National College of Ireland

Securing Financial Sector in the Cloud: A Multi-Cloud Approach to Fraud Detection Using Secure Multi-Party Computation

MSc Research Project
Programme Name

Tanmaya Kumar Dixit
Student ID: x23116668

School of Computing
National College of Ireland

Supervisor: Shaguna Gupta

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Tanmaya Kumar Dixit
Student ID:	x23116668
Programme:	Msc in Cloud Computing **Year:** 2023-2024
Module:	Msc Research Project
Supervisor:	Shaguna Gupta
Submission Due Date:	14/08/2024
Project Title:	Securing Financial Sector in the Cloud: A Multi-Cloud Approach to Fraud Detection Using Secure Multi-Party Computation
Word Count: 10120	**Page Count** 25

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: Tanmaya Kumar Dixit

Date: 14/08/2024

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):	

Securing Financial Sector in the Cloud: A Multi-Cloud Approach to Fraud Detection Using Secure Multi-Party Computation

Tanmaya Kumar Dixit
x23116668

Abstract

In the world of digitalization, banks and other financial institution face huge difficulty in detecting fraudulent transactions such as money laundering, credit card fraud and many other financial frauds. Challenge arises with traditional fraud detection systems within an individual organization and the reason is limited visibility of the data each organization is analysing the data pattern in one isolated which means only the transaction involving its customers which restricts them from detecting any complex fraud patterns that extend over multiple entities. Improving fraud detection in the financial sector especially in this digital world where everything is going on cloud is very crucial due to the rise in online transactions and sophisticated fraud schemes. This research proposes a multi-cloud framework which integrates Secure Multi party computation (SMPC), Homomorphic encryption (HE) and machine learning algorithm specifically Decision tree, random forest and Logistic regression, to address these challenges. The proposed framework will allow different financial institutes to share and analyse data securely without compromising data privacy. SMPC allows multiple party to compute function on encrypted data while keeping the data of each institution involve safe and secure, HE will enhance the security by allowing the computation on encrypted data and decision tree will be used to identify the fraudulent pattern. This combines approach will help In improving fraud detection while maintaining data privacy providing a secure and scalable solution for financial sector.

1 Introduction

1.1 Background & Motivation

Cloud computing technologies have become the latest trendy technologies adopted by the financial sector in the recent past because of the huge benefits they offer in terms of efficiency and connectivity. The growth of the financial sector on digital platforms has made it more vulnerable to complex fraud schemes. Traditional methods of detecting fraud, which worked well in less connected systems, are no longer sufficient to deal with the sophisticated and large-scale frauds we see today. As the financial transactions has increasingly moved online, the complexity and frequency of fraud detection has also increased. Fraudsters attack several organizations at once since current data protection models are poorly integrated. Some of the financial frauds like money laundering, credit card fraud, insurance fraud, etc can occur in various ways within any financial institute, for instance, money laundering makes use of layering to conceal the source of a certain amount of cash while credit card fraud makes use of other people's card details for purchases with intent to defraud. Insurance fraud may entail faking an incident, or exaggerating an occurrence that qualifies for an insurance

claim. Due to dispersed structure of data security, it is nearly impossible for each institution to identify these multiple fraud schemes independently.

In this era of digitalization protecting data while effectively identifying fraudulent activities has become more crucial than ever. As the online financial transaction grows conventional method of detecting fraud is now becoming inadequate. *Sangers et al. (2019)* in his research has highlighted a new method called Secure Multi-party Computation (SMPC), which analysis the transactional data without compromising any private information. With the help of the approach financial institutions were able to identify fraud in secure manner Similarly, *Myalil et al. (2021)* made use of homomorphic encryption to secure the sensitive information during analysis. With this encryption technique analysis was done without decrypting the data. It is proposed that these papers support the establishment of an improved approach to identify fraudulent cases as well as to maintain the confidentiality of the information. Many researchers are employing SMPC and homomorphic encryption in formulating competent tools for safeguarding financial data in this digital financial environment.

However, guaranteeing that each participant would not get access to other participants' data and meeting the strict privacy requirements are the main issues of such an approach. It is important to counter these issues in order to preserve the credibility and reliability among the organization that opt for the collaborative process.

1.2 Problem statement

Increase in complex and large-scale fraud due to rise in online transaction financial sector is facing huge challenges. Institutes make use of traditional methods in which they analyze data within their organization which is not a sufficient approach for detecting different types of frauds. The main challenges is limited visibility as banks only monitor their own transactions, concerns related to data privacy while sharing information and inefficient detection method that cannot handle large volumes of data effectively. To tackle these challenges, the involvement of multiple financial entities working in collaboration will be necessary in the process of identifying fraudulent transactions. Based on the above backgrounds, this research aims to develop a multi cloud framework that incorporates secure multi-party computation, homomorphic encryption and decision tree model. This solution will mean that the different financial parties that are involved will be able to share and analyze the data to detect frauds without any of the parties being aware about the data of the other party.

1.3 Research question

- How Secure Multi-Party Computation can be implemented in a multi-cloud environment to create a reliable and collaborative fraud detection system that maintains data privacy and detects various financial frauds?

1.4 Problem solution

This research proposes a framework that will make use of Homomorphic Encryption (HE) and Secure Multi-Party Computation (SMPC) for secure data sharing and collaborative fraud detection. The framework enables encrypted data stored at different cloud platforms of the financial parties engaged in this process to be safely merged and analysed.

With the usage of SMPC, the system allows computations to be performed directly on encrypted data which ensures privacy and confidentiality of the underlying data and securing it from any security any breaches. This step is very crucial for maintaining trust and legal compliances especially under strict data protection regulation rules such as GDPR.

For detecting fraudulent pattern, we will be building and training a machine learning model. This model will be designed and trained to especially operate on encrypted data ensuring that data decryption is not necessary. Model training will be done on a designated cloud platform, optimizing the use of computational resources while ensuring that all operations adhere to privacy-preserving

protocols. This ensures that the framework not only detects fraud effectively but also enhances data security and maintains compliance with data privacy laws throughout the prediction process.

Objectives

- **Secure multi cloud framework will be developed to detect fraud**: Designing and implementing a secure framework that will allow encrypted data from various financial institutions to be safely accessed and analyzed across multiple cloud platfrom. This will enable organizations to share insights without compromising their data or customer privacy.
- **For maintain data privacy homomorphic encryption will be implemented**: HE will be deployed to make sure that data remains encrypted during transit and analysis. This encryption methods allows complex computations on encrypted data for fraud detection.
- **Implementation of SMPC on encrypted data for collaborative analysis:** SMPC will be used to apply secure and collaborative analysis on data. This will enable the encrypted data that may be stored in different cloud environments to be utilized in fraud analysis while not disclosing the raw data. SMPC also make sure that only the required computation is done and no other information which is sensitive is disclosed.
- **Model Training**: Decision tree, random Forest and Logistic Regression model will be build and trained to operate directly on encrypted data, ensuring data privacy and leveraging computational resources efficiently on a designated cloud platform. This method ensures that model outputs remain secure until properly decrypted by authorized parties involved in the process.

1.5 Structure of document

Introduction- This section discusses about the motivation of research, which relies on advanced fraud detection mechanisms that are required to compete with rapidly changing digital financial environment. It will show the inability of the current fraud detection solutions to address the issues and introduce a collaborative, multi-cloud approach of SMPC, which can help to resolve the mentioned issues.

Literature Review- literature review will involve a synopsis of the research carried out in cloud computing, financial fraud detection, SMPC, HE, and ML algorithms. The methodology section will provide a critical review of the strengths and limitations of the current methods as well as identify research gaps that the proposed approach will target.

Research Methodology and Specifications- Provides an overview of our models methodology, tools, evaluation techniques, and ethical considerations.

2 Related Work

The whole evaluation of this review will focus on the understanding of how SMPC, HE and Decision Tree when linked together ensure that data privacy is maintained and is highly effective in detecting fraudulent activities without making the data less useful which means that while data is encrypted, and privacy preserved its utility is not compromised in detecting fraud. The detailed research allows us to identify the issues with the existing methods and highlight the requirement of the new approach that involves keeping privacy and fraud detection working well together. This review will clear the path for a robust discussion on how cryptography, cloud computing, and finances have influenced each other. Initially we will review prior works and methods to design a multi-cloud fraud detection framework. This platform will ensure that all security, efficiency, and privacy factors are met in the digital finance sector.

2.1 Secure Multi-party Computation (SMPC) in Fraud Detection

A study by *Sangers et.al. (2019)* discusses about the new way to find fraud in the financial world without revealing any private information and for doing so they implemented Secure Multiparty PageRank algorithm for collaborative fraud detection. Methodology section describes the secure multiparty computation framework for the PageRank algorithm which is traditionally used by Google for ranking web searches. For detecting fraudulent activities the algorithm has been adopted by evaluating transaction graphs from multiple organizations while keeping there data secure and confidential. For keeping data secure and confidential the researchers have implemented additively homomorphic encryption, in which when we add two ciphertexts together they give the same result as encrypting the sum of the two plaintexts. The main problem addressed in the paper is detection of fraudulent activities across financial networks without breaching any privacy regulations. However it shows some limitations which is the computational and communication complexity required by SMPC framework. The researchers have suggested some ideas to make the methods easy such as for every new iteration of the graph parties need not to download the encrypted data and using fully homomorphic encryption to reduce the communication complexity.
In another study, *Zhao et.al. (2019)* and their team extensively worked on SMPC, talking about its principles and how they are applicable n various domains including prevention of fraudulent activities in financial sector. The Methodology section of the paper covers the foundational concepts of SMPC in detail, security requirements and techniques for developing SMPC protocols that are applicable on cloud based real world applications. The main problem addressed in the paper is about difficulty faced while performing calculations on private data in distributed systems without compromising privacy. In the era of cloud and mobile computing where data is mostly processed outside the data owner's premises SMPC offers better solution for preserving the integrity of users data and making it secure during analysis. The study addresses the limitation faced by SMPC protocol which was scalability issue and also the challenges they faced while applying SMPC in real world applications such as complex protocol design and infrastructure requirement that supports secure and efficient computation.
The paper *Ramya et.al. (2024)* has addressed the limitations which was mentioned in *Zhao et.al. (2019)* .The authors implemented the ABY2.0 protocol for SMPC which provides the developers the tools for building applications. The methodology involves optimizing memory usage, reducing execution time with third-party Helper node. The problem they tried to solve was related to scalability while implementing SMPC specially for those tasks that are memory and computationally intensive. The study has not only addressed previous limitation but also opens up new possibilities for the implementation of SMPC in data intensive tasks and paving way for real-world applications or sectors that require data privacy such as finance and healthcare

In financial applications, this research by *Byrd, D. and Polychroniadou, A. (2020)* uses differentially private secure MPC in federated learning to enable institutions to train models together while protecting their data. The authors use differential privacy and MPC to allow for private collaborative learning, training of machine learning models on data that has been shared while maintaining privacy. The principal issue discussed is the conflict of interest between data confidentiality and model performance in federated learning settings. But the study also acknowledges the challenge of providing privacy while at the same time maintaining model accuracy and potential loss of precision from models learnt under privacy constraints.

