National College of Ireland

Enhancing Data Integrity in Federated Learning for Cloud Load Prediction Using Blockchain

MSc Research Project
MSc Cloud Computing

Abdullah Khan
Student ID: X23364971

School of Computing
National College of Ireland

Supervisor: Shreyas Setlur Arun

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Abdullah Khan
Student ID:	X23364971
Programme:	MSc Cloud Computing
Year:	2024-2025
Module:	MSc Research Project
Supervisor:	Shreyas Setlur Arun
Submission Due Date:	11/12/2025
Project Title:	Enhancing Data Integrity in Federated Learning for Cloud Load Prediction Using Blockchain
Word Count:	822
Page Count:	11

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Abdullah Khan
Date:	4th December 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Data Integrity in Federated Learning for Cloud Load Prediction Using Blockchain

Abdullah Khan
X23364971

1 Introduction

1.1 Purpose of the Configuration Manual

The presented manual includes specifications of how the federated learning system should be set up, configured, and deployed to the cloud to perform load forecasting and guarantee the integrity of data and transparency of the model used.

1.2 Scope of the Document

The manual includes system architecture, organization of AWS S3 to store data, configuration of the federated learning models, and the implementation of a blockchain-based layer of verification to overcome the effects of adversarial attacks.

1.3 Target Audience

This document is intended for someone who wants to replicate that can also include:

- AI/ML scholars and practitioners
- Software developers and system integrators
- Examiners and academic supervisors of the project
- Managing deployment through cloud infrastructure by clients or stakeholders

1.4 System Overview

The system uses the federated learning (FL) technology that facilitates decentralized training of data but does not invade privacy. Integration of blockchain technology is done to guarantee integrity of updates in the model and data. Scalable data storage is provided through AWS S3, and IAM roles and SigV4 authentication are used to provide access security.

2 System Requirements

2.1 Hardware Requirements

The system has no need of high-end computing hardware to experiment or train the model. My system specifications under which I have worked however are the following:

- **RAM:** 16 GB
- **CPU:** Core(TM) i5-11th Generation
- **Disk Space:** 238GB

2.2 Software Requirements

To install the project and execute its activities, the following software applications, libraries, and environments are needed:

- **Operating System:** Windows 11 Pro Insider Preview 64-bit (Build 26100.7171)
- **Python:** Version 3.11.13
- **Libraries:**
  - Python: 3.11.13
  - NumPy: 1.26.4
  - Pandas: 2.2.3
  - Matplotlib: 3.7.2
  - Scikit-learn: 1.2.2
  - TensorFlow: 2.18.0
  - Keras: 3.8.0
- **Tools:**
  - `Jupyer Note Book`
  - `Kaggle`
  - `AWS IAM` credentials

