FRAUD DETECTION WITH AI-POWERED SYSTEM USING BLOCKCHAIN TECHNOLOGY

A PROJECT REPORT

Submitted by

JERRISH N [211421104111]

MICHAEL JOSIL M [211421104161]

PARUN VIGNESH T [211421104180]

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2025

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"FRAUD DETECTION WITH AI-POWERED SYSTEM USING BLOCKCHAIN TECHNOLOGY"** is the bonafide work of "**JERRISH N, MICHAEL JOSIL M, PARUN VIGNESH T"** who carried out the project work under my supervision.

Signature of the HOD with date

Dr L.JABASHEELA M.E., Ph.D.,
PROFESSOR,

HEAD OF THE DEPARTMENT

Department of CSE,
Panimalar Engineering College,
Chennai - 123

Signature of the Supervisor with date

Mr. A.KARTHIKEYAN B.E., MTech.,
Assistant Professor,

SUPERVISOR

Department of CSE,
Panimalar Engineering College,
Chennai – 123

Certified that the above candidate(s) was examined in the End Semester Project Viva- Voce Examination held on .....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We **JERRISH N (211421104111), MICHAEL JOSIL M(211421104161)** and **PARUN VIGNESH T (211421104180)** hereby declare that this project report titled "**FRAUD DETECTION WITH AI-POWERED SYSTEM USING BLOCKCHAIN TECHNOLOGY**", under the guidance of **Mr. A.KARTHIKEYAN B.E., M. Tech**., is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

JERRISH N [211421104111]

MICHAEL JOSIL M [211421104161]

PARUN VIGNESH T [211421104180]

ACKNOWLEDGEMENT

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., Ph.D**., for his fervent encouragement. His inspirational support proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project.

We want to express our deep gratitude to our Directors, Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and **Dr. SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.,** for graciously affording us the essential resources and facilities for undertaking of this project.

Our gratitude is also extended to our Principal, **Dr. K. MANI, M.E., Ph.D.,** whose facilitation proved pivotal in the successful completion of this project.

We express our heartfelt thanks to **Dr. L. JABASHEELA, M.E., Ph.D.,** Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of project.

We would like to thank our Project Guide **Mr. A. KARTHIKEYAN BE., MTech.,** and our Project Coordinator **Dr. R. JOSPHINELEELA, M.E., Ph.D**., and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

JERRISH N [211421104111]

MICHAEL JOSIL M [211421104161]

PARUN VIGNESH T [211421104180]

ABSTRACT

The rapid adoption of blockchain-based financial systems has introduced new security challenges, particularly in cryptocurrency transactions. This project presents an AI-powered Bitcoin wallet that integrates secure wallet generation, transaction management, and real-time fraud detection. The system employs encryption techniques for private key protection and address generation, ensuring secure wallet creation. Transaction processing features enable seamless sending, receiving, and logging of Bitcoin transactions. An AI-based anomaly detection algorithm continuously monitors transactions to identify suspicious activities, enhancing the security and reliability of digital wallets. This integration of artificial intelligence with blockchain technology improves fraud prevention, promotes transparency, and strengthens trust in decentralized financial systems.

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO
3.2.1	System Architecture	19
3.3.4.1	Level 0 Data Flow Diagram	23
3.3.4.2	Level 1 Data Flow Diagram	24
3.3.4.3	Level 2 Data Flow Diagram	25
3.4.1	UseCase Diagram	26
3.4.2	Class Diagram	27
3.4.3	Activity Diagram	28
3.4.4	Sequence Diagram	29

LIST OF TABLES

TABLE NO.	TABLE NAME	PAGE NO
3.1	AI Model selection	22
5.1	Test case	35

LIST OF ABBRIVIATION

S.NO.	ABBREVIATION	DEFINITION
1.	DCP	Data Collection and Preprocessing
2.	AIMS	AI Model Selection
3.	TAV	Training and Validation
4.	RTTV	Real-time Transaction Validation
5.	AD	Anomaly Detection
6.	CME	Consensus Mechanism Enhancement
7.	EO	Evaluation and Optimization
8.	RF	Random Forest
9.	ANN	Artificial Neural Networks
10.	LSTM	Long Short-Term Memory
11.	IF	Isolation Forest
12.	DFD	Data Flow Diagram
13.	UML	Unified Modeling Language

TABLE OF CONTENTS

CHAPTER NO.		TITLE	PAGE NO.
		ABSTRACT	**v**
		LIST OF FIGURES	**vi**
		LIST OF TABLES	**vi**
		LIST OF ABBREVIATIONS	**vii**
1.		**INTRODUCTION**	1
	1.1	Problem Definition	2
	1.2	Purpose of the Project	3
	1.3	Motivation	4
2.		**LITERATURE REVIEW**	5
3.		**THEORETICAL BACKGROUND**	17
	3.1	Implementation Environment	18
	3.2	System Architecture	19
	3.3	Proposed Methodology	20
		3.3.1 Wallet Generation and Management Module	21
		3.3.2 Data Collection and Preprocessing	21
		3.3.3 AI Model Selection	22
		3.3.4 Module design	23
	3.4	UML Diagrams	26
4.		**SYSTEM IMPLEMENTATION**	30
	4.1	Data Overview	31
	4.2	AI Model Selection	31

	4.3	Training and validation	32
	4.4	Real-time Transaction Validation	32
5.		**RESULTS & DISCUSSION**	33
	5.1	Performance Parameters / Testing	34
	5.2	Results& Discussion	36
6.		**CONCLUSION AND FUTURE WORK**	37
		APPENDICES	40
	A.1	SDG Goals	41
	A.2	Source Code	42
	A.3	Screen Shots	63
	A.4	Plagiarism Report	70
		REFERENCES	81

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION

The rapid growth of blockchain-based financial systems has introduced significant security challenges, particularly in detecting fraudulent transactions due to the decentralized and pseudonymous nature of blockchain networks. Traditional fraud detection mechanisms relying solely on cryptographic security and consensus algorithms often fail to identify real-time anomalies, making blockchain transactions susceptible to money laundering, double-spending, and other illicit activities. This project proposes an **AI-Powered Blockchain Integrity and Fraud Detection System** that integrates **Random Forest (RF), Artificial Neural Networks (ANN), Long Short-Term Memory (LSTM), and Isolation Forest** models to analyze transaction patterns and detect fraudulent activities with high accuracy. By leveraging both supervised and unsupervised machine learning techniques, the system continuously monitors transactions, assigns fraud probability scores, and ensures transparency by logging flagged transactions on the blockchain. Additionally, the system enhances security through an **adaptive consensus mechanism** based on **Byzantine Fault Tolerance (BFT)** to prevent fraudulent transactions from being validated. The integration of AI-driven anomaly detection with blockchain technology significantly improves fraud prevention, enhances transaction security, and strengthens trust in decentralized financial system.

1.2 PURPOSE OF THE PROJECT

The primary objective of this project is to develop an advanced AI-powered fraud detection system for blockchain transactions, leveraging machine learning algorithms to detect and mitigate fraudulent activities in real time. With the increasing adoption of blockchain across various industries, particularly in financial applications like cryptocurrencies, cybercriminals exploit the decentralized and pseudonymous nature of transactions, making fraud detection challenging. Traditional rule-based approaches struggle with evolving fraud tactics and high false positive rates, necessitating more adaptive solutions. By integrating AI-driven anomaly detection models, this project aims to enhance security, transparency, and accuracy in fraud detection by analyzing vast transaction datasets, identifying suspicious patterns, and dynamically adapting to emerging threats. Utilizing techniques such as supervised and unsupervised learning, deep learning, and real-time predictive analytics, the system will provide robust and scalable fraud detection capabilities. Unlike conventional centralized systems prone to single points of failure, this project will implement a decentralized AI-driven framework that aligns with blockchain principles, incorporating smart contracts, decentralized computing, and secure data-sharing mechanisms to ensure efficiency and trust. Ultimately, this project aspires to revolutionize fraud detection in blockchain ecosystems by providing an adaptive, intelligent, and scalable solution that safeguards financial transactions, prevents illicit activities, and fosters greater trust in decentralized digital economies.

1.3 MOTIVATION

In 2024, the cryptocurrency industry witnessed a significant surge in cybercrime, with losses escalating by approximately 21% year-over-year to $2.2 billion. Notably, North Korean hackers were responsible for a substantial portion of these thefts, stealing $1.34 billion and accounting for 61% of the total amount stolen during the year. These alarming statistics underscore the urgent need for more effective fraud detection mechanisms within blockchain networks.Traditional rule-based fraud detection systems, while straightforward to implement, often fall short in adapting to the rapidly evolving tactics employed by cybercriminals. These systems tend to produce high false positive rates and struggle to detect sophisticated fraud patterns. In contrast, machine learning-based approaches offer enhanced capabilities by analyzing vast datasets to identify anomalies and predict fraudulent activities more accurately.The motivation for this project lies in addressing the escalating threats within the cryptocurrency ecosystem by developing an AI-powered fraud detection system tailored for blockchain transactions. By leveraging machine learning techniques, this system aims to provide real-time analysis and detection of fraudulent activities, thereby enhancing the security and integrity of blockchain platforms. This approach seeks to overcome the limitations of traditional methods, offering a more adaptive and robust solution to combat the increasing sophistication of cyber threats in the digital asset landscape.