In this paper *Alghamdi, W. et al. (2023),* author propose the application of SMPC for facilitation of the collaborative analysis of big data while preserving the privacy of the information. In the proposed system, multiple parties can process their integrated data safely with cryptographic methods,

including Yao's Garbled Circuits and FHE, in which no single party can obtain the full information set. The first and biggest issue is how to combine the work with data collected by other participants while keeping individual data anonymous. The approach has limitations of scope and difficult in dealing with varying data structures, as well as high costs of computation and communication of SMPC protocols.

2.2 Homomorphic Encryption (HE) for Privacy-preserving Data Analysis

Homomorphic encryption is an innovative method when it comes to securing data analysis allowing for operations on data while it remains encrypted. This method not only keeps the private data secure but 4 also is an important factor in the filed of financial fraud detection. *Myalil et.al. (2021)* leading the way in this research showing how homomorphic encryption can be applied in logistic regression models to detect fraudulent activities in finance sector. The method part of this study tackles two major problem, first is the uneven spread of the transaction data and the presence if malicious banks. The researchers have divided a dataset into ten parts which represents ten banks and implemented Epsilon Cluster Selection (ECS) algorithm. This algorithm filters out the harmful data during the models update phase making sure the model is robust against attacks from malicious entities. In spite the progress the method had its limitation in distinguishing between irregularities caused by uneven data and those caused by attackers which points more improvement and study for future work.

Embarking upon the path to fraud detection, *Zhuang et.al. (2022)* further explore the use of Homomorphic Encryption in logistic regression for fraud detection. The study has introduced a novel approach for detecting financial fraud in transaction data along with enhancing the privacy. Methodology used in the research paper involves encrypting the data using CKKS homomorphic encryption and allowing logistic regression to be performed on encrypted data and then decrypting it to identify the fraudulent activities. Problem they are trying to solve is to make sure that original sensitive data remains inaccessible to the cloud computing platform where the fraud detection model operates. Mentioned approach greatly works to improve data security and privacy but brings some limitations on to the table which is computational and time overhead due to encryption and decryption process. Handling data in its encrypted form slows down the process when compared to traditional plaintext method but the researchers have accepted this issue as it increases privacy and security of the data.

In today's time credit card fraud is a big problem for bank and their customers making it necessary to have advance fraud detection system. Developing such system requires customers information which leads to privacy concern. *David Nugent et.al. (2022)* has addressed this issue in his study by creating a method to detect such fraud without revealing sensitive data. The paper has used two models, XGBoost and feedforward classifier neural network for fraud detection. These two models are first implemented on plaintext data and after that they are encrypted using homomorphic encryption. While XGBoost model showed better performance but neural network model was easy to deploy. Despite these models are effective but they have their limitation as discussed in the paper these models where not easy to deploy in real world settings and they faced complexity in encrypting and decrypting the data while using these models leaving room for further optimization and research.

Data privacy and system security on cloud-based banking system using Homomorphic encryption which enhances a secure data processing system in cloud environments has been explored by *Mittal, S., Jindal, P. and Ramkumar, K. R. (2021)* in his research. The framework allows arithmetic operations such as additive and multiplicative operations on banking data. This allows the banks to maintain privacy of the customer without exposing it by performing complex calculations on the

customers banking data.The main challenge in cloud-based banking system is to maintain the confidentiality of the private data and preventing the data from data breaches. The framework shows significant improvements in enhancing the privacy and the security during the banking transactions on a cloud-based infrastructure. However, the are some limitations to it due to high computational demands and the arithmetic complexity of homoeomorphic encryption. This can impact the scalability and efficiency of the banking system in real-world scenarios.

Anomaly detection issue in cloud computing focusing on data privacy protection has been addressed by *Alabdulatif et al. (2017).* Authors solution involves integration of lightweight homomorphic encryption with a fuzzy c-means (FCM) clustering algorithm, this will ensure that the data remains encrypted throughout the anomaly detection procedure. This research method identified the anomalies and keeping all the data private. This faced challenges related to execution of time overhead and efficiency due to the complexity of encrypting and process data.

Zhang, P. et al pays attention to the problem of k-means clustering on encrypted data so that the data can be protected from disclosure especially for applications in cloud computing. To address the problem, the authors present the method that allows performing computations on encrypted data using homomorphic encryption and applying k-means clustering while preserving data privacy. The primary concern is to perform a k-means clustering operation and at the same time maintain the privacy of data. The method shows a good result in the context of privacy-preserving clustering, however, it has several drawbacks, such as computational overhead associated with the use of encryption and the efficiency of data encrypted in the process of clustering, which affect scalability and performance

2.3 Enhancing Security and Privacy in Distributed Computing Across MultiCloud Platforms

Business and institution are relying more and more on cloud computing now days so ensuring security and privacy of data is also becoming more important. Similar discussion has been done in the paper *Oscar et.al. (2021),* which introduces an amazing strategy for SMPC in cloud environments. In the study researchers are addressing the issue of securely outsourcing MPC to untrusted cloud environments without compromising integrity and confidentiality of input and output data. For doing so the research improves the SPDZ protocol which enables outsourcing of data and computation of data while making sure that outcomes are correct and private. This includes the use of honest server (HS) which checks the correctness of the work after cloud servers finishes their tasks. The research includes the detailed explanation of handling Message 1Authentication Code (MACs) for confirming the correctness of computation and verifying operations accuracy across nodes by inspecting these MACs.

Aida et.al. (2015) has shed some light on improving security and cloud based operations. The study is inspired by European project named CLARUS, this paper talks about a technique in which statistical analysis an be performed on the data which is on multiple cloud without merging the data hence keeping the data private. The data is broken into smaller and less sensitive fragments and then distributed in multiple clouds. This approach not only makes data private but also makes it secure while merging data all together which is generally an issue in cloud

This study by *Yongkai Fan et al.(2023)* is devoted to the problem of how to perform k-means clustering for sensitive data with enrolment privacy, which is important for the tasks of collaborative data analysis. Thereby, the authors introduce the PPMCK system, which utilises techniques of homomorphic encryption and SMC. The client's collected data is first encrypted at the client-end and then it is transferred to a cloud server where k-means is performed on encrypted data using the

privacy-preserving weighted average problem (PPWAP) protocol. The first requirement is to make k-means clustering secure but information has to remain protected all the time it is being processed. But in this system it is having huge problem of computation and efficiency because of data in the form of cipher text which is not so easy to manage and consumes plenty of time.

2.4 Application of Machine learning Algorithms in Encrypted Data Analysis

In financial sectors like banking anti money laundering is very challenging issue. To prevent it many government policies, procedures and ordinance are designed to put an end to this problem. The study by *Kumar et.al. (2021)* discusses implementation of machine learning techniques particularly linear support vector machine and decision tree to identify illegal transaction within a dataset of 10,000 transactions. The research paper specifically talks about the challenge of detecting money laundering activities from vast pool of legitimate transaction. He researchers follow step by step method that starts by data acquisition, feature selection, dividing dataset, implementing model and then optimizing the model. Models are trained using measures such as precision, recall and accuracy to see how well these models perform in identifying illegal transaction. While the approach is effective and among both models decision tree performed better, the method still had limitation which was related to the use of dataset that may not reflect the complexity of real world transaction which points the need of exploration of diverse dataset and machine learning models to improve performance of AML detection system.

In the context of keeping information secure, a study by *Cong et.al. (2022)* developed a sorting hat system which uses fully homomorphic encryption developed for evaluating decision tree privately in those scenarios where the clients wish to use servers decision tree model for classifying there confidential data without revealing there data to the server. The research introduces an improved transciphering method enhancing communication cost between the client and the server which tackles one of the major challenges in fully homomorphic encryption applications. Sorting hat protocol has two versions first one reduces the computation overhead and the second version called t-SortingHat reduces the communication overhead. Despite its innovative approach the study highlights certain limitations like need of a reliable server for checking results which could be a problem if its not secure, complexity in managing the encryption and decryption process. The protocols ability to protect secure data against sophisticated attacks and its effectiveness with various data arrangements are the areas suggested for further research.

In comparison to all the related works my research will be focusing on introducing a collaborative approach among multiple parties for data analysis without sharing any private information, for this approach I will be utilizing Secure multi-party computation framework for collobration and homomorphic encryption for encrypting sensitive data, with the integration of these two strategies sensitive data of each participants will be secure from each other which will enhance the trust among the participants. Along with this to detect fraud we will be training machine learning model for fraud detection, and this would be majorly done on cloud so that we don't face any sort of computational and memory problem and also with the help of cloud there will be more room for scalability. As previous studies have focused on one type of fraud and only worked on single cloud platform, our research will be using a multi-cloud approach and will be focusing on detecting multiple types of fraud.

