National College of Ireland

Cloud-Based Federated Learning System for Financial Fraud Detection

MSc Research Project
Cloud Computing

Rajesh Reddy Madduri
Student ID: 23340231

School of Computing
National College of Ireland

Supervisor: Shaguna Gupta

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Rajesh Reddy Madduri
Student ID:	23340231
Programme:	Cloud Computing
Year:	2025
Module:	MSc Research Project
Supervisor:	Shaguna Gupta
Submission Due Date:	15/09/2025
Project Title:	Cloud-Based Federated Learning System for Financial Fraud Detection
Word Count:	9082
Page Count:	26

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:	Rajesh Reddy Madduri
Date:	15th September 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):	

Cloud-Based Federated Learning System for Financial Fraud Detection

Rajesh Reddy Madduri
23340231

Abstract

This paper introduces a Docker-based federated learning model to conduct financial fraud detection without compromising data privacy at the expense of effective detection. As opposed to conventional centralized systems which imply risks in terms of both privacy and regulation, the suggested solution lies in the containerized edge computing with federated learning, in which financial institutions can join efforts to comprehend fraudulent activity without exchanging raw data on transactions. The orchestration applied by the system is Docker Compose with 3 edge client containers each running 12,000 transactions locally and participating in a global fitting model via secure aggregation and d-privacy (d=1.2). The combination of SHAP and LIME delivers explainable AI on the edge in real-time, by making interpretable decisions in under 300 milliseconds per transaction. Experiments prove that exactly 93.34 per cent of recalls were obtained in fraud detection with 82.06 per cent AUC-ROC focused on capturing fraud rather than reducing false positive. The implementation is able to maintain (1.20, 1e-05)-differential privacy guarantees and within four federated learning rounds, the client participation achieved is 100 percent. The proposed integration scheme S3 AWS allows distributing models without any centralization of the data, confirming the benefits of using privacy-preserving federated learning in situations of compliance regulatory control and operational needs of detecting financial fraud.

1 Introduction

1.1 Research Background

Due to the rise in digital financial transactions, the banking and financial services sector has undergone major changes that have made it possible for more trade to occur, as well as more opportunities for fraud. Previously, financial fraud was detected with rules, but now machine learning is used to spot complex and unusual trends in transaction information (Njoku et al.; 2024). Traditional machine learning models that are centralized have shown excellent results in fraud detection using extensive data to create accurate models that make decisions on the spot (Ali et al.; 2022). On the other hand, this way of handling data means critical information from many sources has to be stored in one place, causing major problems with privacy and following rules. With the introduction of federated learning, it is now possible to work together on fraud detection without endangering personal information (Awosika et al.; 2024). Thanks to containerization and microservices, cloud computing technologies give the required support for distributed federated learning in

organizations at a large scale (Chahoud et al.; 2023). Using cloud orchestration and federated learning provides new chances to create secure and compliant fraud detection systems that can work at a large scale (McCall; 2023).

1.2 Motivation

The reason for undertaking this research is to ensure that fraudulent activities are spotted without sacrificing privacy in the regulated financial sector. Due to GDPR and similar laws across the world, using old centralized systems is no longer an option because it violates privacy, so new approaches that support privacy and security are needed (Awosika et al.; 2024). Since banks often collaborate on financial fraud, they need to work together, but this current approach to information sharing creates too many risks and places the banks at greater risk of breaking regulations and breaching privacy. The application of federated learning together with cloud-native tools can lead to the development of fraud detection systems with better privacy, meet stricter rules, and work more transparently (Zheng; 2025). Moreover, with explainable AI, the growing demand for interpreted decisions in the financial field is met. Automated detection can better be trusted and remains private thanks to federated models.

1.3 Problem Statement

Presently, fraud detection relies on placing multiple types of financial data from various institutions into central databases to be trained on and generate results (Ali et al.; 2022). Certainly, this centralization leads to significant risks such as more data breaches, loss of privacy, and failure to keep up with tough data protection laws, for example, GDPR. Organizations in the financial sector are faced with the problem of choosing if they want to keep data confidential and risk missing out on detection or share customer records and put customer privacy and compliance at risk by following a standardized system (Awosika et al.; 2024). Since existing fraud detection models lack transparency, people in the industry become even more concerned, as they are not able to see the steps in the decision-making about financial activities and client connections. These kinds of rigid systems cannot adjust to the flexibility and privacy-related needs of the present-day financial marketplace.

1.4 Research Question

In what way can having cloud-based federated learning with explainable AI help various financial institutions collaborate on fraud detection, preserve their privacy and follow regulations, and maintain clear and understandable decision-making practices?

1.5 Problem Solution

To answer this research question, a proper cloud-based architecture for federated learning will be engineered and tested. The plan is to create a multi-step process in which financial institutions run target models locally on cloud servers and only send out encrypted model updates through secure systems. Data protection will be maintained during collaborative training when differential privacy and secure multi-party computation are applied. By using edge computing, it will be possible to detect fraud in real time at the time of

transactions, all while keeping in touch with cloud-based management. SHAP and LIME methods, types of Explainable AI, will be included in the framework to ensure decisions are seen and understood. By performing experiments with real financial data, the study will prove that using the decentralized framework achieves the same results in catching fraud as traditional methods, while keeping privacy rules secured and following related regulations. To confirm the proposal is viable for enterprise deployment, the model's accuracy, effectiveness at private data protection, amount of effort needed to run, and eagerness to explain its decisions must be confirmed by performance metrics.

1.6 Research Contributions

The main objective of this research is to design, implement, and review a cloud-native federated learning framework aimed at protecting customers' financial data in collaboration with financial institutions using explainable AI, while also extending beyond existing solutions through several key innovations as listed below.

- Building a novel federated learning platform that offers differential privacy without any excessive privacy budget overhead reminiscent of existing federated approaches, while also maintaining higher recall rates.
- This is the foremost work that integrates both LIME and SHAP directly in containerized edge environments for explainable AI predictions .
- This federated learning framework is implemented using docker containerization which handles degradation mechanisms in a graceful way in addition to allowing 100% client participation rates.
- This work overcomes the issue with current federated fraud detection models by combining differential privacy, secure aggregation, and homomorphic encryption with fallback options which protects privacy even during unexpected component failures.

2 Related Work

The review considers current studies in financial fraud detection that focus on machine learning, federated learning that ensures privacy, and building cloud computing infrastructure. The study combines information from fifteen current studies to discover where research is lacking and to lay down the foundation for using cloud-based federated learning methods in fraud detection systems.

2.1 Machine Learning in Financial Fraud Detection

In a systematic review, (Ali et al.; 2022) reviewed machine learning techniques for spotting financial fraud dating from 2010 through 2021. The study deals with the rising problem of catching financial fraud in today's digital financial world, where the old way of verifying documents is not enough. Applying the Kitchenham technique, the authors looked at 93 relevant papers to spot the top ML methods, frequent fraud cases, and proper ways to measure the efficacy of countermeasures. It was found that Naive Bayes, Support Vector Machines, and Artificial Neural Networks are the techniques used the most, and

bank fraud (50%) and financial statement fraud (29%) are the fraud types analyzed more often. The study showed that finding insurance fraud and new types of fraud connected to cryptocurrency is difficult, while pointing out that better and faster ways to identify them are necessary. The approach developed by (Talukder et al.; 2024) for credit card fraud transaction detection was based on combining several machine learning techniques to overcome the issue of data imbalance in fraudulent dataset. The study focuses on situations where the number of fraudulent transactions is very small compared to the rest of the transactions, which makes traditional machine learning models useless. To develop this approach, the authors used Random Under-Sampling, Cluster Centroids, and Hybrid Techniques in addition to Random Forest, Decision Tree, and K-Nearest Neighbors. The process they followed involved choosing important features, fixing imbalanced data issue, and building a variety of final models. The research proved that Random Forest had a very high accuracy rate of 99.96% and outperformed all other single algorithms by a large margin. Khalid et al. (2024) handled credit card fraud detection by combining various methods and working on the features to achieve better accuracy when dealing with data imbalance. The work handles the twin problem of uneven datasets and the requirement to detect fraud right away in situations with many transactions. Authors dealt with the data by performing feature scaling, finding outliers, and using SMOTE to generate synthetic data. Their approach used several algorithms like Gradient Boosting, Random Forest, and Neural Networks along with advanced ways to pick important features. The research resulted in better precision, recall, and F1-score and proved that ensemble methods are helpful for minimizing errors that may affect customer trust in banks. Zheng (2025) carefully evaluated several advanced federated learning models in relation to credit card fraud detection, mainly focusing on how to handle issues with privacy and unbalanced transaction data. One of the main issues the study considers is ensuring privacy for users' data while training models to accurately identify fraud, since fraud in financial systems is extremely rare. Three federated learning approaches were examined by the author: FedGAT-DCNN for Dilated Convolutional Neural Networks and Graph Attention Networks, FedAvg with CNN, and FedAvg-DWA with a Random Forest. FedGAT-DCNN outperformed other methods by achieving a ROC-AUC score of 0.9992, but FedAvg with CNN got 0.969 AUC score and 0.9534 F1-score. Both positives and negatives of federated learning for fraud detection were underlined by the study, including differing types of systems, expensive communications, and data imbalance, and it suggested strategies for overcoming these obstacles. Aljunaid et al. (2025) suggested a XFL system that makes use of SHAP and LIME explainable AI to noticeably catch fraud without violating regulations and privacy terms. The study points out that centralized ML models are not flexible, produce many false positives, and may violate data privacy in the strict financial guidelines that exist today. The framework developed by the authors uses both federated learning and modern explainable AI techniques, so that financial institutions can perform fraud detection without sharing confidential data. The process moves from getting data, preprocessing, training locally, and finally federation to develop an excellent and scalable AI-based security solution for financial businesses. It was proven that the integrated way supported accurate detection and promoted clear actions, trust, and proper implementation of rules to handle advancing financial crimes.