CHAPTER 2

LITERATURE SURVEY

[1] **G B Renuka, Pramod Kumar Patjoshi, Upendra Aswal, G. Manikandan, L.N. Jayanthi, Ashish Kaushal** proposed a concept in which AI is integrated into blockchain systems to enhance transparency and accountability. The study focuses on overcoming traditional blockchain limitations such as data tampering and insufficient real-time validation by leveraging AI-driven techniques for fraud detection and transaction validation. Their system improves consensus mechanisms, detects anomalies, and enhances security through real-time monitoring and adaptive algorithms. The proposed AI-enhanced blockchain system significantly boosts fraud detection accuracy, reduces false positives and negatives, and enhances transaction validation performance**.**

"Integrating Reliable AI to Boost Blockchain's Transparency and Accountability" – August 2024, IEEE

Advantages - The system enhances security, scalability, and efficiency in blockchain networks by integrating AI for real-time fraud detection and anomaly identification. It provides a 95% fraud detection rate with only a 5% false positive and negative rate. AI-powered consensus mechanisms improve network adaptability and reduce malicious activities.

Disadvantages - The system requires extensive data collection, preprocessing, and continuous evaluation, which increases computational complexity. Additionally, AI-driven models might introduce biases, and scalability challenges may arise in handling high transaction volumes across diverse blockchain applications.

[2] **Oleksandr Kuznetsov 1,2,3, Paolo Sernani 4, Luca Romeo 5, Emanuele Frontoni 2, Adriano Mancini 6** proposed a paper discussing the integration of Artificial Intelligence (AI) and Blockchain Technology (BCT), focusing on the security implications. They highlight the growing reliance on AI and Blockchain applications and the critical need for secure and trustworthy solutions. This survey aims to bridge the gap in comprehensive studies examining their integration from a security perspective. They propose an analysis of the potential benefits of integrating AI and Blockchain, while also addressing related security concerns.

"On the Integration of Artificial Intelligence and Blockchain Technology: A Perspective About Security - 2024"

Advantages - It proposed an in-depth analysis of the security impact, including mitigation strategies. It also proposed the depiction of the current regulatory environment and its impact on public trust, emphasizing transparency and accountability. The paper addresses the crucial need for security in the growing integration of AI and Blockchain..

Disadvantages - This approach, while providing specific insights, may not offer a complete or unbiased view of the entire research landscape. The paper highlights security concerns, but it does not proposed specific cryptographic solutions.

[3] **Subhi M. Alrubei, Edward Ball, and Jonathan M. Rigelsford** proposed a paper for a secure blockchain platform to support AI-enabled IoT applications at the edge layer. This study designed, developed, and validated a new blockchain protocol and a novel architecture that integrates the advantages of edge computing, artificial intelligence (AI), IoT end-devices, and blockchain. The architecture can monitor the environment, collect data, analyze it, process it using an AI-expert engine, provide predictions and actionable outcomes, and finally share it on a public blockchain platform. The use-case implementation for the proposed system was tested using the COVID-19 pandemic to evaluate its effectiveness**.**

"Secure Blockchain Platform for Supporting AI-Enabled IoT Applications at the Edge Layer June 2023"

Advantages - This platform provides a decentralized, robust, and secure solution for IoT devices. It utilizes edge computing to offer faster data processing and near-real-time actionable outcomes. The platform enhances the security of data with blockchain technology and is both accessible and efficient due to its low-cost and low-power hardware.

Disadvantages - However, the heterogeneous nature of edge and IoT end devices may result in data security issues during transit and storage. Additionally, it requires additional computational power and storage capabilities that some IoT devices may lack.

[4] **Oumaima Fadi, Zkik Karim, El Ghazi Abdellatif, and Boulmalf Mohammed** proposed a system that integrates Blockchain Technology (BT) and Artificial Intelligence (AI) to enhance security and privacy in smart environments. Smart environments consist of sensors, actuators, and computing units, which generate large volumes of data. Traditional data processing systems, such as cloud-based solutions, are prone to security risks and high resource costs. To address these challenges, enterprises are increasingly adopting blockchain technology, a distributed ledger that ensures data reliability and transparency.

"**A Survey on Blockchain and Artificial Intelligence Technologies for Enhancing Security and Privacy in Smart Environments – 2022"**

Advantages - AI-powered anomaly detection enhances security by identifying and mitigating potential cyber threats in real time. It automates various blockchain operations, such as automated sign-ins for nodes, detecting and isolating compromised elements, and predicting potential cybercrime activities. The combination of AI and blockchain ensures transparency, improves data security, and enhances the efficiency of smart environments by reducing fraud and unauthorized access.

Disadvantages - The high computational power required for AI-driven security solutions increases processing costs and energy consumption. Additionally, while blockchain ensures data security, its implementation in IoT networks faces scalability challenges due to the vast amount of data being processed. Privacy concerns also arise as AI models require access to sensitive data for analysis, which could lead to potential data breaches.

[5] **Subhi M. Alrubei, Edward Ball, and Jonathan M. Rigelsford** proposed a decentralized architecture that integrates blockchain technology with Distributed Artificial Intelligence (DAI) in IoT systems. The system utilizes IoT hardware as a distributed neural network, where each IoT device functions as a neuron in the AI model. This architecture leverages blockchain to facilitate secure, trusted interactions between distributed AI nodes, ensuring data integrity and decentralized computation.

"The Use of Blockchain to Support Distributed AI Implementation in IoT Systems – 2022"

Advantages - The integration of blockchain into IoT-based AI systems enhances security and trust by enabling decentralized computation, ensuring that interactions and data exchanges between distributed AI nodes remain tamper-proof and reliable. This approach reduces dependence on centralized cloud processing, thereby lowering communication overhead and improving efficiency.

Disadvantages - However, the implementation of blockchain in distributed AI for IoT comes with challenges. High computational and energy costs make large-scale adoption difficult, particularly for resource-limited IoT devices. The reliance on PoW-based consensus mechanisms further increases power consumption and processing time, impacting efficiency.

[6] **Beatrice Ietto, Kerstin Eisenhut, Robert Muth, Jochen Rabe, Florian Tschorsch** proposed BBBlockchain, a blockchain-based decentralized application (DApp) for citizen participation in urban planning. This system aims to enhance transparency, trust, and accountability in urban development decision-making by securely storing and managing documents such as land-use plans, approval processes, and contracts. The study investigates blockchain's role in improving transparency through timestamping and document management.

"Transparency in Digital-Citizens Interfaces Through Blockchain Technology: BBBlockchain for Participation Processes in Urban Planning – 2019"

Advantages - BBBlockchain provides an immutable and transparent record of urban planning decisions, improving citizen trust and engagement. It ensures secure document storage and reduces risks of corruption by allowing real-time monitoring of government actions.

Disadvantages - Blockchain alone cannot fully ensure transparency, as certain transparency dimensions remain outside its scope. Additionally, the system relies on technical infrastructure like Ethereum, which may introduce scalability and cost concerns.

[7] **Joe Abou Jaoude, Raafat George Saade** proposed a systematic literature review on blockchain technology, analyzing its adoption across various domains. The paper highlights blockchain's evolution from its initial application in cryptocurrency to its impact on industries such as finance, healthcare, energy, government, and the Internet of Things. The study identifies blockchain's key features, including decentralization, transparency, privacy, security, and immutability, as major factors driving its adoption in these fields.

"Blockchain Applications – Usage in Different Domains (2019)"

Advantages - Blockchain provides a decentralized system that removes the need for intermediaries, ensuring transparent and secure transactions. It enhances trust through cryptographic verification, making fraud and data manipulation nearly impossible. Additionally, blockchain enables anonymous yet publicly auditable transactions, fostering efficiency in various industries.

Disadvantages - Despite its benefits, blockchain technology has faced skepticism due to hype-driven fraudulent enterprises. The finance and technology sectors have witnessed erosion of trust, and challenges such as scalability, regulatory concerns, and adoption barriers remain unresolved. Furthermore, the technology requires significant computational resources and lacks standardization across industries.

[8] **Wu, M., Wang, K., Cai, X., Guo, S., Guo, M., & Rong, C.** proposed a comprehensive survey of blockchain technology, analyzing its evolution from cryptocurrency to IoT applications and beyond. The paper structures blockchain technologies into four layers and conducts an extensive study on consensus strategies, networking aspects, and applications across various domain.

"A Comprehensive Survey of Blockchain: From Theory to IoT Applications and Beyond (2019)"

Advantages - Blockchain achieves decentralization through distributed network architecture with ledgers maintained by numerous nodes. It establishes trust without costly third-party authorities through consensus mechanisms. The technology provides transparency as all transactions are public and visible.

Disadvantages - Despite its benefits, blockchain technology faces significant scalability challenges, with Bitcoin supporting only 7 transactions per second (TPS) and Ethereum just 15 TPS. The technology is vulnerable to various security threats including replay attacks, Sybil attacks, and DDoS attacks.

[9] **Sunny Pahlajani, Avinash Kshirsagar, and Vinod Pachghare** conducted a survey on private blockchain consensus algorithms, examining how these mechanisms establish trust in blockchain networks. The paper classifies consensus algorithms into two main categories: voting-based consensus and proof-based consensus, exploring their applications in private blockchain environments.

"Survey on Private Blockchain Consensus Algorithms (2019)"

Advantages - Blockchain operates as a distributed database existing synchronously across multiple network nodes, creating an immutable ledger where new blocks are added irreversibly. This structure makes it impossible to manipulate previously recorded transactions. The technology ensures independence, transparency, and security while reducing costs and risks.

Disadvantages - Despite its benefits, consensus algorithm selection for private blockchains faces limitations in theoretical support and data. The paper highlights that inadequate research exists on selecting suitable consensus mechanisms for private blockchain environments. Additionally, the implementation complexity of consensus algorithms presents challenges for organizations seeking to adopt this technology.