Author/Year	Problem Addressed	Method	Algorithm	Dataset	Tools	Results	Limitations

Ramya et al., 2024	Scalability and efficiency in data-intensive tasks	ABY2.0 protocol for SMPC	ABY2.0 protocol	Not specified	ABY2.0 protocol suite	Enhanced scalability and efficiency	Not specified
Wajdi Alghamdi et al., 2023	Secure MPC for collaborative data analysis	Secure Multi-Party Computation (MPC)	Yao's Garbled Circuits, SMC-BC, FHE	Not specified	SMPC, cryptographic protocols	Showed how SMPC can safely analyze data across different fields without sharing actual data	Challenges in scaling and managing complex data and user scenarios
Yongkai Fan et al., 2023	Privacy-preserving multi-party computing for K-means clustering	Lightweight Homomorphic Encryption (HE)	K-means clustering	Simulated data	Lightweight HE tools	Applied lightweight encryption to enable private data clustering	Limited by the type of computations and potential scalability issues
Kun Liu. et al., 2022	Privacy-preserving Naive Bayes classification based on secure two-party computation	Systematic review of SMPC	Naive Bayes classification	Financial data	SMPC, review of existing approaches	Highlighted effective fraud detection using SMPC without revealing private data	Needs better scalability and efficiency in real-life applications
Cong et al., 2022	Secure data analysis	Fully homomorphic encryption	Decision trees	Secure data sets	Homomorphic encryption tools	Reduced computation and communication overhead	Need for a reliable server and complex management
David Nugent et al., 2022	Credit card fraud detection	Homomorphic encryption with ML	Machine learning models	Credit card data	ML frameworks, encryption tools	Effective without compromising data privacy	Complex deployment in real-world settings
Zhuang et al., 2022	Transaction data security	CKKS homomorphic encryption	Logistic regression	Transaction data	CKKS encryption method	Enhanced privacy during fraud detection	Time and computational overhead for encryption/decryption
Myalli et al., 2021	Financial fraud detection	Homomorphic encryption	Logistic regression	Financial records	CKKS scheme, encryption tools	Robust model against attacks	Difficulty distinguishing between data irregularities
Oscar et al., 2021	Outsourced cloud computing	Improved SPDZ protocol	SPDZ protocol	Cloud data	SPDZ protocol suite	Ensures data privacy and integrity	Reliance on a single honest server as a potential failure point
Sonam Mitta	Data privacy and system security for	Homomorphic Encrypti	Not specified	Banking data	Homomorphic encrypti	Improved data security and privacy for	Complex and resource-intensive to

l et al., 2021	banking using Homomorphic Encryption	on			on techniques	cloud banking systems	implement
Kumar et al., 2021	Anti-money laundering in the banking sector	Machine learning	Decision trees	Banking transactions	ML libraries	Effective in identifying illegal transactions	Dataset may not reflect real-world complexity
David et al., 2020	Federated learning in financial applications using differential privacy and SMPC	Differential Privacy, Secure MPC	Logistic regression, SMPC algorithms	Credit card fraud data	Open-source simulation platform	Improved collaborative learning without exposing sensitive client data, optimized for finance	Complexity in balancing data privacy with model accuracy, risk of reduced model precision
Sangers et al., 2019	Collaborative fraud detection	Secure multiparty computation	PageRank algorithm	Not specified	Custom SMPC framework	Secure fraud detection without privacy compromise	High computational and communication complexity
Zhao et al., 2019	Privacy in financial services domains	SMPC protocol development	Custom SMPC protocols	Not specified	SMPC tools	Improved application of SMPC principles	Scalability issues and complex protocol design
A. Alabdulatif et al., 2017	Privacy-preserving anomaly detection in cloud computing	Lightweight homomorphic encryption	Fuzzy c-means (FCM) clustering algorithm	Simulated data sets	Cloud computing environment, encryption tools	High accuracy in anomaly detection, preserved data privacy	Execution time overhead, efficiency concerns
Aida et al., 2015	Data privacy in cloud computing	Statistical analysis	Statistical analysis methods	Cloud data sets	Statistical tools	Maintained data privacy without dataset merging	Potential for data reconstruction by colluding providers

Table 1: Summarization of related works

3 Research Methodology

Steps of Research Methodology for Securing Financial Sector in the Cloud: A Multi-Cloud Approach to Fraud Detection Using Secure Multi-Party Computation is as follows :

- **Research Understanding:** The study aims to bring multiple financial parties to work together in a collaborative manner without revealing any sensitive data to each other and to detect various types of financial fraud, such as Money laundering and Credit card fraud, by using Homomorphic Encryption (HE) and Secure Multi-party Computation (SMPC) framework. For achieving this goal, each party will encrypt their datasets locally and then store it on their respective cloud platforms and then securely transfer the data to AWS S3 bucket. Then SMPC framework will be created which would facilitate the extraction of necessary information or features from each datasets and then based on this information

machine learning model will be trained and evaluation will be done accordingly.

- **Data Collection and Exploration:** For conducting our research we will be using two datasets both are sourced from Kaggle, one dataset will be based on anti-money laundering (https://www.kaggle.com/datasets/berkanoztas/synthetic-transaction-monitoring-dataset-aml) and another will be about credit cardfraud (https://www.kaggle.com/datasets/iabhishekofficial/creditcard-fraud-detection/data). Structure of both datasets will be examined to get insight into their features, quality and potential issues. This involves identifying the importance of various features for fraud detection and determining which information will be encrypted.

- **Data Cleaning and Pre-processing:** This step involves preparing each dataset to be used for model training. We will clean each dataset by filling in or removing any missing values, fixing inconsistencies or errors, dealing with outliers that might impact our results. For maintaining consistency across the datasets, we will convert categorical data into numerical data and further normalize it.

- **Dataset Encryption:** For maintaining security of data each participant involved in this approach will encrypt there data locally and only that information which is sensitive and private this could be customers name, account number etc. By doing this data will be safe during analysis.

- **Dataset storage and Transfer:** After each party encrypt their data locally they will they upload it to there respective cloud platform and from these they will be migrated to AWS S3 bucket, fulfilling the multi-cloud approach taken in the study.

- **SMPC Framework creation:** We plan to create a Secure Multi-Party Computation framework that will allow multiple parties to work together on encrypted data without sharing there sensitive information. This framework will apply such cryptographic techniques as secure integer and fixed-point arithmetic to compute on the encrypted data such that data is never disclosed during the computation process.

- **Model training:** After obtaining processed datasets from SMPC framework, we will use decision tree, random forest and logistic regression to train our machine learning model. These models will be used to detect frauds such as money laundering and credit card fraud in financial sector. The training will be done on encrypting data so that the data remains secure and without needing to be decrypted.

- **Evaluation:** After machine learning model is trained we will check their performance using standard metric such as F1 score, R2 score, root mean square , precession , recall and accuracy. With the help of these evaluation we will be able to determine how effectively the models can detect frauds like money laundering and credit card fraud. By comparing the performance metrics we can identify which model works best in detecting fraud in financial sector.

3.1 Dataset Description

Anti-Money-Laundering-Dataset

This research uses anti-money laundering dataset which is gotten from Kaggle (Source). Money laundering is a problem that affects everyone, and necessitates strengthening methods of transaction monitoring, which is achieved by this dataset. The dataset It contains 9,504,852 records of transactions spanning from October 7, 2022, to August 23, 2023, with detailed information such as Time, Date, Sender_account, Receiver_account, Amount, Payment_currency, Received_currency, Sender_bank_location, Receiver_bank_location, Payment_type, Is_laundering, and Laundering_type. In total there are 12 features and 28 typologies: Among them, there are eleven normal cases and seventeen suspicious ones, which creates a huge database for the evaluation of the new methods of enhanced monitoring. These typologies owe a great deal in the structures of subnets as these enhance the relevance of the dataset to research.

Credit-Card-Fraud-Dataset

Credit card fraud dataset is our second dataset which we have also obtained from Kaggle (Link). The dataset used in this study consists of two primary files: cc_info. csv and transactions. csv. The cc_info. Header row of .csv file describes details of credit card; it has fields like credit_card (card number to differentiate between all the cards), city, state, zipcode, and credit_card_limit. The transactions. csv file of transactions includes credit_card-the card number associated with the transaction, date-time when the transaction occurred, transaction_dollar_amount-amount of money involved in the transaction, Long and Lat-coordinates of the place of the transaction. These files are combined on the credit_card column to create a large data set that contains both the cardholder's information and the transaction description, which is useful in identifying fraudulent behavior. The combined dataset has 294,588 records and 9 columns, and none of the fields holds any missing values.

Reason for selecting these two datasets is that both datasets have sensitive information like anti money landering dataset has 'senders_account', 'recivers_account' information and the second dataset has 'credit_card number', 'long', 'lat' information. Both of the information present in each datasets are sensitive and each party involved in this collaborative approach would not like to share such private data with each other. Therefore, it will be encrypted using homomorphic encryption and then ML model will be trained on this encrypted dataset for fraud detection without needing to decrypt the datasets. This ensures privacy and security while using datasets for fraud detection.

3.2 System Architecture

The system architecture of our study is designed to securely manage two different datasets which we are assuming belong to each financial institute. Institute A is having anti money laundering datasets, and they are operating on Google cloud platform and institute B is having credit card fraud dataset and they are operating on Azure cloud platform. AWS will be used as centralised system for storing all the migrated data and model training. Therefore, our research is leveraging multi cloud approach and advance machine learning techniques on a centralized cloud platform to identify fraudulent activities. Below is the detail break down of the system architecture:

- **Data Storage and Initial Processing**:
    - **Azure Path**:
        - **Azure Blob Storage**: Institute A creates a blob storage in azure and then uploads the locally processed and encrypted dataset using homomorphic encryption.
        - **Secure Transfer to AWS**: The encrypted dataset is transferred from Azure Blob Storage to AWS S3 using a python script which handles the

downloading of the file from Azure and then it uploads it to specified AWS S3 bucket ensuring efficient data migration between two cloud platforms.

