

A.2.1. Creation of Virtual Machines on Nectar

The first step is to create Virtual Machines (VMs) on the Nectar. As there are four nodes; one server and three clients participating in the federated learning setup, we need to create four VMs. The specific steps required to create a VM is mentioned at the official website of Nectar:

https://tutorials.rc.nectar.org.au/cloud-starter/02-tutorials

The steps are also illustrated at AusCAT documentation: https://australiancancerdatanetwork.github.io/auscatverse/simulation/NECTAR.html

We will be creating from Ubuntu image and therefore need to generate cryptographic key pairs; the public key will be used at the time of VM creation and private key will be used at the time of logging in.

A.2.2. Login and Copying Files

To login into the VM, use the following syntax:

```
ssh -i ~/.ssh/your-private-ssh-key ubuntu@your-vm-ip
```

To copy files from the local machine into the VM, use the following syntax:

```
scp /path/to/local/file ubuntu@your-vm-ip:/path/to/remote/directory
```

We need to copy the relevant files to the VMs. For the server VM, we need to copy the server python file and pyproject.toml file (which lists all the required packages to be installed). For each of three client VMs, we need to copy the client python file, pyproject.toml file and data (csv) file.

A.2.3. Run Python Files

Once the relevant files are copied to the VM, we need to install the relevant packages listed in pyproject.toml for all the server and three client VMs. After this, run the server python file first, once the server is up and running, run the client python files from the client VMs (Note: make sure to enter the server's VM's IP address and port number in each of the client python file).

A.3. Implementation on Nectar Cloud using Docker

This section describes the required steps to implement the above federated learning setup using Docker instead of raw python files.

A.3.1. Docker Installation

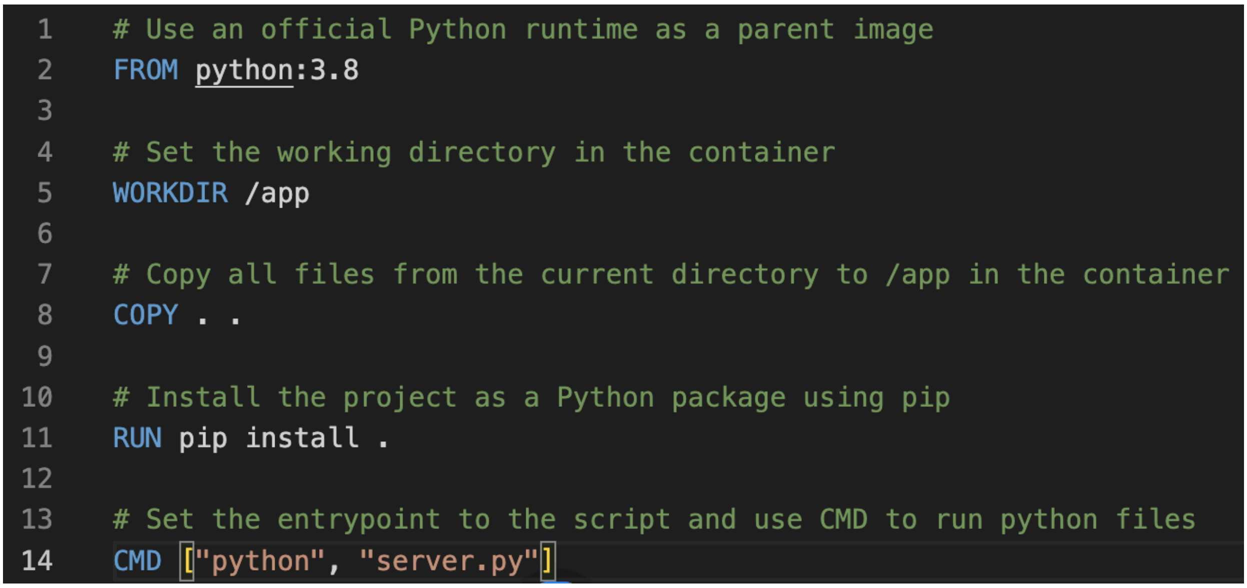
We need to install Docker at each of four VMs. The detailed steps for the installation of Docker in Ubuntu VM are mentioned at AusCAT documentation:

https://australiancancerdatanetwork.github.io/auscatverse/simulation/DOCKER_PORTAINER.html

A.3.2. DockerFiles

Once the docker is installed, we need to create docker images on the VMs using DockerFiles. The DockerFile for the server and the client will be a bit different; though a same DockerFile will be used for all three clients.