[10] **Khaled Salah, M. Habib Ur Rehman, Nishara Nizamuddin, and Ala Al-Fuqaha** conducted a comprehensive survey on the intersection of blockchain and artificial intelligence, reviewing emerging blockchain applications, platforms, and protocols specifically targeting AI. The paper discusses how these two disruptive technologies can be integrated to create decentralized AI systems that operate on trusted, digitally signed data.

"Blockchain for AI: Review and Open Research Challenges (2019)"

Advantages - Blockchain provides a trusted platform for storing the massive volumes of data required by AI systems. Smart contracts enable programmable governance of transactions among participants involved in AI decision-making processes. The integration creates decentralized learning environments that facilitate secure sharing of knowledge and decision outcomes across autonomous agents. Blockchain's immutable nature ensures that AI decisions cannot be refuted and can be traced, tracked, and verified by all participating entities.

Disadvantages - Conventional blockchain is expensive for storing large amounts of data, with limitations such as Bitcoin's one-megabyte block size restriction. The centralized nature of traditional AI makes data vulnerable to tampering, hacking, and manipulation. Current AI systems often lack guaranteed data provenance and authenticity of sources, which can lead to erroneous, risky, or dangerous decision outcomes.

[11] **Priyanka Bothra, Raja Karmakar, Sanjukta Bhattacharya, and Sayantani De** conducted a comprehensive survey examining how blockchain and artificial intelligence can be integrated to enhance Internet of Things (IoT) systems. The paper discusses existing research that combines these technologies to create automated, secure, and robust IoT models, while highlighting the strengths and limitations of current approaches.

"**How can applications of blockchain and artificial intelligence improve performance of Internet of Things? – A survey (2023)"**

Advantages - The integration of blockchain provides IoT systems with a distributed ledger technology that enables secure, immutable, and anonymous transactions without central authorities. This addresses security vulnerabilities in resource-constrained IoT devices and eliminates the need for third parties in financial transactions. AI implementation in IoT allows for intelligent systems that can automatically adapt to different environments with minimal human intervention, reducing errors and enabling faster decision-making.

Disadvantages - Despite the potential benefits, the integration of blockchain with cloud or edge computing in IoT increases design complexity, making efficient data handling more challenging. The paper notes that achieving minimal data loss while ensuring security constraints remains difficult. Resource limitations of IoT devices continue to pose challenges for implementing computationally intensive blockchain and AI technologies. The survey identifies that thorough research is still needed to fully understand the impacts of integrating these technologies, indicating that current implementations may not be optimized.

CHAPTER 3

THEORETICAL BACKGROUND

3.1 IMPLEMENTATION ENVIRONMENT

The implementation environment of our email app for visually impaired software were stated in the following statements. It was developed by considering the widely used category of devices, mobile phones. Our application can run in Android OS in mobiles. Since, most of the projects were developed for the system purpose. We focused on creating application for mobile users with feasibility. Also, it can be able to run and support on the system (windows OS). Our application won't be able to run on Android OS less than version 5. In windows os, it won't support for versions below window 7. For smooth execution, it requires atleast windows 8 with 4GB RAM and in Android, it requires 4GB RAM with version above 10 for smooth execution. Speech recognition are used separately for both windows and Android. The application requires internet permission to fetch data and perform necessary activities such as authentication, send and receive emails. Storage can be used, if the user wishes to store email seperately. For Android , the application requires access to storage for file attachments in mail.

Hardware Requirements :

- System Device : Mobile or Computer
- Processor (RAM) : 4GB (Both mobile and computer)
- Computer Storage : 512 GB
- Mobile Storage: 32 GB
- Internet Connection

Software Requirements :

- Mobile OS : Android version 10
- Computer OS : Windows 8
- Computer Clock Speed : 4.90 GHz

3.2 SYSTEM ARCHITECTURE

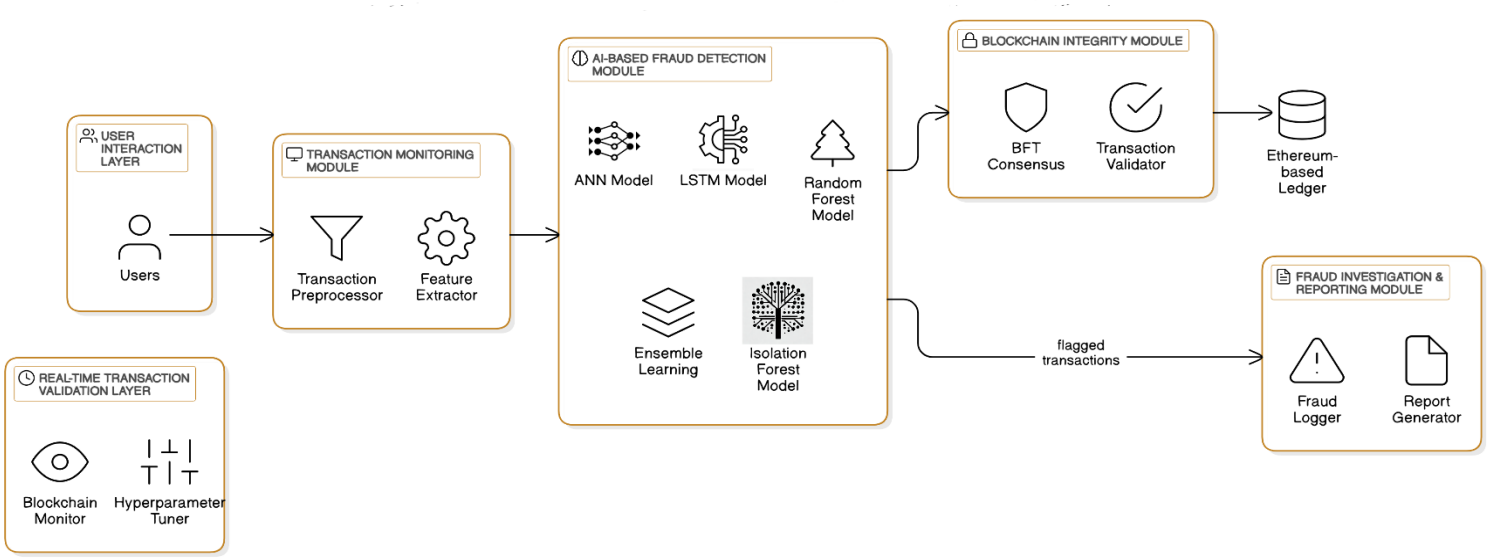


Fig no.3.2.1 System Architecture

The design and structure of this AI-powered blockchain fraud detection system outline a systematic approach to ensuring transaction integrity and security. The architecture integrates machine learning models with blockchain technology to detect fraudulent transactions in real time. The system comprises multiple interconnected modules, including a transaction monitoring module that preprocesses and extracts relevant features from transactions, an AI-based fraud detection module utilizing ANN, LSTM, Random Forest, and Isolation Forest models for anomaly detection, and a blockchain integrity module that employs Byzantine Fault Tolerance (BFT) for secure transaction validation. Flagged transactions undergo further scrutiny in the fraud investigation and reporting module, which logs suspicious activities and generates detailed reports. Additionally, a real-time transaction validation layer continuously monitors blockchain activity and refines fraud detection models for enhanced accuracy. The Ethereum-based ledger ensures transparency and immutability, reinforcing the security and reliability of the entire system.

3.3 PROPOSED METHODOLOGY

The proposed system in this project introduces an AI-Powered Blockchain Integrity and Fraud Detection System that enhances the reliability, transparency, and security of blockchain networks using Artificial Intelligence (AI) techniques. Traditional blockchain systems rely solely on cryptographic algorithms and decentralized consensus mechanisms, making them vulnerable to fraudulent transactions, data tampering, and inefficiency in real-time validation. The proposed system leverages AI-based algorithms to address these limitations, offering real-time anomaly detection, fraud prevention, and adaptive consensus mechanisms.

The system consists of three primary components: Transaction Monitoring Module, AI-Based Fraud Detection Module, and Blockchain Integrity Module. The Transaction Monitoring Module continuously scans incoming transactions, extracting relevant features such as transaction amount, sender-receiver relationship, and frequency of transactions. These extracted features are then passed to the AI-Based Fraud Detection Module, where Support Artificial Neural Network (ANN), Random Forest (RF), Isolation Forest (IF), and Long Short-Term Memory (LSTM) neural networks work collaboratively to classify transactions as legitimate or fraudulent. The final decision is taken based on an ensemble approach, where multiple models vote on the classification result. If a transaction is flagged as fraudulent, it is immediately logged onto the blockchain for transparency, and further actions are taken, such as blocking or requiring additional verification.

3.3.1 Wallet Generation and Management

The wallet module is responsible for creating and managing Bitcoin wallets for users. It provides secure private key generation, transaction signing, and storage management to prevent unauthorized access. The wallet ensures cryptographic security through elliptic curve cryptography (ECC), offering users a secure and reliable platform for handling digital assets. Multi-signature authentication and hierarchical deterministic (HD) wallets are also supported to enhance security and usability.

3.3.2 Data Collection and Preprocessing

- **Data Cleaning:** Removal of missing values, duplicate transactions, and irrelevant features.
- **Outlier Detection:** Identification and removal of anomalous transactions using statistical and AI-based methods.
- **Feature Engineering:** Extraction of key transaction features such as transaction amount, frequency, sender-receiver behavior, and time-based patterns.
- **Feature Scaling and Normalization:** Standardizing transaction features for better model performance.
- **Label Encoding:** Converting categorical features into numerical form for AI model compatibility.

.

