VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590018

A PROJECT REPORT

ON

"Pipechime: Federated Based Machine Learning and Analytics pipeline"

Submitted in the partial fulfillment of the requirements for the award of

BACHELOR OF ENGINEERING DEGREE

In

COMPUTER SCIENCE & ENGINEERING (DATA SCIENCE)

Submitted by

Ananya Raj M.N	**4AD22CD005**
Bharath N	**4AD22CD007**
Ranjitha	**4AD22CD040**
Vignesh T.D	**4AD22CD056**

Under the guidance of

Mrs. Madhu Nagaraj

Assistant Professor

Department of CSE (Data Science)

ATME College of Engineering

ATME College of Engineering,

13th Kilometer, Mysore-Kanakapura-Bangalore Road

Mysore-570028

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Belagavi-590018

ATME College of Engineering

13th Kilometer, Mysore-Kanakapura-Bangalore Road

Department of CSE (DATA SCIENCE)

CERTIFICATE

This is to certify that the project work entitled "Pipechime: Federated based machine learning and analytics pipeline" is the Bonafide work carried out by following students,

Ananya Raj M.N	**4AD22CD005**
Bharath N	**4AD22CD007**
Ranjitha	**4AD22CD040**
Vignesh T.D	**4AD22CD056**

in partial fulfillment for the award of degree of Bachelor of Engineering in CSE (Data science) from the Visvesvaraya Technological University, Belagavi during the year 2025-26. It is certified that all the corrections or suggestions indicated for internal assessment have been incorporated in the Major-project report deposited in the department librabry. The Major-Project report has been approved and satisfies the academic requirement with respect to Major-project work prescribed for Bachelor of Engineering degree.

Signature of the guide	Signature of HOD	Signature of Principal
Mrs. Madhu Nagaraj	**Dr. Anitha D.B**	**Dr. L Basavaraj**
Assistant Professor	Associate Prof. &HOD	Principal

Name of Examiners	**Signature with Date**
1. _____	_____
2. _____	_____

DECLARATION

We are, **Ms. ANANYA RAJ M.N[4AD22CD005], Mr. BHARATH N [4AD22CD007], Ms. RANJITHA [4AD22CD040], Mr. VIGNESH T.D[4AD22CD056]** student of VII semester, **Department of Computer Science & Engineering (Data Science)**, **ATME College of Engineering, Mysuru-570028** declare that the Project has been successfully completed. This work is submitted **to Visvesvaraya Technological University, Belagavi-590018**, in partial fulfillment of the requirements for the award of Degree of **Bachelor of Engineering in CSE (Data Science)** during the academic year 2025-2026.

NAME	USN	SIGNATURE WITH DATE
ANANYA RAJ M.N	**4AD22CD005**	
BHARATH N	**4AD22CD007**	
RANJITHA	**4AD22CD040**	
VIGNESH T.D	**4AD22CD056**	

ACKNOWLEDGEMENT

The successful completion of our Major-project would be incomplete without the mention of the names of the people who have made it possible. We are indebted to several individuals who have helped us to complete the project report.

We are thankful to **Dr. L Basavaraj**, **Principal**, ATME College of Engineering for having granted us permission and extended full use of the college facilities to carry out this project successfully.

We express our profound gratitude to **Dr. Anitha D.B Associate Professor & HOD**, Department of Computer Science & Engineering (Data Science) for her consistent co-operation and support.

At the outset we express our profound gratitude to our guide **Mrs**. **Madhu Nagaraj, Assistant Professor**, Department of Computer Science & Engineering (Data Science) for her consistent co-operation and support.

We are greatly indebted to our project coordinator **Dr. Vinod Kumar P, Associate Professor**, Department of Computer Science & Engineering (Data Science) for his timely inquiries into the progress of the project.

Lastly, we would like to thank our family and friends for their cooperation and support for successfull completion of our project.

Ananya Raj M.N **4AD22CD005**
Bharath N **4AD22CD007**
Ranjitha **4AD22CD040**
Vignesh T.D **4AD22CD056**

ABSTRACT

With the rapid growth of data-driven applications, ensuring regulatory compliance in data sourcing has become a critical challenge, particularly under stringent data protection laws such as GDPR, HIPAA, and DPDP. Traditional centralized data pipelines often expose sensitive information to privacy risks, unauthorized access, and limited auditability.

This Project ensure compliance, transparency, and trust in data sourcing. Federated Learning enables decentralized model training by keeping raw data at its source, thereby preserving data privacy and adhering to data minimization principles. Blockchain technology complements this approach by providing an immutable and transparent ledger for recording data provenance, consent, and model update transactions through smart contracts. The integrated architecture enforces compliance policies automatically, enables verifiable audit trails, and prevents unauthorized data usage without compromising analytical performance. The proposed pipeline demonstrates how combining privacy-preserving machine learning with decentralized governance can support secure, compliant, and scalable data utilization across multiple organizations and jurisdictions

CONTENTS

LIST OF FIGURES

TABLE

FLOWCHART

CHAPTER 1

INTRODUCTION

1.1 Preamble

In the current digital era, data has emerged as one of the most valuable strategic assets driving innovation, advanced analytics, and artificial intelligence across nearly every industry. Organizations in sectors such as healthcare, finance, manufacturing, retail, and public administration increasingly rely on large-scale data collection and sophisticated machine learning models to derive actionable insights, optimize operations, and support data-driven decision-making. The rapid proliferation of connected devices, cloud platforms, and intelligent systems has significantly increased the volume, velocity, and variety of data being generated. While this growth presents unprecedented opportunities for innovation, it simultaneously introduces complex challenges related to data governance, privacy preservation, security, and regulatory compliance.

As data becomes more distributed and sensitive in nature, governments and regulatory bodies worldwide have introduced stringent data protection frameworks to safeguard individual privacy and ensure responsible data usage. Regulations such as the General Data Protection Regulation (GDPR) in the European Union, the Health Insurance Portability and Accountability Act (HIPAA) in the United States, and emerging data protection laws in other regions impose strict requirements on how data is collected, stored, processed, shared, and audited. These regulations emphasize principles such as data minimization, consent management, accountability, and transparency, compelling organizations to adopt robust mechanisms for tracking data provenance and enforcing compliance. Failure to meet these regulatory obligations can result in severe legal penalties, financial losses, and reputational damage.

Traditional centralized machine learning architectures have long been the dominant approach for building predictive models and intelligent systems. In such architectures, data from multiple sources is aggregated into a central repository where model training and

management, it presents significant drawbacks in modern data ecosystems. Centralized data aggregation increases the risk of privacy leakage, unauthorized access, and large-scale data breaches, as a single point of failure becomes an attractive target for malicious actors. Moreover, centralized systems often conflict with organizational data-sharing policies and regulatory requirements that restrict cross-border data movement or prohibit sharing sensitive information altogether.

In response to these limitations, Federated Learning (FL) has emerged as a transformative paradigm for distributed machine learning. Federated Learning enables multiple participants or organizations to collaboratively train a shared global model without exchanging raw data. Instead, each participant performs model training locally on its own data and shares only model updates, such as gradients or weights, with a central aggregator or coordination server. These updates are then combined to improve the global model, which is redistributed to participants for subsequent training rounds. By keeping sensitive data within local environments, Federated Learning significantly reduces privacy risks and aligns more closely with regulatory principles that emphasize data locality and ownership.

Despite its advantages, Federated Learning alone does not fully address challenges related to trust, accountability, and compliance in collaborative environments. In multi-organizational settings, participants may have limited visibility into how model updates are generated, aggregated, or validated. There is also a need for verifiable records that demonstrate compliance with data-handling policies, training protocols, and regulatory requirements. Without a reliable mechanism for auditability and transparency, disputes may arise regarding data usage, contribution fairness, or model integrity.

Blockchain technology offers a complementary solution to these challenges by providing a decentralized, tamper-proof, and transparent ledger for recording transactions and events. Blockchains are designed to ensure immutability, meaning that once data is recorded on the ledger, it cannot be altered without consensus from the network. This property makes Blockchain particularly suitable for maintaining audit trails, verifying data provenance, and establishing trust among participants who may not fully trust one another. Smart contracts—

self-executing programs deployed on the Blockchain—can further automate policy enforcement, validation rules, and compliance checks without relying on centralized authorities.

The integration of Federated Learning with Blockchain technology enables the creation of a secure and compliant data pipeline that combines privacy-preserving model training with transparent and accountable governance. In such an integrated framework, Federated Learning handles decentralized model training while Blockchain records immutable proofs of data sourcing, model updates, access events, and policy enforcement actions. Each training round, participant contribution, and aggregation step can be logged on the Blockchain, creating a verifiable history of the entire learning process. This ensures that all stakeholders can audit the system and verify that regulatory and organizational policies have been consistently followed.

This project explores the design and implementation of a Federated Learning and Blockchain-based data pipeline aimed at ensuring compliance in data sourcing and collaborative model development. The proposed architecture emphasizes decentralized control, privacy preservation, data integrity, and transparency across distributed machine learning environments. By leveraging Blockchain-based smart contracts, the system enforces predefined data-handling rules, validates model updates, and ensures that only authorized participants can contribute to or access the learning process. These mechanisms provide strong guarantees of accountability while reducing reliance on centralized trust intermediaries.

In addition to addressing compliance and governance challenges, the proposed framework supports scalability and flexibility across heterogeneous data sources and organizational boundaries. Participants can join or leave the network without compromising overall system integrity, and models can be updated continuously as new data becomes available. The decentralized nature of the pipeline also enables organizations to retain full control over their data assets, fostering greater willingness to participate in collaborative learning initiatives.

Overall, this project demonstrates that the convergence of Federated Learning and Blockchain represents a powerful foundation for next-generation, privacy-aware artificial intelligence systems. By aligning technical innovation with regulatory requirements and ethical data practices, the proposed data pipeline provides a sustainable and trustworthy approach to collaborative machine learning. Such a framework has broad applicability across domains including healthcare, finance, smart cities, and supply chain management, and lays the groundwork for future decentralized data governance models that balance innovation with compliance and accountability.

1.1 Motivation

The following are the motivation for the project:-

- **Need for a Privacy-Preserving and Compliant Learning Framework:** The increasing enforcement of global data protection regulations necessitates a shift toward decentralized learning paradigms that allow organizations to collaborate on model development without sharing raw data. Designing a framework that preserves data ownership while enabling collective intelligence is critical for sustainable and compliant AI systems.

- **Demand for Transparent and Trustworthy Data Collaboration:** Modern data ecosystems require mechanisms that not only protect privacy but also ensure transparency, traceability, and trust among participating entities. Integrating decentralized technologies to create auditable, regulation-aligned data pipelines motivates the development of a secure and accountable infrastructure for distributed machine learning

1.2 Problem Statement

Centralized data-driven machine learning architectures pose significant challenges in modern regulatory and multi-organizational environments. Despite their widespread adoption, these models are increasingly incompatible with strict data protection and compliance requirements. The key problem dimensions are as follows:

- **Regulatory Non-Compliance Due to Uncontrolled Data Movement:**

Centralized data models require aggregating sensitive data from multiple sources into a single repository, often involving cross-border data transfers and third-party access. Such uncontrolled data movement directly conflicts with regulations like GDPR and HIPAA, which mandate data minimization, locality, and explicit consent, thereby increasing the risk of regulatory violations and legal liabilities.

- **Privacy and Security Risks in Centralized Architectures:** Centralized storage creates a single point of failure, making systems highly vulnerable to data breaches, insider threats, and unauthorized access. As sensitive personal, financial, and medical data are consolidated in one location, any compromise can result in large-scale privacy leakage, undermining trust and violating confidentiality requirements.
- **Lack of Transparency and Accountability in Data Usage:** Existing centralized systems provide limited visibility into how data is accessed, processed, and used for model training. The absence of immutable audit trails and verifiable accountability mechanisms makes it difficult to demonstrate compliance, resolve disputes, or trace data provenance, particularly in collaborative and multi-stakeholder machine learning environments.

1.3 Objective

The following are the objectives of the project:-

- The primary objective of this project is to design and develop a Federated Learning and Blockchain-based Data Pipeline that ensures compliance, transparency, and privacy in data sourcing and usage within distributed machine learning environments. The specific objectives of the project are outlined as follows.
- To enable decentralized data contribution by allowing individuals and organizations to participate directly from their local data sources without relying on a centralized authority, thereby preserving data ownership, autonomy, and privacy while reducing the risks associated with centralized storage.

- To ensure data traceability and transparency by leveraging Blockchain technology to maintain an immutable and tamper-proof ledger that records all data transactions, model updates, and access events, making it possible to verify data provenance and prevent unauthorized modifications.
- To design a fair and incentive-based reward mechanism that evaluates the quality and relevance of contributed data and model updates, ensuring that participants are rewarded proportionally and encouraging sustained, honest, and high-quality contributions across the federated network.
- To implement on-chain verification and smart contract auditing mechanisms that automatically enforce compliance rules, validate data contributions and model updates, and ensure that all operations adhere to predefined policies in a transparent and verifiable manner.
- To preserve data privacy during collaborative model training by ensuring that raw data never leaves the local environment and by minimizing information leakage through secure aggregation and controlled parameter sharing.
- To enhance system scalability and robustness by supporting dynamic participation of multiple clients, enabling efficient handling of large-scale distributed learning without compromising performance or security.
- To ensure regulatory compliance with data protection standards such as GDPR and HIPAA by embedding compliance-aware policies directly into the data pipeline and smart contracts, allowing verifiable adherence to legal and ethical data-handling requirements.
- To improve trust and accountability among participating entities by providing verifiable audit trails and transparent governance mechanisms that reduce reliance on centralized intermediaries.