The server DockerFile is illustrated in the following figure:

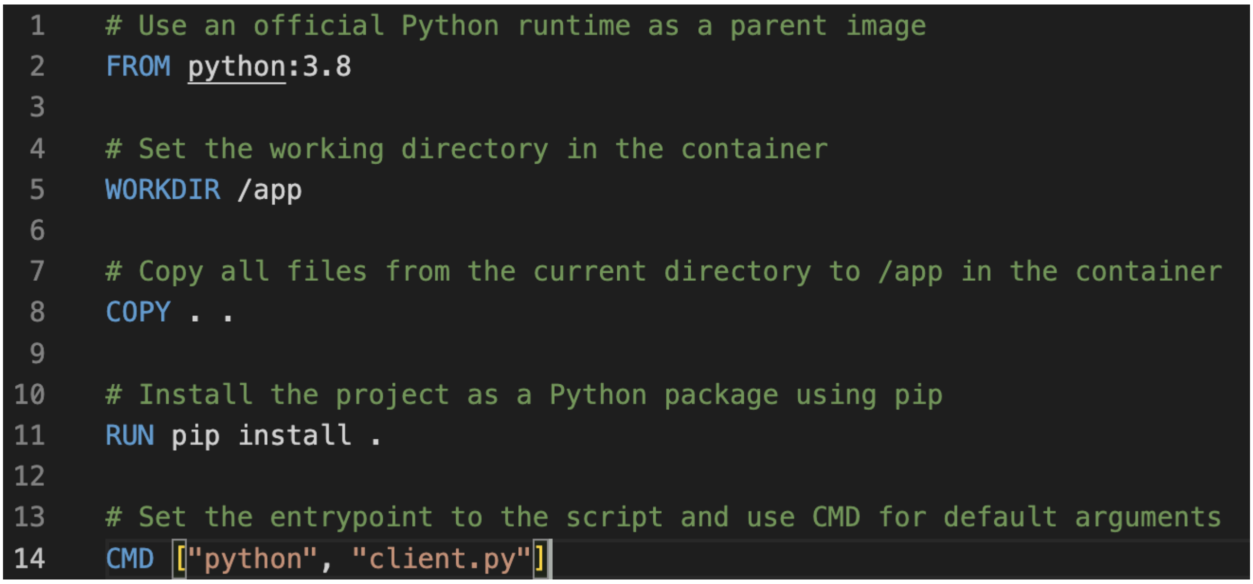


```
# Use an official Python runtime as a parent image
FROM python:3.8

# Set the working directory in the container
WORKDIR /app

# Copy all files from the current directory to /app in the container
COPY . .

# Install the project as a Python package using pip
RUN pip install .

# Set the entrypoint to the script and use CMD to run python files
CMD ["python", "server.py"]
```

First, we are using a python base image to install it. The working directory of the container is set to /app (this will be used when we run the container of the image). Next, we are copying all the files from the local machine current directory to the container current directory (which is /app set in the previous line); need to make sure we have all the required files (client python file, data file, pyproject.toml and

DockerFile). Then, we are installing the required packages mentioned in pyproject.toml. Finally, the servor python file is being run at the end.

The client DockerFile is illustrated in the following figure:



```
# Use an official Python runtime as a parent image
FROM python:3.8

# Set the working directory in the container
WORKDIR /app

# Copy all files from the current directory to /app in the container
COPY . .

# Install the project as a Python package using pip
RUN pip install .

# Set the entrypoint to the script and use CMD for default arguments
CMD ["python", "client.py"]
```

The only difference between is the client python file being run at the last line as compared to the server DockerFile.