- **GCP Path**:
  - **Google BigQuery**: Institute B first creates a table in GCP BigQuery and a bucket in google cloud storage. Dataset is locally processed and is encrypted using homomorphic encryption.
  - **Secure Transfer to AWS**: For securely transferring the data from GCP to AWS S3 we unload the data from GCP BigQuery table into the GCP bucket and then using cloud shell we create a configuration file which has necessary credentials for the specified AWS S3 bucket and then by running the script we securely migrate the data from one cloud to another.

- **Data Processing on AWS**:
  - **AWS EC2 Instance:**
    Setup: An EC2 instance is created to handle the downloading and processing of encrypted datasets from the AWS S3 bucket.
    SMPC Framework: To perform secure computation on the encrypted data, SMPC framework is constructed using MPyC library in Python.
  - **Feature Extraction and Enhancement:**
    Features are extracted from that sensitive information which is encrypted in each of the datasets for further processing and model training.
    Two new enhanced datasets are derived by adding more features to the existing encrypted columns of the initial datasets.
- **Model Training:**
  - **AWS Sage Maker:** AWS sagemaker is used for training and evaluating the machine learning models on the enhanced and encrypted dataset, providing a complete platform for building and testing models**.**

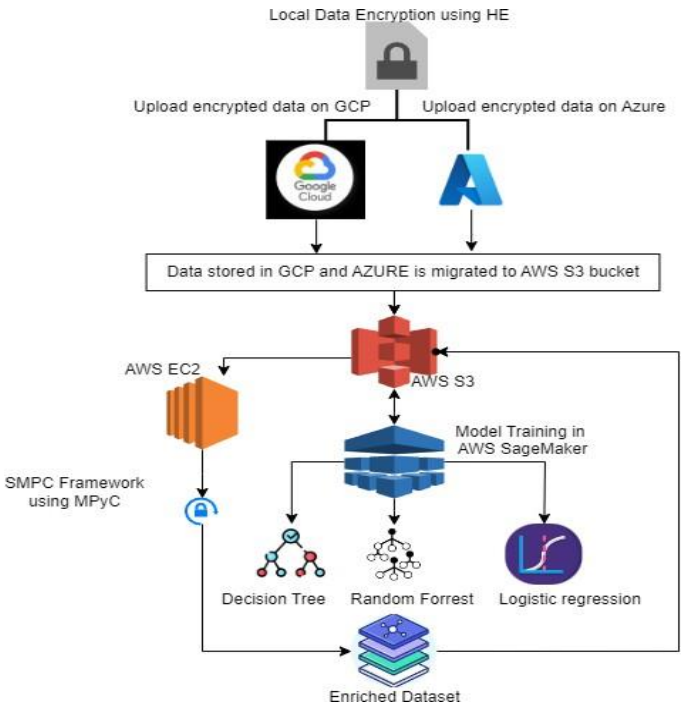


Figure 1: System Architecture

3.3 Proposed Approach

For our research, we are using various machine learning algorithms to detect fraudulent activities within financial datasets. The focus is on three key algorithms: Decision Tree, Random Forest, and Logistic Regression.

i. Decision Tree (DT):

Decision Tree a supervised learning algorithm which is used for both classification and regression tasks. Here, dataset is splitted into subsets based on most significant attribute, using tree like model decisions. In this research, this algorithm helps in identifying the attributes that contributes the most to the classification of transactions as fraudulent or non-fraudulent. This provides a clear decision path, and results can be transparent and interpreted easily.

ii. Random Forest (RF):

It is an ensemble learning method which builds multiple decision trees and joins them to get more accurate and stable predictions. With the help of this method overfitting tendency of each decision trees can be corrected by averaging the result. This method has been used in this research to leverage the power of multiple decisions trees to increase the accuracy and robustness of fraud detection. Also, handles vast number of features well and provides insights into feature importance, which are curial for understanding the drivers of fraudulent transitions.

iii. Logistic Regression (LR):

Logistic Regression is a statistical method which used for analysing dataset that contains one or more independent variable that determines an outcome. Outcomes are measured with dichotomous variable where there only two possible outcomes. Also, estimates the probability that give input point which belong to a certain class. Here, LR is used to model the probability of fraudulent transactions. Despite its simplicity, it is very powerful and interpretable, especially for binary classification problems.

4 Design Specification

The Design specification describes architecture and requirements for creating a secure and efficient system for detecting fraudulent activities from financial datasets obtained from different institutions. The research utilises multi cloud approach in which institute A is storing its Anti-money laundering dataset in Google cloud platform and institute B is storing its credit card fraud dataset in Azure. The data is however sensitive and to ensure its protection the data is encrypted using Homomorphic Encryption before being stored in the cloud. Data preparation is done progressively and efficiently and to ensure that even when in encrypted format the analysis can be done. Once each institute upload their encrypted data in their respective cloud platform, they are then securely migrated to central AWS S3 bucket. An AWS EC2 instance is then utilised to process these encrypted datasets so that necessary features can be extracted from encrypted information without needing to decrypt them in a secure environment and this would be done using Secure Multi-Party computation framework, resulting in enriched datasets which will be then used to train machine learning models.

Decision Tree, Random Forest, and Logistic Regression are the machine learning models being used in this research and these models are chosen for their effectiveness in classification tasks. These models are trained and tested using AWS SageMaker a platform that makes it easier and faster to work with large datasets and compare different models. Each model performance is measured using metrics like F1 score, accuracy, precession and recall. By comparing these results, the system identifies which model will be best in detecting fraudulent activities. This entire process from encrypting data to training ML model on the encrypted data is carried out in a secure way so that each participant involved in this collaborative approach can trust that their sensitive information remains

confidential. This comprehensive and secure approach not only enhances fraud detection but also fosters trust and collaboration among the institutions involved.

5 Implementation

This research project employs Python programming language extensively for data encryption, secure data transfer, and training machine learning models within sagemaker.

- **Data Encryption and Secure Transfer**:
  The study begins by each participant properly processing their datasets by doing data cleansing, applying normalization and feature engineering, to make sure the data is ready for encryption. As we know institute A which has AML dataset, will encrypt the data using Homomorphic encryption with SEAL library in python. Now instead of encrypting the whole dataset only that information will be encrypted which is sensitive like in AML 'senders account' and 'recovers account' this two information are very confidential data about customer. After encryption is done new encrypted dataset will be uploaded to GCP bucket. In the same manner institute B using HE will also encrypt only sensitive information about their customers data which is 'credit card', 'long', 'lat', and upload the new encrypted data to Azure blob storage.
  Now for securely transferring data to AWS where rest of the process will be done , GCP will be using BigQuerry for transferring data to S3 bucket as data transfer option in GCP only allows to move data within GCP only and for Azure a python script will be written to securely migrate the data from blob storage to specified S3 bucket.

- **Feature Extraction Using SMPC**:
  Once the datasets are securely migrated from each cloud platforms to AWS S3 bucket then an EC2 instance is deployed on AWS where encrypted datasets are downloaded from s3 bucket. Now for extracting features from each encrypted datasets a Secure Multi-Party Computation (SMPC) framework is implemented with the MPyC library in Python to perform privacy preserving computations on the encrypted data. In credit card fraud dataset, the MPC script counts transactions, sums amounts, and computes average transaction amounts by location without decrypting the sensitive information needed for the computation. Similarly in anti-money laundering dataset, the MPC script confidentially counts the number of interactions and totals the sum of transactions between each sender-receiver encrypted pair. These scripts also make it possible for data to be encrypted right from the computation process and only decrypted at point of output, thus protecting data privacy. The SMPC techniques ensure that the data that is to be computed is encrypted all the time within the computation and decryption is only carried out at the end of the computation process. This approach is useful in ensuring the privacy of the data while coming up with enriched datasets that include newly extracted features from the encrypted information.

- **Machine Learning Model Training and Evaluation**:
  After features are extracted, enriched datasets are uploaded to AWS S3 bucket for training machine learning models. ML model will be trained and evaluated using AWS sagemaker which has in built juypter notebook. Now our models will be trained and tested on three datasets, which are enriched AML dataset, enriched credit card fraud dataset and lastly combined datasets of both and all these three datasets will be trained using Random Forest, Decision tree and logistic regression algorithms.

1. Combined Dataset:
   The combined dataset was created by merging the enriched AML and enriched credit card fraud datasets. Missing columns were filled with zeroes to ensure consistency. The dataset was then divided into testing and validation sets. SMOTE (Synthetic Minority Over-sampling Technique) was used to handle class imbalance. Three machine learning models—Random Forest, Decision Tree, and Logistic Regression—were trained on this dataset using bagging techniques to enhance their robustness. These models were evaluated using metrics like accuracy, precision, recall, F1-score, and confusion matrices. The evaluation results confirmed that the models were effective in detecting fraudulent activities.
2. Enriched Credit Card Fraud Dataset & AML Dataset
   Both of this enriched dataset is now processed by excluding encrypted data and instead of that we use extracted features as these features are generated from the encrypted information only, so this would allow us to train our model efficiently and easily without increasing any computational complexity. Dataset is further processed by excluding unnecessary columns and encoding categorical variables using one-hot encoding, handling missing splitting dataset into training and validation sets. For handling class imbalance SMOTE was utilised. The models (Random Forest, Decision Tree, and Logistic Regression) were trained using bagging techniques, and their performance was evaluated using the same metrics. The models showed high accuracy, demonstrating their ability to identify fraudulent transactions.

5.1 Experimental Setup

i. **Data Sources and Encryption**:
- Two datasets AML dataset and credit card fraud dataset were used in this research and were available on Kaggle.
- For securing the private information each datasets were encrypted using homomorphic encryption via SEAL library in python.
- Both encrypted datasets, AML and credit card fraud datasets were stored in GCP and Azure blob storage.

ii. **Secure Data Transfer**:
- AML dataset was migrated from GCP to AWS S3 using google BigQuerry and credit card fraud dataset was migrated from Azure blob storage to AWS S3 bucket using python script.

iii. Feature Extraction Using Secure Multi-Party Computation (SMPC):
- An AWS EC2 instance was created to process the encrypted datasets stored in AWS S3.
- To perform computations on the encrypted data SMPC framework was created using MPyC library in python and necessary features were extracted from the encrypted information.

iv. Dataset Preparation:
- After extracting features, both enriched datasets were prepared for training ML model.
- A combined dataset was created by merging the enriched AML and enriched credit card fraud datasets, ensuring that all features were consistent across both datasets.

v. **Model Training and Evaluation**:
- The obtained datasets were divided into training and testing data, and SMOTE was used in order to handle issues with class imbalance.
- Three machine learning algorithms of Random Forest, Decision Tree, and Logistic Regression were implanted using bagging for better generalization of the models.