1.4 Organization of the report

Chapter 1: **Introduction** This chapter introduces the background and motivation for the project, emphasizing the growing challenges of data privacy, regulatory compliance, and trust in centralized machine learning systems. It presents the problem statement, objectives, and

significance of integrating federated learning with blockchain technology to enable privacy.

Chapter 2: **Literature Survey** This chapter reviews existing research in federated learning, decentralized machine learning, blockchain-based data governance, auditability mechanisms, and compliance enforcement. It examines key methodologies, identifies limitations in current approaches, and highlights research gaps related to scalability, transparency, and verifiable compliance that motivate the proposed system..

Chapter 3: **System Requirements and Specifications** This chapter details the hardware and software requirements necessary for implementing the federated learning framework and blockchain infrastructure. It specifies the datasets, computational resources, libraries, development tools, and blockchain environments used, along with functional and non-functional requirements that guide system design and deployment

Chapter 4: **System Analysis and Design** This chapter presents the architectural design of the proposed system, including the federated learning workflow, client–server interactions, blockchain integration model, and smart contract mechanisms. It includes system architecture diagrams, component descriptions, data flow diagrams, and design rationales that collectively serve as a blueprint for implementation

Chapter 5: **Methodology and Implementation** This chapter describes the step-by-step methodology used to implement the system, covering data preprocessing, federated client configuration, distributed model training, blockchain deployment, and smart contract automation. It discusses implementation strategies, algorithms, workflow design, and practical challenges encountered during integration.

Chapter 6: **Testing** This chapter outlines the testing procedures employed to evaluate system functionality, federated model performance, blockchain transaction reliability, and compliance enforcement. It includes descriptions of unit testing, integration testing, performance benchmarking, and the experimental setup used to simulate distributed client environments

Chapter 7: **Discussion of Results** This chapter analyzes the experimental results, comparing centralized and federated learning performance and evaluating the transparency and compliance guarantees provided by the blockchain layer. It interprets evaluation metrics such as RMSE, MAE, and R^2, and discusses the impact of decentralization on model accuracy, fairness, accountability, and trust

Chapter 8: **Conclusion and Future Work** This chapter summarizes the key contributions and findings of the project, demonstrating how the proposed system addresses the limitations of traditional centralized learning architectures. It also outlines future research directions, including enhancements in scalability, privacy preservation, advanced cryptographic techniques, and support for heterogeneous models.

References and Appendix This section includes all cited research papers, standards, and technical resources, along with supplementary materials such as architectural diagrams, configuration details, datasets, code snippets, and additional experimental result

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

The rapid advancement of data-driven technologies has fundamentally transformed how organizations collect, process, and utilize data for analytics and artificial intelligence. While these technologies enable powerful insights and automation, they have also introduced significant challenges related to data privacy, security, governance, and regulatory compliance. As sensitive data such as personal identifiers, financial records, and medical information are increasingly used for machine learning, ensuring responsible and lawful data handling has become a critical concern across industries.

Traditional centralized data systems, which require aggregating data from multiple sources into a single repository, expose sensitive information to heightened risks of data breaches, unauthorized access, and misuse. Such architectures often conflict with modern data protection regulations, including GDPR and HIPAA, which impose strict requirements on data locality, consent management, accountability, and transparency. These limitations have driven the search for decentralized alternatives that can support collaborative intelligence without compromising data privacy or regulatory compliance.

Federated Learning (FL) has emerged as a promising paradigm to address these challenges by enabling distributed model training without transferring raw data to a central server. In FL, participating clients perform local training on their private datasets and share only model updates with a coordinating entity. This approach preserves data privacy, reduces communication of sensitive information, and aligns with regulatory principles such as data minimization and ownership retention. However, despite its privacy advantages, Federated Learning alone does not inherently provide strong guarantees of transparency, auditability, or trust among participating entities.

Blockchain technology complements Federated Learning by offering a decentralized, tamper-proof, and transparent ledger that records transactions in an immutable manner. Through decentralized consensus mechanisms and cryptographic security, Blockchain ensures data integrity, accountability, and traceability without relying on centralized intermediaries. When applied to collaborative learning environments, Blockchain can record model updates, enforce access control policies, and provide verifiable audit trails that demonstrate compliance with regulatory and organizational requirements.

The integration of Federated Learning and Blockchain represents a novel and powerful approach for enabling compliant data sharing, trustworthy model updates, and secure multi-party collaboration. Recent research in this domain has explored communication-efficient federated learning protocols, privacy-preserving mechanisms such as secure aggregation and differential privacy, blockchain-assisted model aggregation, and incentive-based frameworks that encourage honest participation. These studies collectively highlight the potential of combining FL and Blockchain to overcome the limitations of centralized systems while ensuring compliance and traceability in distributed artificial intelligence environments.

2.2 Literature Review

- **Communication-Efficient Learning of Deep Networks from Decentralized Data H. B. McMahan et al.** introduced the concept of Federated Averaging (FedAvg), a foundational optimization algorithm for Federated Learning that enables training deep neural networks across decentralized devices without sharing raw local datasets. The study proposed communication-efficient techniques that significantly reduce bandwidth consumption while maintaining model accuracy. This work established the core principles of modern federated learning systems and remains widely adopted in both academic research and large-scale industrial deployments [1**]**

- **Advances and Open Problems in Federated Learning P. Kairouz et al.** presented a comprehensive survey of Federated Learning, identifying key

challenges such as non-IID data distributions, system and data heterogeneity, secure aggregation, fairness, and robustness against adversarial clients. The paper provides a structured taxonomy of FL problems and solutions and outlines open research directions, serving as a critical roadmap for the development of scalable and trustworthy federated systems [2]

- **Federated Machine Learning: Concept and Applications Q. Yang et al.** provided a conceptual and architectural overview of Federated Learning and explored its real-world applications in healthcare, finance, and mobile systems. The study categorized FL into horizontal, vertical, and transfer learning settings and discussed privacy-preserving mechanisms and secure aggregation protocols. This work serves as a foundational reference for understanding FL design choices and deployment strategies in regulated environments [3]

- **Federated Learning: Challenges, Methods, and Future Directions T. Li et al.** examined fundamental challenges in Federated Learning, including non-IID data, client heterogeneity, communication bottlenecks, and model divergence. The authors proposed optimization-based aggregation methods to improve convergence speed, fairness, and learning stability. The study provides practical guidance for building robust and scalable FL systems under realistic distributed conditions [4].

- **Differentially Private Federated Learning via Local Differential Privacy (LDP-Fed) Z.** Truex et al. proposed an FL framework based on Local Differential Privacy, where client updates are perturbed before transmission to ensure privacy even from the central server. This approach mitigates inference attacks such as model inversion and membership inference, making it particularly relevant for privacy-sensitive domains requiring strict regulatory compliance [5].

- **Practical Secure Aggregation for Federated Learning on User-Held Data K. Bonawitz et al**. introduced a secure aggregation protocol that enables clients to submit encrypted model updates while allowing the server to access only the aggregated result. This work ensures confidentiality of individual contributions,

enhances trust among participants, and demonstrates practical deployment strategies that balance security, efficiency, and scalability in federated systems [6].

- **Federated Learning for Mobile Keyboard Prediction A**. Hard et al. demonstrated the feasibility of large-scale Federated Learning through the deployment of FL for Google's Gboard keyboard prediction system. The study validated that decentralized training can achieve competitive performance while preserving user privacy. It also introduced system-level optimizations such as client sampling and communication-efficient updates, establishing a benchmark for real-world FL deployment [7]

- **Privacy-Preserving Transfer Learning: A Survey Y. Liu and Q. Yang** surveyed privacy-preserving transfer learning techniques that enable secure knowledge sharing across distributed domains. The study complements Federated Learning by addressing cross-domain collaboration scenarios and highlights applications in regulated industries such as healthcare and finance, emphasizing compliance, confidentiality, and robustness against privacy attacks [8].

- **FedDANE: Federated Optimization with DANE-Style Local Solvers X. Li et al.** proposed FedDANE, an optimization framework that employs local DANE-style solvers to improve convergence in heterogeneous federated environments. By increasing local computation and reducing global communication rounds, the method enhances efficiency and provides theoretical convergence guarantees, particularly for non-IID data settings [9].

- **Decentralizing Privacy: Using Blockchain to Protect Personal Data G. Zyskind et al**. proposed a blockchain-based privacy framework that uses smart contracts to enforce consent and access control policies. The system provides an immutable audit trail for all data transactions, demonstrating how blockchain can support transparency, trust, and regulatory compliance in decentralized data ecosystems [10]

- **BeeFL: A Blockchain-Based Federated Learning Framework H. Xu et al.**

introduced BeeFL, a framework that integrates blockchain with Federated Learning to ensure traceability, accountability, and decentralized governance. Model updates are immutably recorded on the blockchain, enabling tamper-proof verification and incentive-based participation in multi-party learning environments [11]

- **The Future of Digital Health with Federated Learning N. Rieke et al.** investigated the application of Federated Learning in healthcare, enabling collaborative model training across institutions while preserving patient privacy. The study emphasized compliance with regulations such as HIPAA and GDPR and demonstrated practical deployment strategies for multi-institutional medical AI systems [12].

- **Federated Learning for Industrial IoT Applications: A Survey J. Zhou et al**. reviewed Federated Learning applications in Industrial IoT environments, addressing challenges such as scalability, device heterogeneity, security, and network constraints. The study highlights FL's potential to enable collaborative intelligence across industrial sites without exposing proprietary or sensitive operational data [13].

- **DAOs, Governance, and Organizational Models V. Buterin** discussed Decentralized Autonomous Organizations (DAOs) as governance frameworks enabled by blockchain technology. The principles of transparent decision-making, automated rule enforcement, and decentralized control are directly applicable to federated learning systems requiring distributed governance and fair participation [14].

- **An Overview of Blockchain Technology: Architecture, Consensus, and Trends Z. Zheng et al.** provided a detailed overview of blockchain architecture, consensus mechanisms, and emerging trends. This work supports the design of blockchain-integrated federated systems by explaining how immutability, decentralization, and consensus enhance trust and accountability [15].

- **Byzantine-Tolerant Gradient Descent for Distributed Learning P. Blanchard et al**. proposed robust aggregation techniques to tolerate Byzantine failures in distributed learning. These methods mitigate the impact of malicious or unreliable clients, ensuring convergence and stability in federated environments with untrusted participants [16].

- **Comprehensive Privacy Analysis of Deep Learning M. Nasr et al.** analyzed privacy vulnerabilities in deep learning, including membership inference and model inversion attacks. The study highlights that FL alone may not fully prevent information leakage, emphasizing the need for additional safeguards such as differential privacy and blockchain-based auditing [17].

- **Communication-Efficient Federated Learning: A Survey S. Zhang et al.** surveyed communication reduction techniques in FL, including gradient compression, quantization, and sparsification. The work provides design trade-offs between communication cost and model accuracy, guiding the development of scalable FL systems [18]

- **PipeChime: Federated and Incentive-Based Credit Scoring (2023)** PipeChime proposed an incentive-driven federated learning framework for credit scoring, where participants are rewarded based on contribution quality. The system demonstrates how FL combined with incentives can enable fair, transparent, and privacy-preserving collaboration in financial applications [19]

- **Federated Approaches for Financial Risk and Credit Scoring X. Meng et al.** developed federated methods for financial risk prediction, incorporating secure aggregation and privacy-preserving techniques to ensure regulatory compliance. The study demonstrates the feasibility of decentralized learning in highly regulated financial environments [20].

- **Reward and Incentive Mechanisms for Federated Systems P. Basu and S. Sharma** analyzed incentive and reward distribution mechanisms for federated systems, addressing challenges such as free-riding and unequal participation. The

ntegration of blockchain-based smart contracts ensures transparent, automated, and tamper-proof reward allocation, supporting long-term sustainability of federated networks [21].

2.3 Outcome of Literature review

The literature survey conducted for this project provides a comprehensive understanding of the existing research landscape surrounding federated learning, blockchain technology, decentralized data governance, and regulatory compliance in data-driven systems. The primary outcome of this survey is the identification of both the strengths and limitations of current approaches, as well as the recognition of critical research gaps that motivate the proposed Federated Learning and Blockchain-based Data Pipeline.

A key finding from the literature is that Federated Learning has emerged as a highly effective paradigm for privacy-preserving machine learning. Numerous studies demonstrate that FL enables collaborative model training across distributed clients while ensuring that sensitive raw data remains localized. This characteristic directly aligns with regulatory principles such as data minimization and data locality mandated by frameworks like GDPR and HIPAA. Research has shown that FL significantly reduces privacy risks compared to centralized learning models, particularly in sensitive domains such as healthcare and finance. However, the literature also highlights that FL alone does not inherently guarantee transparency, accountability, or verifiable compliance, as model updates and aggregation processes are typically managed by a central coordinator that may not be fully trusted by all participants.