3.3.3 AI Model Selection

The following Machine Learning Algorithms are selected to build the hybrid fraud detection system:

Algorithm	**Technique**	**Purpose**
Random Forest (RF)	Supervised	Detects non-linear fraud patterns by combining multiple decision trees.
Artificial Neural Network (ANN)	Supervised	Learns deep transaction patterns for complex fraud detection.
Long Short-Term Memory (LSTM)	Supervised	Captures sequential dependencies in transaction behavior to detect time-based fraud patterns.
Isolation Forest (IF)	Unsupervised	Identifies anomalies and outliers in blockchain transactions.

Table 3.1 AI Model Selection

3.3.4 MODULE DESIGN

DATA FLOW DIAGRAMS

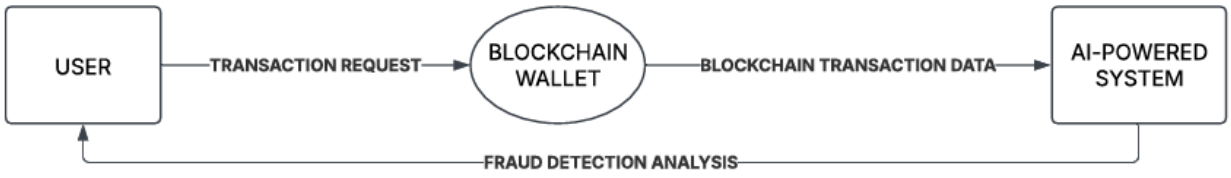


Fig no.3.3.4.1 Level 0 DFD

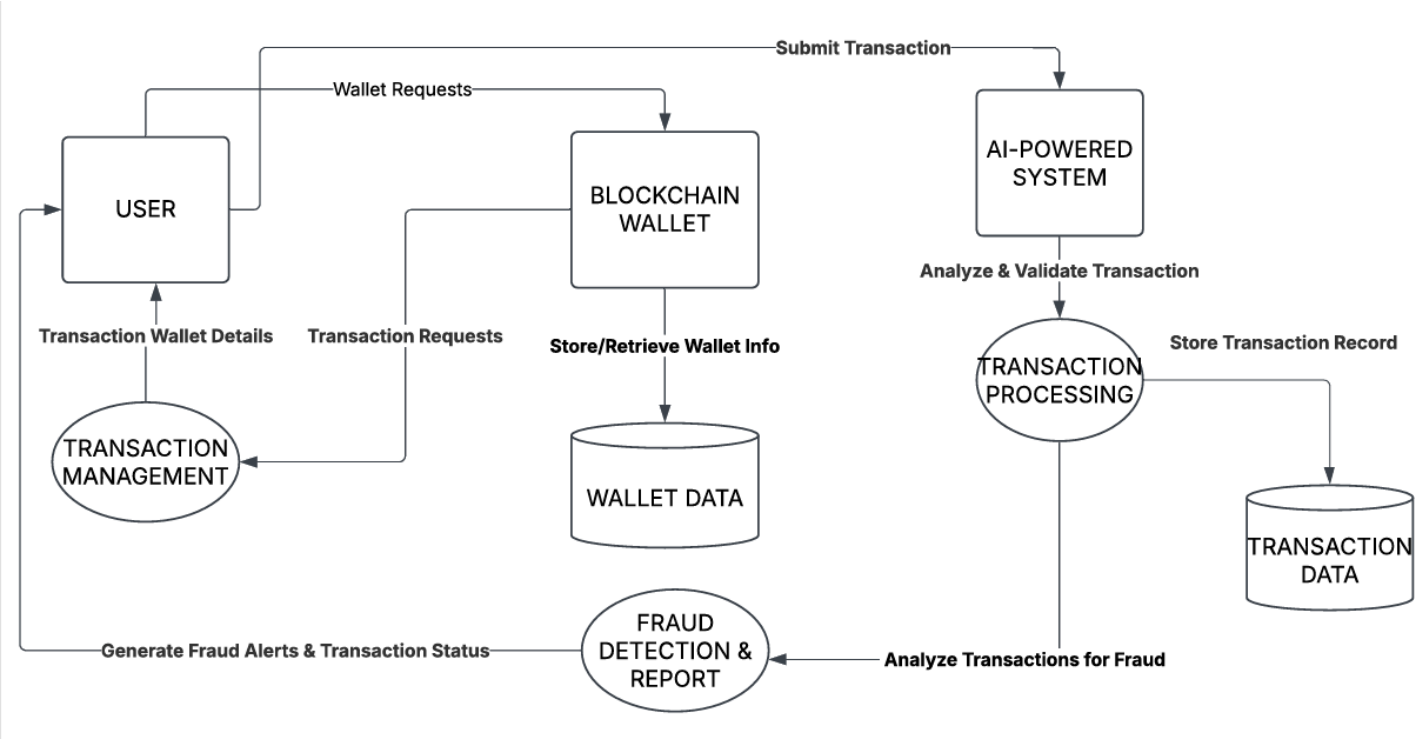


Fig no. 3.3.4.2 Level 1 DFD

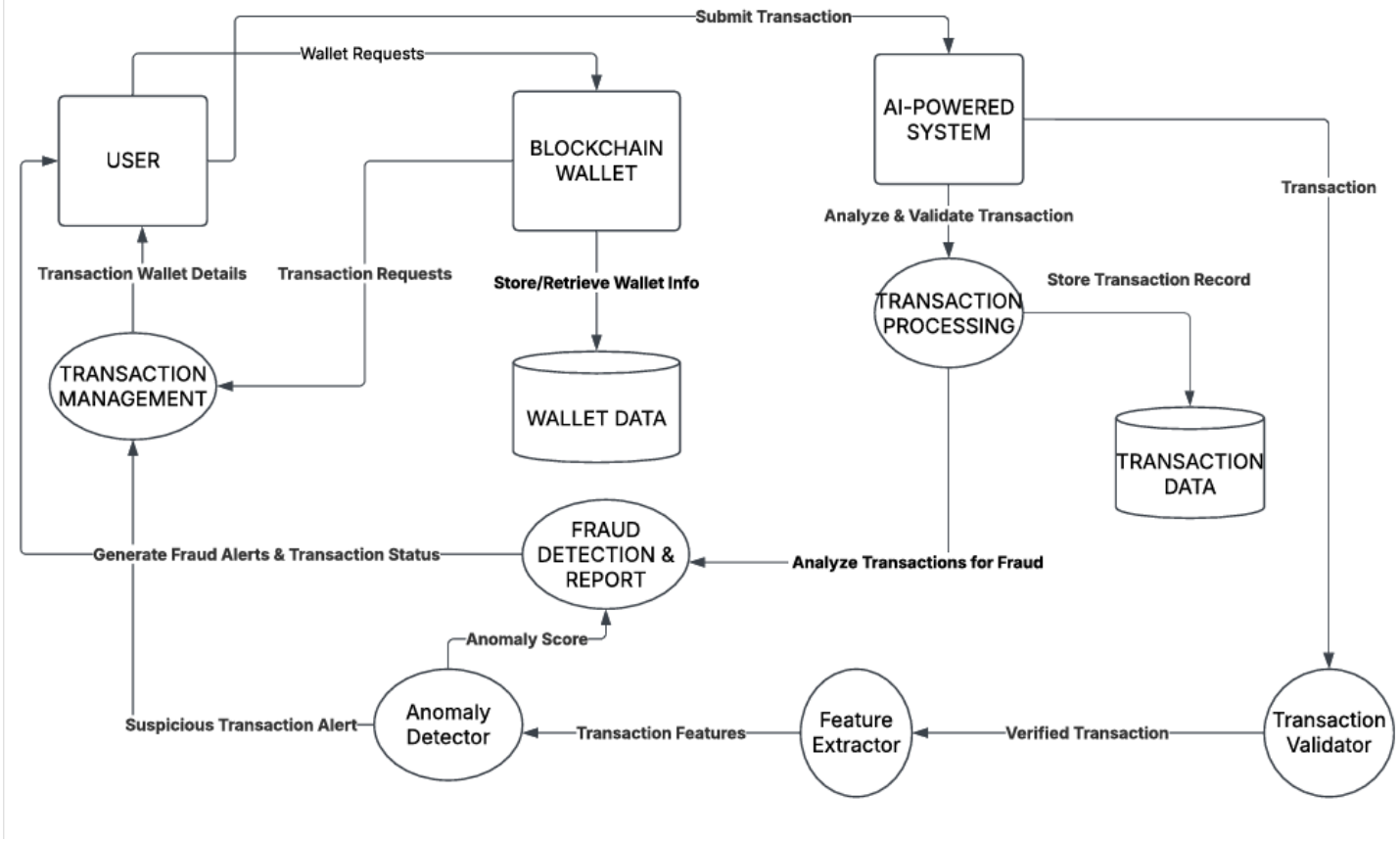


Fig no. 3.3.4.3 Level 2 DFD

3.4 UML DIAGRAMS

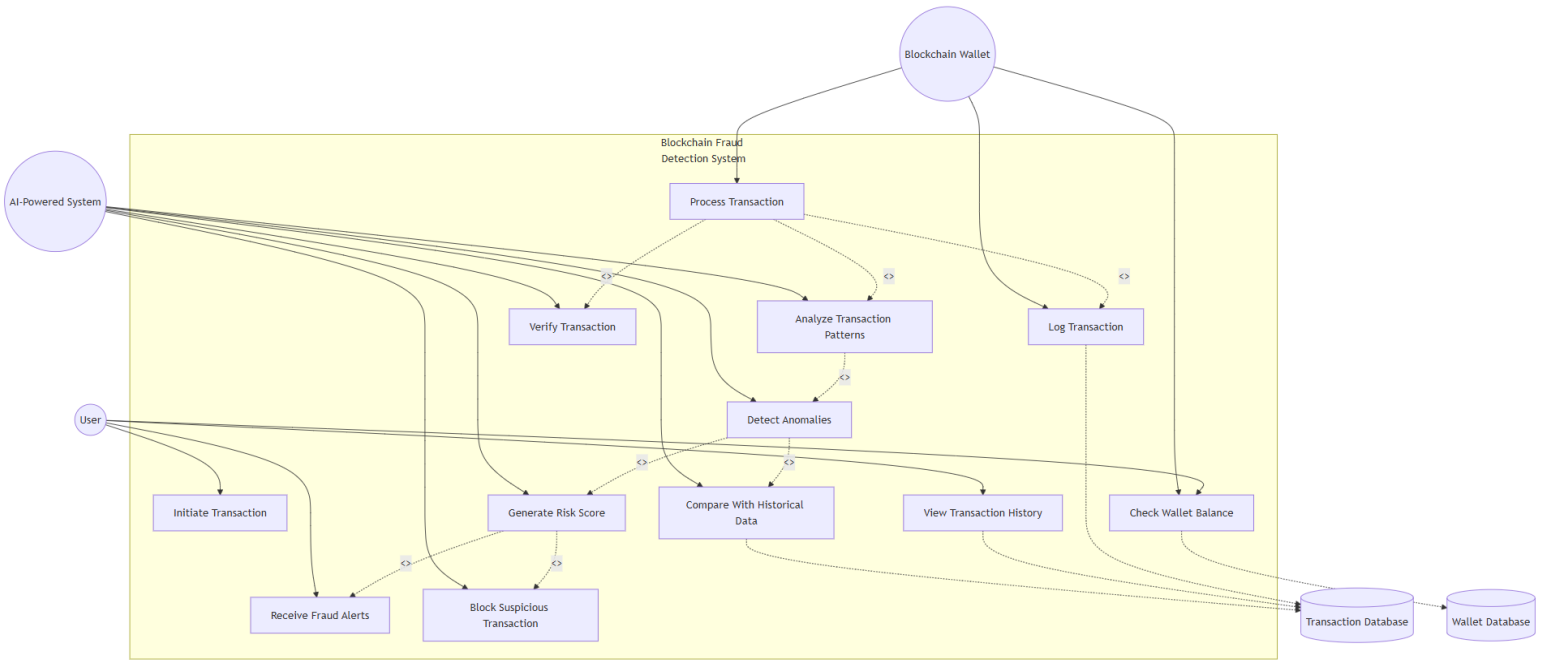


Fig no.3.4.1 Use Case Diagram

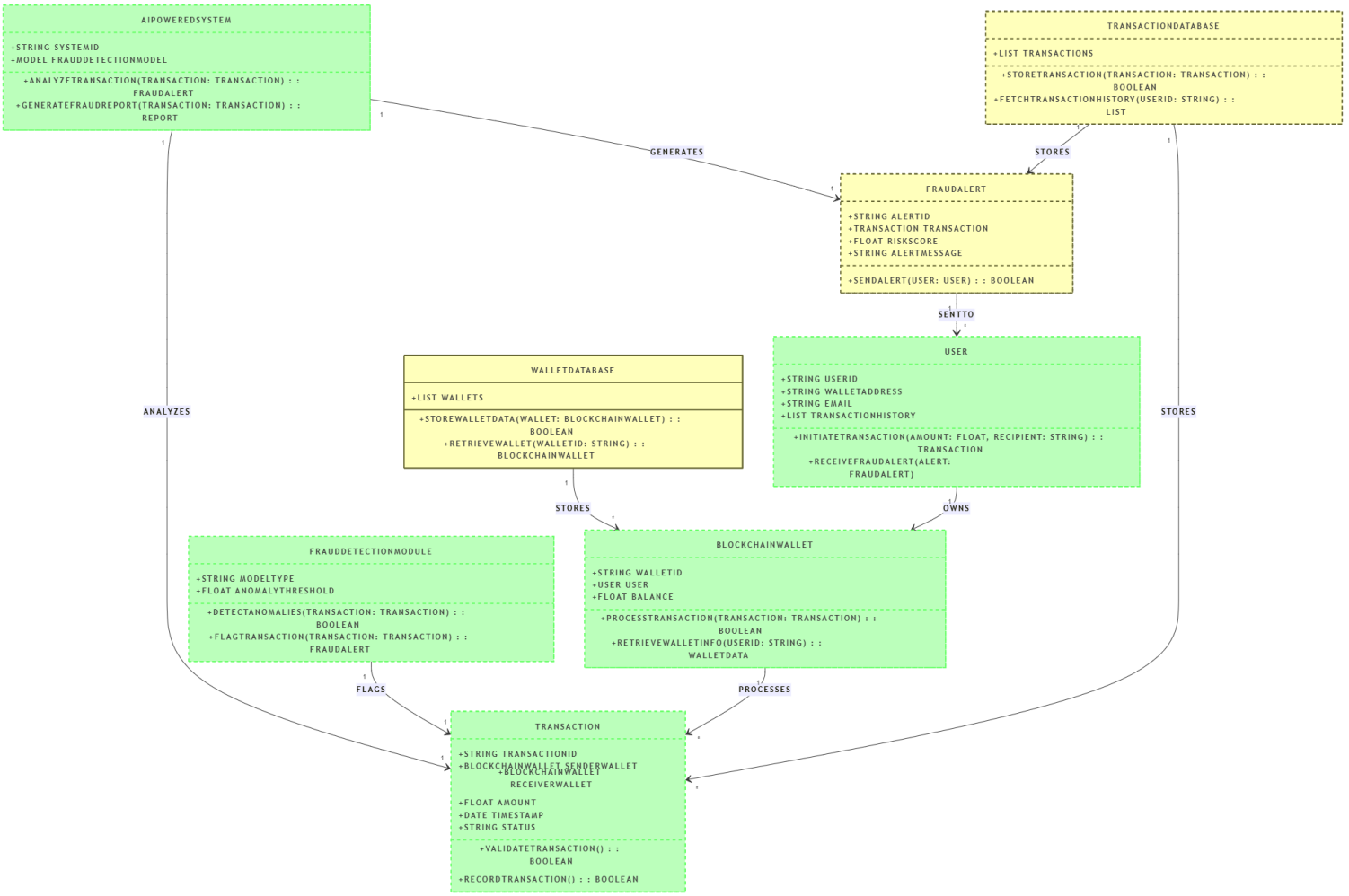


Fig no. 3.4.2 Class Diagram

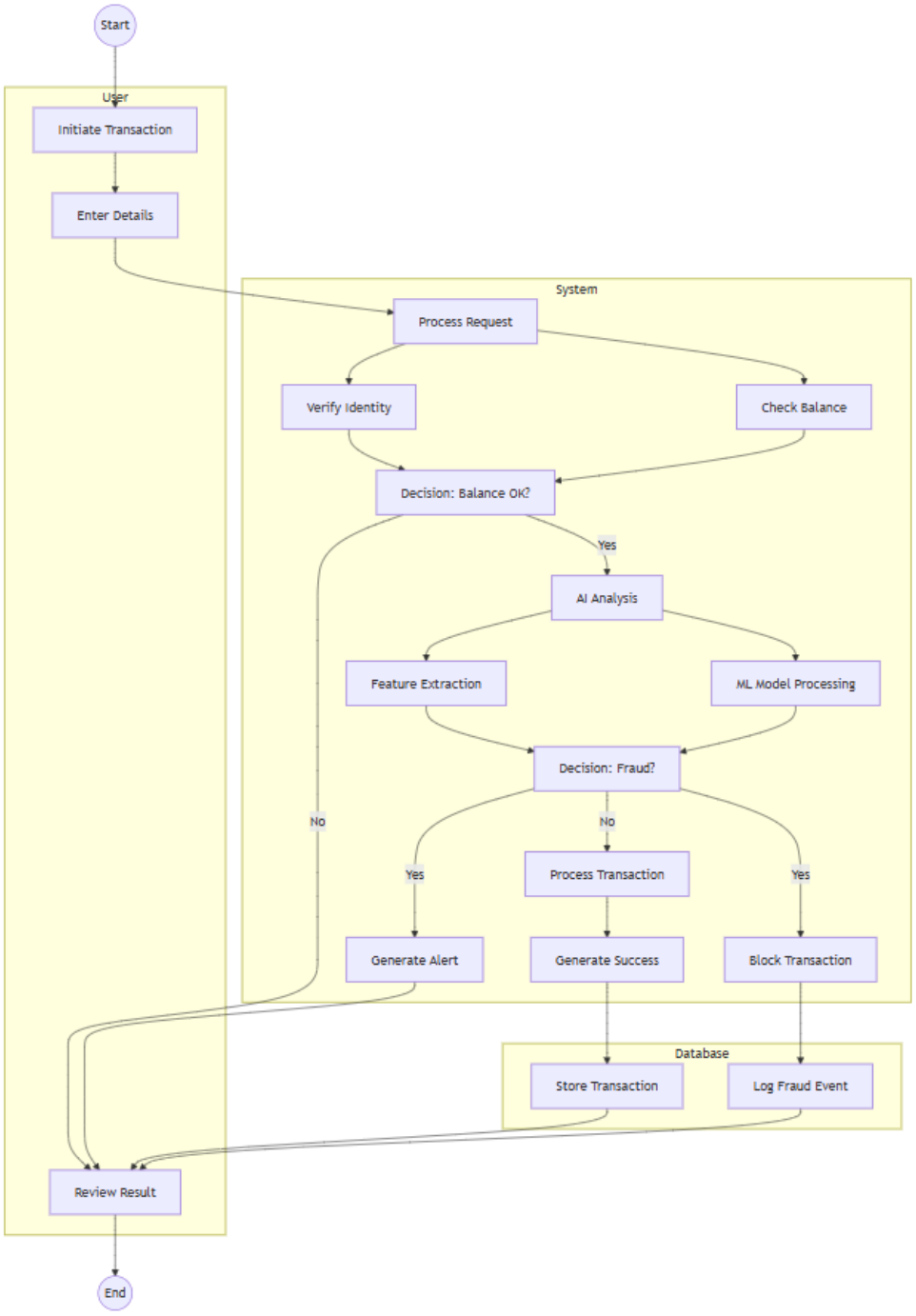


Fig no. 3.4.3 Activity Diagram

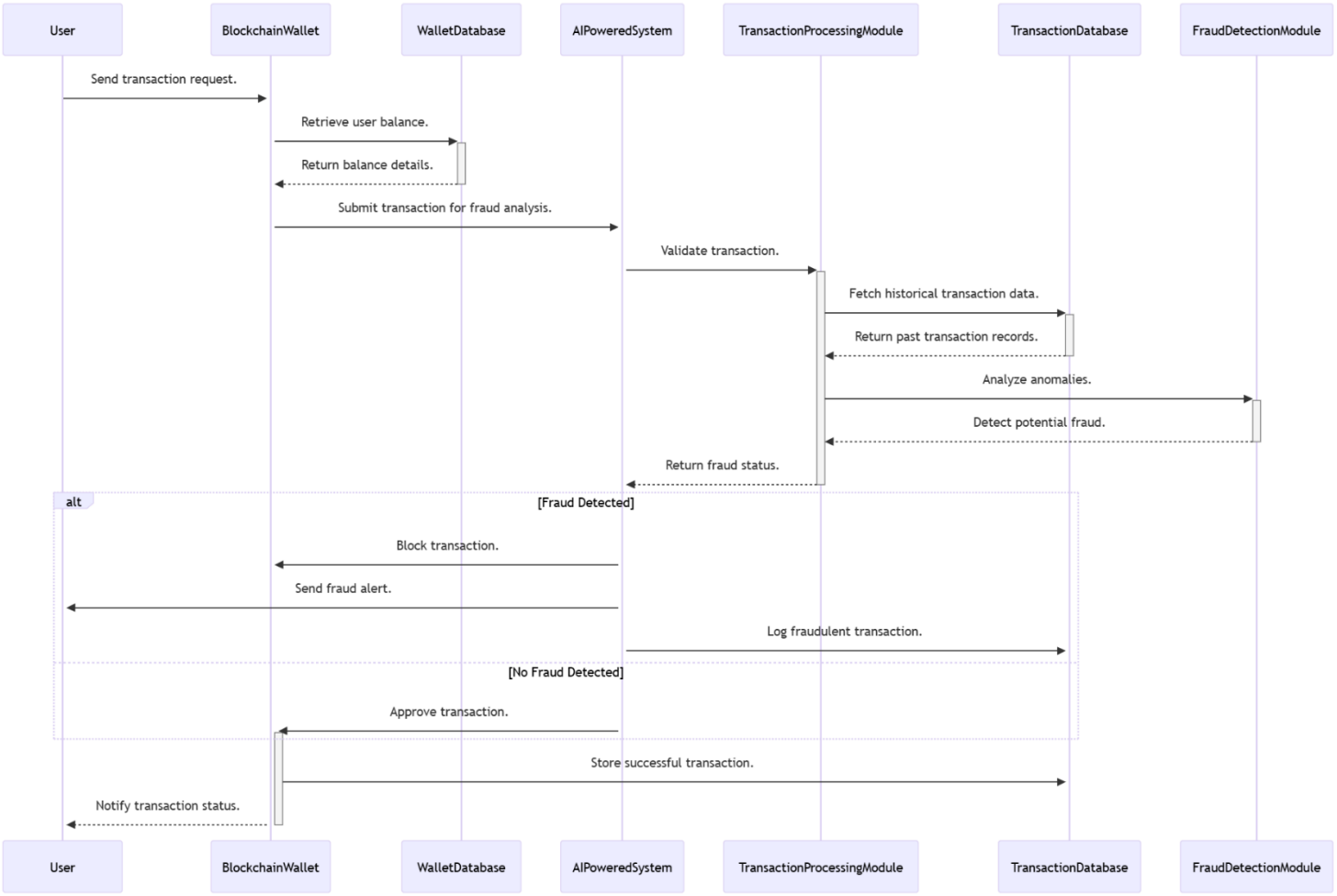


Fig no. 3.4.5 Sequence Diagram

CHAPTER 4

SYSTEM IMPLEMENTATION

4.1. Data Overview

The first phase of the system implementation focuses on data collection and preprocessing, which is critical for ensuring the integrity and quality of the dataset used for model training. This process involves gathering transaction data from various blockchain sources, including historical transaction records and user-generated data. The gathered data undergoes thorough cleaning to remove duplicates and inconsistencies, ensuring that it is accurate and reliable. Missing values are addressed through imputation or removal, and the data is normalized to maintain uniformity across features. Additionally, exploratory data analysis (EDA) is conducted to visualize the distribution and relationships within the dataset, which further informs the subsequent feature engineering phase. This foundational work sets the stage for the effective training of artificial intelligence models in detecting fraudulent transactions.

4.2. AI Model Selection