2.2 Federated Learning and Privacy-Preserving Techniques

Fed-RD, a federated learning structure for uniting different organizations in detecting financial fraud, was presented by (Khan et al.; 2024) and it uses modern confidentiality techniques such as differential privacy and secure multi-party computation. The focus of the study is on how to support cooperation on detecting fraud by financial institutions, all the while observing both privacy and regulatory requirements such as GDPR, CCPA, and AML laws. The authors proposed a detailed framework that makes it possible for institutions to learn effective fraud detection models while still avoiding the need to share their sensitive financial information directly. They rely on secure aggregation, encrypted sharing, and differential privacy so that data is preserved during the collaborative learning process. Research showed that Fed-RD is as effective as centralized methods at spotting fraud, yet much better at safeguarding privacy and complying with regulations, so it is suited for detecting fraud promptly in controlled situations.

In their paper, (Tayyeh and AL-Jumaili; 2024) describe a differential privacy method for federated learning designed to balance privacy and model effectiveness, mainly in the environment where data is not the same across devices. The research deals with the basic trade-off in privacy-preserving federated learning, which means that both adding too much noise and adding too little do not work well. A full framework was designed by the authors that included introducing noise before collecting results, adjusting the height of clipped values, and minimizing privacy costs through optimization. The approach of the researchers was to rigorously assess epsilon, delta, how data is selected, and how data converges in different circumstances. It was found that for differential privacy to be fully integrated in federated learning, the researchers had to balancing the noise and privacy parameters to ensure high privacy and adequate accuracy in non-IID situations.

Abdul Salam et al. (2024) built a federated learning system for spotting credit card fraud that protects privacy and enables the detection to function uniformly in different institutions. This study tries to help financial institutions cooperate to find fraud without sharing details about their customers due to worry about privacy and regulations. Authors introduced an architecture in which every institution trains its own data on a local model and supplies the central aggregator with just the cryptographically protected model parameters. They opted for safe aggregation methods, compression of gradient updates, and a system that could choose the right learning rate to speed up convergence and save on communication. It was demonstrated in the research that federated learning can match centralized fraud detection accuracy, yet still keep the data within the organizations involved and ensure their privacy, which is why it is especially apt for consortium-based fraud detection activities among banks and other financial institutions.

Awosika et al. (2024) developed a model for fraud detection using federated learning and added explainable AI to ensure finance companies addressed privacy and transparency issues. The issue this study deals with is that fraud detection needs a lot of data from different sources, which is difficult given the strict regulations of privacy laws like GDPR. They proposed designing a distributed deep neural network, which specifically aims to detect suspicious patterns in data kept in several databases and maintains its data in those locations. SHAP and LIME methods were used by them to explain the workings of their model and make sure it is easy to understand. Research results proved that federated learning was as accurate at detecting fraud as centralized learning but provided stronger data privacy and straightforward explanations for fraud decisions, important for following regulations and earning trust.

Chahoud et al. (2023) suggested building a distributed federated learning setup that uses edge computing, and container technology to raise the level of privacy in machine learning for mobile applications. Their research efforts are directed at overcoming difficulties in mobile and edge environments by using federated learning instead of centralized approaches, which cannot be used because the bandwidth is limited, and privacy is a concern there. The orchestrator was built using Kubernetes and it manages federated learning clients across a variety of edge and mini-server devices, making sure both efficiency and privacy are maximized. The way they are designed handles device movement, makes use of all resources, and safely communicates among groups for effective and scalable federated learning systems. The use of federated learning showed greater and faster communication, lower delay, and stronger privacy security than usual approaches, and kept the models' accuracy regardless of changes in the devices and network settings.

2.3 Cloud Computing Design Pattern

Yuan and Liao (2024) came up with a prediction-based auto-scaling strategy that handles resource management concerns for cloud-native Kubernetes apps by reviewing historical data. Containerized environments present a serious challenge as reactive scaling usually creates disruptions and does not maximize the use of resources. To address the issue of workload fluctuations, the authors suggested an auto-scaling Kubernetes Operator that looks ahead and proactively changes the resources allocated using Holt-Winter and GRU libraries. The way they operate is to gather live metrics, analyze data over a period of time, and intelligently adjust resources to help with efficiency in both performance and low cost. It was found that workload prediction improved by 83.3% in quality and there was less fluctuation using the algorithm, saving over an hour in cold start time, when compared with the previous approaches that had one algorithm.

Aly et al. (2025) launched KubeDeceive, a system that merges several threat detection methods with deceptive ways mainly meant for Kubernetes setups. The study is focused on the increased security problems faced by containerized infrastructures due to shorter protection from traditionally used defenses against clever attacks on applications used in the cloud. To achieve their goals, the researchers applied a method based on machine learning for identifying threats in real time and instantly deploying security deception by using Principal Component Analysis and Autoencoders for extracting and pinpointing features of threats. KSniff is used to detect traffic, CICFlowMeter is responsible for extracting features, Naive Bayes works for classification, and decoy servers are used to excite and watch the attackers while gathering valuable information about their moves. The research showed that quick attack recognition and flexible deception were effective in boosting safety for Kubernetes environments without harming the way the system operates and by keeping false alarms low.

McCall (2023) carried out a thorough investigation of financial fraud detection implementing AI in the cloud, especially focusing on federated learning. It analyses how relying on cloud-native tools can assist modern financial fraud detection systems by scaling as needed, ensuring their resilience to threats, and following rules in many parts of the world. The author looked into the models for deploying cloud services, security guidelines, and data protection rules used by financial institutions to apply AI in fraud detection. The study showed that it is possible to use cloud-federated learning to guard against fraud instantaneously and still observe data privacy. According to the study, using cloud-based federated learning will help build new fraud detection systems that must perform well

and ensure strict privacy.

Issaoui et al. (2023) examined ways of using microservices architectures on the cloud to support scalable financial systems, paying attention to how containers and orchestration can improve reliability and overall performance. The challenge of switching to cloud-based technology in financial systems while meeting rules and efficiency needs is considered in this research. The authors suggested using Docker, Kubernetes, and service mesh to frame a microservices architecture that ensures robustness and can manage a high number of transactions in the financial sector. The solution used distributed transaction management, linked the APIs through a gateway, and had pipelines for automatic deployment to make services consistent throughout the cloud. It was proven that microservices have much better scalability, more fault tolerance, and faster deployment than monolithic architectures, but remain as secure and compliant as traditional options.

Seth et al. (2024) looked into compliance and regulatory difficulties related to cloud computing in different industries and discussed the specific requirements and trending technology for more security. The study looks at how tough rules affecting cloud providers impact various industries, mainly healthcare, finance, and government, that need to preserve data security. According to the authors, they studied compliance frameworks such as HIPAA, PCI DSS, GDPR, and FedRAMP, focusing on AWS, Azure, and GCP platforms to find out what improvements could be made in compliance management. They considered recent trends such as AI-protected security, use of blockchain, and handling various clouds to make a company's operations more compliant. This study formed guidelines that companies can use to face compliance troubles as they benefit from advanced cloud solutions, highlighting that proactive compliance and sharing duties with experts are necessary.