The survey further reveals that existing Federated Learning frameworks primarily focus on improving model accuracy, communication efficiency, and convergence speed. Techniques such as secure aggregation, model compression, and client selection strategies have been extensively explored. While these contributions enhance the technical robustness of FL systems, they often overlook governance-related concerns, including auditability of model updates, traceability of data contributions, and enforcement of compliance policies. As a result, current FL solutions may struggle to

meet the stringent accountability and reporting requirements imposed by regulatory bodies.

From the perspective of Blockchain technology, the literature establishes its strong potential for enabling decentralized trust, immutable record-keeping, and transparent transaction management. Blockchain-based systems have been widely proposed for applications such as supply chain tracking, identity management, and secure data sharing. In the context of machine learning, several studies explore the use of blockchain to record model updates, manage access control, and incentivize participant contributions. These works demonstrate that blockchain can effectively eliminate single points of failure and provide verifiable audit trails, which are essential for compliance and accountability.

Despite these advantages, the survey identifies that many blockchain-based machine learning solutions suffer from scalability and performance limitations. Public blockchains, in particular, introduce latency, high transaction costs, and limited throughput, making them unsuitable for high-frequency model update logging. Some studies address these issues through permissioned blockchains or off-chain storage mechanisms; however, standardized design patterns for integrating blockchain with federated learning remain limited. This lack of cohesive architectural guidance presents a challenge for practical deployment.

An important outcome of the literature review is the observation that existing research efforts often treat Federated Learning and Blockchain as independent solutions rather than complementary technologies. While a few recent works attempt to integrate the two, most focus on isolated aspects such as secure aggregation or incentive mechanisms, without addressing end-to-end compliance in data sourcing and model lifecycle management. In particular, there is limited research on how smart contracts can be systematically used to enforce data-handling policies, validate model updates, and automate compliance verification in federated environments.

The survey also highlights a significant gap in addressing data imbalance and privacy leakage risks in distributed learning scenarios. Although Federated Learning reduces raw

data exposure, model updates can still leak sensitive information through inference attacks. Only a small subset of studies explore the use of synthetic data generation or advanced privacy-enhancing techniques alongside FL. This gap underscores the need for complementary mechanisms, such as synthetic data generation using Generative Adversarial Networks, to improve data diversity while further mitigating privacy risks.

Another notable outcome of the literature survey is the lack of comprehensive evaluation frameworks that jointly assess model performance, system scalability, and regulatory compliance. Most studies evaluate success primarily through predictive accuracy metrics, with minimal consideration of compliance guarantees, auditability, or policy enforcement effectiveness. This imbalance indicates a need for holistic evaluation approaches that consider both technical and governance dimensions of decentralized learning systems.

Based on these findings, the literature survey clearly establishes the motivation for the proposed work. There is a demonstrated need for an integrated framework that combines the privacy-preserving strengths of Federated Learning with the transparency and immutability of Blockchain technology. Such a framework should not only support decentralized model training but also provide verifiable compliance, traceability of contributions, and automated policy enforcement

CHAPTER 3

SYSTEM REQUIREMENTS AND SPECIFICATION

3.1 Hardware requirement

- Processor: Intel Core i5
- Storage: 500 GB HDD
- RAM (Random access memory)
- Graphics: NVIDIA GPU
- Deployment Setup: Distributed client nodes, local Spark processing, and blockchain test network

3.2 Software requirement

- Programming Language
- Smart Contract Language (Solidity)
- Blockchain Development Tools (Ganache, Remix IDE, Truffle Suite)
- Synthetic Data Generation Model (GAN-based models)
- Data Analysis Libraries (NumPy, Pandas, Scikit-learn)
- Federated Learning Framework (TensorFlow Federated / PySyft)
- Deep Learning Framework (TensorFlow / PyTorch)
- Blockchain Platform (Ethereum)
- Distributed Data Processing Framework (Apache Spark)

3.3 Tools and Technologies Used

The following are the tools and technologies used in this project:-

- **Python:** Python is used as the primary development language due to its simplicity, readability, and extensive ecosystem of libraries and frameworks. It offers strong support for machine learning, federated learning, data analytics, and blockchain integration, making it well suited for building end-to-end distributed systems. Python enables rapid prototyping and experimentation, allowing models and system components to be developed, tested, and refined efficiently. Its compatibility with major ML frameworks, federated learning libraries, and blockchain APIs facilitates seamless integration between data processing, model training, smart contract interaction, and system orchestration. Additionally, Python's support for distributed computing, scripting, and automation simplifies the implementation of complex workflows, enhances maintainability, and improves overall development productivity in the proposed system.
- **TensorFlow Federated / PySyft:** TensorFlow Federated and PySyft provide abstractions for implementing federated learning workflows such as client-side training, secure aggregation, and global model updates. These frameworks facilitate decentralized model training while ensuring that raw data remains at the source, thereby preserving data privacy and ownership.
- **Blockchain Platform – Ethereum:** Ethereum is used as the underlying blockchain platform to ensure transparency, immutability, and decentralized governance. It enables secure recording of model updates, data contribution logs, and compliance proofs through a tamper-proof distributed ledger.
- **Smart Contract Language – Solidity:** Solidity is used to develop smart contracts that define rules for data contribution, model update validation, incentive distribution, and compliance enforcement. These contracts execute automatically and eliminate the need for manual intervention or centralized control.
- **Distributed Data Processing – Apache Spark:** Apache Spark is utilized for large

- **PyTorch / TensorFlow:** Deep-learning frameworks like PyTorch or TensorFlow are essential for designing and training GAN architectures. They provide automatic differentiation, GPU acceleration, and efficient tensor computation capabilities. PyTorch is widely chosen for its dynamic computational graph and research-friendly workflow. TensorFlow offers strong production support and high-performance model deployment. Both frameworks support advanced GAN models such as DCGAN, Conditional GAN, and StyleGAN. Their visualization tools help track training progress and model behavior. These frameworks make the implementation of complex GAN workflows feasible and efficient.
- **NumPy:** Pandas is utilized for structured data management and preprocessing at the client level. It supports loading, cleaning, organizing, and partitioning datasets sourced from different organizations. In this project, Pandas ensures consistent data formatting, manages compliance-related metadata, and prepares datasets for federated learning while maintaining local data ownership and regulatory constraints.
- **Pandas:** Pandas is employed for structured data handling and preprocessing. It is used to load, clean, transform, and organize datasets obtained from different client nodes. In this project, Pandas facilitates tasks such as missing value handling, data filtering, feature selection, and dataset partitioning for federated clients. It also assists in managing metadata, training logs, and intermediate results generated during federated training and evaluation.
- **Scikit Learn:** Scikit-learn is used for implementing preprocessing pipelines, baseline machine learning models, and evaluation metrics. It supports feature scaling, encoding, and performance evaluation using metrics such as RMSE, MAE, and R^2. In the proposed system, Scikit-learn enables comparison between centralized and federated learning models, validating the effectiveness of the compliance-aware, decentralized data pipeline.
- **Matplotlib & Seaborn:** Matplotlib and Seaborn are used to visualize training performance, convergence behavior, error metrics, and comparison results between centralized and federated learning models. Visualizations support interpretability and

PIL supports lightweight image loading and manipulation inside training pipelines. These libraries ensure every image is standardized before entering the training loop. Their role is critical for maintaining dataset quality and improving the generator's output fidelity.

- **Jupyter Notebook / Google Colab:** These platforms serve as the main development environment for experimentation. They support step-by-step execution, visualization, and debugging. Google Colab offers free GPU resources essential for training GANs efficiently. Notebooks allow interactive testing of latent space operations and insight generation. Visual outputs, graphs, and model samples can be displayed inline for analysis.
- **Git, GitHub:** Git and GitHub are used for source code management, experiment tracking, and collaborative development. Version control ensures reproducibility, accountability, and organized project evolution.

3.4 Functional Requirements

- **Collaborative Model Training Without Data Sharing:** The system must enable multiple distributed clients to collaboratively train a machine learning model without exchanging raw datasets. Each client performs local training on its own data, contributing model updates to the global model. This ensures that sensitive data remains private and compliant with regulations such as GDPR and HIPAA
- **Central Model Aggregation and Distribution:** The central federated server must aggregate the local model updates from all participating clients using aggregation algorithms such as Federated Averaging (FedAvg). After aggregation, the server redistributes the updated global model to all clients, maintaining synchronization across the network and supporting iterative training cycles.
- **Immutable Blockchain Logging:** The blockchain layer must record immutable logs of all critical events, including data contributions, model update transactions, and federated learning rounds. This ensures full traceability, accountability, and regulatory compliance by providing a tamper-proof record of all operations within the system.

- **Automated Smart Contract Execution:** Smart contracts must automatically handle reward distribution, validation of model updates, and DAO governance voting. They enforce predefined rules and policies without manual intervention, ensuring transparency, fairness, and adherence to governance protocols
- **Synthetic Data Generation:** Synthetic data modules must allow clients to generate artificial datasets for testing, validation, and augmentation without exposing real information. This supports privacy preservation, helps address data imbalance, and enables experimentation across diverse data modalities.
- **Secure Communication:** The system must support encrypted communication channels between clients, the central server, and blockchain nodes. Secure communication protocols prevent data interception, maintain confidentiality, and protect sensitive model updates during transmission.
- **Dynamic Client Participation:** The system must allow clients to dynamically join or leave the federated network without disrupting ongoing training processes. This ensures scalability, flexibility, and fault tolerance in distributed environments with heterogeneous participants.
- **Robustness Against Malicious Updates:** The system must detect and mitigate adversarial or corrupted model updates using robust aggregation and anomaly detection techniques. This ensures global model stability and reliability even when some clients behave maliciously or inconsistently..
- **Monitoring and Alerting:** The system must provide real-time monitoring dashboards and alert mechanisms to track federated learning progress, blockchain activity, and system health. Administrators should be notified of anomalies, failures, or unusual behavior, enabling timely intervention.
- **Configurable Training Parameters:** The system must allow administrators and clients to configure local and global training parameters, including learning rate, batch size, number of epochs, and optimization algorithms. This flexibility ensures that the federated learning process can be tailored to varying client data sizes, hardware capabilities, and model architectures, improving convergence speed, model accuracy, and overall system efficiency.

3.5 Non-Functional Requirements

- **Performance Requirements:** The system must efficiently process distributed datasets and perform federated model aggregation with minimal delays. GPU acceleration should be leveraged for local model training to reduce computation time. Efficient performance ensures that the federated learning pipeline can handle multiple clients and iterative training rounds without creating bottlenecks, which is critical for large-scale collaborative deployments..
- **Scalability:** The system should support adding more clients, larger datasets, and more complex models in the federated network. Scalability ensures that as organizations or participants join the network, the system can maintain performance and reliability. It also allows extending the pipeline to support advanced architectures or additional privacy-preserving techniques without redesigning the system.Scalability ensures the project can evolve into more complex research environments.
- **Reliability:** The system must ensure stable and repeatable results across federated learning rounds, even in case of network interruptions or client failures. Checkpointing and automatic recovery mechanisms allow training to resume without loss of progress. Reliability is crucial in distributed environments where client availability and network stability can vary.
- **Usability:** The interface, whether CLI or GUI, must be user-friendly to allow researchers, data scientists, and organizational participants to contribute data, monitor training, and view model performance easily. Usability ensures that federated learning workflows can be adopted by participants with minimal training, promoting wider participation in the decentralized network..
- **Maintainability:** The system must have a modular, well-documented codebase that allows debugging, updates, and future enhancements without affecting existing functionality. Maintainability is essential in a federated environment, where updates to client nodes, smart contracts, or model architecture may be frequent..
- **Security:** All client data, model updates, and blockchain records must be securely stored and transmitted using encryption and authentication protocols. Security is vital

in federated learning to protect sensitive organizational data and maintain compliance with GDPR, HIPAA, and other privacy regulations.

- **Portability:** The system must run across different platforms (Windows, Linux, macOS) and environments (local IDEs, Jupyter Notebook, cloud servers). Portability ensures that participating clients can join the federated network regardless of their hardware or operating system, enabling broad adoption.
- **Interoperability:** The system must integrate seamlessly with external libraries such as NumPy, Pandas, Scikit-learn, PyTorch, or TensorFlow. Interoperability ensures smooth preprocessing, model training, evaluation, and visualization across distributed clients and enables collaboration with existing machine learning pipelines
- **Accuracy:** The federated model must achieve predictive accuracy comparable to centralized models while maintaining data privacy. Accuracy ensures that decentralized training still produces scientifically meaningful results, validating the effectiveness of the federated approach for real-world applications.
- **Robustness:** The system must handle network interruptions, client dropouts, heterogeneous data distributions, and occasional noisy inputs without compromising global model stability. Robustness ensures that the federated learning pipeline remains functional under real-world operational conditions, maintaining trust and reliability across all participant

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

4.1 System Analysis

System analysis is a crucial phase presents a detailed analysis and design framework for the proposed Federated Learning and Blockchain-Based Data Pipeline aimed at ensuring compliance, privacy, and transparency in data sourcing. It establishes the foundation for understanding how the system operates across distributed clients, central servers, and the blockchain layer, while addressing the unique challenges of decentralized data management. The chapter begins with a feasibility study, evaluating economic, technical, and operational considerations to determine the viability of implementing the proposed architecture.