- Each machine learning model's performance on each dataset were evaluated using accuracy, precision, recall, and F1-score.

5.2 Tools and Technology Stack:

1. Python: The primary programming language used for implementing encryption, SMPC framework creation, and for training machine learning models.
2. SEAL (Microsoft SEAL Library): A homomorphic encryption library used to securely encrypt sensitive data before any computations are performed.
3. MPyC (Multi-Party Computation in Python): A Python library which allows multiple parties to collaboratively compute on encrypted data while preserving the confidentiality of the data.
4. Google cloud platform- GCP BigQuerry and container storage for storing and migrating data.
5. Azure – Blob storage is used for storing and migrating data.
6. AWS S3: Utilized for cloud storage, including storing encrypted datasets and saving model outputs after training.
7. Google collab is used for pre processing datasets.
8. AWS SageMaker:
9. Notebook Instance Type: ml.m5.2xlarge instance used for training, evaluating, and deploying machine learning models on the processed datasets. SageMaker provides a fully managed environment with built in juypter notebook and easily integrates with other AWS services,
10. AWS EC2:
11. Instance Configuration: EC2 instances t2. large was used for handling encrypted data processing and SMPC computations.
12. IAM Role Configuration:
    SageMaker and EC2 IAM Role: A unified IAM role with S3 Full Access and SageMaker Full Access permissions was created and used for both the EC2 instance and SageMaker notebook instance. This role facilitated secure access to S3 buckets for data storage and retrieval, and full capabilities for SageMaker operations.
13. Google Cloud Platform (GCP) and Azure: Initially used for data storage and processing before secure transfer to AWS. Each cloud platform handled the specific datasets (anti-money laundering and credit card fraud) relevant to their respective financial institutions.
14. Pandas, NumPy, Scikit-learn, Matplotlib: Essential Python libraries for data manipulation, data visualization., preprocessing, and implementing machine learning algorithms, used extensively throughout the project.

6 Evaluation

In this research project we have conducted our implementation using three datasets which are anti-money laundering dataset, combined dataset of both and credit card fraud dataset. We have individually trained and evaluated machine learning models on each datasets and evaluated their performance through metrics such as F1 score, Recall, Precision, MSE and RMSE.

6.1 Performance Metrics

. . .

Metric	Formula	Definition
Accuracy	$Accuracy = TP + TN + FP + FNTP + TN$	Accuracy measures the proportion of true results (both true positives and true negatives) among the total number of cases examined. It tells us how often the model is correct overall.
Precision	$Precision = TP + FPTP$	Precision is the ratio of correctly predicted positive observations to the total predicted positives. It indicates how many of the predicted positive cases are actually positive. High precision indicates a low false positive rate
Recall	$Recall = TP + FNTP$	Recall (also known as sensitivity or true positive rate) is the ratio of correctly predicted positive observations to all observations in the actual class. It indicates how many of the actual positive cases the model is capturing. High recall indicates a low false negative rate.
F1 Score	$F1 Score = 2 \times Precision + RecallPrecision \times Recall$	The F1 Score is the weighted average of Precision and Recall. It considers both false positives and false negatives, making it useful when the class distribution is imbalanced. The F1 Score balances the precision-recall trade-off, providing a single metric that considers both.
MSE	$MSE = n1 \sum i = 1n (yi - y\hat{} i)2$	Mean Squared Error (MSE) is the average of the squared differences between the predicted values and the actual values. It measures the average magnitude of errors in the predictions, penalizing larger errors more heavily due to the squaring process. A lower MSE indicates a better fit of the model to the data.
RMSE	$RMSE = n1 \sum i = 1n (yi - y\hat{} i)2$	Root Mean Squared Error (RMSE) is the square root of the Mean Squared Error (MSE). It provides the error metric in the same units as the target variable, making it easier to interpret. Like MSE, a lower RMSE indicates a better fit of the model to the data. RMSE is more sensitive to large errors than MSE because the errors are squared before averaging.

6.2 Model Evaluation on all Three Datasets:

6.2.1 Dataset 1: AML

Initial model training was done using SMOTE technique, see figure 2 and in this Random Forest model performed well with 100 % accuracy, recall, precision and F1 score on training set and 99.99% of high accuracy in validation set. Confusion metrics show minimum misclassification which demonstrates models' strong ability to capture complex patterns in data. The decision tree model also performed well with accuracy of 99.93% on both training and validation set but it had more misclassification than random forest along with maintaining high precision and recall, proving an effective mode for detecting money laundering activity. On the other hand, logistic model performed poorly with an accuracy of 51.8% on both training and validation set.

The near perfect performance of both Decision tree and Random Forest model signifies the case of overfitting and class imbalance, so to reduce it and to improve robustness and accuracy, bagging technique was used see the results in figure 3. Additionally logistic regression which performed poor in initial performance was also subjected to Bagging technique to enhance its performance. The bagging decision tree maintained high accuracy and achieved 99.94% on the training set and 99.93% on the validation set. It reduced misclassification and showed strong recall and F1 score. bagging random forest achieved 99.99% accuracy on both sets, while it maintained perfect precession and recall on training set its performance declined on validation set with 90.05% recall but overall, still performed well. On the other logistic regression performed poorly despite using bagging technique. It got accuracy of 51.5% and extremely low precision and F1 score. Due to the linear nature of logistic regression model, it could not handle the complexity of the AML dataset, and this is the reason this model performed poorly in both cases.

6.2.2 Dataset2: Credit card fraud

The evaluation of the Decision Tree, Random Forest, and Logistic Regression models on the credit card fraud dataset demonstrated outstanding performance across all models. Each model effectively handled the significant class imbalance present in the dataset, largely due to the application of SMOTE during preprocessing. The Decision Tree and Random Forest models both achieved perfect results, with 100% accuracy, precision, recall, and F1 scores on both training and validation sets. The confusion matrices showed no misclassifications, indicating that these models have fully captured the patterns in the data. However, this perfection also suggests a potential risk of overfitting, where the models may perform less well on unseen data. Logistic Regression, while not perfect, still achieved near-perfect results with an accuracy of 99.99% on both training and validation sets. To reduce overfitting Bagging Technique was applied in all three models. The bagging decision tree and random forest maintained same results which shows there strong generalization capabilities . on the other hand logistic regression also showed near perfect results with accuracy of 99.99% on both training and testing sets as it did previously also along with slight decline in precision, recall and F1 score but it still remained highly effective in detecting fraudulent transactions. Figure 4 shows the results of both.

6.2.3 Combined Dataset:

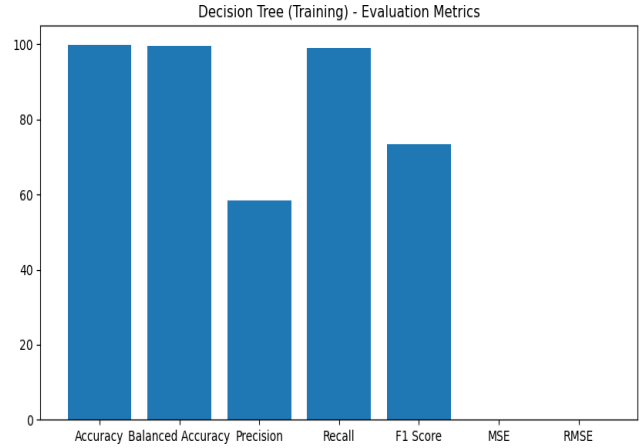
Before applying SMOTE models showed overall high accuracy but their ability to correctly classify the minority class such as fraudulent and non-fraudulent transactions, varied significantly. Random forest model with an accuracy of 99.96% performed well with perfect precision and recall for majority class but recall for minority class was slightly lower at 92% and F1 score was 0.96 for that class. The Decision tree model also performed well with an accuracy of 99.99% but struggled less with minority class. It showed precision 100% and F1 score was 0.98 despite getting recall of 97%. Logistic regression model performed very poorly; it got accuracy of 78.35%. precision for majority class was perfect but for minority

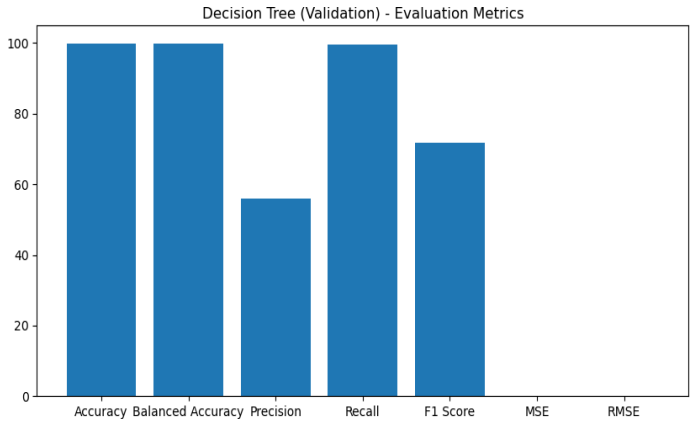
class it was only 2% and F1 score was 0.04 for minority class. So, to handle this class imbalance issue by oversampling the minority class, we utilised SMOTE technique.