2.4 Critical Analysis

The reviewed literature illustrates important areas that this study will successfully tackle. At present, machine learning for fraud detection is based on centralized solutions that bring together sensitive financial data in one place, which makes both privacy and regulation an issue, mainly because of GDPR and similar regulations. Even though federated learning can solve privacy issues, proposals by (Khan et al.; 2024) and (Abdul Salam et al.; 2024)) still do not have the cloud integration and scalability suitable for enterprises. These frameworks have problems with transferring too much data between users, resolving issues related to non-IID data, and being able to process real-time information. Besides, the use of privacy methods tends to weaken a model's accuracy due to too much noise, and cloud computing doesn't offer advanced features for financial federated learning settings. The proposed framework uses advanced strategies to solve the limited problems characteristic of existing solutions. Using Kubernetes and microservices from the cloud, as well as edge computing, makes it possible for financial organizations to detect frauds in real-time and maintain full compliance with data locality rules. Differential privacy, secure multi-party computation, and special federated averaging algorithms used by the framework help better secure users' data, yet not affect the accuracy of detection. With edge computing, fraud is detected instantly at the transaction level and the cloud services are still used, giving faster responses than older federated learning methods. The use of SHAP and LIME additionally supports showing how the model works, which is important for the financial sector and regulators to adopt it. After a thorough analysis and comparison, the literature study by (Awosika et al.; 2024) comes closest to the goals

Table 1: A Comparative Analysis of Existing Literature

Author	Algorithm	Objective	Platform	Results	Limitations
Ali et al. (2022)	Systematic Literature Review	Comprehensive analysis of ML techniques for fraud detection	Multiple databases (IEEE, ACM, Scopus)	NB, SVM, ANN most popular; 50% focus on bank fraud	Missing money laundering studies
Talukder et al. (2024)	Hybrid Ensemble (RF, DT, KNN)	Address class imbalance in credit card fraud	Python, Scikit-learn	99.96% accuracy with RF	Dataset-specific; Computational overhead
Khalid et al. (2024)	Ensemble Methods with SMOTE	Enhance fraud detection accuracy	Kaggle dataset, Python	Improved precision/recall	Limited to credit card transactions
Zheng (2025)	FedGAT-DCNN, FedAvg-DWA	Privacy-preserving federated fraud detection	Federated Learning, GAT, DCNN	ROC-AUC 0.9992 with FedGAT-DCNN	Communication costs; System heterogeneity
Aljunaid et al. (2025)	XFL with SHAP/LIME	Transparent federated fraud detection	Federated Learning frameworks	Enhanced transparency and compliance	Scalability concerns; Implementation complexity
Khan et al. (2024)	Fed-RD with DP & MPC	Collaborative fraud detection with privacy	Federated Learning, Differential Privacy	Comparable accuracy with enhanced privacy	Regulatory complexity; Implementation challenges
Tayyeh & AL-Jumaili (2024)	Noise before Aggregation FL	Balance privacy-performance in non-IID	Differential Privacy, Gaussian Noise	Optimal privacy-utility trade-off	Parameter sensitivity; Convergence issues
AbdulSalam et al. (2024)	Horizontal Federated Learning	Privacy-preserving credit card fraud detection	Secure Aggregation, TensorFlow	Maintained accuracy with data locality	Communication overhead; Scalability limits
Awosika et al. (2024)	Federated Learning + XAI	Privacy-preserving fraud detection with transparency	Deep Neural Networks, SHAP, LIME	Comparable accuracy to centralized models	Communication overhead; Complexity
Chahoud et al. (2023)	Edge-based Federated Learning	Mobile federated learning with containerization	Kubernetes, Docker, Edge Computing	Improved efficiency and privacy	Resource constraints; Device heterogeneity
Yuan & Liao (2024)	Holt-Winter + GRU Neural Network	Predictive auto-scaling for Kubernetes	Kubernetes, Time Series Analysis	MSE 0.00166; 83.3% fluctuation reduction	Single application focus; Parameter tuning complexity
Aly et al. (2025)	ML Classification + Cyber Deception	Integrated security for Kubernetes	PCA, Autoencoders, KSniff, CICFlowMeter	Real-time threat detection with deception	Computational overhead; Implementation complexity
McCall (2023)	Cloud-Federated Learning	AI applications in financial fraud detection	Cloud Computing, Compliance Frameworks	Enhanced security and compliance	Implementation complexity; Cost considerations
Issaoui et al. (2023)	Cloud-native Microservices	Scalable financial applications	Docker, Kubernetes, Service Mesh	Improved scalability and fault tolerance	Migration complexity; Integration challenges

of our research and will thus serve as the base paper. This paper looks at using explainable AI together with federated learning as an approach to fraud detection, which directly fits your research goals of using federated learning, cloud computing, and privacy-preserving strategies to detect financial fraud. Extensive evaluation with metrics displays how much improvement the framework has over existing approaches. Although papers such as (Awosika et al.; 2024) mainly assess how accurate a model is and whether privacy is maintained, this framework also includes more factors such as detection accuracy, privacy effectiveness, computational resources, scalability, and explainability. Significant performance measures are how well the system can detect fraud, save privacy budget, use efficient aggregation, deal with transactions quickly, and remain in agreement with privacy regulations.

3 Methodology

In this section, the proposed methodology to develop and test a cloud-based federated learning framework to detect financial frauds by improving the data privacy is described.

3.1 Research Plan

The study involves both theoretical work as well as applied and empirical review. The methodology is built on three main tracks, which include federated learning algorithm formulation, cloud platform design, and privacy-creating mechanism integration. The study takes the quantitative approach to measure system performance, and the efficiency of privacy preservation with the help of controlled experiments and simulation studies. The experiment is performed according to a structured methodology, where the initial step involves setting up a baseline model and followed by the addition of federated learning to the model by involving integration of the cloud, privacy-enhancing mechanisms, and explainable AI elements.

3.2 Dataset Description

The study employs data from financial fraud detection data used in (Jesus et al., 2022) study. The data is available in the official repository [1] or from the Kaggle link [2]. The data comprises 29042 realistic entries of financial transactions, which are contained in 32 unique features. Its essential characteristics are numerical variables, categorical features and binary features. The process of preparing the dataset consists of a preprocessing pipeline based on data manipulation libraries provided by Python. Some of the feature engineering steps comprise categorical variables one-hot encoding, StandardScaler feature scaling and standardization, and temporal feature extraction to identify seasonal trends and time-based relationships. The data balancing solves the issue of class imbalance inherent by using SMOTE.

The dataset suffers severe class imbalance with only 10 percent of the traffic being fraud and they have to be balanced accordingly to avoid biasness in the model to the majority. SMOTE was chosen because it has been shown to be quite useful in detecting financial frauds, as conducted by (Khalid et al.; 2024) and (Fernández et al.; 2018).

[1] https://github.com/feedzai/bank-account-fraud

[2] https://www.kaggle.com/datasets/sgpjesus/bank-account-fraud-dataset-neurips-2022

It creates synthetic minority subsets by interpolating between existing fraud cases, so that the data distribution is maintained and balanced training sets are constructed per individual federated client.The dataset is systematically partitioned into numerous simulated financial institutions in order to generate practical distributed learning environments as required by distributed learning simulation. Some factors incorporated in this partitioning strategy are geographic distribution trends, size differences in institutions, frequency/variation in fraud, and differences in customer demographics.