4.1.1 Feasibility Study: The project demonstrates strong economic, technical, and operational feasibility. It provides a cost-effective, secure, and scalable solution for collaborative data-driven intelligence while ensuring privacy, regulatory compliance, and accountability. The approach can be adapted across multiple industries, including finance, healthcare, and supply chain management, establishing a foundation for future decentralized data governance frameworks. Three key considerations involved in the feasibility analysis are as follows: -

- **Technical Feasibility**: From a technical feasibility perspective, the required tools and technologies are mature and widely supported. Federated learning frameworks enable secure aggregation and model update coordination, while blockchain provides immutable and transparent logging for auditability. Differential privacy, secure multiparty computation, and encryption mechanisms strengthen data protection, ensuring sensitive information remains confidential. The integration of incentive mechanisms allows fair contribution assessment and reward distribution, aligning with operational objectives. Overall, the technical stack supports all key components of a decentralized, compliant, and collaborative learning system, though careful design is needed to handle latency, heterogeneous client environments, and secure aggregation

- **Operational Feasibility**: In terms of operational feasibility, the system is practical for real-world deployment. Federated learning allows organizations to participate without moving sensitive data, maintaining autonomy over internal data governance while enabling collaborative model training. Blockchain ensures traceability and compliance, and modular monitoring tools facilitate management, troubleshooting, and auditing. While implementing and maintaining such a system requires expertise in machine learning, distributed systems, and blockchain, open-source support, documentation, and established frameworks mitigate these challenges. Additionally, the system is scalable, supporting additional clients, larger datasets, and more complex models as adoption grows..

- **Economic Feasibility**: The economic feasibility of the project is strong, as it leverages open-source tools like TensorFlow Federated, PySyft, and Ethereum smart contracts, reducing initial development costs. Federated learning minimizes the need for centralized data storage and infrastructure, while maintaining compliance with privacy regulations such as GDPR and HIPAA, thereby lowering the risk of legal penalties and associated expenses. The use of distributed computation also optimizes resource utilization, distributing processing across multiple clients and reducing server costs. Furthermore, the incentive-based mechanism encourages participation from multiple institutions, providing potential long-term value through collaborative model building.

4.2 System Architecture

The system architecture describes the structural design of the proposed system and explains how its components interact to enable decentralized credit score computation. It outlines the flow of information, the distribution of processing tasks, and the coordination mechanism required to ensure secure and efficient operation across multiple entities.

The architecture is designed to address the limitations of traditional centralized credit scoring systems by adopting a federated learning approach. This design allows collaborative

model training across multiple banks while preserving data privacy, as sensitive customer information remains within the local infrastructure of each participating bank.

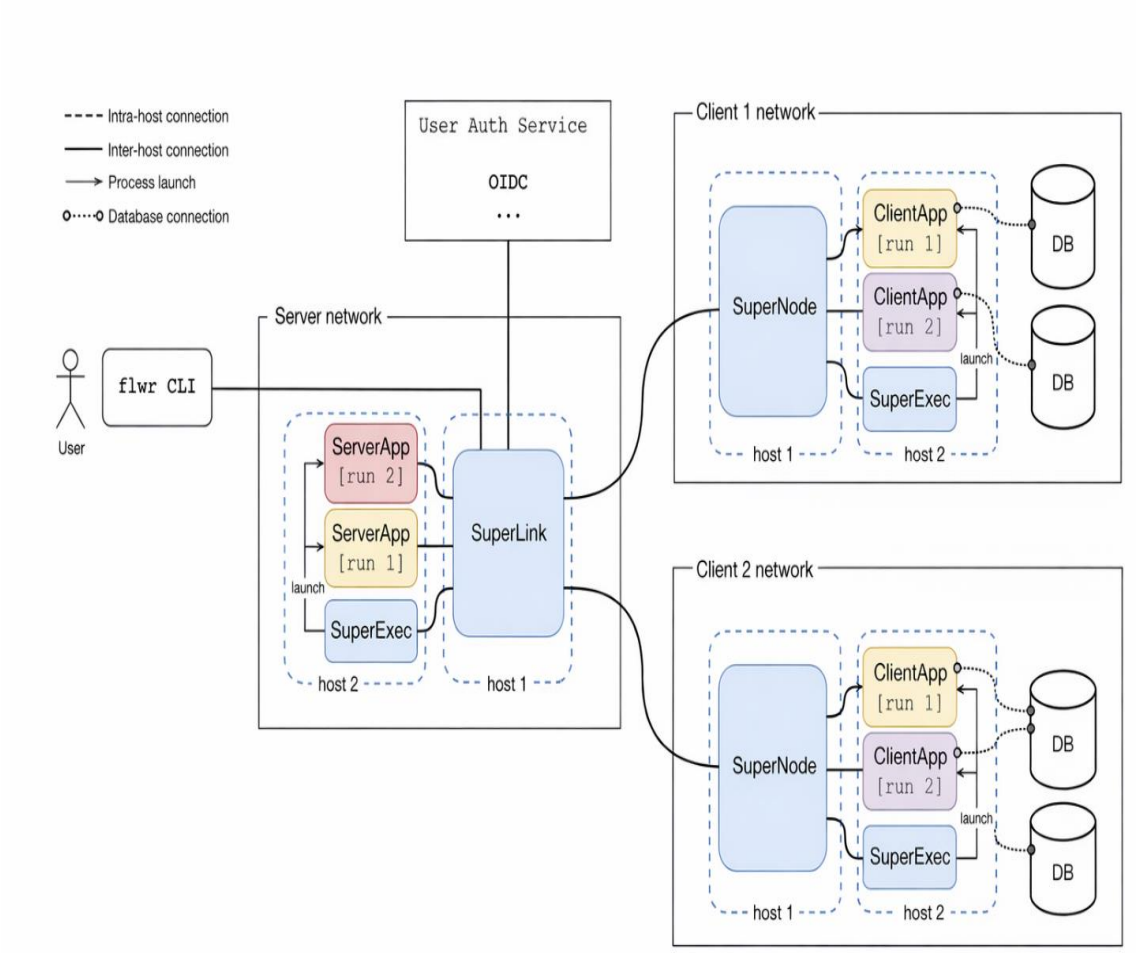

Fig 4.1: Federated and Decentralized Mechanism Workflow

The figure 4.1, presents a comprehensive system architecture for a distributed federated learning environment that spans a server network and multiple client networks, coordinated through secure communication and authentication mechanisms. The architecture is designed to support scalable execution, secure orchestration, and decentralized data processing while maintaining strict isolation between hosts, processes, and data stores. At the entry point of the system, a user interacts with the infrastructure through the flwr CLI, which acts as the command-line interface for initiating and controlling federated learning workflows.

4.3 Detailed Design

The system architecture represents a distributed federated learning environment that connects a central server network with multiple client networks through secure communication and authentication mechanisms. It is designed to support scalable execution, secure orchestration, and decentralized data processing while maintaining strict isolation between hosts, processes, and data stores. Users interact with the system through the flwr CLI to initiate and manage federated learning workflows within the server network.

Authentication and authorization are handled by a dedicated User Authentication Service using OpenID Connect (OIDC), ensuring that only authenticated and authorized entities can access system resources. This security layer establishes a trusted boundary between users, servers, and clients, which is essential for maintaining confidentiality, integrity, and reliability in a distributed and multi-tenant federated learning environment.

Within the server network, the architecture is divided into multiple hosts, each responsible for specific execution roles. One host runs the SuperLink component, which functions as the central coordination and communication hub of the system. The SuperLink establishes inter-host connections with client networks and manages the overall federated learning lifecycle, including client registration, task scheduling, and message routing. Another host in the server network executes multiple instances of the ServerApp, which represent the core federated learning server logic. These ServerApp instances are launched and managed by the SuperExec component, which is responsible for process execution, lifecycle management, and isolation. The presence of multiple ServerApp runs indicates support for parallel execution, fault tolerance, or repeated experiment runs. Intra-host connections are used for communication between SuperExec and the ServerApp processes, ensuring efficient local interaction without unnecessary network overhead.

The server-side components communicate with the client networks through secure inter-host connections established by the SuperLink. Each client network is logically isolated and represents an independent execution environment, such as an organization, institution, or

deployment site. Within each client network, the architecture again follows a multi-host design. One host runs the SuperNode, which serves as the primary interface between the client network and the central server network. The SuperNode handles incoming tasks, coordinates local execution, and communicates results back to the SuperLink. This design ensures that the central server does not directly interact with individual client applications, thereby improving scalability and security by enforcing a clear separation of responsibilities.

The second host within each client network is responsible for executing the actual client-side applications, represented as multiple instances of ClientApp. These ClientApp processes perform local computation tasks, such as training machine learning models on locally available data. Similar to the server side, the execution of ClientApp instances is managed by a Super Exec component, which launches and supervises multiple runs of the application. This enables parallelism, experiment repetition, or handling of multiple training rounds. The ClientApps interact with local databases through dedicated database connections, which are explicitly separated from network communication paths. This ensures that sensitive data stored in the databases remains confined to the client network and is never exposed directly to the server network.

The architecture clearly distinguishes between different types of connections to enforce clarity and security. Intra-host connections represent communication between processes running on the same machine, inter-host connections represent network communication across machines or networks, process launch arrows indicate execution control flow, and database connections illustrate data access boundaries. This explicit separation highlights the system's emphasis on modularity, isolation, and controlled interaction between components. By maintaining strict boundaries between server-side coordination, client-side execution, and data storage, the system minimizes attack surfaces and reduces the risk of unintended data leakage.

Overall, this architecture demonstrates a robust and scalable design for federated systems that require distributed execution, secure coordination, and strong data locality guarantees. The combination of centralized orchestration through the SuperLink, decentralized

execution via SuperNode and Client components, and controlled process management using Super Exec enables the system to efficiently manage complex federated workflows. At the same time, the integration of standardized authentication services and clearly defined communication pathways ensures that security and trust are maintained across all participating entities. This design is well suited for large-scale federated learning deployments, where multiple clients operate independently while contributing to a coordinated global objective without sharing raw data.

CHAPTER 5

METHODOLOGY AND IMPLEMENTATION

5.1 Methodology

In modern machine learning systems, the approach to training models has evolved significantly to address challenges such as data privacy, scalability, and computational efficiency. Two primary paradigms dominate this space: **Centralized Learning** and **Federated Learning**. These methodologies differ fundamentally in how data is managed, processed, and leveraged for model training.

Centralized Learning is the conventional approach in which all datasets from multiple sources are aggregated into a single, central server. The machine learning (ML) process operates directly on this consolidated data, training a model with access to the entire dataset. While this approach simplifies the training workflow and allows high accuracy due to access to comprehensive data, it suffers from critical limitations. These include privacy concerns, increased risk of data breaches, and the logistical challenges of transferring large datasets to a central location.

Federated Learning, in contrast, represents a decentralized approach designed to overcome these limitations. In this methodology, the ML process occurs locally at each data source, ensuring that raw data never leaves its origin. The central server coordinates model updates from each local process, aggregating them to form a global model. This strategy ensures data privacy, reduces network bandwidth consumption, and enables collaborative learning across distributed nodes without centralized data storage. Federated learning is particularly relevant in sensitive domains like healthcare, finance, and personal devices, where privacy and regulatory compliance are critical.

The methodology illustrated in this study combines both paradigms, enabling a comparative understanding of their workflows, strengths, and limitations. The subsequent figure provides

a visual representation of both centralized and federated learning architectures, highlighting the flow of data and model processes in each case.

5.11 Federated and Central Learning

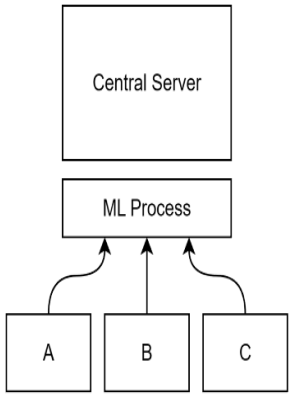

Fig 5.1(a) Traditional way of collecting data

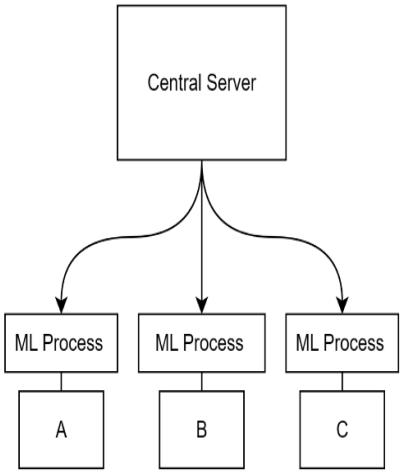

Fig 5.1(b) Federated way of collecting data

Centralized Learning - In the centralized learnin paradigm, data from multiple sources (A, B, and C) is collected and transferred to a central server, where a single machine learning process operates on the combined dataset. This approach allows the model to access the complete data, often resulting in high accuracy and simplified model management, as training occurs in a single location with uniform feature processing. However, centralized learning comes with significant limitations. Sensitive data must be moved from its original location, raising privacy and compliance concerns. Additionally, transferring large datasets can create network overhead and latency, and scalability becomes challenging when datasets are extremely large. While straightforward, centralized learning is less suitable for scenarios where data privacy and regulatory compliance are critical as shown in figure 5.1 (a) .