Once the data is prepared, the project moves into the AI model selection phase, where various machine learning algorithms are evaluated to identify the best performers for the task of fraud detection. The models investigated include Random Forest, Support Vector Machines (SVM), Artificial Neural Networks, and more specialized algorithms like Isolation Forest. Each model is subjected to rigorous testing, using metrics such as accuracy, precision, recall, and F1-score to assess their performance. Cross-validation techniques, particularly k-fold validation, are employed to avoid overfitting and ensure that the selected models generalize well to unseen data. This phase emphasizes the importance of selecting the right algorithm, as the effectiveness of fraud detection directly relies on the chosen model's performance characterist.

4.3. Training and Validation

The third stage encompasses training and validation, which is essential for optimizing the selected models. During this phase, the models are trained on a subset of the dataset, allowing them to learn the underlying patterns that distinguish fraudulent transactions from legitimate ones. The training process involves fine-tuning hyperparameters to enhance model performance. Post-training, the models are validated using a separate validation dataset to assess their predictive capabilities. This includes monitoring various performance metrics and adjusting the models accordingly to strike a balance between precision and recall. By the end of this phase, each model is evaluated not only for its predictive accuracy but also for its robustness in real-world applications, ensuring that they are ready for effective implementation.

4.4. Real-time Transaction Validation

In the final operational phase, trained models are deployed for real-time transaction validation, a critical feature of the system aimed at providing swift and automatic responses to potential fraudulent activities. The application analyzes incoming transaction data, applying the selected machine learning models to classify transactions as either legitimate or fraudulent in real-time. This process incorporates anomaly detection techniques to flag suspicious transactions quickly, allowing for immediate alerts and actions to prevent fraud. The system is designed to balance high-speed performance with accuracy, minimizing false positives while maintaining sensitivity to true fraud cases. Continuous monitoring and feedback mechanisms are in place to refine the model's performance over time, ensuring that the system adapts to emerging patterns of fraudulent behavior effectively.

CHAPTER 5

RESULTS & DISCUSSION

5.1 TESTING

Software Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding, Testing presents aninteresting anomaly for the software engineer