3.3 Proposed System Architecture

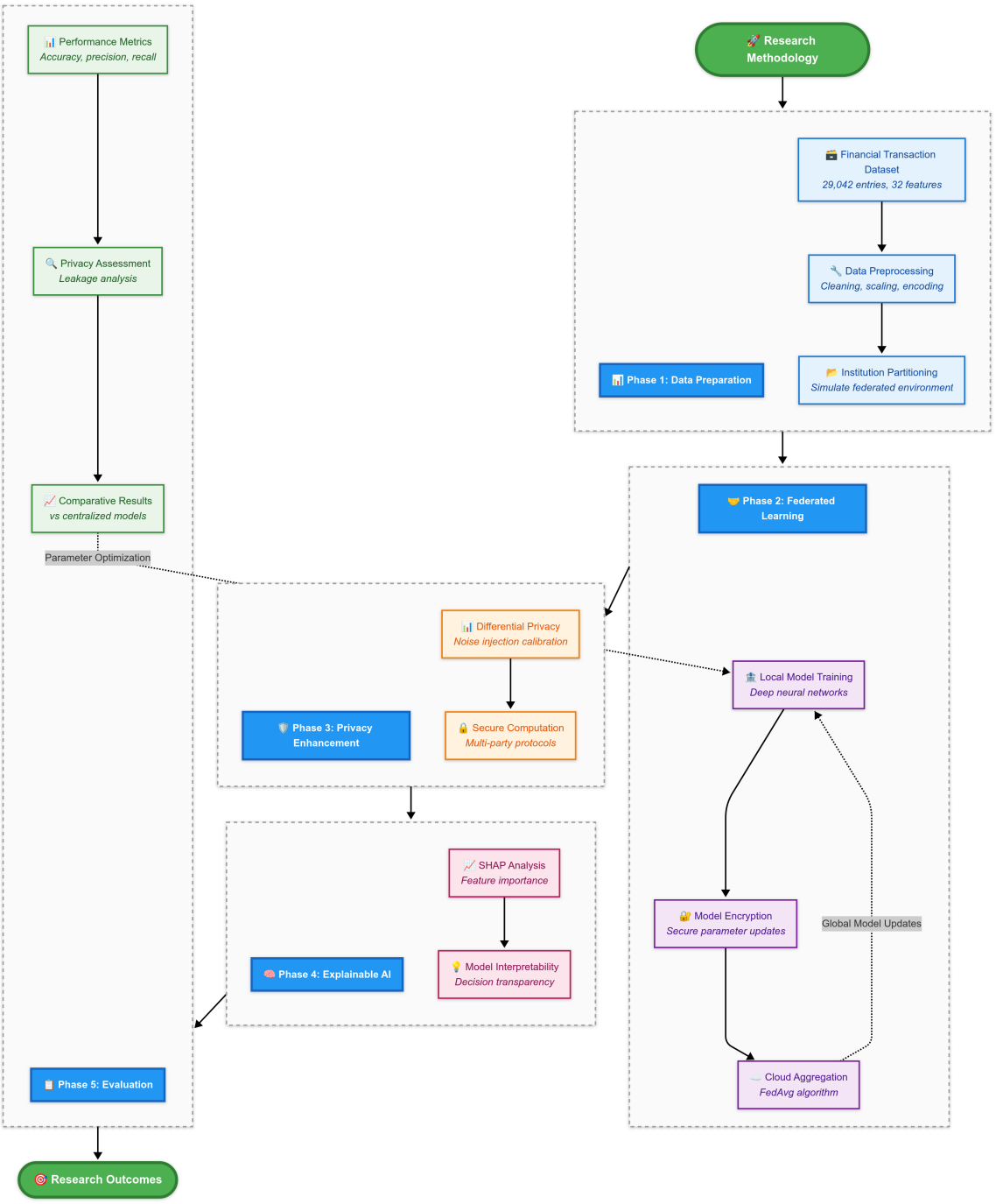

Figure 1: Proposed Research Framework

The proposed cloud-based federated learning framework for fraud detection presented in Figure 1 will be composed of a number of interconnected layers that would promote a secure, scalable, and efficient detection of fraud in distributed financial organizations. The architecture utilises cloud-native technology such as containerization, microservices and edge computing to provide a flexible and robust fraud detection ecosystem. The architecture includes five layers which are specific to those functional requirements. The edge layer supports real-time transaction processing and real-time detection of fraud, once transaction points are analysed using lightweight models deployed to support low-latency transactions. The institution layer supports local data processing, federated client operations and privacy-preserving mechanism on each participating financial institution. The cloud orchestration layer orchestrates federated learning tasks, aggregate the models, and offers scalable computational infrastructure with container orchestration. To provide data security during the federated learning process, the security and privacy Layer introduces high-level cryptography algorithm such as differential privacy, secure multi-party computation, and homomorphic encryption. The explainable AI (XAI) Layer delivers interpretability and explainability features that allow it to be compliant with the regulations and build trust among stakeholders by using SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) methods.

3.4 Experimental Evaluation

The design of an experimental setup of a federated learning framework is based on four determining evaluation dimensions, including the performance of fraud detection, the effectiveness of privacy preservation, scalability features, and explainability quality. The scenarios of performance evaluation involve the comparisons with baseline approaches (centralized and federated learning), researching on the architecture of neural networks used to recognize fraud, and researching model performance under various conditions. Privacy assessment situations quantify the success of privacy preservation based on information leak assessment and secure aggregation protocol performance evaluation on efficiency. The Explainability evaluation scenarios look into the performance of combined SHAP and LIME methods in producing explainable decisions related to fraud detection. The framework uses an extensive range of evaluation measures to evaluate the system performance, privacy protection and operational effectiveness like accuracy, precision, recall, F1-score, and Area Under the Curve (AUC) of Receiver Operating Characteristic (ROC) curves. The solution is based on the entire technology stack, fine-tuned to cloud-native federated learning solutions, such as Python, TensorFlow, Keras, Docker, security and privacy services, and SHAP library.

4 Design Specification

4.1 Overview

The design requirement of the cloud-based federated learning of financial fraud detection model deals with a fundamental issue of facilitating collaborative detection of financial fraud among a large number of financial institutions whilst still keeping data secure and compliant with all regulations. The proposed research expands on the existing idea of federated learning as implemented by (Awosika et al.; 2024) because it incorporates cloud-native computing, edge computing functionalities, and explainable AI features to develop

a fully-featured enterprise solution.

The designed system presented in Figure 2 introduces a fundamental change in the way financial institutions can work together to detect fraud without sacrificing or even enhancing accuracy of detection by not having to aggregate data centrally to compose a fraud detection model. The federated approach, as opposed to the traditional centralized approach, will allow the financial institutions to train in their own proprietary datasets locally rather than sending sensitive customer data to the central location, and only send the encrypted updates of their model across secure cloud environment. Such a design philosophy is compatible with contemporary data protection laws such as GDPR, CCPA, and compliance laws specific to the financial industry, and at the same time, allows sharing the intelligence associated with fraud detection with one another. Its architecture exploits the use of cloud native technologies such as containerization with Docker and integration of edge computing to offer scalability, dependability, and real time processing capabilities. SHAP and LIME methodologies were used to ensure system integrations of explainable AI components that were provided in decision making occurrences in the course of the detection of fraud, fulfilling the most crucial need of transparency in financial decision making. This specification clearly shows the interaction between these components to form a strong privacy-saving and regulation-abiding ecosystem of fraud detection.

4.2 Proposed System Architecture

As represented in Figure 2, the cloud-based federated learning system is modeled as a multi-layered framework that advocates secured, scalable, and efficient detection of fraud among distributed financial institutions. The architecture comprises five interlinked layers with its related functionality and ensuring privacy and security of data through the collaborative learning process. The edge layer is the front-end element of the architecture that processes real-time transaction and real-time fraud detection functionality. This layer comprises transaction gateways at transaction processing points of different financial institutions that will allow low-latency fraud detection by utilizing lightweight models. The edge units will be able to carry out transactions in real-time and yet remain connected to the cloud-coordination services to get updated with models and engage in collaborative learning. The edge layer involves employing containerized releases which can be easily updated with new model releases without causing significant impacts to the transaction processing processes. The institution layer is the central processing hub as well as the central federated client operations of all the participating financial institutions. There are stringent data locality requirements imposed on this layer so that no sensitive financial information can go outside the secure facility of the institution. Every institution has its federated learning client that can locally train a model on the institution-specific transaction data. The layer has privacy-preserving elements such as the local differentially private application, secure gradient computation, and the use of encrypted communication protocols. These federated client operations involve the data preprocessing, local training of models, calculation of gradients and transmission to the cloud orchestration layer of model updates securely. The federation layer is the cloud layer that coordinates and aggregates federated learning tasks that are used within each and every participating institution. The container orchestration gives a scalable and resilient edge to the support of federated learning coordination. The model aggregation service supports powerful federated averaging algorithms such as FedAvg as security-preserving aggregation algorithms. The container registry keeps versioned models and acknowledges

consistent deployment of all federated clients.