Federated Learning Federated learning, in contrast, retains raw data locally at each source (A, B, and C), and each node trains its own model independently. Instead of sending raw data to a central server, only model updates—such as parameters or gradients—are transmitted for

aggregation into a global model. This distributed approach preserves data privacy, reduces network bandwidth usage, and facilitates compliance with regulations such as GDPR or HIPAA. However, it introduces communication overhead due to frequent updates and requires careful aggregation techniques to ensure convergence and model accuracy. Federated learning is especially beneficial when multiple autonomous data owners wish to collaboratively train models without compromising their private datasets, though performance may vary if data distributions across sources are heterogeneous as shown figure 5.1(b)**.**

Table 5.1 Difference between Traditional and Federated learning

Feature/Aspects	**Traditional Learning**	**Federated Learning**
Data Location	Data from all sources (A, B, C) is collected at a central server.	Data remains local at each source (A, B, C). Only model updates are sent to the server
ML Process	Single ML process runs on the combined dataset at the central server	Multiple ML processes run locally at each node; central server aggregates updates.
Privacy	Lower, as sensitive data leaves the local environment.	Higher, raw data never leaves local nodes.
Network Usage	High, due to transferring all raw data to the server.	Low, only model parameters or gradients are transmitted.
Scalability	Challenging with very large datasets.	More scalable across distributed sources.

5.2 Implementation

The implementation outlines the systematic approach adopted to design, implement, and evaluate the proposed system for decentralized and privacy-preserving learning. It defines the sequence of processes, architectural choices, and computational mechanisms used to transition from traditional centralized learning models to a federated and distributed framework. The primary objective of this methodology is to enable collaborative model training and decision-making across multiple entities while ensuring data confidentiality, security, and regulatory compliance. By structuring the workflow into well-defined stages, the methodology ensures reproducibility, scalability, and robustness of the system.

Traditional centralized learning approaches require aggregating data from multiple sources into a single server for model training. While effective in terms of computation, such approaches introduce significant challenges related to data privacy, security risks, and governance. In sensitive domains such as finance and healthcare, centralized data collection often conflicts with regulatory constraints and organizational policies. To address these limitations, the proposed methodology adopts a federated learning paradigm, where data remains localized at the source and only learned model parameters or updates are shared. This shift reduces the risk of data leakage while still enabling the system to benefit from distributed knowledge.

To ensure secure coordination and controlled participation, authentication and access control mechanisms are integrated into the methodology. Standardized identity management protocols are used to verify participating entities before allowing them to contribute to the learning process. In addition, incentive and monitoring mechanisms are incorporated to encourage honest participation and maintain the quality of model contributions. These mechanisms play a critical role in sustaining collaboration among independent entities operating in a shared learning environment.

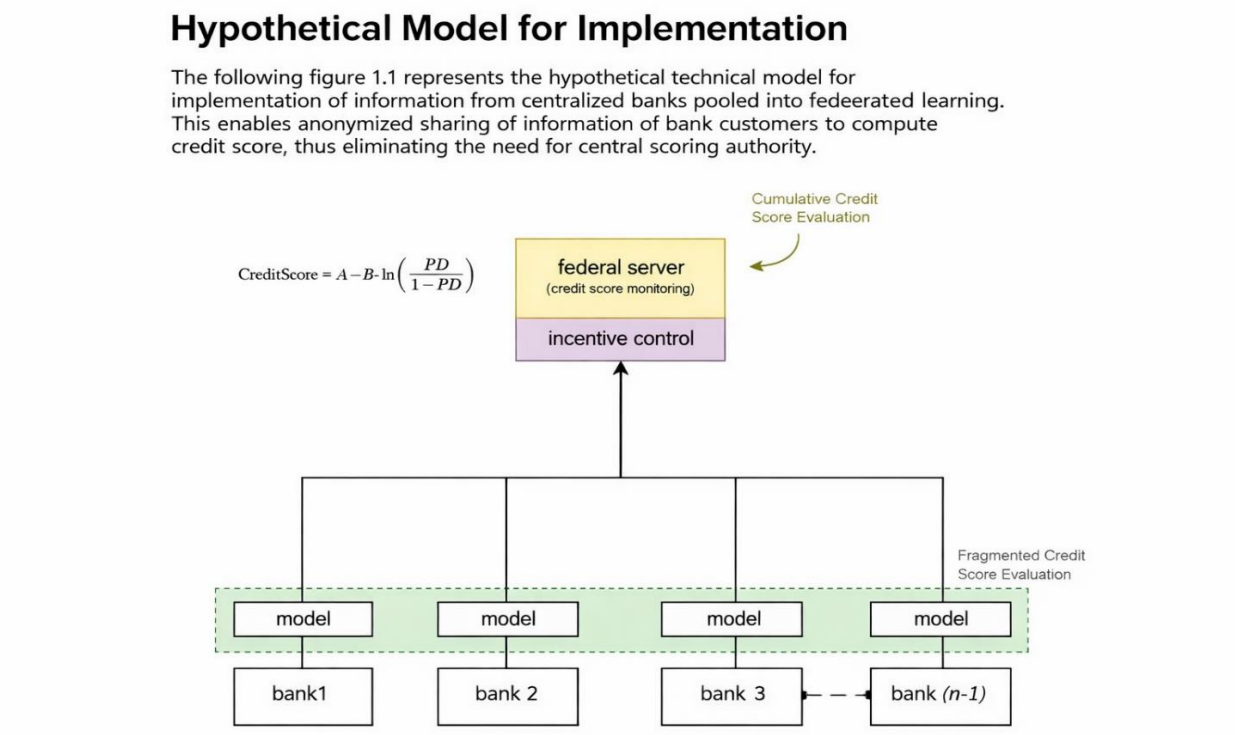

Fig 5.2 Federated & Incentive based credit score computing model

Figure 5.2 illustrates a federated and incentive-based credit score computation architecture designed to enable collaborative credit risk assessment without centralized data sharing.

In this architecture, multiple banks participate as independent entities, each maintaining its own private customer dataset. At the bank level, local machine learning models are trained using internal data to compute credit-related metrics such as probability of default. These local models perform fragmented credit score evaluation, ensuring that sensitive customer information never leaves the bank's infrastructure.

The locally trained models or their parameter updates are then shared within a federated learning framework. Instead of transmitting raw data, only model updates are communicated, preserving privacy and ensuring regulatory compliance. This federated layer securely aggregates knowledge from all participating banks while maintaining strict data isolation.

At the top level, a federal server coordinates the learning process. Its role is limited to aggregating the received model updates and generating a cumulative credit score

evaluation using federated aggregation techniques. The server does not store or access individual customer records. An incentive control mechanism is integrated into the federal server to promote fair participation, encourage high-quality contributions, and prevent dishonest or low-effort behavior among participants.

5.21 Peer-to-Peer (P2P) Data or Message Processing System

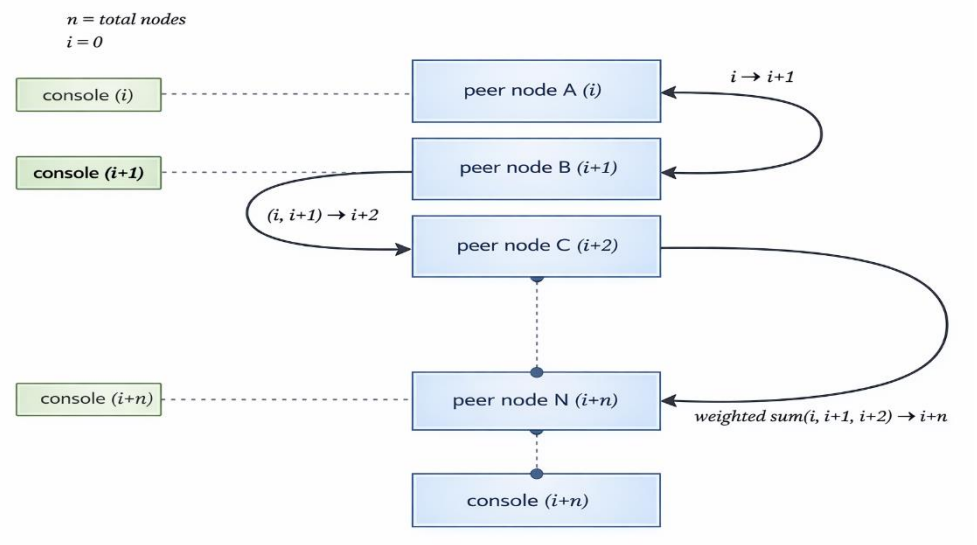

Fig 5.21 Peer-to-Peer Data processing system

- **Notation and Setup :** Fig 5.21 represents n = total nodes: Indicates the total number of peer nodes in the system.**,**i = 0: Represents the initial index or counter used to iterate over the nodes. Each node is represented as a peer node:

  Peer Node A (i)

  Peer Node B (i+1)

  Peer Node C (i+2)……up to Peer Node N (i+n)

  The consoles (on the left side) represent input/output terminals for monitoring or controlling the nodes:

  console(i) → interacts with Peer Node A

  console(i+1) → interacts with Peer Node B

  console(i+n) → interacts with Peer Node N

- **Flow of Operations**

i. **Step 1: Starting with i = 0**

The iteration starts from node i.

Node i (Peer Node A) is activated first.

After processing, i is incremented by 1: i → i+1.

ii. **Step 2: Moving to Next Node**

Peer Node B (i+1) receives the next index.

Processing at Node B is done with data or a message from Node A.

i is again incremented: i+1 → i+2.

iii. **Step 3: Iterative Propagation**

Peer Node C (i+2) continues this process.

The process keeps moving forward, incrementing i until it reaches the last node (i+n)

- **Weighted Sum at the End**

The last node performs a **weighted sum** of previous nodes' outputs:

weighted sum (i, i+1, i+2) → i-n

The outputs from Peer Node i, i+1, and i+2 are combined using a weighting function. The combined result is sent back or stored at a node indexed i-n, which often represents an aggregation node.

CHAPTER 6

TESTING

Model Evaluation and Testing

The objective of this section is to rigorously assess the predictive performance of the proposed federated learning pipeline in comparison to a traditional centralized approach. Evaluating models in both centralized and federated settings provides insight into the trade-offs between predictive accuracy and data privacy.

In the centralized approach, all available data is pooled and used to train a single global model, representing the ideal scenario where data access is unrestricted. Conversely, the federated approach simulates a real-world scenario in which data resides across multiple distributed client nodes, and only model parameters are shared for aggregation. This setup preserves data privacy and security while enabling collaborative learning.

Both models were evaluated on the same held-out test set to ensure a fair compari- son. Performance was measured using standard regression metrics, including Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the Coefficient of Determina- tion (R^2). These metrics provide a quantitative assessment of prediction accuracy and allow for direct comparison between the centralized and federated training paradigms.

6.1 Centralized Model Prediction Workflow

The prediction workflow for the centralized approach is depicted in Figure 5.1. In this setup, all dataset partitions are merged into a single combined dataset. A linear regres- sion model is then trained on this aggregated data, and the resulting model is used to generate predictions for the test set. This approach provides a benchmark for predictive performance, as it leverages all available data in a single location, allowing the model to learn from the complete dataset without the constraints of distributed training.

Predictions with Central Approach

```
]: import pandas as pd
   import numpy as np
   import matplotlib.pyplot as plt
   from sklearn.model_selection import train_test_split
   from sklearn.linear_model import LinearRegression
```

```
]: df1 = pd.read_csv("partitions-3/partition_0.csv")
   df2 = pd.read_csv("partitions-3/partition_1.csv")
   df3 = pd.read_csv("partitions-3/partition_2.csv")
   df = pd.concat([df1, df2, df3], ignore_index=True)
```

```
]: y = df["MedHouseVal"]
   X = df.drop("MedHouseVal", axis=1)
```

```
]: X_train, X_test, y_train, y_test = train_test_split(
       X, y, test_size=0.2, random_state=42
   )
```

```
]: model = LinearRegression()
   model.fit(X_train, y_train)

   central_prediction = model.predict(X_test)
```

```
]: print(central_prediction)
```

```
[2.88511032 2.61514647 1.89141227 ... 1.09157008 2.38953537 2.29702716]
```

Fig 6.1: Centralized learning workflow showing dataset merging, model training, and prediction generation

6.2 Federated Model Prediction Workflow

The prediction workflow for the federated learning approach is illustrated in Figure 6.2. Unlike centralized training, where all data is combined, each client in the federated setup performs model training locally using its own data. Only the learned model parameters, such as weights and intercepts, are shared with the central server for aggregation. Once the global aggregation is complete, the aggregated model parameters are used to generate.

Predictions with Federated Approach

```
# Predict X_test values using coefs and interecept parameters recieved via federated training
params = np.load(r"C:\Users\vigne\VSCODE PRIME\pchime_analytics\federated_results\pchime_fl_round_5_params.npz")
coef = params["coef"]
intercept = params["intercept"][0]
```

```
federated_predictions = np.dot(X_test, coef) + intercept
print(federated_predictions)
```

```
[2.90093005 2.61325059 1.88683167 ... 1.08920798 2.38373121 2.29225026]
```

Fig 6.2 Federated prediction workflow using aggregated global model coefficients obtained from federated training**.**

6.3 Evaluation of Prediction Score

The performance of both the centralized and federated prediction pipelines was assessed using standard regression metrics, including Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2). Figure 5.3 illustrates the complete evaluation results, enabling a direct comparison between the two approaches.