5.1.1 Testing objectives

1. Testing is a process of executing a program with the intent of finding an error
2. A good test case is one that has a probability of finding an as yet undiscovered error
3. A successful test is one that uncovers an undiscovered error

5.1.2 Testing Principles

- All tests should be traceable to end user requirements
- Tests should be planned long before testing begins
- Testing should begin on a small scale and progress towards testing in large
- Exhaustive testing is not possible
- To be most effective testing should be conducted by a independent third party

5.1.3 TEST CASES

S.NO	Test Cases	Test Procedure	Test Input	Expected Result	Actual Result
1	Data Upload	Upload a valid CSV file containing transaction data.	Valid transaction CSV file	Data should be uploaded successfully.	Data uploaded successfully.
2	Invalid Data Upload	Attempt to upload an invalid file type.	Invalid file (e.g., .txt)	Error message indicating invalid file type should appear.	Error message indicating invalid file type appeared
3	Preview Data	After uploading data, click on "Preview" button.	-	Display a table with the first few rows of the data.	Table displayed with data preview
4	Feature Engineering	Execute feature transformation after uploading.	Valid transaction data	New features should be generated and displayed.	New features generated successfully.
5	Model Training	Select model and click "Train."	Selected model	Model should be trained, and performance metrics should be displayed.	Model trained successfully.
6	Training with Insufficient Data	Attempt to train a model with insufficient data.	Inadequate training data	Error message indicating insufficient data should appear.	Error message about insufficient data appeared.

7	Real-time Validation	Submit a transaction for validation.	New transaction data	Transaction should be classified as legitimate or fraudulent.	Transaction classified correctly.
8	Anomaly Detection	Analyze transaction patterns for anomalies.	Historical transaction data	Outliers should be identified and flagged.	Anomalies identified successfully.
9	Consensus Mechanism Check	Test validation process with multiple transactions.	Multiple transactions	All transactions should reach consensus without errors.	Consensus reached without issues.
10	Model Evaluation	Evaluate the trained model on a test dataset.	Test dataset	Performance metrics (accuracy, precision) displayed correctly.	Metrics displayed correctly.