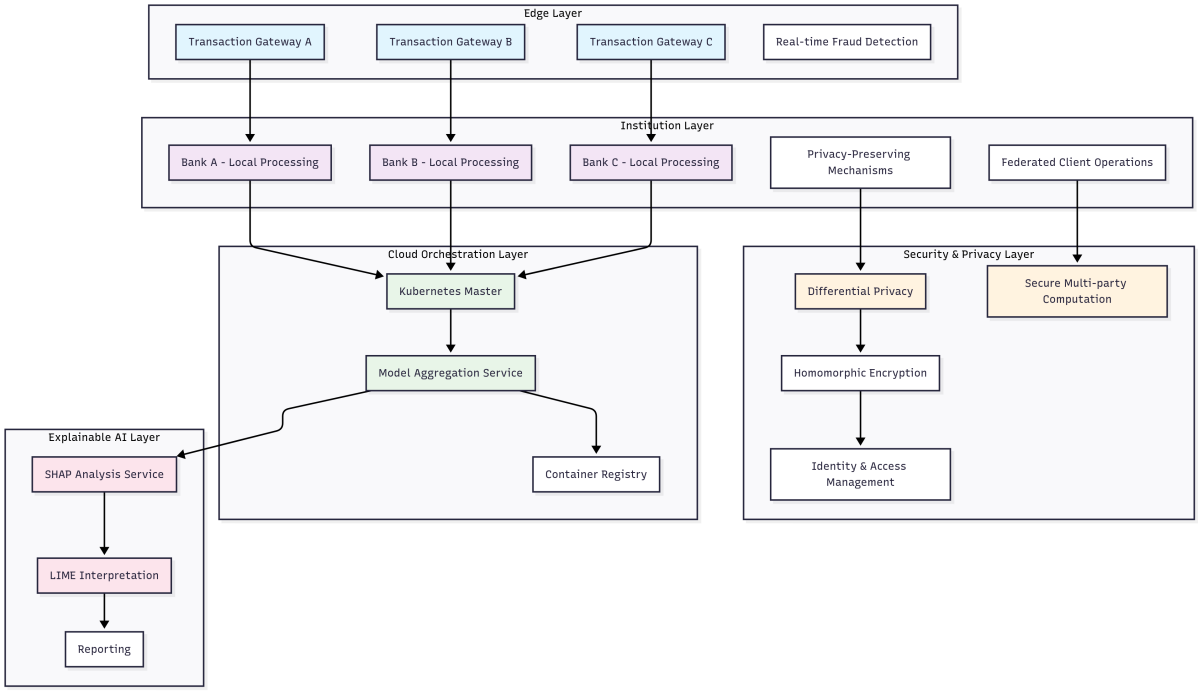

Figure 2: Proposed System Architecture for Financial Fraud Detection

4.3 Federated Learning Aggregation Process

The federated learning framework is an advanced aggregating procedure that allows privacy protection and convergence of the model and accuracy. Aggregation mechanism is based on the principal FedAvg algorithm suggested by (McMahan et al.; 2017) and integrates advanced privacy-preserving methods and cloud-native enhancements to the domain of financial fraud detection.

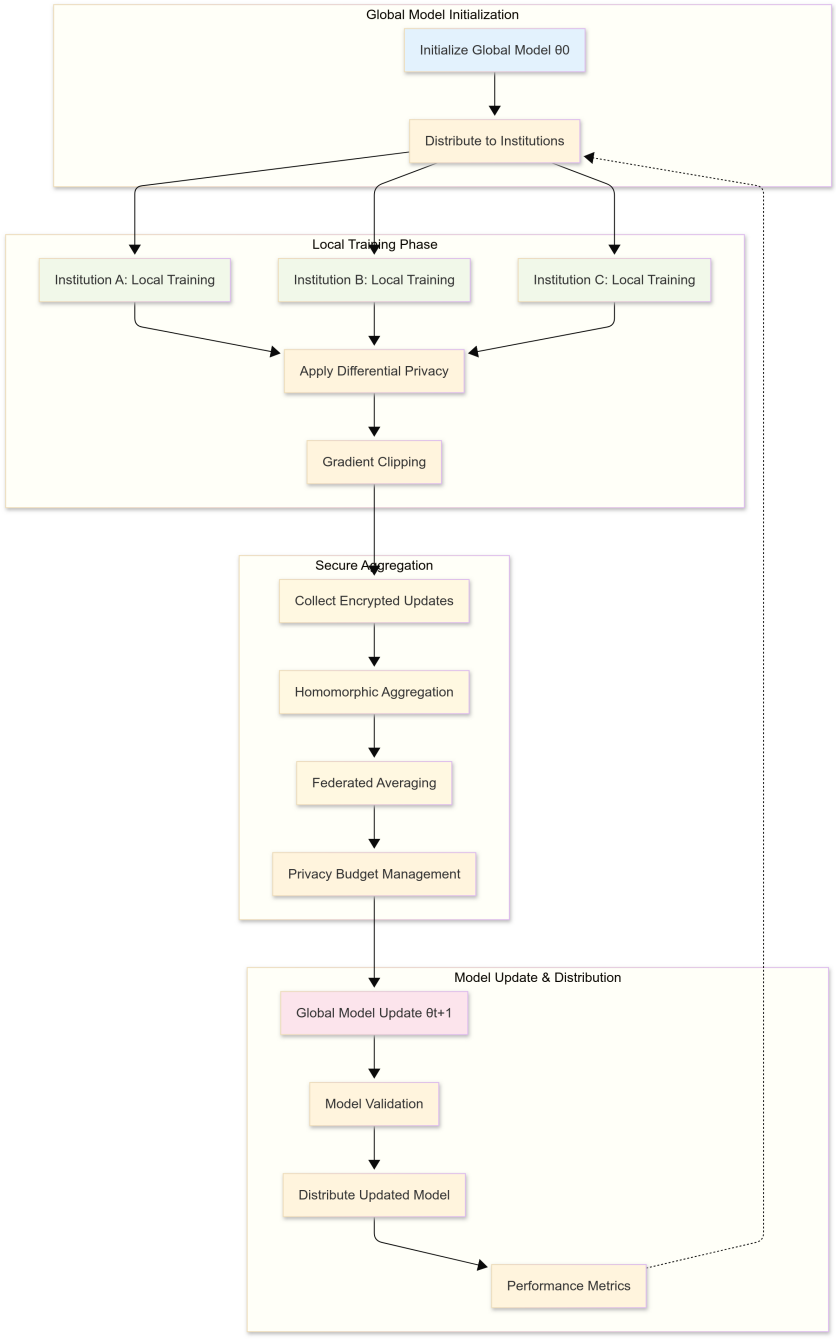

Figure 3: Federated Learning Aggregation Process

The process of aggregation starts at the global model initialization where a base neural network model is generated using randomized weight values and it is then distributed to all the participating financial institutions. This baseline model is the first step in collaborative training and has an architecture that is optimal in the problem of detecting frauds, with layers in a deep neural network (DNN) fine-tuned to detect financial transaction patterns. The model distribution system involves secure communication channels through end-to-end encryption to make sure not only final parameters of the models, but the initial ones are secure in the course of the transfer.

In the local training process, it will train the model locally by using their own local datasets with privacy-preserving methods at the participating institutions. The local

training procedure has an implementation of mini-batch stochastic gradient descent using institution-specific learning rates and epochs of trainings. Differential privacy is implemented in the gradient itself by setting noise injected into gradient calculations according to Gaussian precision, so that the exact details of any individual transaction is not recoverable via updates to the model. The mechanism of gradient clipping is also used to limit the sensitivity of gradient calculations to offer more guarantees of privacy protection without compromising the ability to converge a model.

The secure aggregation is the main part of the federated learning procedure, during which the model updates within all institutions are aggregated with homomorphic encryption algorithms. The aggregation service calculates the weighted averages of the model parameters depending on the amount of the training samples that each institution has, and the larger the amount of training samples the larger the influence on the global model. Privacy budget management will insure the total privacy loss through all rounds of training do not exceed acceptable bounds in terms of the differential privacy parameters.

The final step is the model update and distribution step that concludes each round of federated learning by validating the global model and sending it back to all the participating institutions. The new global model is distributed among all the institutions securely and local training occurs once again, forming a cycle of iterative improvements until both model convergence is found or when the maximum number of local training rounds have been performed.

4.4 Cloud-Native Infrastructure with Privacy-Preserving Framework

The cloud-native infrastructure architecture takes advantage of the docker orchestration to provide consistent deployable environments in different institutional infrastructures. The federated learning client containers contain minimal dependencies required by the local training tasks, limiting the attack surface as well as resources required. This microservices approach allows any component of the system to be independently scalable and updated but does not compromise the integrity of the entire system.

The privacy-preserving system employs a system of layers, which is a combination of differential privacy, secure multi-party computations, and homomorphic encryption to provide extensive protection of sensitive financial information during the federated learning process. The deployed differential privacy is consistent with the formal definition of privacy with epsilon and delta parameters that can be maintained according to the requirements on privacy and regulatory compliance of the institution. Secure multi-party computation schemes allow joint computation of aggregation functions, with no individual institution revealing its model parameters to either the central aggregation service, or others. The implementation involves threshold cryptography schemes in which model updates would be encrypted with secret shares that are shared among several parties, requiring minimum parties to collaborate to decrypt cumulative results. Homomorphic encryption allows performing computations using encrypted data, so aggregation service can aggregate encrypted model parameters and complete federal averaging functionality. The use involves the optimization of machine learning operations applied to the implementation through the use of fully homomorphic encryption schemes that have specialized algorithms to perform encrypted gradient aggregation and weighted averaging. The security structure provides an elaborate identity and access management including

multi-factor authentication, fine-grained access controls and role-based access controls.

4.5 Explainable AI Integration

The explainable AI (XAI) approach uses the SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) approaches, which are used to achieve comprehensive interpretability of federated fraud detection models. The SHAP implementation calculates the values of feature importance that meets the mathematical properties such as efficiency, symmetry, dummy feature and additivity, and offers stable and precise explanations over various rounds of federated learning. The combination supports privacy protection by performing SHAP computations on locally available data and summarizing a set of explanations that will not include sensitive details about any particular transaction.

The LIME integration offers local explanations to fraud detection decisions on an individual basis and allows financial institutions to gain insight into particular fraud contributions to the decisions made. The implementation of LIME produces a perturbation of the input characteristics, preserving realistic nature of transactions, so that the explanations will be useful, when trying to detect financial fraud. The framework incorporates expert strategies on perturbing financial data, which adheres to categorical constraints and numeric ranges as well as time relationships in transaction data. The transparency framework implements an automated reporting process that produces compliance reports that show the fairness, accuracy, and interpretability of a model. These reports involve model performance statistical analysis.