A.3.3. Build and Run Images

To build the Docker Image using DockerFile, use the following syntax:

```
docker build -t [name-of-the-image] -f [name-of-the-DockerFile]
```

Once the images are created/build, we need to run the containers for these images on the respective VMs using the following syntax:

```
docker run -it --rm -v $(realpath ../../data):/app/data f [name-of-the-image]
```

-it: it is for interactive mode

-v: to mount the host’s directory to the container’s directory.

Bibliography

Beutel, D. J., Topa, T., Mathur, A., Qiu, X., Fernandez-Marques, J., Gao, Y., . . . Lane, N. D. (2020). Flower: A Friendly Federated Learning Research Framework. *arXiv:2007.14390*.

Amazon. (2024). *What is Cloud Native?* Retrieved from https://aws.amazon.com/what-is/cloud-native/#:~:text=Cloud%20native%20is%20the%20software,quickly%20to%20meet%20customer%20demands.

ARDC. (2024). *E-Research Institutional Cloud Architecture (ERICA)*. Retrieved from https://ardc.edu.au/project/e-research-institutional-cloud-architecture-erica/

Australia, C. (2024). *Australian Cancer Plan*. Retrieved from https://www.canceraustralia.gov.au/sites/default/files/publications/pdf/2023_ACP%20Summary%20Report%20DIGITAL_V9.pdf

Chen, J. Q., & Benusa, A. (2017). HIPAA security compliance challenges: The case for small healthcare providers . *International Journal of Healthcare Management , 10*(2), 135-146.

Close, T. G., Ward , P. G., Sforazzini, F., Goscinski, W., Chen, Z., & Egan, G. F. (2020). A Comprehensive Framework to Capture the Arcana of Neuroimaging Analysis . *Neuroinformatics, 18*(1), 109-129.

CNCF. (2024). *Do I need Kubernetes?* Retrieved from https://www.cncf.io/blog/2022/07/07/do-i-need-kubernetes/

Ekmefjord, M., Ait-Mlouk, A., Alawadi, S., Åkesson, M., Singh, P., Toor, S., & Hellander, A. (2022). Scalable federated machine learning with FEDn. Taormina, Italy: 2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid).

Fereidooni, H., Marchal, S., Miettinen, M., Mirhoseini, A., Möllering, H., Nguyen, T. D., . . . Zeitouni, S. (2021). SAFELearn: Secure Aggregation for private FEderated Learning. 2021 IEEE Security and Privacy Workshops (SPW).

Field , M., Thwaites, D. I., Carolan , M., Delaney, G. P., Lehmann, J., Sykes, J., . . . Holloway, L. (2022). Infrastructure platform for privacy-preserving distributed machine learning development of computer-assisted theragnostics in cancer. *Journal of Biomedical Informatics, 134*.

Field, M., Vinod , S., Delaney, G. P., Aherne , N., Bailey, M., Carolan, M., . . . Holloway, L. (2024). Federated Learning Survival Model and Potential Radiotherapy Decision Support Impact Assessment for Non-small Cell Lung Cancer Using Real-World Data. *Clin Oncol (R Coll Radiol), 36*(7), e197-e208.

Field, M., Vinod, S., Aherne, N., Carolan , M., Dekker, A., Delaney, G., . . . Thwaites, D. (2021). Implementation of the Australian Computer-Assisted Theragnostics (AusCAT) network for radiation oncology data extraction, reporting and distributed learning. *J Med Imaging Radiat Oncol, 65*(5), 627-636.

Goddard, M. (2017). The EU General Data Protection Regulation (GDPR): European Regulation that has a Global Impact. *International Journal of Market Research, 59*(6), 703-705.

Hansen, C. R., Price, G., Field, M., Sarup, N., Zukauskaite, R., Johansen, J., . . . Brink, C. (2022). Larynx cancer survival model developed through open-source federated learning. *Radiother Oncol*, 176:179.