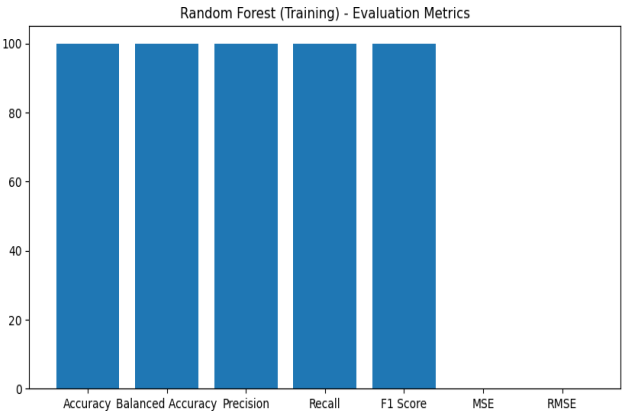
After Applying SMOTE there was no significant improvement recorded for random forest model and showed consistent result of accuracy 99.95%. The Decision tree model also did not show any big but only slight improvement with its accuracy of 99.99% and F1 score of

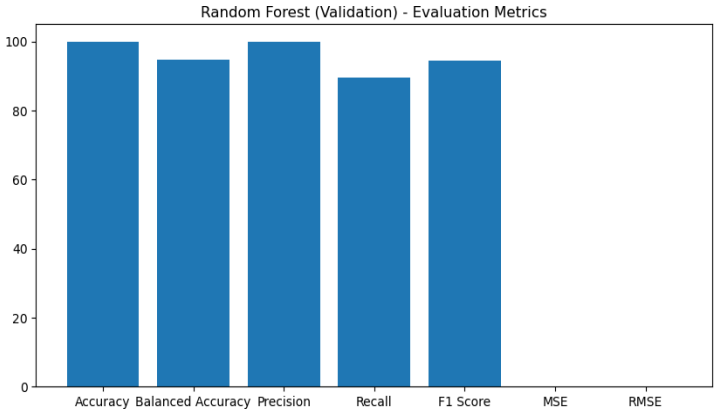
0.99 for minority class. On the other hand, the Logistic regression model showed no improvement even after applying SMOTE which showcases models' inability to handle complex, non-linear data even when class imbalance is mitigated. Figure 5 shows the results of both as the results did not change much.