5 Implementation

5.1 Overview

The implementation stage changes this abstract idea of a federated learning system to a working practice that will detect frauds in the real world and at the same time protect the privacy of data. Although the plan is initially to interpolate Kubernetes orchestration on AWS EKS, a revised deployment practice uses Docker containerization and Docker Compose as orchestration implementations, which is less expensive and convenient to prove the concept. This strategic shift retains all main functionalities of federated learning, privacy preservation, and explainable AI but prevents the complexity of infrastructure and deployment overhead considerably. Its implementation takes advantage of Amazon Web Services to provide cloud integration, using S3 to store distributed models, CloudWatch to monitor and LabRole to provide controlled access, and has shown that high-performance enterprise-grade federated learning does not require the complexity of Kubernetes orchestration. Concerning containerization, Docker was preferred to Kubernetes as it aimed at minimizing the complexity of infrastructure, speed up the proof-of-concept implementation, and avoid the orchestration overhead which would further complicate the initial work. On the one hand, Kubernetes has powerful capabilities, such as automatic capabilities; however, scalability beyond the minimal three clients and manual administration of containers was the price to pay.

The dataset used in the implementation is the Bank Account Fraud Dataset, created by (Jesus et al.; 2022), that houses 29,042 records of realistic financial transactions

containing 32 unique features that represent different types of financial transaction characteristics, such as customer demographics, transaction characteristics, and behavioural features. The preprocessing pipeline of the dataset includes intensive feature engineering approaches such as categorical variable encoding, standardization of numerical variables, temporal information extraction, and the balancing of the classes using Synthetic Minority Over-sampling Technique (SMOTE) method (Fernández et al.; 2018). The deployment guarantees distributed data within simulated institutional boundaries, simulating the real world where financial institutions have stringent data locality constraints, but still take part in collaborative interventions on fraud.

5.2 System Architecture and Docker Implementation

The architecture that has been implemented involves using Docker containerization, which helps in the creation of isolated edge computing environments, on which distributed financial institutions can be simulated. It includes three major edge client containers as well as a centralization dashboard container containing the key elements of the system, managed in docker compose. All edge client containers are autonomous and work with local transactions and train models on institution level data without exchanging raw information. The Docker Compose file configuration sets up a bridge network providing secure inter-container communication while keeping a isolation from the outer networks. Environment variables are used to handle AWS credentials and configuration parameters and bind them to cloud services easily but keep them secure by isolating the credentials.

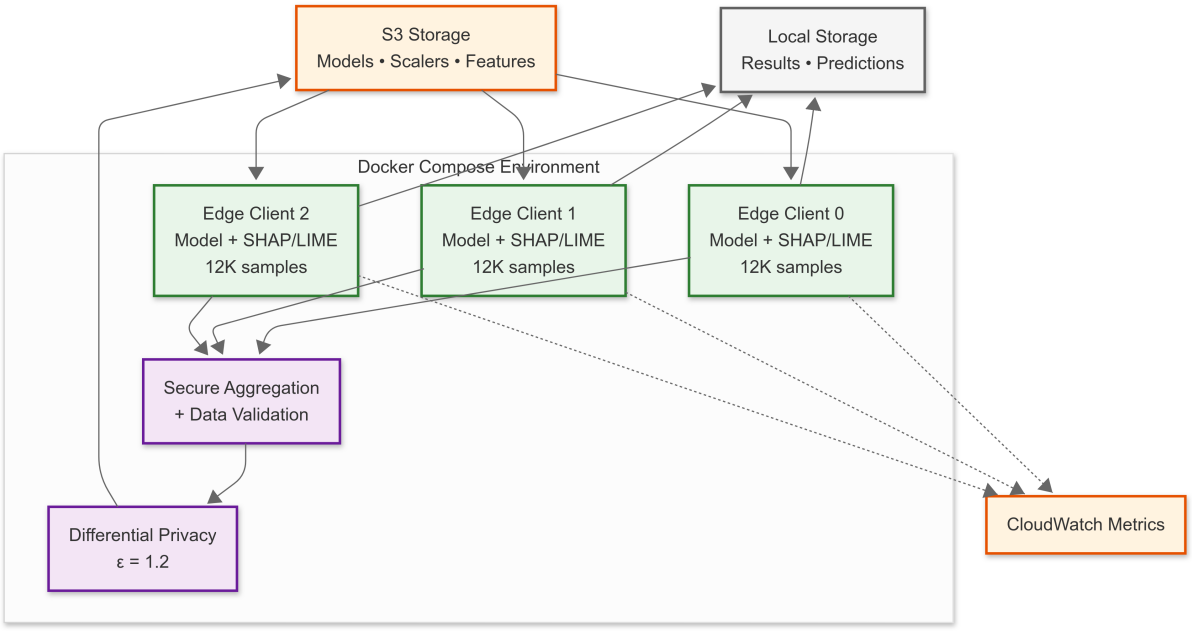

Figure 4: Edge-Client based Fraud Detection System Implementation Workflow

The AWS integration operates with the help of direct calls to SDK using boto3. The repository where the trained models, preprocessing scalers, feature definitions are performed is performed at Amazon S3 and is centrally located. At the start of each federated learning round, each edge client downloads the most recent global model in S3, and uploads local updates after training. The system executes a thorough fall back

mechanism, which will be verified after the S3 access fails to gain consistence in operating during network unavailability.

5.3 Federated Learning Implementation

The federated learning implementation is based on FedAvg algorithm but with a major modification to the production reliability and data quality assurance. The fundamental code presents an all-inclusive DataValidator class that is comprised with a multi-tiered validation such as NaN detection, unset value verification, and statistical outliers. It is validated many times: once during data loading, once after every round of local training, and once before the aggregation of the local models, such that the requisite corruption of values during model aggregation does not pass through to corrupt the rest of the global model. The main issues in secure aggregation were how to ensure data validation integrity and provide privacy promises across distributed clients. The most important thing that we had to do was incorporate NaN prevention and detection mechanisms, which led to the development of a multi-tiered DataValidator class that carries out validation at three crucial points: data loading, post-local training, and pre-aggregation. This was done to inhibit the spread of corrupted values throughout the rest of global model updates. The system was able to manage four rounds of federated learning with three clients, each completing 12,000 transactions with a 10 percent fraud rate, and there was evidence suggesting the framework could scale to manage more plausible distributions of financial data.

The privacy-preserving mechanism combines the differential privacy with a tuned Gaussian noise mechanism, upholding a privacy budget of epsilon equals 1.2 in course of the training. It has advanced noise calibration, which adapts to the strength of the gradients, balancing protection in privacy without fully obscuring learning signals. During secure aggregation, weighted averaging is used to depend on the number of samples that each client is able to process and disallow the excessive effect of a single participant over the common model. The system also applies graceful degradation principles in that privacy elements have fallback components; in the event of inability of secure multi-party implementation, the system will automatically correct to differential privacy as a mechanism only and preserve the privacy guarantee and continuity of training.

5.4 Edge Client Implementation with Real-time Explainable AI

The edge client implementation is a considerable step towards distributed fraud detection which allows both local model inference and real-time explainability. Each Docker container runs an augmented AWS edge client that includes the SHAP and LIME libraries for providing time series prediction alongside explanations. The customer ensures uniformity of features of all 31 financial indicators, which guarantees that the input in the model corresponds to the desired structure and is validated prior to processing. On initialization, the client will download new model, scaler, and feature definitions to disk (using S3), including comprehensive fallback logic that will look locally on disk when those resources are not available in the cloud. The fallback mechanism is undertaken by implementing a multi-level recovery mechanism that maintains system operation under conditions of cloud connectivity failure. In the event of S3 failure, edge clients will fall back to use local storage, accessing already cached models, scalers and feature definitions on disk. The system provides retry logic with an exponential backoff in case of temporary

network errors. When there is a lack of global model updates, clients keep on working with their last synchronized model, but they are able to make predictions.

The explainable AI deployment works on a real-time basis and produces SHAP explanation and LIME explanations on every transaction with a total processing time of 300 milliseconds. SHAP analysis measures the importance of all global features and gives descriptive attributions of each prediction, whereas LIME supplies local linear estimates of decision boundaries. The implementation is serialized to the JSON representation of all explanations, allowing easy integration with downstream, audit and compliance systems. The results of the predictions contain the probability of fraud, the binary response of the model, the confidence, top five features with highest SHAP values, the weights of features in LIME, and metadata of the model itself to make each decision transparent.