Li, L., Fan, Y., Tse, M., & Lin, K.-Y. (2020). A review of applications in federated learning. *Computers & Industrial Engineering*, 106854.

Ludwig, H., Baracaldo, N., Thomas, G., Zhou, Y., Anwar, A., Rajamoni, S., . . . Abay, A. (2020). IBM Federated Learning: an Enterprise Framework White Paper V0.1. *arXiv:2007.10987*.

Mansouri, M., Önen, M., Jaballah, W. B., & Conti, M. (2023). SoK: Secure Aggregation Based on Cryptographic Schemes for Federated Learning. Lausanne: Proceedings on Privacy Enhancing Technologies.

Moncada-Torres , A., Martin, F., Sieswerda, M., Soest, J. V., & Geleijnse, G. (2021). VANTAGE6: an open source priVAcy preserviNg federaTed leArninG infrastructurE for Secure Insight eXchange . AMIA Annu Symp Proc.

Moore, H. C., Guiver , T., Woollacott, A., Klerk, N. d., & Gidding, H. F. (2016). Establishing a process for conducting cross-jurisdictional record linkage in Australia . *Aust N Z J Public Health ., 40*(2), 159-64.

ONS. (2024). *Secure Research Service* . Retrieved from https://www.ons.gov.uk/aboutus/whatwedo/statistics/requestingstatistics/secureresearchservice

QLD. (2024). Cancer Alliance Queensland. *https://cancerallianceqld.health.qld.gov.au/*.

Reina, G. A., Gruzdev, A., Foley, P., Perepelkina, O., Sharma, M., Davidyuk, I., . . . Bakas, S. (2021). OpenFL: An open-source framework for Federated Learning. *arXiv:2105.06413*.

Renton, A. I., Dao, T. T., Johnstone, T., Civier, O., Sullivan, R. P., Spitz, G., . . . Bollmann, S. (2024). Neurodesk: an accessible, flexible and portable data analysis environment for reproducible neuroimaging. *Nature Methods, 21*, 804–808.

Riedel, P., Schick, L., Schwerin, R. v., Reichert, M., Schaudt, D., & Hafner, A. (2024). Comparative analysis of open-source federated learning frameworks - a literature-based survey and review. *International Journal of Machine Learning and Cybernetics*, 1-22.

Roth, H. R., Cheng, Y., Wen, Y., Yang, I., Xu, Z., Hsieh, Y.-T., . . . Feng, A. (2022). NVIDIA FLARE: Federated Learning from Simulation to Real-World. New Orleans: 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