Table 5.1 Test Cases

5.2 RESULT

The primary objective of this project is to develop a robust system for detecting fraud in blockchain transactions, ensuring enhanced security and trust in digital financial operations. The implementation starts with data collection and preprocessing, where transaction data is meticulously gathered and cleaned, laying a solid foundation for analysis. Following this, various AI models are selected based on their efficacy in fraud detection, which are then trained and validated using k-fold cross-validation techniques to ascertain their predictive power. Once the models are refined, real-time transaction validation is activated, allowing for the swift classification of transactions as legitimate or fraudulent. Anomaly detection further strengthens the system by identifying unusual transaction patterns, while enhancements to the consensus mechanism ensure that transactions are validated securely and efficiently. Continuous evaluation and optimization processes are established to refine performance based on user feedback and emerging fraud tactics, creating a dynamic and adaptive framework that effectively safeguards users against fraudulent activities.

CHAPTER 6

CONCLUSION & FUTURE SCOPE

6.1 FUTURE SCOPE

A. Enhanced User Interface:

Improving the user interface to make it more intuitive will facilitate easier navigation for users of all levels. Incorporating customizable dashboards and visual analytics could empower users to interpret data insights effectively.

B. Cross-Chain Transactions Support:

Expanding the system's capabilities to support cross-chain transactions would significantly enhance its utility. This integration would enable users to transact seamlessly across different blockchain networks, broadening the scope of fraud detection.

C. Advanced Machine Learning Technique:

Implementing advanced machine learning techniques, such as deep learning and reinforcement learning, will enhance the system's ability to detect sophisticated fraud patterns. Continuous learning mechanisms could be introduced to adapt to evolving fraud tactics.

D. Privacy-Preserving Technologies:

Adopting privacy-preserving technologies, such as zero-knowledge proofs, would allow users to verify transactions without exposing sensitive information. This enhancement would bolster user confidence and data security.

E. Integration with Financial Institutions:

Collaborating with financial institutions to develop an integrated fraud detection system could lead to more comprehensive financial oversight. Partnerships would enhance the system's legitimacy and scope in identifying and mitigating financial crimes.

F. User Feedback Mechanism:

Establishing a user feedback mechanism will enable continuous improvement based on real-world experiences. Gathering insights from users can trigger updates and refinements to the detection algorithms, ensuring the system evolves alongside emerging threats.

6.2 CONCLUSION

The integration of artificial intelligence (AI) with blockchain technology marks a significant advancement in securing digital financial transactions. This project presents an AI-Powered Blockchain Integrity and Fraud Detection System that utilizes models such as Artificial Neural Networks (ANN), Long Short-Term Memory (LSTM), Random Forest (RF), and Isolation Forest (IF) to effectively identify fraudulent transactions. Each model contributes uniquely, capturing anomalies, sequential patterns, and isolating suspicious activities, while ensemble learning further enhances detection performance. The system also features a secure Bitcoin wallet utilizing elliptic curve cryptography (ECC) for transaction management, along with real-time monitoring and adaptive hyperparameter tuning for optimized fraud detection. Additionally, a Blockchain Integrity Module employs Byzantine Fault Tolerance (BFT) to validate transactions on an Ethereum ledger, ensuring only legitimate entries are recorded. The system showcases high accuracy in detecting fraudulent transactions and presents an intelligent approach to fraud prevention through behavioral analysis. While challenges exist, such as the need for continuous model updates and computational demands, future enhancements may focus on scalability, privacy-preserving techniques, and federated learning to better protect user privacy. Overall, this work illustrates the synergistic potential of AI and blockchain to bolster transaction security and trust in cryptocurrency ecosystems.

APPENDICES

A.1 : SDG GOALS

The integration of our AI-Powered Blockchain Integrity and Fraud Detection System significantly contributes to achieving Sustainable Development Goal (SDG) 9: Industry, Innovation, and Infrastructure in various ways:

1. Innovative Technology Application:

Our system employs cutting-edge AI and machine learning techniques to enhance fraud detection in blockchain transactions. This innovative approach not only bolsters security but also sets a new standard for technological applications in the financial sector.

2. Strengthening Industry Resilience:

By implementing robust fraud detection mechanisms, our system strengthens the resilience of blockchain-based services, reducing vulnerabilities and fostering trust within the industry. This resilience is crucial for attracting investment and driving innovation.

3. Enhancing Financial Infrastructure:

The secure Bitcoin wallet integrated into our system promotes reliable transaction management. By ensuring cryptographic security and privacy, we enhance the financial infrastructure, making it more secure for users and stakeholders alike.

4. Encouraging Public-Private Partnerships:

The system provides opportunities for public and private sector collaboration in enhancing blockchain security. By engaging financial institutions and regulatory bodies, we create a collective approach to combating fraud, which can stimulate further innovations in the industry.

5. Promoting Sustainable Investment:

Our project demonstrates the potential for utilizing technology to address financial crimes, thereby encouraging responsible investments in blockchain and cryptocurrency markets. This alignment with sustainable practices can attract a wider range of investors, including those focused on ethical standards.

By aligning with SDG 9, our system not only addresses immediate fraud detection needs but also promotes a sustainable and innovative ecosystem within the blockchain industry, paving the way for future advancements in technology and infrastructure