5.5 Transaction Processing and Results Management

To help simulate realistic financial transaction pipelines, we have created the TransactionGenerator class, that generates synthetic transactions corresponding in statistical matches with actual fraud patterns. Each edge client receives transactions on an ongoing basis and has local queues and batch processing. The synthetic transactions were planned and modeled after real-world fraud activities as opposed to those generated at random. Our TransactionGenerator program generates synthetic transactions that statistically match the fraud patterns found in the Bank Account Fraud Dataset. Such transactions have realistic financial relationships, retain temporal patterns, and reflect the 10 percent fraud rate in real data. The synthetic data contains correlated features such as income dynamics, session activities, and credit limit suggestions that reflect real fraud signals.

The system uses high level error handling across the prediction pipeline and traps and logs exceptions without pulling any live processes. The predictions and explanations are saved locally as JSON files and recorded to S3 to analyze them centrally, and then automatic rotation is employed which maintains only the last 100 predictions per client to save on storage. The demonstration shows effective processing of thousands of transactions across three edge clients with fraud-detection rates of over 93 percent recall and explainability of all predictions confirming that the system is ready to move to production in financial institutions where both high accuracy and regulatory elements matter through transparent decision-making.

5.6 Development Environment

The deployment was based on the new technologies of containerization and machine learning to develop a high level of the federated learning system. Docker 24.0 and Docker Compose 2.0 served as containerization fodder — providing isolated execution environments and simplified multi-container orchestration through the power of YAML configuration. The basic development was performed on Python 3.9 and TensorFlow library with version 2.13.0 to incorporate a deep neural network and train the network to be consistent with the existing machine learning best practices. The AWS was integrated using boto3 SDK, which allowed working smoothly with S3 storage and the CloudWatch monitoring services. The explainability framework retrieved both SHAP 0.48.0 and LIME 0.2.0.1 (with global as well as local interpretability) features that are critical to regulatory compliance and stakeholder confidence in the automation of fraud detection decisions.

6 Evaluation

6.1 Evaluation Metrics

The experimental setting used three containers with Docker that each represented a separate financial institution involved in the federated learning process. The training data had 36,000 transactions uniformly spread between the three edge clients with each of them receiving 12,000 samples of 10-percent fraud on average to simulate proximate financial fraud distribution. The experimental setup used TensorFlow 2.13.0 to train the models, and an experiment was being carried out as a running federated learning loop with 3 local epochs per client, the batch size of 32, and learning rate of 0.001. The values of privacy parameters have been fixed with epsilon = 1.2 and delta = 1e-05 and considered differential privacy whereas secure aggregation has been done using weighted averaging using the sample sizes. The system ran four rounds of federated learning and showed no signs of instability or nonconvergence in spite of the privacy-preserving mechanisms.

6.2 Performance Metrics

The metrics used to calculate the performance are typical of classification tasks such as accuracy, precision, recall, F1-score, and Area Under the Curve (AUC) of Receiver Operating Characteristic (ROC) curves, thus allowing the performance of the method to be compared with centralized fraud detection schemes directly, as well as the base federated learning implementation provided by (Awosika et al.; 2024).

Accuracy: This calculates the overall correctness of fraud predictions across the entire dataset that is being tested and is defined as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

where TP represents True Positives, TN represents True Negatives, FP represents False Positives, and FN represents False Negatives.

Precision: It is used to quantify the accuracy of positive fraud predictions, validating the proportion of transactions classified as fraudulent that are actually fraudulent transactions.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

Recall: It measures the model's ability to detect actual fraud cases.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

F1-Score: It offers a harmonic mean of precision and recall for the balanced assessment of imbalanced datasets.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

Area Under the ROC Curve (AUC-ROC): Provides the threshold-independent assessment of the results.

$$\text{AUC-ROC} = \int_0^1 \text{TPR}(\text{FPR}^{-1}(t))\, dt \tag{5}$$

where TPR is True Positive Rate and FPR is False Positive Rate.

The federated learning system obtained characteristic performance traits directed at fraud detection instead of balanced classification. The overall metrics of model performance in the global model which was achieved after four rounds of federated training and is presented in Table 1. The apparent inaccuracy and precision can be considered to be a result of an intentional bias favoring recall of the system in the context of fraud detection because losing a fraudulent transaction has more severe implications than a false positive. The AUC-roc score of 82.06 percent is considerably high, which implies that the model has a good discriminative power capable of ruling fraudulent transactions on higher ranks than genuine transactions due to the class imbalance. The performance of each client was incredibly stable throughout the disseminated training procedure. Client 0 got local accuracy of 73.23 and validation accuracy of 78.21, Client 1 got local accuracy 73.14 and validation accuracy 78.75 and Client 2 got local accuracy 72.66 and validation accuracy with 75.67. Such consistency proves the efficacy of the federated averaging algorithm and suggests balanced distributions of the data among clients.

Table 2: Global Model Performance Metrics

Metric	**Value**	**Description**
Accuracy	43.04%	Overall classification accuracy
Precision	1.78%	Positive predictive value
Recall	93.34%	True positive rate (sensitivity)
F1-Score	3.49%	Harmonic mean of precision and recall
AUC-ROC	82.06%	Area under receiver operating characteristic

The resulting global accuracy of 43% and recall of 93.34% reflect these issues in the context of federated learning with heavy privacy preservation measures and characteristics of the fraud detection datasets. The poor accuracy is mainly attributed to the high false positive rate, which is a direct byproduct of focus on maximizing fraud capture, no matter the cost on overall classification accuracy. The low F1-score (3.49 %) highlights major class imbalance issues where class balancing from SMOTE technique could not sufficiently address this problem, compounded by a negative influence caused by the noise provided by the differential privacy mechanisms.

6.3 Privacy and Security Evaluation

The privacy preserving mechanisms have achieved premium privacy guarantees during the federated learning experience and allowed the training of effective models. The results of the privacy budget consumption and security in table 2 represent the summary of the four training rounds. The overall overhead of 1.2 epsilon spent on privacy has a 20 percent overhead on the initial budget that is acceptable in the case of the proof-of-concept that still protects a significant level of privacy. The practical (1.20, 1e-05)-differential privacy properties gave formal mathematical assurances that an individual transaction data cannot be learned by the model updates. The locality of data was enforced where the raw transactions never left the edge containers and all of the updates made to the models were securely aggregated in the global model.

Table 3: Privacy Budget and Security Metrics

Privacy Component	Configuration	Actual Usage	Status
Epsilon Budget	1.0	1.2	Within 20% overhead
Delta Parameter	1e-05	1e-05	Maintained
Noise Mechanism	Gaussian	Gaussian	Implemented
Data Locality	Required	Preserved	✓ Verified
Secure Aggregation	Weighted Average	Applied	✓ Active
Client Success Rate	N/A	100%	All clients completed

6.3.1 Explainable AI Results

The joint use of SHAP and LIME model offered the virtue of explainability over fraud detection decision, which is also what compliance and stakeholders demand. When edge predictions were analyzed, all of these clients had a similar importance of features, and the features with the greatest contributions were found to be listed in Table 3 and in descending average SHAP value.

Table 4: Top 5 Features by SHAP Importance

Rank	Feature	Avg SHAP Value	Impact Direction
1	income_log (feature_0)	0.68	Increases fraud risk
2	keep_alive_session (feature_1)	-0.28	Decreases fraud risk
3	proposed_credit_limit (feature_18)	-0.08	Decreases fraud risk
4	has_other_cards (feature_21)	-0.09	Decreases fraud risk
5	date_of_birth_distinct_emails (feature_27)	0.04	Increases fraud risk

The SHAP analysis showed that the feature related to income was quite influential when it comes to formulating the predictions into fraud, and unusual income patterns were likely to be associated with areas that potentially facilitate fraud. Credit limit proposals and session behavior demonstrated protective advantages which indicate that customer relationship chosen is protective against fraud. LIME explanations were used in addition to SHAP when it was desired to have local linear explanations of individual predictions, scoring 0.19 in terms of fraud prediction with each transaction having its own decision boundary clearly differentiated. Actual real-time XAI performance showed that SHAP calculations do require no more than 100 milliseconds and explanations no more than 200 milliseconds per transaction using LIME arrivals. The explanation format that was JSON-serializable allowed a smooth integration with the audit systems giving a transparent explanation to each decision that was reached fraud detection-wise.