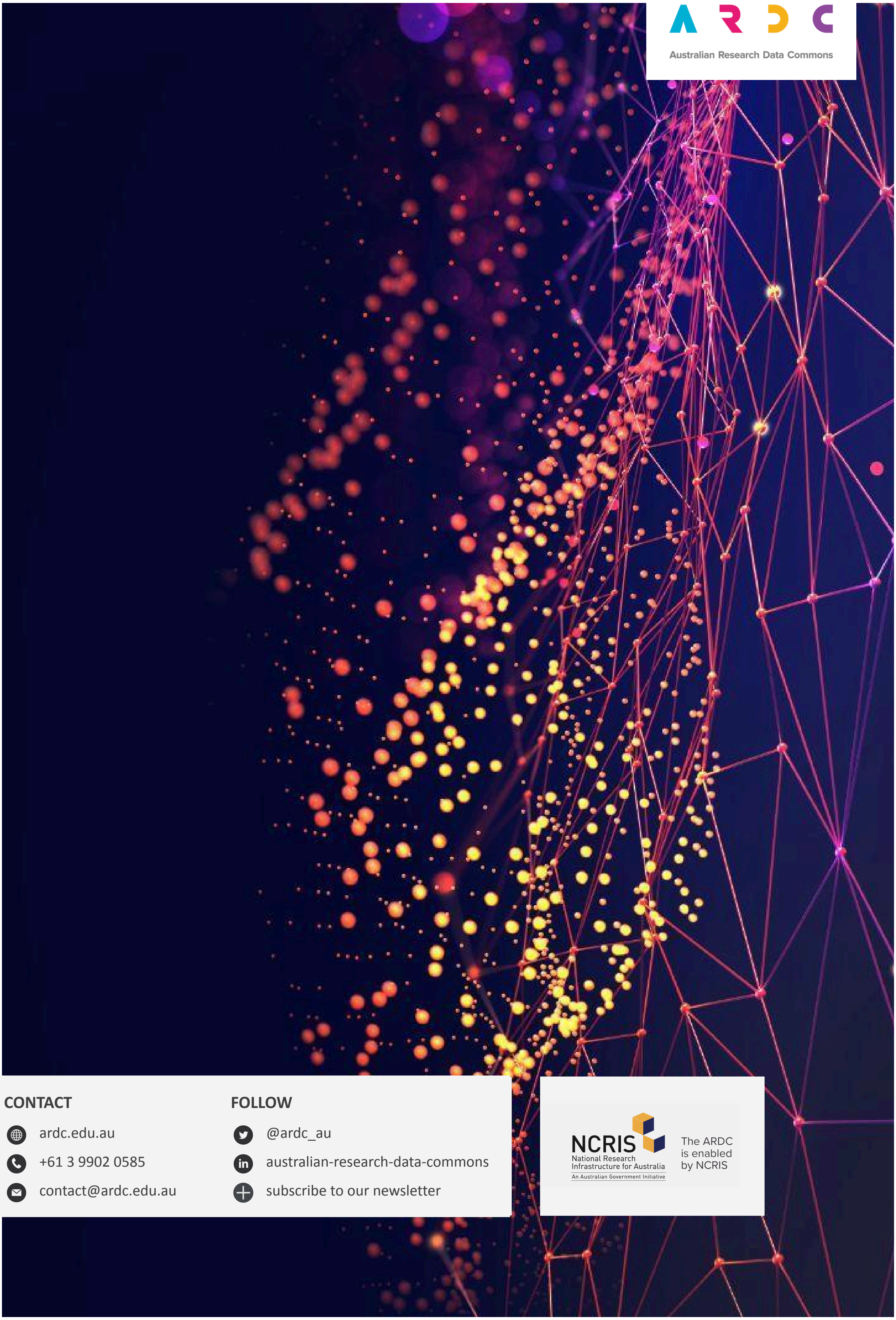
Theophanous, S., Lønne, P.-I., Choudhury, A., Berbee, M., Dekker, A., Vassiliou, V., . . . Appelt, A. L. (2022). Development and validation of prognostic models for anal cancer outcomes using distributed learning: protocol for the international multi-centre atomCAT2 study. *Diagn Progn Res., 6*(14), 1-11.

Veritis. (2024). *Kubernetes Adoption: The Prime Drivers and Challenges*. Retrieved January 2024, from https://www.veritis.com/blog/kubernetes-adoption-the-prime-drivers-and-challenges/

Wibawa, F., Catak, F. O., Sarp, S., & Kuzlu, M. (2022). BFV-Based Homomorphic Encryption for Privacy-Preserving CNN Models . *Cryptography, 6*(3), 34.

Zhang, Y., Lu, Y., & Liu, F. (2023). A Systematic Survey for Differential Privacy Techniques in Federated Learning. *Journal of Information Security, 14*(2), 111-135.

Ziller, A., Trask, A., Lopardo, A., Szymkow, B., Wagner, B., Bluemke, E., . . . Kaissis, G. (2021). PySyft: A Library for Easy Federated Learning. In *Federated Learning Systems* (pp. 111-139). Springer.



Australian Research Data Commons

CONTACT

ardc.edu.au

+61 3 9902 0585

contact@ardc.edu.au

FOLLOW

@ardc_au

australian-research-data-commons

subscribe to our newsletter

NCRIS
National Research
Infrastructure for Australia
An Australian Government Initiative

The ARDC
is enabled
by NCRIS