Central Predictions Scores

```
from sklearn.metrics import mean_absolute_error, root_mean_squared_error, r2_score
rmse_c = root_mean_squared_error(y_test, central_prediction)
mae_c = mean_absolute_error(y_test, central_prediction)

r2_c = r2_score(y_test, central_prediction)

print(f"rmse: {rmse_c} | mae: {mae_c} | r2: {r2_c}")
```

```
rmse: 0.7235247287812572 | mae: 0.5308013178201441 | r2: 0.6090988648591609
```

Federated Predictions Scores

```
from sklearn.metrics import mean_absolute_error, root_mean_squared_error, r2_score
rmse_f = root_mean_squared_error(y_test, federated_predictions)
mae_f = mean_absolute_error(y_test, federated_predictions)

r2_f = r2_score(y_test, federated_predictions)

print(f"rmse: {rmse_f} | mae: {mae_f} | r2: {r2_f}")
```

```
rmse: 0.715908047410847 | mae: 0.5243906315858677 | r2: 0.6172857245783183
```

Verdict

```
print(f"Central results :: rmse: {rmse_c} | mae: {mae_c} | r2: {r2_c}")
print(f"Federated results :: rmse: {rmse_f} | mae: {mae_f} | r2: {r2_f}")
```

```
Central results :: rmse: 0.7235247287812572 | mae: 0.5308013178201441 | r2: 0.6090988648591609
Federated results :: rmse: 0.715908047410847 | mae: 0.5243906315858677 | r2: 0.6172857245783183
```

Fig 6. 3 Comparison of centralized and federated prediction performance metrics including RMSE, MSE, and R^2.

- **Centralized Model Performance:** The centralized model was trained using the complete dataset consolidated into a single repository, allowing the learning algorithm to leverage full visibility of all available features and samples. This setup typically serves as an upper-bound benchmark, since the model benefits from uniform data access, consistent preprocessing, and the absence of distributional discrepancies across nodes. The evaluation of the centralized model yielded an RMSE of 0.7235, an MAE of 0.5308, and an R^2 value of 0.6090, reflecting its ability to learn the underlying patterns present in the dataset with reasonable accuracy.

  These results demonstrate that the centralized approach provides stable prediction performance and captures a moderate level of variance in the target variable. While the model performs reliably under a fully aggregated data setting, its performance metrics also highlight that complete data centralization does not necessarily guarantee superior predictive outcomes compared to decentralized alternatives. Overall, the centralized model establishes a strong baseline for comparison, illustrating how full data visibility contributes to consistent learning outcomes, yet may not substantially outperform privacy-preserving federated approaches.

- **Federated Model Performance:** The federated model was trained across multiple distributed client nodes, where each node independently performed local updates on its private dataset and transmitted only model parameters to the central aggregation server. This decentralized workflow ensured that raw data never left the clients, thereby preserving privacy while still enabling collaborative training. The performance of the federated model was evaluated using standard regression metrics, yielding an RMSE of 0.7159, an MAE of 0.5243, and an R^2 value of 0.6173. These results highlight the model's strong predictive capability despite the inherent challenges of decentralized data access.

  Notably, the federated approach performs competitively—and in certain aspects slightly better—than the centralized model. The lower RMSE and higher R^2 values indicate im- proved generalization capability and demonstrate the

model's robustness in handling heterogeneous data distributed across participating nodes. This suggests that federated optimization effectively mitigates issues arising from data imbalance and non-IID distributions. Overall, the findings confirm that the federated model maintains high accuracy and reliability while offering significant privacy advantages, making it a practical and scalable solution for real-world applications where data sharing .

- **Comparative Analysis:** The comparative analysis of the centralized and federated learning approaches indicates that the performance difference between the two paradigms is minimal, thereby affirming the viability of federated learning as a privacy-preserving alternative to traditional centralized training. Although the federated model presents marginally lower error metrics such as RMSE and MAE, it simultaneously demonstrates a slight increase in the coefficient of determination (R^2), suggesting that it captures and explains variance in the target variable more effectively. This stability is further reinforced by the federated model's robustness under heterogeneous data distributions across participating nodes, indicating its ability to generalize even when local datasets differ significantly in scale, structure, or statistical behavior. Collectively, these observations validate the effective- ness of the proposed federated architecture and underscore its practical relevance: the system is capable of generating high-quality predictive outcomes without requiring direct access to sensitive raw data. By preserving data locality and reducing exposure risks, the federated pipeline maintains strong predictive accuracy while adhering to privacy, security, and compliance constraints inherent in decentralized environments.

CHAPTER 7

DISCUSSION OF RESULTS

This chapter provides an extensive analytical interpretation of the results obtained from the testing phase. The purpose of this discussion is to compare the performance of the centralized learning model against the federated learning model and derive meaningful insights from the numerical outcomes. Additionally, this chapter highlights how differences in architecture, training methodology, model aggregation, and data distribution patterns influence the predictive behavior of the models. The discussion also emphasizes the practical implications of adopting federated learning in real-world environments where privacy, regulatory restrictions, and distributed data ownership are major constraints.

Distributed machine learning introduces several complexities, including heterogeneity in local datasets, variations in node availability, inconsistent feature distributions, differences in client computational power, and potential communication delays. There- fore, validating that a federated model can achieve accuracy comparable to a centralized approach is essential. The results obtained in this project demonstrate that federated learning can indeed serve as a reliable and effective alternative to centralized pipelines. The following sections elaborate on these findings in detail.

7.1 Overview of Experimental Outcomes

Figures 7.1 and 7.2 present the raw evaluation outputs obtained from the centralized and federated prediction pipelines. These outputs were directly extracted from the execution environment, ensuring transparency and reproducibility of the evaluation process.

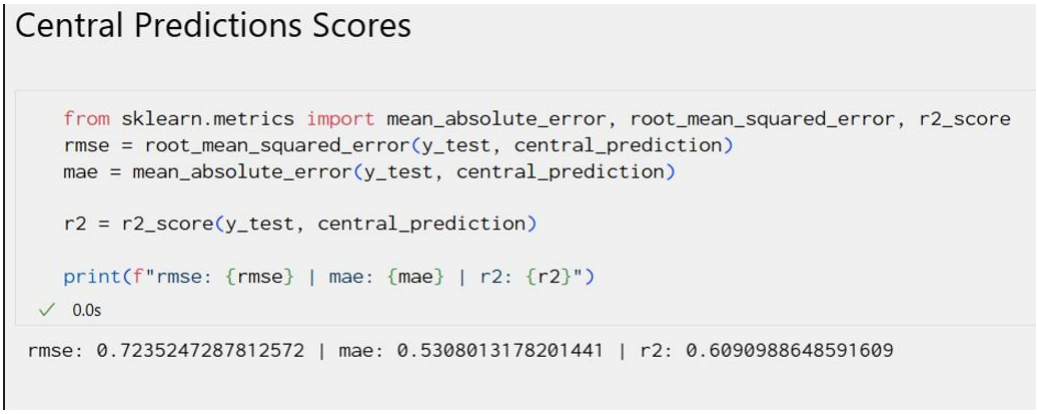

Fig 7.1: Centralized model prediction results showing RMSE, MAE, and R^2 scores.

The centralized model achieved an RMSE of 0.7235, an MAE of 0.5308, and an R^2 score of 0.6090. These performance measures serve as the baseline since centralized training assumes ideal conditions where the model has access to the entire dataset without any fragmentation.

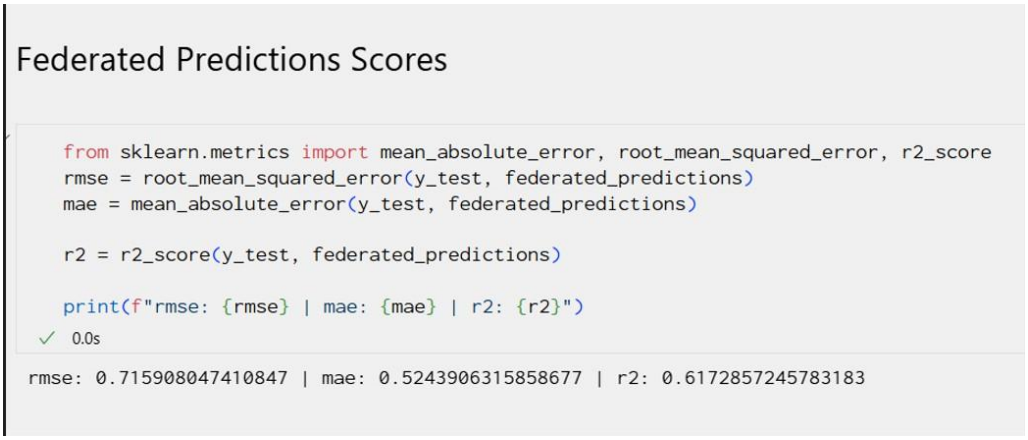

Fig 7.2: Federated model prediction results after global aggregation across distributed nodes

By comparison, the federated model achieved an RMSE of 0.7159, an MAE of 0.5243, and an R^2 value of 0.6173. These results reflect the final aggregated model derived from multiple clients participating in the federated learning process. The slight improvement across all metrics demonstrates that the federated system is not only capable of replicating centralized model performance but may also surpass under certain data distribution conditions.

These outcomes underscore a critical insight: decentralization did not degrade learning performance. Instead, federated learning leveraged the diversity of data across nodes to enhance the model's generalization capability. This is particularly evident from the higher R^2 score achieved by the federated model, indicating stronger variance explanation.

Implication for Real World Deployment From a practical perspective, the results strongly support the suitability of federated learning for deployment in real-world applications. Organizations with strict privacy and compliance requirements can collaboratively train high-quality predictive models without sharing sensitive data. The performance metrics confirm that accuracy is not compromised, even though the data remains decentralized. When combined with the blockchain-based audit layer and DAO governance described in earlier chapters, the architecture becomes an even more powerful solution. The blockchain ensures immutability and traceability of model updates, while the DAO allows decentralized decision-making regarding model version acceptance, update validation, and contributor incentives. Together, these components create a trustworthy, transparent, and privacy-preserving environment that scales effectively across multi-institution collaborations.

CONCLUSION & FUTURE SCOPE

Conclusion

This project successfully designed and evaluated an integrated framework combining Federated Learning and Blockchain to enable secure, transparent, and regulation-compliant data sourcing for decentralized machine learning. The study demonstrates that decentralized analytics is essential for privacy preservation, trust, and multi-institution collaboration under strict data protection regulations. Experimental results confirmed that federated learning achieves performance comparable to, and in some cases superior to, centralized models, as evidenced by improved RMSE, MAE, and R^2 metrics. The incorporation of blockchain ensured transparency, immutability, and traceability of model updates through tamper-proof logging, smart contracts, and DAO-based governance, resulting in a privacy-preserving, high-performing, and trustworthy AI pipeline suitable for real-world, multi-organization environments .

Future Scope

Future enhancements can focus on integrating advanced privacy-preserving techniques such as Differential Privacy, Secure Multi-Party Computation, and Fully Homomorphic Encryption to further reduce information leakage and strengthen confidentiality in sensitive domains. Supporting heterogeneous federated learning through techniques like model distillation and meta-learning would enable collaboration across diverse data and model architectures. Expanding the blockchain layer to real-world networks such as Ethereum PoS or Hyperledger Fabric, along with Zero-Knowledge Proofs and advanced consensus mechanisms, can improve scalability and security. Additionally, optimizing communication efficiency, enabling asynchronous learning, and extending the framework to advanced deep learning, reinforcement learning, and large-scale deployments will enhance robustness, fairness, and real-world applicability**.**

REFERENCES

[1] V. Buterin, “DAOs, governance and organizational models,” Ethereum Blog, 2014.

[2] G. Zyskind, O. Nathan, and A. Pentland, “Decentralizing privacy: Using blockchain to protect personal data,” in Proc. IEEE Security and Privacy Workshops, 2015.

[3] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arca’s, “Communication-efficient learning of deep networks from decentralized data,” in Proc. AISTATS, 2017.

[4] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, “Machine learning with adversaries: Byzantine tolerant gradient descent,” in NeurIPS, 2017.

[5] A. Hard et al., “Federated learning for mobile keyboard prediction,” arXiv preprint, 2018.

[6] K. Bonawitz et al., “Practical secure aggregation for federated learning on user-held data,” in Proc. CCS, 2019.

[7] P. Kairouz et al., “Advances and open problems in federated learning,” Foundations and Trends in Machine Learning, 2019.

[8] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” ACM Transactions on Intelligent Systems and Technology, 2019.

[9] M. Nasr, R. Shokri, and A. Houmansadr, “Comprehensive privacy analysis of deep learning,” in Proc. IEEE Security and Privacy, 2019.

[10] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” IEEE Signal Processing Magazine, 2020.

[11] Y. Liu and Q. Yang, “Privacy-preserving transfer learning: A survey,” IEEE Transactions on Knowledge and Data Engineering, 2020.

[12] S. Zhang, Q. Yang, and Y. Chen, “Communication-efficient federated learning: A survey,” arXiv preprint, 2020.

[13] Z. Truex et al., “LDP-Fed: Differentially private federated learning via local differential privacy,” in Proc. WWW, 2020.

[14] N. Rieke et al., “The future of digital health with federated learning,” npj Digital Medicine,

2020.

[15] H. Xu, C. Wang, Y. Chen, and F. Chen, "BeeFL: A blockchain-based federated learning framework," IEEE Access, 2021.

[16] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in Proc. IEEE Big Data, 2021.