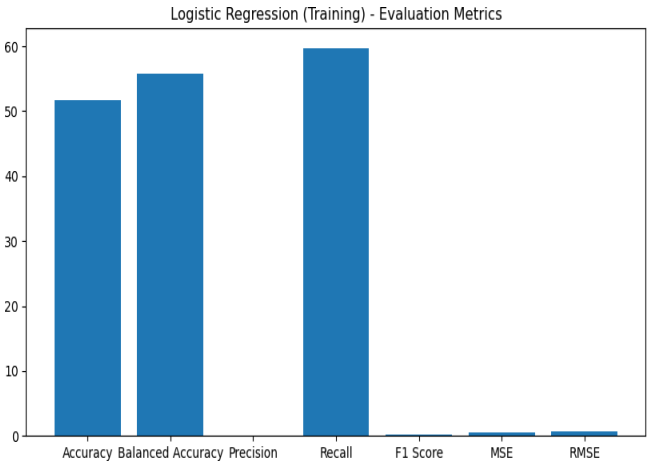
Figure 2: With SMOTE

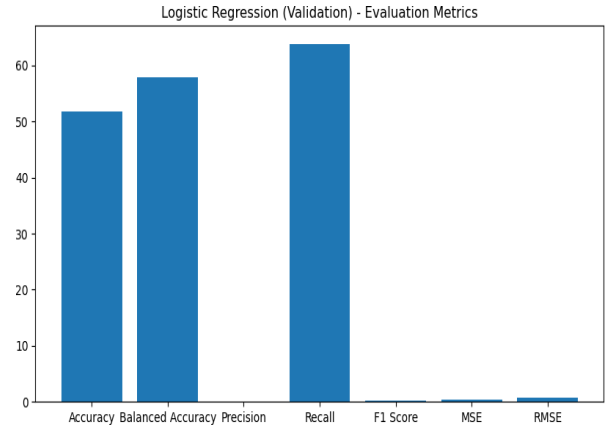


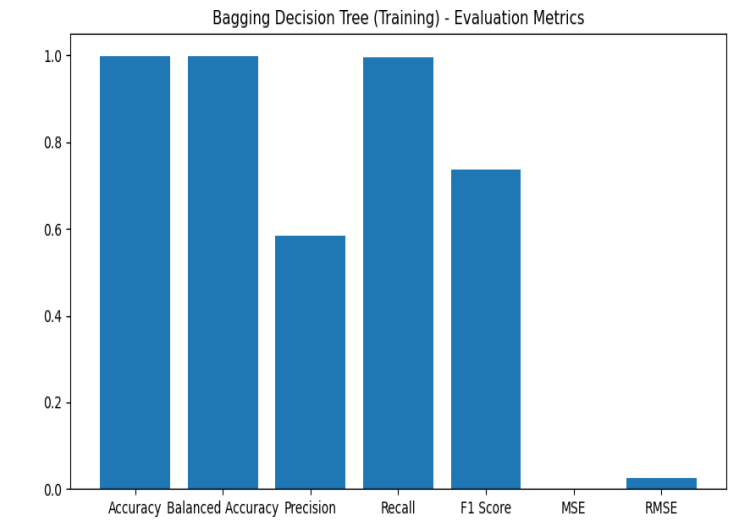


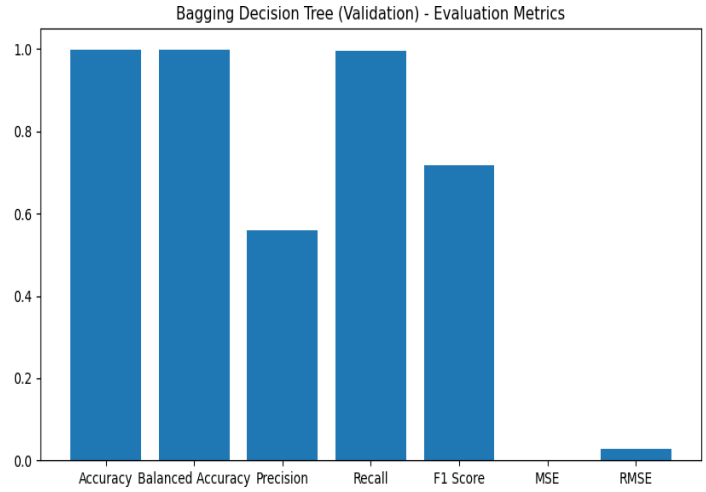


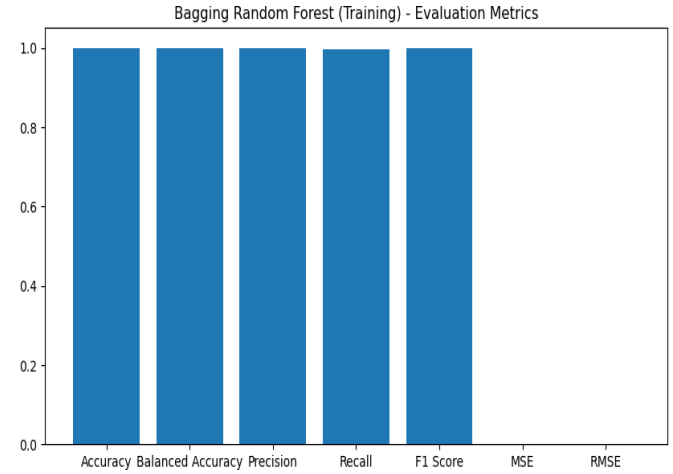


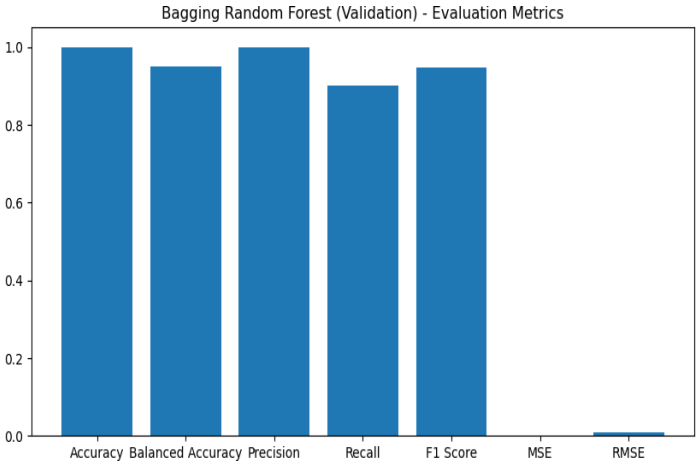


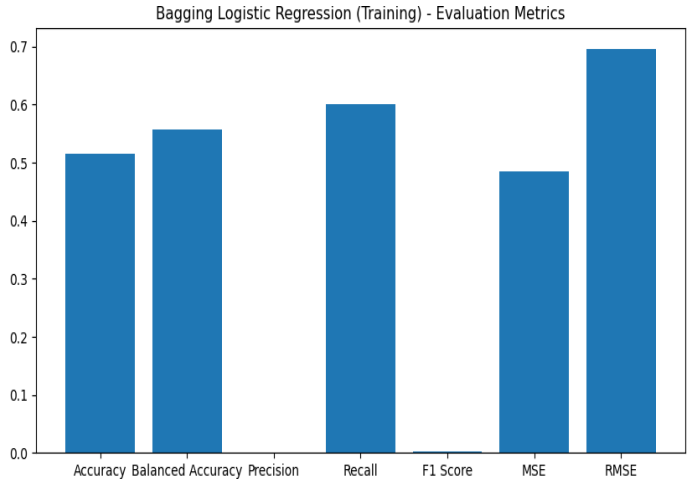












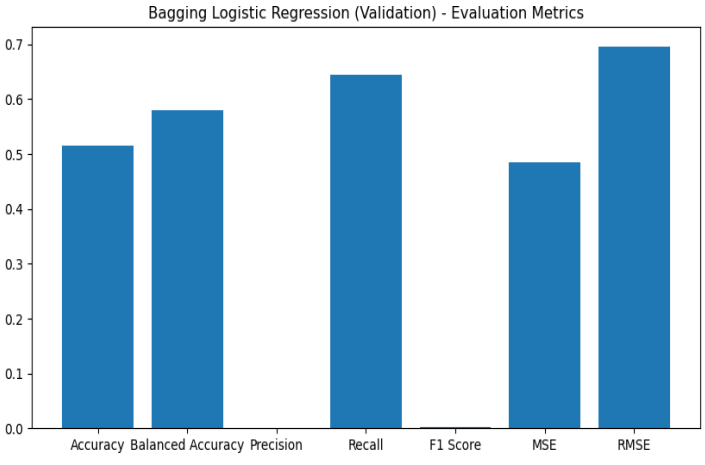
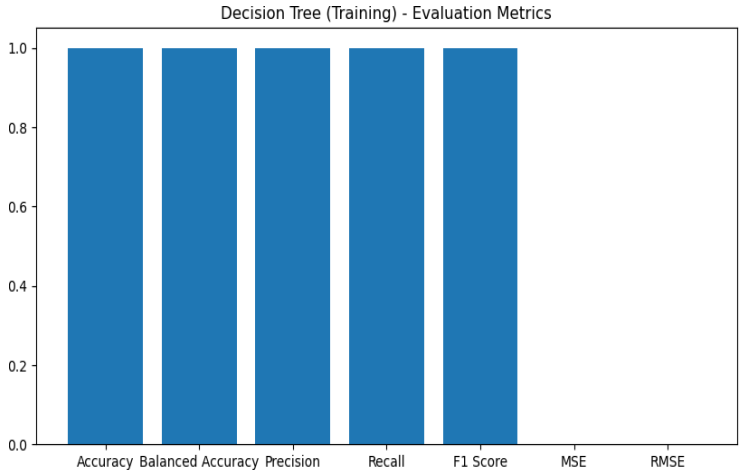
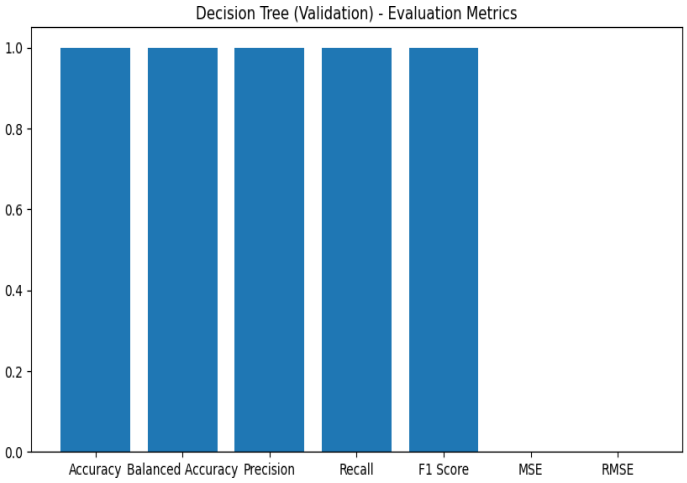
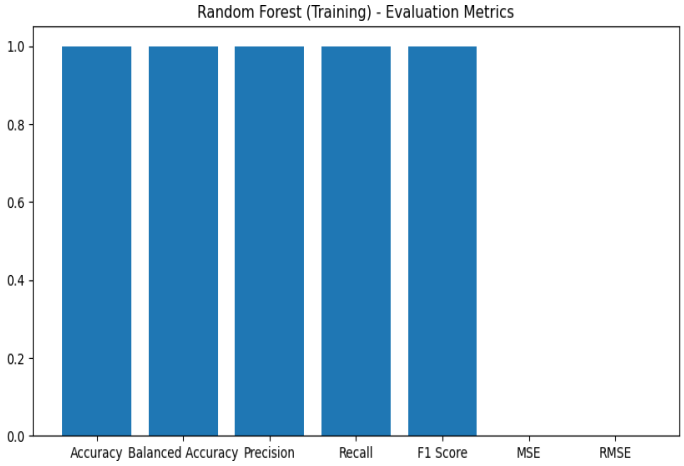
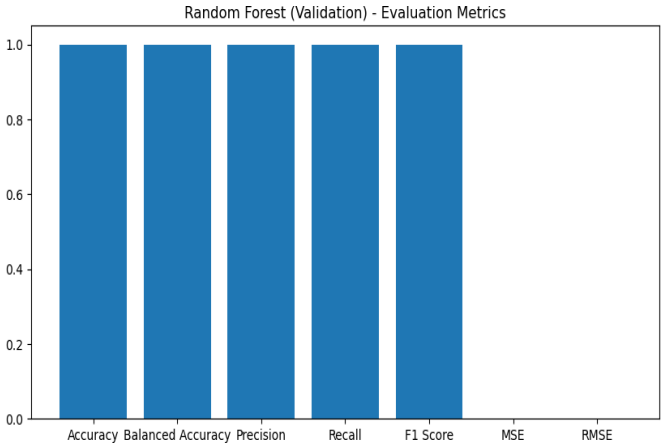


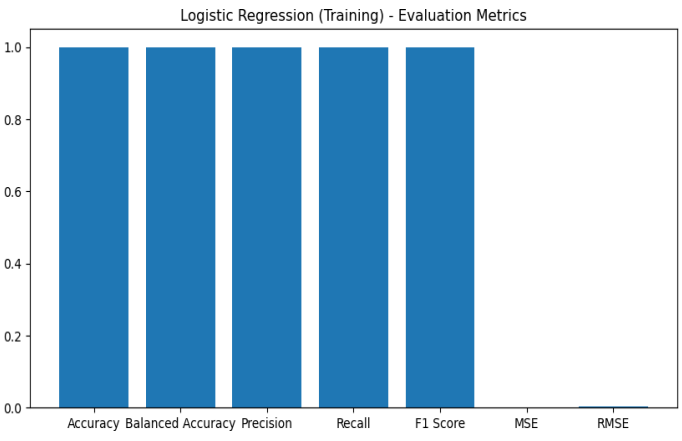
Figure 3: With SMOTE and BAGGING


Decision Tree (Training) - Evaluation Metrics
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
Balanced Accuracy
Precision
Recall
F1 Score
MSE
RMSE


Decision Tree (Validation) - Evaluation Metrics
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
Balanced Accuracy
Precision
Recall
F1 Score
MSE
RMSE


Random Forest (Training) - Evaluation Metrics
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
Balanced Accuracy
Precision
Recall
F1 Score
MSE
RMSE


Random Forest (Validation) - Evaluation Metrics
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
Balanced Accuracy
Precision
Recall
F1 Score
MSE
RMSE


Logistic Regression (Training) - Evaluation Metrics
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
Balanced Accuracy
Precision
Recall
F1 Score
MSE
RMSE

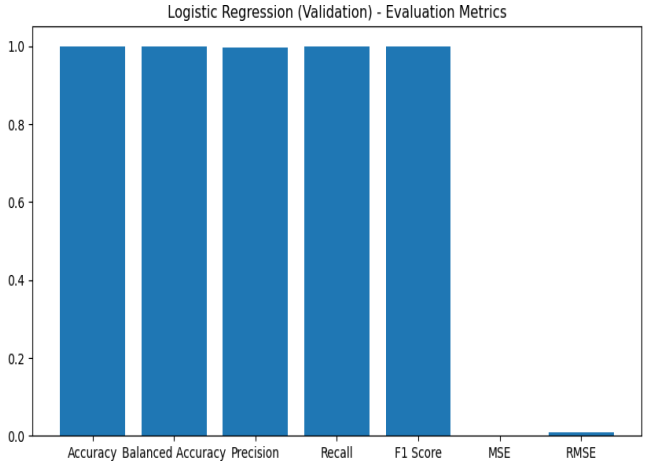
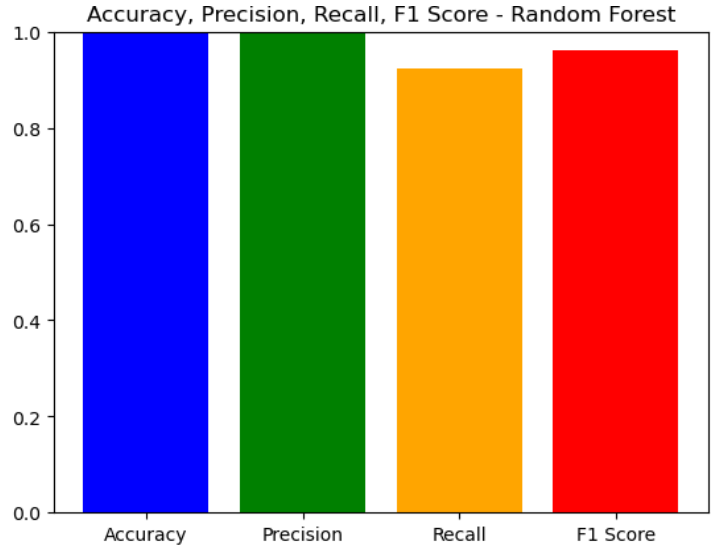

Logistic Regression (Validation) - Evaluation Metrics
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
Balanced Accuracy
Precision
Recall
F1 Score
MSE
RMSE

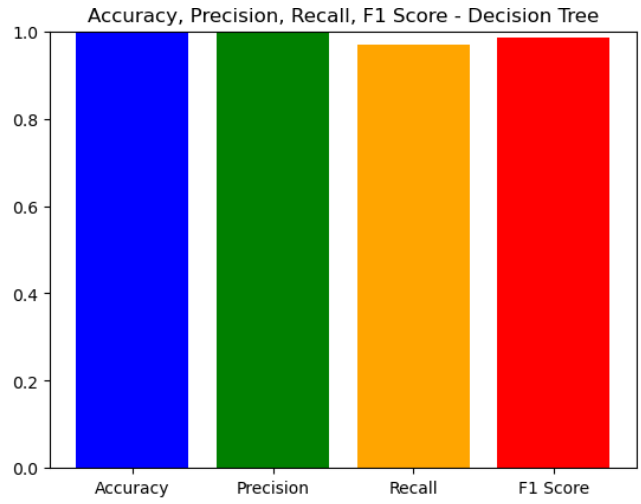
Figure 4 :With SMOTE & BAGGING



Classification Report (Test) - Random Forest Model:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	267346
1	1.00	0.92	0.96	1287
accuracy			1.00	268633
macro avg	1.00	0.96	0.98	268633
weighted avg	1.00	1.00	1.00	268633