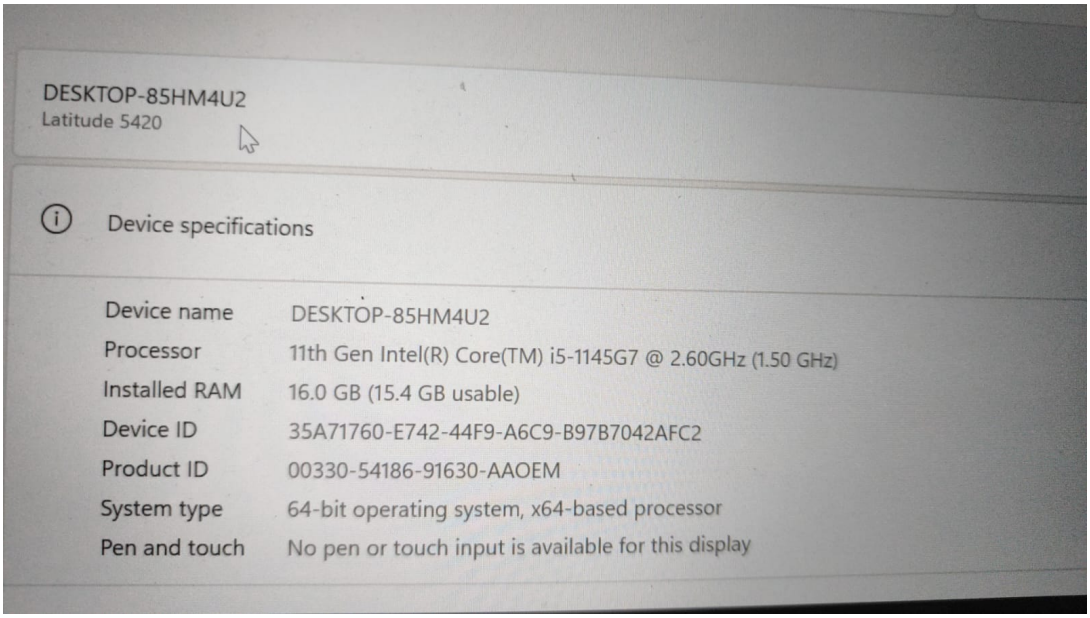


Figure 1: System Information

2.3 Network Requirements

To run the project the following tasks need an active internet connection:

- Downloading Python packages and dependencies
- Accessing the Kaggle runtime environment and datasets
- To facilitate effective and timely data synchronization with the cloud storage

For AWS deployment Amazon Web Services (2023), make sure that the IAM user is given the required permissions and policies to access S3, and endpoint services.

3 Installation Instructions

This section details how the necessary environment and dependencies required by the system can be set up.

3.1 Prerequisites

The prerequisites to be followed before the system is installed are as follows:

- Kaggle account for model development and training
- AWS IAM credentials
- Connection to the Internet to install the necessary libraries

To install the necessary libraries in Python, the command below would be used when in Kaggle or a local machine:

Install Required Libraries: Install the necessary libraries using `pip`.

```
pip install numpy==1.26.4 pandas==2.2.3 matplotlib==3.7.2
pip install scikit-learn==1.2.2 tensorflow==2.18.0 keras==3.8.0
```

Install AWS SDK: Install the AWS interactive library known as `boto3`.

```
pip install boto3
```

Check Installation: The libraries versions can be checked by the following commands Ramírez (2023); Pedregosa et al. (2011); Chen et al. (2020); Manaswi (2018):

```
import numpy, pandas, matplotlib, sklearn, tensorflow, keras
print(numpy.__version__)
print(pandas.__version__)
print(matplotlib.__version__)
print(sklearn.__version__)
print(tensorflow.__version__)
print(keras.__version__)
```

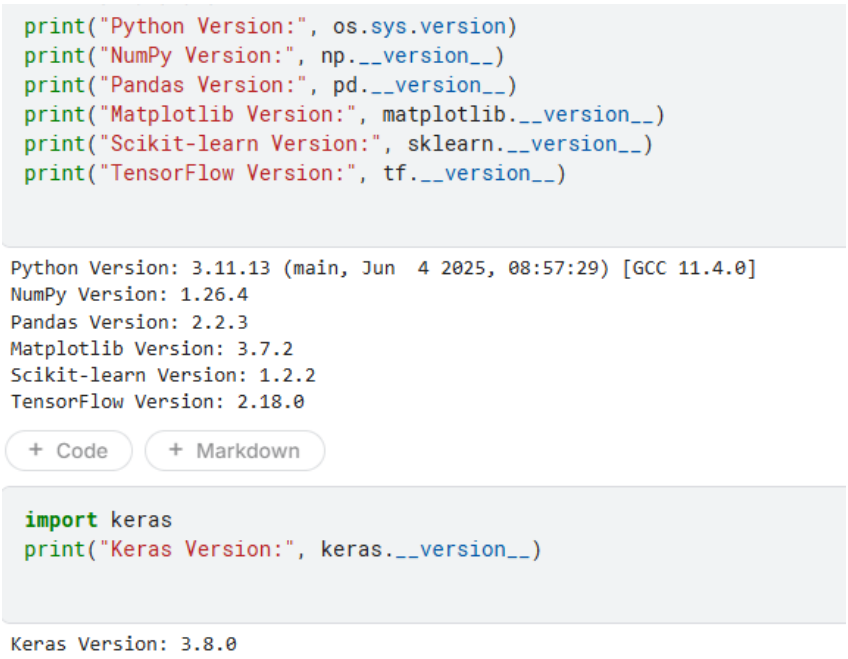


Figure 2: Verify Libraries

3.2 Installation Steps

The model was developed and tested using Kaggle. The following steps were taken:

1. Log in to the kaggle and make a new Notebook.
2. Upload dataset using the **Add Data** panel of the Kaggle notebook interfaces.
3. Run all cells and get the results

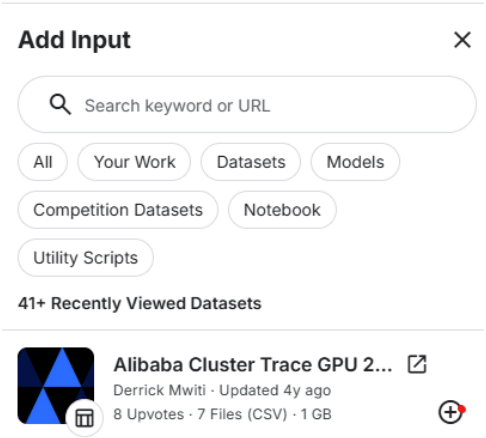


Figure 3: Uploading dataset to Kaggle

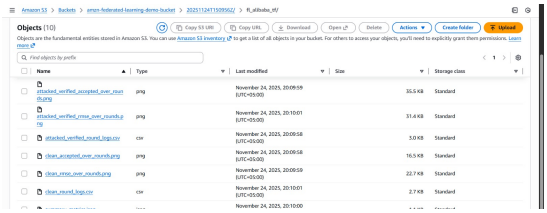
Figure 4: Running the model training and evaluation pipeline

4. S3 Bucket: Provision in the region (e.g., eu-north-1 in Stockholm) an S3 bucket where all the outputs of ML/federated-learning are going to be stored.

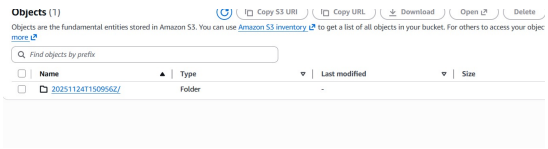
5. Create IAM Role: This step will create a programmatic IAM user/role to access the services in AWS.

6. Set up SDK Client: Set up an AWS SDK client (boto3) using IAM credentials and region (eu-north-1).

7. Validate Persistence: Call the ListObjectsV2 API with the help of which one ensures the upload and data persistence.

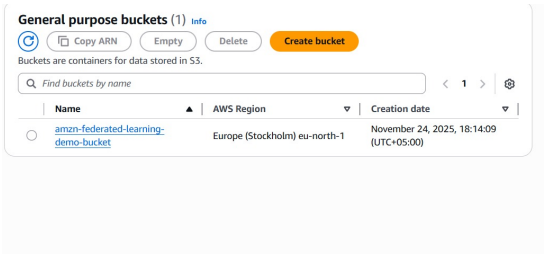


(a) S3 bucket object listing with experiment files

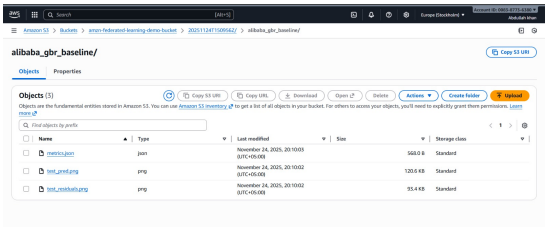


(b) Overview of the S3 bucket for federated learning

Figure 5: (a) AWS Deployment



(a) S3 folder view with timestamped directory



(b) S3 object listing for Alibaba GBR baseline model

Figure 6: (b) AWS Deployment

References

Amazon Web Services (2023). Boto3 documentation, `https://boto3.amazonaws.com/v1/documentation/api/latest/index.html`.

Chen, J., Zhao, F., Sun, Y. and Yin, Y. (2020). Improved xgboost model based on genetic algorithm, *International Journal of Computer Applications in Technology* **62**: 240–245. **URL:** */doi/pdf/10.1504/IJCAT.2020.106571*

Manaswi, N. K. (2018). Understanding and working with keras, *Deep Learning with Applications Using Python* pp. 31–43. **URL:** *https://link.springer.com/chapter/10.1007/978-1-4842-3516-4*$_2$

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine learning in python, *Journal of Machine Learning Research* **12**: 2825–2830.

Ramírez, S. (2023). Fastapi: Web framework for building apis with python 3.6+ based on standard python type hints, `https://fastapi.tiangolo.com/`.