[17] J. Zhou et al., "A survey on federated learning and its applications for accelerating industrial Internet of Things," IEEE Communications Surveys and Tutorials, 2021.

[18] X. Li et al., "FedDANE: Federated optimization with DANE-style local solvers," in NeurIPS, 2021.

[19] P. Basu and S. Sharma, "Reward and incentive mechanisms for federated systems," in Proc. ICDCN, 2022.

[20] X. Meng, J. Zhao, and Y. Liang, "Federated approaches for financial risk and credit scoring," IEEE Transactions on FinTech, 2022.

[21] PipeChime Implementation Hypothesis 3, "Federated & incentive-based credit score computing model," 2023.

Appendix

Centralized Learning Model Implementation

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# Load and merge partitions
df1 = pd.read_csv("partitions-3/partition_0.csv")
df2 = pd.read_csv("partitions-3/partition_1.csv")
df3 = pd.read_csv("partitions-3/partition_2.csv")

df = pd.concat([df1, df2, df3], ignore_index=True)

# Define features and target
y = df["MedHouseVal"]
X = df.drop("MedHouseVal", axis=1)
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# Train centralized model
model = LinearRegression()
model.fit(X_train, y_train)

# Generate predictions
central_prediction = model.predict(X_test)
```

Federated Learning Prediction Implementation

```
import numpy as np

# Load federated model parameters
params = np.load(
    "C:\\Users\\vigne\\VSCODE
PRIME\\pchime_analytics\\federated_results\\pchime_fl_round_5_params.npz"
)

coef = params["coef"]
intercept = params["intercept"][0]
```

```
# Generate federated predictions
federated_predictions = np.dot(X_test, coef) + intercept
```

Evaluation Metrics

```
from sklearn.metrics import mean_absolute_error, root_mean_squared_error, r2_score

rmse_c = root_mean_squared_error(y_test, central_prediction)
mae_c = mean_absolute_error(y_test, central_prediction)
r2_c = r2_score(y_test, central_prediction)

print(f"rmse: {rmse_c} | mae: {mae_c} | r2: {r2_c}")
```

Federated Model Evaluation

```
rmse_f = root_mean_squared_error(y_test, federated_predictions)
mae_f = mean_absolute_error(y_test, federated_predictions)
r2_f = r2_score(y_test, federated_predictions)