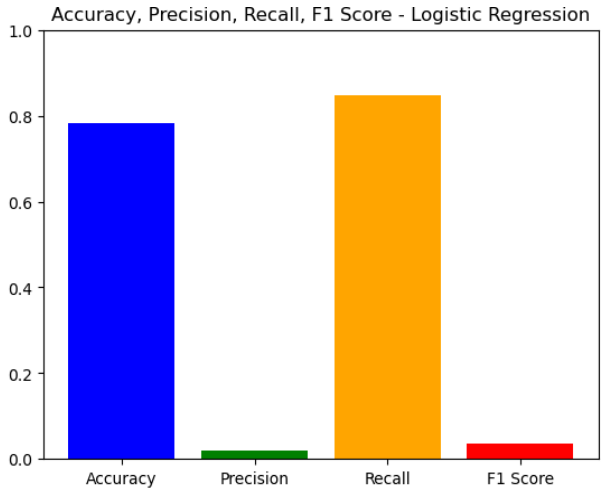
Accuracy Score (Test) - Random Forest Model:
99.96%



Classification Report (Test) - Decision Tree Model:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	267346
1	1.00	0.97	0.98	1287
accuracy			1.00	268633
macro avg	1.00	0.98	0.99	268633
weighted avg	1.00	1.00	1.00	268633

Accuracy Score (Test) - Decision Tree Model:
99.99%



Classification Report (Test) - Logistic Regression Model:

	precision	recall	f1-score	support
0	1.00	0.78	0.88	267346
1	0.02	0.85	0.04	1287
accuracy			0.78	268633
macro avg	0.51	0.82	0.46	268633
weighted avg	0.99	0.78	0.87	268633

Accuracy Score (Test) - Logistic Regression Model:
78.33%

Figure 5: Combined dataset

In summarizing our evaluation, we successfully achieved our goal in which we encrypted private information from each parties involved in collaboration, then used SMPC for extracting necessary features out of encrypted data and making sure data was encrypted throughout the process and then used the enriched datasets for training machine learning model for fraud detection. In comparison to previous studies which implemented SMPC and HE separately and in one cloud platform, in my project I have utilised SMPC and HE together and integrated it in a multi cloud environment and trained machine learning model on encrypted data ensuring private data is secure throughout the process and no decryption is done.

6.3 Discussion

In this research we successfully implemented what we were aiming for A secure collaborative approach for fraud detection. We successfully encrypted each of our datasets sensitive information using Homomorphic encryption, migrated these data from one cloud platform to another, created a SMPC framework in which participants involved for the noble approach, there encrypted data were securely analyzed within a secure environment ensuring decryption never occurs while extracting necessary features from the encrypted information, hence full filling our goal of training machine learning model on encrypted data. Using this approach of extraction made ML training and testing quite straight forward.

Regarding Model training and evaluation, Random Forest and Decision Tree model both performed exceptionally well Logistic Regression model did not perform well on all the three datasets, enriched AML dataset, credit card fraud Dataset and combined dataset of both. For dataset 1 we came to conclusion that with class imbalance issue and usage of SMOTE, Random forest and Decision tree gave near perfect performance indicating overfitting and With the usage of Bagging technique with SMOTE ,to handle class imbalance there was still no changes in both model mainly due to the reason that fraudulent data was more as compared to non-fraudulent and model we training more on the majority class, on the other hand Logistic regression performed poorly in both the scenario mainly due to his non-linear nature its not able to identify the complex relationships.

For dataset 2 all three-model performed well and gave near perfect performance, this was also indicating to overfitting due to class imbalance. But the interesting part was Logistic regression performed well in this because the dataset had more linear relationship between features and target variable. For combined dataset we firstly trained models without SMOTE and got to see random forest and decision tree performed well but logistic did not and with smote also there was not much improvement in random forest only slight improvement in decision tree in handling minority class.

Another finding from this research was that although training ML model was straight forward but it was computationally intensive and time consuming majorly all three datasets were large and sagemakar kernel failed many times, this is one of the reason why apart from SMOTE and Bagging no other techniques were used to handle class imbalance and overfitting issue.

7 Conclusion and Future Work

In this research project, a comprehensive exploration of fraud detection within the financial sector using a multi-cloud approach was conducted. The study focused on integrating Secure Multi-Party Computation (SMPC) and Homomorphic Encryption (HE) with machine learning models, specifically Decision Trees, Random Forests, and Logistic Regression, to securely analyze financial data across multiple cloud platforms. By employing advanced encryption techniques, the project ensured that sensitive data remained protected throughout the analysis process, addressing significant concerns about data privacy and security. The application of SMOTE for handling class imbalance and bagging

techniques to enhance model robustness provided further insights into the effectiveness of these models. Despite the strong performance of Decision Trees and Random Forests, especially after applying SMOTE, Logistic Regression struggled to handle complex, non-linear data, even with these enhancements. The findings underline the importance of collaborative approaches in fraud detection, where financial institutions can work together securely to detect fraudulent activities more effectively. Overall while the research demonstrates the potential of secure collaborative approach in detecting multiple fraud there is still more scope to this research. For future work we will incorporate more types of fraud such as bank loan fraud, experiment with more machine learning models that are better in handling imbalanced datasets. We will also explore how advance models like CNN and RNN can capture complex data pattern. For more secure computation we will plan to integrate SMPC with blockchain this will further enhance the transparency, data privacy and trustworthiness of the system among the participants. To address the computational challenge, we encountered in this research we will utilise more powerful cloud resources to reduce processing time and increase efficiency. additionally, we will focus on creating an user friendly application that would stream line the entire process from encryption to fraud detection.

8 Video presentation Demo

https://youtu.be/W315qi496cU

References

Alabdulatif, A. et al. (2017) "Privacy-preserving anomaly detection in cloud with lightweight homomorphic encryption," Journal of computer and system sciences, 90, pp. 28–45. doi: 10.1016/j.jcss.2017.03.001.

Alghamdi, W. et al. (2023) "Secure Multi-Party Computation for Collaborative Data analysis," E3S web of conferences, 399, p. 04034. doi: 10.1051/e3sconf/202339904034.

Bautista, O., Akkaya, K. and Homsi, S. (2021) "Outsourcing secure MPC to untrusted cloud environments with correctness verification," in 2021 IEEE 46th Conference on Local Computer Networks (LCN). IEEE, pp. 178–184

Burra, R., Tandon, A. and Mittal, S. (2023) "Empowering SMPC: Bridging the gap between scalability, memory efficiency and privacy in neural network inference," arXiv [cs.CR]. Available at: http://arxiv.org/abs/2310.10133 (Accessed: April 16, 2024)

Byrd, D. and Polychroniadou, A. (2020) "Differentially private secure multi-party computation for federated learning in financial applications," in Proceedings of the First ACM International Conference on AI in Finance. New York, NY, USA: ACM.

Calvino, A., Ricci, S. and Domingo-Ferrer, J. (2015) "Privacy-preserving distributed statistical computation to a semi-honest multi-cloud," in 2015 IEEE Conference on Communications and Network Security (CNS). IEEE, pp. 506–514.

Chen, Z., Cai, M. and Wang, Z. (2022) "Research on privacy fraud detection of Logistic regression based on homomorphic encryption," in 2022 IEEE 4th International Conference on Power, Intelligent Computing and Systems (ICPICS). IEEE, pp. 36–40.

Cong, K. et al. (2022) "SortingHat: Efficient private decision tree evaluation via homomorphic

encryption and transciphering." Available at: https://eprint.iacr.org/2022/757

Das, S., Kumar, A., Shaw, R., Tyagi, V. and Ghosh, A. (2021) 'Analysis of classifier algorithms to detect anti-money laundering', in Advances in Data and Information Sciences. Singapore: Springer, pp. 161-170. doi: 10.1007/978-981-16-0407-2_11.

Fan, Y. et al. (2021) "PPMCK: Privacy-preserving multi-party computing for K-means clustering," Journal of parallel and distributed computing, 154, pp. 54–63. doi: 0.1016/j.jpdc.2021.03.009.

Mittal, S., Jindal, P. and Ramkumar, K. R. (2021) "Data privacy and system security for banking on clouds using homomorphic encryption," in 2021 2nd International Conference for Emerging Technology (INCET). IEEE, pp. 1–6.

Myalil, D., Rajan, M.A., Apte, M. & Lodha, S., 2021. Robust Collaborative Fraudulent Transaction Detection using Federated Learning. In: Proceedings of the 20th IEEE

International Conference on Machine Learning and Applications (ICMLA). Pasadena, CA, USA, pp. 373-378. Available at: http://dx.doi.org/10.1109/ICMLA52953.2021.00064

Nugent, D. (2022) "Privacy-preserving credit card fraud detection using homomorphic encryption," arXiv [cs.CR]. Available at: http://arxiv.org/abs/2211.06675 (Accessed: April16, 2024).

Sangers, A. et al., 2019. Secure Multiparty PageRank Algorithm for Collaborative Fraud Detection. In: I. Goldberg and T. Moore, ed., Financial Cryptography and Data Security. FC 2019. Lecture Notes in Computer Science, vol 11598. Cham: Springer, pp. [page range].
Available at: https://doi.org/10.1007/978-3-030-32101-7_35

Zhang, P. et al. (01 May-June 2023) "Privacy-preserving and outsourced multi-party K- means clustering based on multi-key fully homomorphic encryption," IEEE transactions on dependable and secure computing, 20(3), pp. 1–12. doi: 10.1109/tdsc.2022.3181667.

Zhao, C., Zhao, S., Zhao, M., Chen, Z., Gao, C.-Z., Li, H., & Tan, Y.-a., 2019. Secure Multi- Party Computation: Theory, practice and applications. Information Sciences, 476, pp.357- 372. https://doi.org/10.1016/j.ins.2018.10.024