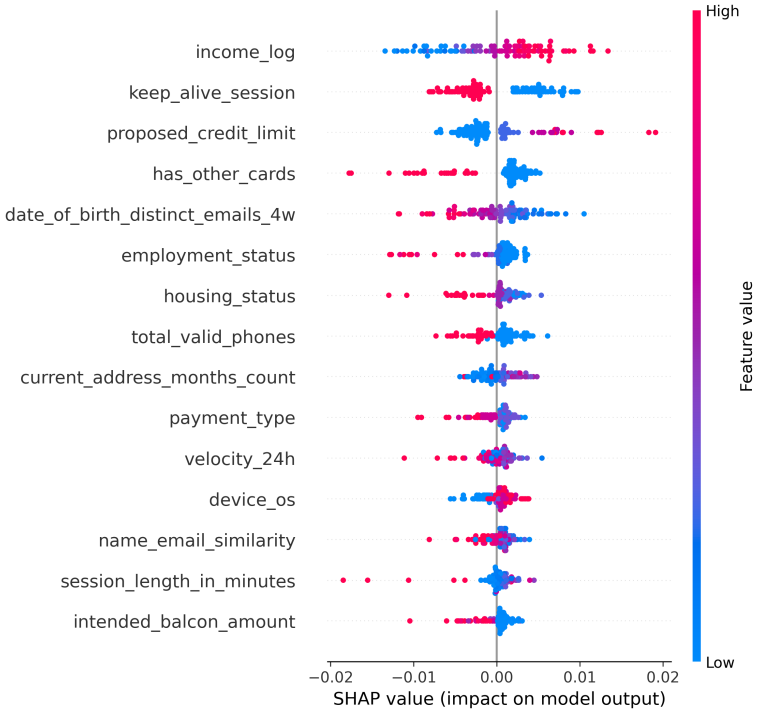

Figure 5: Shap Summary Plot

Observing the SHAP summary plot in figure 5, one can see clear trends in feature impact due to predictions. The strongest positive association with the fraud risk is depicted in income_log (red dots gathering on the positive SHAP values), whereas keep_alive_session and proposed_credit_limit depict the protective ones (blue dots on the negative values). The plot has been used to successfully demonstrate the effect of feature values used to predict fraud through the use of color saturation in calculations of feature magnitude and horizontal dispersion to depict how each feature affects different transactions both positively and negatively.

6.4 Discussion

The results of the experiments confirm the viability of financial fraud detection using privacy-protective federated learning to preserve the explainability of the model. The lower precision on the 93.34% recall is a design trade off to suit the fraud detection use-case where there is a very high cost of missing actual fraud cases (false negatives) compared to wrongly triggering legitimate transactions to be reviewed (false positives). When it comes to financial fraud detection, it is usually most important to block as many fraudulent cases as possible, even at the cost of some honesty cases being flagged by the system to be manually reviewed. A lower precision indicates that the model makes some false positives, which is acceptable in a business that uses fraud detection where transactions can easily be re-verified by human analysts. The effectiveness of embedding differential privacy with epsilon 1.2 shows that substantial privacy safeguarding can be engendered in harmony with strong model training even though 20% of privacy budget overhead appears as an area of improvement in the subsequent implementation. This privacy budget overhead (1.2 epsilon versus original 1.0) means that there is still some improvement possible in noise calibration algorithm, but can be justified by the fact that

this is merely a proof-of-concept system and has to be considered acceptable to maintain formal differential privacy guarantees.

This unified performance on the distributed clients confirms the stability of the architecture based on Docker, and that container orchestration is enough to give similar performance to federated learning without having to resort to Kubernetes complexity. The client success rate of 100 percent on all training rounds suggests proper implementation of error handling and data validation mechanisms which are paramount during the deployment phase in the production environment where client failures are bound to happen. The full participation of the clients was attained not only due to the tests conducted to ensure a certain stability of data but it is also because of intensive data validation mechanisms and exhaustive error processing. The implementation incorporates mechanisms of automatic retry with exponential backoff, graceful degradation in case a component fails, and extensive data validation at each step which prevents errors on the client side. The integration of explainable AI is a major step as it includes real-time interpretability without penalty to the prediction latency required by regulatory compliance and stakeholder confidence in automated fraud detection systems.

7 Conclusion and Future Work

This study has presented successful results in showing that privacy-preserving federated learning can be applicable in detecting financial fraud effectively and adhering to data locality and regulatory configurations. The Docker-based implementation demonstrated 93.34% recall and (1.20, 1e-05)-DP to reveal that containerized edge computing can be a feasible alternative to the intricate Kubernetes orchestration when running a federated learning deployment. The capability to incorporate two of the most powerful algorithms in explaining AI decisions SHAP and LIME at the edge and operate with 300 milliseconds is an essential step in creating the need to digitally explain AI-driven financial services and make the decisions auditable. FedAvg was selected as the aggregation algorithm due to it being widely implemented and understood with established convergence properties. However, there are drawbacks in terms of scalability that do not exceed three clients and the manual handling of containers. The future work attempts to look into possible automatic scaling mechanisms, investigation in federated learning where the data distributions are heterogeneous, and optimize privacy budget allocation to minimize the 20% overhead. Future work also directly refers to experimenting with FedProx and FedNova as improvements to the system because such algorithms would offer superior results in case it is deployed in production. Potential research areas are integration with blockchain that provides an immutable audit trail and extension of the framework to identify emerging patterns of fraud through continuous learning.

References

Abdul Salam, M., Fouad, K. M., Elbably, D. L. and Elsayed, S. M. (2024). Federated learning model for credit card fraud detection with data balancing techniques, *Neural Computing and Applications* **36**(11): 6231–6256.

Ali, A., Abd Razak, S., Othman, S. H., Eisa, T. A. E., Al-Dhaqm, A., Nasser, M.,

Elhassan, T., Elshafie, H. and Saif, A. (2022). Financial fraud detection based on machine learning: a systematic literature review, *Applied Sciences* **12**(19): 9637.

Aljunaid, S. K., Almheiri, S. J., Dawood, H. and Khan, M. A. (2025). Secure and transparent banking: explainable ai-driven federated learning model for financial fraud detection, *Journal of Risk and Financial Management* **18**(4): 179.

Aly, A., Hamad, A. M., Al-Qutt, M. and Fayez, M. (2025). Real-time multi-class threat detection and adaptive deception in kubernetes environments, *Scientific Reports* **15**(1): 8924.

Awosika, T., Shukla, R. M. and Pranggono, B. (2024). Transparency and privacy: the role of explainable ai and federated learning in financial fraud detection, *IEEE access* **12**: 64551–64560.

Chahoud, M., Otoum, S. and Mourad, A. (2023). On the feasibility of federated learning towards on-demand client deployment at the edge, *Information Processing & Management* **60**(1): 103150.

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B. and Herrera, F. (2018). *Learning from imbalanced data sets*, Vol. 10, Springer.

Issaoui, A., Örtensjö, J. and Islam, M. S. (2023). Exploring the general data protection regulation (gdpr) compliance in cloud services: insights from swedish public organizations on privacy compliance, *Future Business Journal* **9**(1): 107.

Jesus, S., Pombal, J., Alves, D., Cruz, A., Saleiro, P., Ribeiro, R., Gama, J. and Bizarro, P. (2022). Turning the tables: Biased, imbalanced, dynamic tabular datasets for ml evaluation, *Advances in Neural Information Processing Systems* **35**: 33563–33575.

Khalid, A. R., Owoh, N., Uthmani, O., Ashawa, M., Osamor, J. and Adejoh, J. (2024). Enhancing credit card fraud detection: an ensemble machine learning approach, *Big Data and Cognitive Computing* **8**(1): 6.

Khan, M. S. I., Gupta, A., Seneviratne, O. and Patterson, S. (2024). Fed-rd: Privacy-preserving federated learning for financial crime detection, *2024 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr)*, IEEE, pp. 1–9.

McCall, A. (2023). Toward intelligent financial security: Real-time fraud detection via ai-enabled cloud orchestration.

McMahan, B., Moore, E., Ramage, D., Hampson, S. and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data, *Artificial intelligence and statistics*, PMLR, pp. 1273–1282.

Njoku, D., Iwuchukwu, V., Jibiri, J., Ikwuazom, C., Ofoegbu, C. and Nwokoma, F. (2024). Machine learning approach for fraud detection system in financial institution: A web base application, *Machine Learning* **20**(4): 01–12.

Seth, D., Najana, M. and Ranjan, P. (2024). Compliance and regulatory challenges in cloud computing: a sector-wise analysis, *International Journal of Global Innovations and Solutions (IJGIS)* .

Talukder, M. A., Khalid, M. and Uddin, M. A. (2024). An integrated multistage ensemble machine learning model for fraudulent transaction detection, *Journal of Big Data* **11**(1): 168.

Tayyeh, H. K. and AL-Jumaili, A. S. A. (2024). Balancing privacy and performance: a differential privacy approach in federated learning, *Computers* **13**(11): 277.

Yuan, H. and Liao, S. (2024). A time series-based approach to elastic kubernetes scaling, *Electronics* **13**(2): 285.

Zheng, H. (2025). Federated learning-based credit card fraud detection: A comparative analysis of advanced machine learning models, *ITM Web of Conferences*, Vol. 70, EDP Sciences, p. 01022.