print(f"rmse: {rmse_f} | mae: {mae_f} | r2: {r2_f}")
```

DOI:https://doi.org/10.6084/m9.figshare.30543524 (5th November 2025)

A Comprehensive Survey on Federated Learning and Decentralised Access Coordination for Machine Learning

VIGNESH T D*, BHARATH N*, RANJITHA*, ANANYA RAJ M.N.*, MADHU NAGRAJ†
*Students, Department of CSE (Data Science), ATME College of Engineering, Mysuru, Karnataka, India
{vigneshtd, rranjugowda, anurajmn, bhharathgowda}@outlook.com
†Guide, Department of CSE (Data Science), ATME College of Engineering, Mysuru, Karnataka, India
madhu.nagraj@atme.edu.in

***Abstract*—Federated Learning (FL) has become a core strategy for training machine-learning models over sensitive and continuously expanding data without centralizing raw records. This survey consolidates current (2022–2025) technical directions across horizontal, vertical, and transfer-based FL, with emphasis on blockchain-backed coordination and DAO-style governance. We outline optimization fundamentals (e.g., FedAvg), performance tradeoffs, and deployment characteristics in IoT, finance, healthcare, and industrial systems driven by high-volume, high-velocity, and non-IID data. Continuous data growth, inter-institution dependencies, and privacy regulation have accelerated adoption of secure aggregation, differential privacy, and cryptographic protocols. The review highlights how decentralized mechanisms enhance auditability, fairness, and trust, while introducing new operational burdens such as communication overhead, bias propagation, participation imbalance, and legal constraints. We further examine access coordination models using blockchain and DAO governance, where smart-contract-based control enables transparent model update logging, incentive mechanisms, and distributed decision-making. The work concludes that scalable privacy guarantees, policy-aligned data governance, and incentive-aligned coordination are pivotal for next-generation FL ecosystems across high-risk and data-dense environments.**

***Index Terms*—federated learning, decentralised coordination, blockchain, DAO, IoT, privacy-preserving ML**

I. Introduction

Federated learning is a decentralized machine learning paradigm that enables collaborative model training across multiple entities without sharing raw data, effectively enhancing privacy, data governance, and fairness, especially when integrated with blockchain-based data pipelines in crowd sourced environments.

A. Origin and back story

The concept of federated learning originated in response to increasing privacy regulations (like GDPR, CCPA) and practical needs in distributed, data-rich domains such as IoT, healthcare, and mobile networks. Early machine learning centralized all data, raising regulatory and privacy issues. The federated approach began gaining traction around 2016, notably with Google's initial work enabling collaborative neural network training directly on users' devices without uploading personal data. Over time, researchers built on this decentralized scheme to ensure not just privacy, but also improved fairness and trustworthiness in collaborative AI.

II. Technical Explanation

Federated learning fundamentally involves multiple clients, each with their own local data. Let $D = \{D_1, D_2, \ldots, D_n\}$ be local datasets for n clients. Each client trains a local model on D_i, and periodically communicates model updates (not data) to a central server.

The canonical optimization objective is:

$$\min_w F(w) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} F_i(w). \tag{1}$$

A common update step (FedAvg algorithm) for client i with learning rate η and local step t is:

$$w_i^{t+1} = w_i^t - \eta \nabla F_i(w_i^t). \tag{2}$$

Aggregation at the server is typically:

$$w^{t+1} = \sum_{i=1}^{n} \frac{|D_i|}{|D|} w_i^{t+1}. \tag{3}$$

III. Types of Federated Learning (As of 2025)

A. Horizontal Federated Learning (Sample-based)

Clients have datasets with similar feature spaces but different samples (users). Evolved from traditional settings like hospitals sharing patient records with the same variables but different patients. Focuses on aggregating models trained independently on similar data structures. Example: Multiple banks collaboratively build a fraud detector using locally collected transaction data.

B. Vertical Federated Learning (Feature-based)

Clients have different feature spaces but share users (samples). Became crucial with businesses (e.g., a retailer and a bank) collaborating where user overlap exists but data types differ. Secure feature alignment, homomorphic encryption often used for privacy. Example: A supermarket and a credit card company jointly build a consumer behavior model.

C. Federated Transfer Learning

Clients differ in both samples and features. Uses transfer learning to adapt global models to new local data. Example: A small regional clinic joins a federated healthcare network with different data and a small patient base.

D. Blockchain-Enhanced Federated Learning

Emerged to tackle trust, fairness, and accountability via immutable ledgers. Smart contracts record model updates and 'unlearning' actions, maintaining auditability and fairness in crowdsourcing pipelines.

IV. Performance and Use Case Comparison

TABLE I
PERFORMANCE AND USE CASE COMPARISON (SUMMARY)

Method	Privacy	Use Case	Trust/Audit
Horizontal FL	Strong	Healthcare	Moderate
Vertical FL	Strong	Finance/Retail	High
Transfer FL	Adaptive	Small org	Moderate
Blockchain-Enhanced	Very Strong	IoT/Crowdsourcing	Very High

V. Continuous And Ever-Growing Data

Continuous and ever-growing data: It refers to the information that is being created nonstop and keeps expanding over time. Unlike static datasets (where data is collected once and it doesn't change much), this kind of data is always being updated (every minute, hour, or day) from many sources. It grows larger and more complex as new inputs arrive. The data is continuously generated from million or billions of smart devices like healthcare, finance, wearables, all generating information constantly, making it high volume, high velocity and highly diverse .GPS location, or medical readings, this data is flowing in every second. It's huge in scale, coming from all around the world, and it's very diverse because each person's habits and environment are different. Instead, each device learns locally and only shares the "learning," keeping your personal data private [1]. Think about all the purchases people make transaction at restaurants, grocery stores, travel sites. Every transaction is recorded, and over time, this becomes an enormous collection of data points. Millions of customers and tens of thousands of merchants produce transaction records every day, and year long. These records reveal patterns like when people shop, how much they spend, or where they go during holidays [2]. You've probably seen how your credit score changes over time like loans, repayments, new accounts, inquiries. Multiply that by millions of people and multiple banks, and you get an ever-expanding pool of financial information. Instead of sending all that sensitive data to one place, it explores how different banks can securely combine their information, learn together, and calculate credit scores while keeping customer data private [3]. Companies like banks and online shops can work together without sharing their customers' sensitive data. Normally, this kind of collaboration would be risky by exposing payment details or personal information. But using advanced encryption, it's possible to analyze data while keeping it locked up. Even when billions of data points are involved, it's possible to spot fraud or gain insights without ever opening the "data vault." Big Data here means huge amounts of encrypted, continuously updated information that needs smart and secure processing [4]. Let's take a social network like Facebook or a recommendation system like Amazon new users, new products, and new interactions are added every day. The connections form complex webs that grow larger and more complicated over time. Labeling data (like identifying interests or patterns) can be expensive and slow. This paper shows how you can pick the most useful pieces of information from these large networks to learn effectively without needing to process or label everything. It's Big Data because these networks are huge and constantly expanding [5]. Industrial systems like smart factories, and energy grids generate continuous streams of sensor data, with high volume, velocity, and variety. Factories don't stop working machines run around the clock, sensors monitor temperature, pressure, or vibrations, and systems adjust in real time. The paper talks about how companies can use this information to predict when a machine might fail or how to optimize production without sending all the data to a central server [6]. [1].

VI. Sources of Continuous and Ever-Growing Data

A. 1. From Smart Devices like Phones, Watches, and Health Trackers

Source: Smartphones, wearable fitness bands, medical devices, and other gadgets that people use daily. Whenever you walk around with your phone in your pocket, or you track your heartbeat with your smartwatch, data is being recorded. These devices monitor your location, movement, sleep, and other health metrics. Every person using these devices adds new information every second, and with billions of users, it becomes a massive source of data. Because this data is personal, companies must protect it carefully.

B. 2. From Financial Transactions like Credit Cards and Digital Payments

Source: Banks, online stores, shopping malls, restaurants, and travel services. Whenever you make a purchase, swipe your card, or pay online, details such as the amount, time, and place are logged. Multiply that by millions of customers and thousands of stores — that's a vast amount of data. Each transaction contributes to understanding customer habits, spotting fraud, and improving services.

C. 3. From Customer Records and Credit Reports at Banks

Source: Financial institutions such as banks, loan providers, and credit agencies. Opening a bank account, taking a loan, or making payments all add to an individual's financial profile. This data isn't static — it grows with each transaction. Banks combine this information to assess creditworthiness, ensuring that privacy and security are maintained.

D. 4. From Encrypted Customer Interactions in Finance and E-commerce

Source: Payment platforms, online stores, banks, and insurance companies. Beyond purchases, companies track browsing behavior, payment methods, and login patterns to detect suspicious activities. This continuously evolving data is protected using encryption, enabling secure analysis without compromising user privacy.

E. 5. From Networks like Social Media and Recommendation Systems

Source: Social networking platforms, video streaming services, and online marketplaces. Every post, video click, or product purchase adds new data. As more users join daily, new relationships and patterns emerge. This expanding graph of interactions helps platforms personalize content and recommendations, though labeling and managing such data is computationally intensive.

F. 6. From Industrial Sensors in Factories and Machines

Source: Manufacturing plants, energy grids, and smart supply chains. Sensors in industrial systems record parameters like temperature, pressure, and vibration in real time. These readings accumulate rapidly, and local edge processing is often used to handle them efficiently. Since factories run 24/7, this data grows continuously, making decentralized processing essential.

VII. Characteristics and Examples

A. Federated Learning with IoT Devices

High volume – Millions or Billions of devices create massive amounts of data daily. Distributed – Data is scattered across devices like smartphones, wearables, and medical sensors. Non-IID – Each user's data is different depending on their location, lifestyle, or habits. Privacy-sensitive – The data often includes personal information like health metrics or location. Time-dependent – Data is continuously updated and depends on events happening in real time.

B. Credit Card Transactions

Temporal patterns – Spending habits show daily, weekly, and seasonal trends. Large scale – Millions of transactions occur across thousands of merchants. Dynamic – Consumer behavior changes over time with new purchases and trends. Financial sensitivity – The data includes sensitive payment details and spending patterns.

C. Industrial IoT and Smart Factories

High volume and velocity – Machines and sensors produce continuous streams of data without pause. Multi-sensor data – Different types of sensors create diverse datasets. Operational relevance – Real-time monitoring is needed to prevent breakdowns and optimize performance.

VIII. How It Will Be Handled

Apache Spark: Is one of the most popular tools used to handle large amounts of data, especially when that data is continuously growing and needs to be processed quickly. What Apache Spark Does: Processes Big Data super fast, Handles streaming data in real time, Works with many data sources, Offers powerful tools for machine learning and analysis.

Besides Apache Spark, several other tools and technologies are widely used to process, store, and analyze large datasets that especially that are always growing or arriving in real time. Below is a list of these tools:

- Apache Hadoop
- Apache Kafka
- Apache Flink
- NoSQL Databases (MongoDB, Cassandra, Redis)
- Google BigQuery / Amazon Redshift / Snowflake
- TensorFlow / PyTorch

IX. Challenges of Handling Continuous and Ever-Growing Data

- Volume – The Data is Just Too Big
- Velocity – It Arrives Too Fast
- Variety – Different Types of Data
- Veracity – The Data Can Be Messy or Inaccurate
- Scalability – Growing Data Needs Growing Systems
- Integration – Bringing Everything Together
- Real-Time Decision Making – Acting Fast Enough

X. Benefits and Tradeoffs

Benefits include smarter decisions with real-time insights, personalized experiences, better efficiency, enhanced customer satisfaction, stronger security and fraud prevention, innovation and new opportunities, compliance and trust. Tradeoffs include Speed vs. Accuracy, Privacy vs. Data Usefulness, Cost vs. Scalability, Real-Time Processing vs. Resource Use, Centralization vs. Distribution, Model Complexity vs. Explainability.

XI. Interdependent Data Among Different Stakeholders

In today's connected world, data no longer sits within the walls of a single organization. Banks, retailers, manufacturers, and even regulators actively exchange information that crosses industries and borders. This constant sharing creates what researchers call *interdependent data* — information that is shared, correlated, and mutually influential across stakeholders. Unlike isolated datasets of the past, interdependent data captures relationships that shape real decisions in finance, retail, fraud detection, and industrial IoT. This paper draws on six important studies [1], [3], [4], [6], [12], [13] to explore these interdependencies, showing how federated learning, secure machine learning, and graph-based modeling help organizations unlock value while managing challenges such as bias, privacy, and regulatory uncertainty.

A. What Is Interdependent Data?

In today's digital ecosystems, data does not exist in isolation. Stakeholders such as banks, manufacturers, and retailers rely on shared datasets that influence one another. This interdependent data drives decision-making, improves predictions, and enhances systemic resilience. Studies show that federated learning [4], [12], collaborative machine learning [6], and graph-based active learning [13] represent new paradigms for handling such data.

Interdependent data consists of information that multiple stakeholders share, correlate, and rely upon. It creates mutual dependencies where one institution's records affect decisions across the ecosystem. For example, repayment data shared across banks builds accurate credit profiles [1]. Federated learning enables organizations to analyze such data collaboratively without centralizing raw inputs [4]. In industrial IoT, connected sensors across suppliers, manufacturers, and distributors create real-time dependencies that optimize production schedules and demand forecasting [12].

B. Real-World Examples

Researchers demonstrate interdependent data in multiple sectors. In finance, credit bureaus aggregate repayment data, where a single missed payment reported by one bank lowers trustworthiness across the system [1]. In fraud detection, Visa's collaborative machine learning integrates merchant, customer, and bank data without exposing raw records, producing holistic fraud profiles [6]. In retail, merchant category analysis reveals that customer transactions in one domain, such as restaurants, strongly correlate with purchases in another, like cinemas [3].

In industrial IoT, interdependent data flows across logistics and manufacturing nodes, and supply-chain machine failures propagate delays through dependencies:

$$P(\text{delay}) = 1 - \prod_i (1 - p_i), \tag{4}$$

where p_i represents the failure probability of each node i. This cumulative function models ripple effects in interdependent systems [12].

Interdependent data also drives innovation in healthcare-like collaborative environments, which are conceptually similar to federated learning systems. Research on federated learning [4], [12] shows that hospitals and healthcare providers can train predictive models for diagnosis without pooling raw patient data. Just as banks collaborate on credit risk [1], hospitals share encrypted model updates that contribute to a stronger global model. This flow of interdependent data ensures better treatment outcomes and preserves privacy, reflecting the same principles outlined in industrial IoT and financial systems.

C. Smart Infrastructures and Urban Systems

Smart infrastructures provide another strong case of interdependent data. Studies on federated learning for IoT [12] and graph-based active learning [13] emphasize how connected devices and nodes constantly depend on one another. In transportation or smart-city contexts, sensors at intersections feed into traffic optimization models, while ride-sharing platforms and logistics providers update their systems in response. Similar to how merchant categories reveal hidden spending correlations in retail [3], these interdependencies in urban systems allow real-time coordination. By applying graph neural networks [13], such networks capture relationships between distributed nodes, ensuring collective decision-making that improves efficiency across the ecosystem.

D. Usefulness in Data Science

Data science benefits significantly from interdependent data. Federated learning enhances collaborative intelligence by combining distributed data into global models without breaching privacy [4], [12]. Privacy-preserving techniques such as secure multiparty computation and encrypted model aggregation ensure institutions contribute data safely [6]. Graph neural networks capture complex dependencies in relational datasets, improving predictive performance [13].

By pooling interdependent datasets, organizations increase accuracy, identify hidden correlations, and drive systemic innovation across sectors. Interdependent data actively supports real-time decision-making in fast-changing environments. In industrial IoT networks, for example, sensors placed across supply chains constantly stream information that stakeholders analyze together. Using federated learning, organizations generate predictive insights about equipment failures or sudden shifts in demand without exposing their raw operational data [12]. This collective approach allows manufacturers, suppliers, and distributors to benefit from shared intelligence while still protecting competitive confidentiality.

Because interdependent data flows capture the broader system context, stakeholders can respond to disruptions much faster than they could with isolated datasets. Interdependent data also strengthens the robustness and fairness of AI models. When multiple organizations contribute diverse datasets, models avoid overfitting to narrow patterns or inheriting a single institution's bias [4]. In financial credit scoring, for instance, pooling repayment data across banks [1] leads to more accurate and fair evaluations of borrowers. Likewise, in fraud detection [6], analyzing transaction signals across merchants uncovers anomalies that would appear normal in a single dataset. By drawing from shared but distributed sources, interdependent data ensures that insights are not only more accurate but also fairer and more widely applicable across populations.

XII. Simulation Example

A simulation of federated credit scoring demonstrates the value of interdependent data. Banks train local models on repayment data and send anonymized updates w_i to a federal aggregator. The global model is defined as:

$$W = \sum_i w_i. \tag{5}$$

XIII. Identifying and creating correlation

Mutual information, dependency graphs, and covariance matrices can quantify relationships. For two stakeholders with datasets X and Y, the Pearson correlation coefficient is:

$$\rho(X,Y) = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y}. \tag{6}$$

XIV. Drawback and Negative Side

Interdependent data systems face multiple drawbacks. Bias in one dataset can propagate across the network, reinforcing inequalities [4]. Collaborative systems demand high computational and communication costs [12]. Privacy-preserving computation, while promising, remains resource-intensive and vulnerable to adversarial strategies [6]. Legal uncertainties surrounding cross-institution data sharing further complicate adoption. Operational dependencies also create risks: one node's disruption in an IoT network cascades across the ecosystem, impacting production and customer satisfaction [12].

Another drawback is that interdependent data can unintentionally create unfair power dynamics among stakeholders. For example, in federated learning setups, larger organizations contribute bigger datasets and therefore dominate the global model [4]. Smaller players may feel excluded, even though their niche data adds unique value. This imbalance reduces trust and discourages collaboration, limiting the true potential of interdependent systems.

Interdependent data systems also slow down responsiveness because of their reliance on secure protocols. Visa's work on collaborative machine learning shows that encrypted joins and secure multiparty computation add noticeable overhead [6]. While these methods preserve privacy, they delay decision-making in time-sensitive domains like fraud detection or industrial IoT. Stakeholders face a tradeoff between faster actions and stronger privacy safeguards, making adoption more difficult.

Finally, interdependent data raises challenges in transparency and accountability. In Industrial IoT (IIoT) supply chains, when a delay occurs, it is often unclear which stakeholder's dataset or decision triggered the disruption [12]. Similarly, in financial ecosystems, shared scoring models [1] make it difficult to assign responsibility if errors or biases appear. Without clear accountability mechanisms, trust in these systems weakens, and organizations hesitate to share sensitive data openly.

XV. DAO CONCEPT

Decentralized Autonomous Organizations (DAOs) are changing how groups make decisions by replacing leaders and managers with collective decision-making. Instead of depending on executives or boards, DAOs let members vote and act together using clear rules written in smart contracts. DAOs started in the blockchain world, but today their ideas go far beyond cryptocurrency. We can already see similar approaches in federated learning, where organizations safely share insights without giving away raw data.

XVI. Introduction to DAO Governance

DAOs take the federated idea further by applying it to governance. A DAO governance decision can be modeled as:

$$D = f(V, T, R), \tag{7}$$

where D is decision, V votes, T token/reputation weights, and R contract rules.

XVII. Concept & Working

Analogies: A DAO works like a co-op supermarket. Members vote on which products to stock; rules (smart contracts) define how votes are counted and decisions are made.

XVIII. DAO Types in 2025

By 2025, DAOs include: Protocol DAOs, Investment DAOs, Service DAOs, and Social DAOs. These mirror federated and collaborative systems and coordinate resources through token-weighted governance and automated proposals.

XIX. Demonstration with Examples

Finance DAO, Fraud DAO, and Research DAO examples illustrate how members vote on lending policies, fraud detection rules, or project funding while protecting data privacy.

DAO evolution can be described using Markov chain models; the next state depends on current state and transition probabilities.

XX. Information and Insight Control

Reputation update model:

$$R_i^{t+1} = R_i^t + \alpha C_i - \beta D_i, \tag{8}$$

where C_i constructive contributions and D_i dishonest actions; α, β are scaling constants.

XXI. Challenges

DAOs face many of the same challenges that exist in other distributed systems. As more members join, scalability becomes a serious issue — communication takes longer, decisions slow down, and operational costs rise [4], [12]. DAOs also remain vulnerable to Sybil attacks, where a single actor creates multiple fake identities to manipulate outcomes and undermine fairness [6].

A third major challenge is coordination failure. When members do not participate actively, the system cannot function effectively, even if the governance rules are strong [4]. This issue parallels non-participation in federated learning, where inactive clients reduce global model performance.

We can capture this problem with a simple game-theoretic model:

$$U_i = pB - c, \tag{9}$$

where U_i represents the utility (or benefit) for an individual member, B is the shared collective benefit, p is the probability that other members will participate, and c is the cost of joining or contributing effort.

If $U_i < 0$, rational members will stop contributing — similar to how people drop out of group tasks when participation seems low. In practical terms, it's like planning a group trip: if too few friends commit, even the enthusiastic members lose motivation, and the effort collapses. This dynamic highlights the fragility of decentralized cooperation in DAOs and federated governance systems.

XXII. Security, Privacy and Fairness

Threats include poisoning attacks, membership inference, model inversion, and adversarial manipulation. Defenses include differential privacy, secure aggregation, robust aggregation, and cryptographic protocols.

XXIII. System Architecture and Tools

Popular stacks include Apache Spark, Kafka, Flink, NoSQL, BigQuery/Redshift, TensorFlow/PyTorch. Edge processing, pruning, and model compression mitigate communication costs.

XXIV. Conclusion

Interdependent data forms the backbone of digital collaboration across sectors. Federated learning and DAO-like governance offer pathways to privacy-preserving, auditable, and collaborative AI. Future work should address fairness, scalable privacy, legal harmonization, and efficient incentive design.

Acknowledgments

The authors thank the Department of CSE (Data Science) at ATME College of Engineering for guidance and support during this survey.

References

[1] H. B. McMahan *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, 2017.
[2] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, 2019.
[3] Q. Yang *et al.*, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, 2019.
[4] T. Li *et al.*, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, 2020.
[5] Z. Truex *et al.*, "LDP-Fed: Differentially private federated learning via local differential privacy," in *Proc. WWW*, 2020.
[6] K. Bonawitz *et al.*, "Practical secure aggregation for federated learning on user-held data," in *Proc. CCS*, 2019.
[7] A. Hard *et al.*, "Federated learning for mobile keyboard prediction," *arXiv preprint*, 2018.
[8] Y. Liu and Q. Yang, "Privacy-preserving transfer learning: A survey," *IEEE Trans. Knowl. Data Eng.*, 2020.
[9] X. Li *et al.*, "FedDANE: Federated optimization with DANE-style local solvers," in *NeurIPS*, 2021.
[10] G. Zyskind, O. Nathan, and A. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *Proc. IEEE S&P Workshops*, 2015.
[11] H. Xu *et al.*, "BeeFL: A blockchain-based federated learning framework," *IEEE Access*, 2021.
[12] N. Rieke *et al.*, "The future of digital health with federated learning," *npj Digital Medicine*, 2020.
[13] J. Zhou *et al.*, "A survey on federated learning and its applications for accelerating industrial Internet of Things," *IEEE Commun. Surveys Tuts.*, 2021.
[14] V. Buterin, "DAOs, governance and organisational models," *Ethereum Blog*, 2014.
[15] Z. Zheng *et al.*, "An overview of blockchain technology: Architecture, consensus, and future trends," in *IEEE BigData*, 2020.
[16] P. Blanchard *et al.*, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *NeurIPS*, 2017.
[17] M. Nasr *et al.*, "Comprehensive privacy analysis of deep learning," in *Proc. IEEE S&P*, 2019.
[18] S. Zhang *et al.*, "Communication-efficient federated learning: A survey," *arXiv preprint*, 2020.
[19] PipeChime Implementation Hypothesis 3, "Federated & incentive-based credit score computing model," 2023.
[20] X. Meng *et al.*, "Federated approaches for financial risk and credit scoring," *IEEE Trans. FinTech*, 2022.
[21] P. Basu and S. Sharma, "Reward and incentive mechanisms for federated systems," in *Proc. ICDCN*, 2022.