A.2 : SOURCE CODE

Main.py :

```
import streamlit as st
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier, IsolationForest
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.neural_network import MLPClassifier
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM, Dropout
import io
import base64
import pickle

st.set_page_config(page_title="Blockchain Fraud Detection", layout="wide")

st.markdown("""
<style>
    .main-title {
        font-size: 42px;
        font-weight: bold;
        color: #1E88E5;
        text-align: center;
        margin-bottom: 30px;
    }
    .section-title {
        font-size: 26px;
        font-weight: bold;
        color: #333;
        margin-top: 20px;
```

```
        margin-bottom: 10px;
    }
    .card {
        background-color: #f8f9fa;
        border-radius: 10px;
        padding: 20px;
        box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
        margin-bottom: 20px;
    }
    .metric-card {
        background-color: #ffffff;
        border-radius: 5px;
        padding: 15px;
        box-shadow: 0 2px 4px rgba(0, 0, 0, 0.05);
        text-align: center;
    }
    .highlight {
        color: #E53935;
        font-weight: bold;
    }
</style>
""", unsafe_allow_html=True)

st.markdown('<div class="main-title">Blockchain Transaction Fraud Detection System</div>',
unsafe_allow_html=True)

tab1, tab2, tab3, tab4, tab5 = st.tabs([" 📊 Data Overview", " 🔍 Feature Engineering", " 🛠️ Model
Training", " 📈 Evaluation", " 🔮 Prediction"])

@st.cache_data
def load_data(uploaded_file):
    if uploaded_file is not None:
        try:
            data = pd.read_csv(uploaded_file)
            return data
        except Exception as e:
            st.error(f"Error loading data: {e}")
            return None
    return None
```

```
def get_download_link(model, filename="blockchain_fraud_model.pkl"):
    buffer = io.BytesIO()
    pickle.dump(model, buffer)
    buffer.seek(0)
    b64 = base64.b64encode(buffer.read()).decode()
    href = f'<a href="data:application/octet-stream;base64,{b64}" download="{filename}">Download Trained Model</a>'
    return href


def preprocess_data(df):
    if 'Index' in df.columns:
        df = df.drop('Index', axis=1)
    if 'Address' in df.columns:
        df = df.drop('Address', axis=1)
    df = df.fillna(0)
    df['sent_received_ratio'] = df['Sent tnx'] / (df['Received Tnx'] + 1)
    df['avg_value_per_tx'] = (df['total Ether sent'] + df['total ether received']) / (df['Sent tnx'] + df['Received Tnx'] + 1)
    df['contract_interaction_rate'] = df['Number of Created Contracts'] / (df['total transactions (including tnx to create contract'] + 1)
    df['unique_address_ratio'] = (df['Unique Sent To Addresses'] + df['Unique Received From Addresses']) / (df['Sent tnx'] + df['Received Tnx'] + 1)
    df['balance_to_transaction_ratio'] = df['total ether balance'] / (df['total Ether sent'] + df['total ether received'] + 1)


    token_cols = ['ERC20 uniq sent token name', 'ERC20 uniq rec token name', 'ERC20 most sent token type', 'ERC20_most_rec_token_type']
    for col in token_cols:
        if col in df.columns:
            df[col] = df[col].astype(str)
            df[col] = df[col].astype('category')


    return df


def extract_features_targets(df):
    X = df.drop('FLAG', axis=1)
    y = df['FLAG']
    categorical_features = X.select_dtypes(include=['category']).columns.tolist()
    numeric_features = X.select_dtypes(include=['float64', 'int64']).columns.tolist()
```

```
    numeric_transformer = StandardScaler()
    categorical_transformer = OneHotEncoder(handle_unknown='ignore')
    preprocessor = ColumnTransformer(
        transformers=[
            ('num', numeric_transformer, numeric_features),
            ('cat', categorical_transformer, categorical_features)
        ],
        remainder='passthrough'
    )
    return X, y, preprocessor

def train_models(X, y, preprocessor):
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
    rf_model = Pipeline([
        ('preprocessor', preprocessor),
        ('classifier', RandomForestClassifier(n_estimators=100, random_state=42))
    ])
    nn_model = Pipeline([
        ('preprocessor', preprocessor),
        ('classifier', MLPClassifier(hidden_layer_sizes=(100, 50), max_iter=500, random_state=42))
    ])
    iso_model = Pipeline([
        ('preprocessor', preprocessor),
        ('classifier', IsolationForest(contamination=0.1, random_state=42))
    ])
    rf_model.fit(X_train, y_train)
    nn_model.fit(X_train, y_train)
    iso_model.fit(X_train)

    if len(X_train) > 50:
        X_train_array = preprocessor.fit_transform(X_train)
        if hasattr(X_train_array, "toarray"):
            X_train_array = X_train_array.toarray()
        X_train_lstm = X_train_array.reshape(X_train_array.shape[0], 1, X_train_array.shape[1])
        X_test_array = preprocessor.transform(X_test)
        if hasattr(X_test_array, "toarray"):
            X_test_array = X_test_array.toarray()
        X_test_lstm = X_test_array.reshape(X_test_array.shape[0], 1, X_test_array.shape[1])
```

```
        lstm_model = Sequential([
            LSTM(units=50, return_sequences=True, input_shape=(1, X_train_array.shape[1])),
            Dropout(0.2),
            LSTM(units=30),
            Dropout(0.2),
            Dense(units=16, activation='relu'),
            Dense(units=1, activation='sigmoid')
        ])
        lstm_model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
        lstm_model.fit(X_train_lstm, y_train, epochs=10, batch_size=32, validation_split=0.1, verbose=0)

        return rf_model, nn_model, iso_model, lstm_model, X_test, y_test, X_test_lstm
    else:
        return rf_model, nn_model, iso_model, None, X_test, y_test, None

def evaluate_models(rf_model, nn_model, iso_model, lstm_model, X_test, y_test, X_test_lstm):
    evaluation_results = {}
    rf_pred = rf_model.predict(X_test)
    rf_accuracy = accuracy_score(y_test, rf_pred)
    evaluation_results['Random Forest'] = (rf_accuracy, confusion_matrix(y_test, rf_pred),
classification_report(y_test, rf_pred, output_dict=True))

    nn_pred = nn_model.predict(X_test)
    nn_accuracy = accuracy_score(y_test, nn_pred)
    evaluation_results['Neural Network'] = (nn_accuracy, confusion_matrix(y_test, nn_pred),
classification_report(y_test, nn_pred, output_dict=True))

    iso_pred = iso_model.predict(X_test)
    iso_pred = np.where(iso_pred == -1, 1, 0)
    iso_accuracy = accuracy_score(y_test, iso_pred)
    evaluation_results['Isolation Forest'] = (iso_accuracy, confusion_matrix(y_test, iso_pred),
classification_report(y_test, iso_pred, output_dict=True))

    if lstm_model is not None and X_test_lstm is not None:
        lstm_pred_prob = lstm_model.predict(X_test_lstm)
        lstm_pred = (lstm_pred_prob > 0.5).astype(int).reshape(-1)
        lstm_accuracy = accuracy_score(y_test, lstm_pred)
        evaluation_results['LSTM'] = (lstm_accuracy, confusion_matrix(y_test, lstm_pred),
```

```
classification_report(y_test, lstm_pred, output_dict=True))

    return evaluation_results

def plot_feature_importance(model, feature_names):
    if hasattr(model, 'named_steps') and hasattr(model.named_steps['classifier'], 'feature_importances_'):
        importances = model.named_steps['classifier'].feature_importances_
        indices = np.argsort(importances)[::-1]
        num_features = len(importances)
        top_n = min(15, num_features)
        top_indices = indices[:top_n]
        top_importances = importances[top_indices]
        feature_names_array = np.array(feature_names)
        if np.max(top_indices) < len(feature_names_array):
            top_features = feature_names_array[top_indices]
        else:
            st.error("Top indices exceed the number of available features.")
            return None
        fig = go.Figure()
        fig.add_trace(
            go.Bar(
                y=top_features,
                x=top_importances,
                orientation='h',
                marker_color='rgba(55, 83, 109, 0.7)',
                marker_line_color='rgba(55, 83, 109, 1.0)',
                marker_line_width=1
            )
        )
        fig.update_layout(
            title='Top Feature Importances',
            xaxis_title='Importance',
            yaxis_title='Feature',
            height=600,
            template='plotly_white'
        )
        return fig
    else:
```

```
        return None

with tab1:
    st.markdown('<div class="section-title">Upload Your Blockchain Transaction Dataset</div>',
unsafe_allow_html=True)
    uploaded_file = st.file_uploader("Choose a CSV file", type="csv")
    use_sample = st.checkbox("Use provided sample data (for demonstration)")

    if use_sample:
        sample_data = {
            'Index': [1] * 1000,
            'Address': ['0x00009277775ac7d0d59eaad8fee3d10ac6c805e8'] * 1000,
            'FLAG': np.random.choice([0, 1], size=1000, p=[0.7, 0.3]),
            'Avg min between sent tnx': np.random.normal(844.26, 100, 1000),
            'Avg min between received tnx': np.random.normal(1093.71, 150, 1000),
            'Time Diff between first and last (Mins)': np.random.normal(704785.63, 10000, 1000),
            'Sent tnx': np.random.poisson(721, 1000),
            'Received Tnx': np.random.poisson(89, 1000),
            'Number of Created Contracts': np.random.poisson(1, 1000),
            'Unique Received From Addresses': np.random.poisson(40, 1000),
            'Unique Sent To Addresses': np.random.poisson(118, 1000),
            'min value received': np.zeros(1000),
            'max value received': np.random.normal(45.81, 5, 1000),
            'avg val received': np.random.normal(6.59, 1, 1000),
            'min val sent': np.zeros(1000),
            'max val sent': np.random.normal(31.22, 3, 1000),
            'avg val sent': np.random.normal(1.20, 0.2, 1000),
            'total transactions (including tnx to create contract': np.random.poisson(810, 1000),
            'total Ether sent': np.random.normal(865.69, 100, 1000),
            'total ether received': np.random.normal(586.47, 70, 1000),
            'total ether balance': np.random.normal(-279.22, 50, 1000),
            'Total ERC20 tnxs': np.random.poisson(265, 1000),
            'ERC20 total Ether received': np.random.normal(35588543.78, 1000000, 1000),
            'ERC20 total ether sent': np.random.normal(35603169.52, 1000000, 1000),
            'ERC20 uniq sent addr': np.random.poisson(30, 1000),
            'ERC20 uniq rec addr': np.random.poisson(54, 1000),
            'ERC20 uniq sent token name': np.random.choice(['Cofoundit', 'Ethereum', 'Bitcoin', 'Tether'],
1000),
            'ERC20 uniq rec token name': np.random.choice(['Numeraire', 'Ethereum', 'Bitcoin', 'Tether'],
```

```
1000),
        'ERC20 most sent token type': np.random.choice(['Cofoundit', 'Ethereum', 'Bitcoin', 'Tether'],
1000),
        'ERC20_most_rec_token_type': np.random.choice(['Numeraire', 'Ethereum', 'Bitcoin', 'Tether'],
1000)
    }
    df = pd.DataFrame(sample_data)
elif uploaded_file is not None:
    df = load_data(uploaded_file)
else:
    st.info("Please upload a CSV file or use the sample data to proceed.")
    df = None

if df is not None:
    st.markdown('<div class="section-title">Dataset Overview</div>', unsafe_allow_html=True)
    st.markdown('<div class="card">', unsafe_allow_html=True)
    col1, col2, col3, col4 = st.columns(4)

    with col1:
        st.markdown('<div class="metric-card">', unsafe_allow_html=True)
        st.metric("Total Transactions", len(df))
        st.markdown('</div>', unsafe_allow_html=True)

    with col2:
        st.markdown('<div class="metric-card">', unsafe_allow_html=True)
        if 'FLAG' in df.columns:
            fraud_count = df['FLAG'].sum()
            fraud_percent = (fraud_count / len(df)) * 100
            st.metric("Fraudulent Transactions", f"{fraud_count} ({fraud_percent:.2f}%)")
        else:
            st.warning("FLAG column not found in dataset")
        st.markdown('</div>', unsafe_allow_html=True)

    with col3:
        st.markdown('<div class="metric-card">', unsafe_allow_html=True)
        if 'total Ether sent' in df.columns:
            total_ether = df['total Ether sent'].sum()
            st.metric("Total Ether Sent", f"{total_ether:.2f} ETH")
        else:
```

```
            st.warning("total Ether sent column not found")
        st.markdown('</div>', unsafe_allow_html=True)

    with col4:
        st.markdown('<div class="metric-card">', unsafe_allow_html=True)
        if 'total transactions (including tnx to create contract' in df.columns:
            avg_tx = df['total transactions (including tnx to create contract'].mean()
            st.metric("Avg Transactions per Address", f"{avg_tx:.2f}")
        else:
            st.warning("total transactions column not found")
        st.markdown('</div>', unsafe_allow_html=True)

    st.markdown('</div>', unsafe_allow_html=True)
    st.markdown('<div class="section-title">Data Preview</div>', unsafe_allow_html=True)
    st.dataframe(df.head())
    st.markdown('<div class="section-title">Basic Statistics</div>', unsafe_allow_html=True)
    st.write(df.describe())
    st.markdown('<div class="section-title">Transaction Distributions</div>',
unsafe_allow_html=True)
    col1, col2 = st.columns(2)

    with col1:
        if 'Sent tnx' in df.columns and 'Received Tnx' in df.columns:
            fig = px.histogram(df, x='Sent tnx', color_discrete_sequence=['#1E88E5'])
            fig.update_layout(title='Distribution of Sent Transactions', template='plotly_white')
            st.plotly_chart(fig, use_container_width=True)

    with col2:
        if 'Sent tnx' in df.columns and 'Received Tnx' in df.columns:
            fig = px.histogram(df, x='Received Tnx', color_discrete_sequence=['#43A047'])
            fig.update_layout(title='Distribution of Received Transactions', template='plotly_white')
            st.plotly_chart(fig, use_container_width=True)

    st.markdown('<div class="section-title">Feature Correlation Matrix</div>',
unsafe_allow_html=True)
    numeric_df = df.select_dtypes(include=['float64', 'int64'])
    corr = numeric_df.corr()
    fig = px.imshow(corr, text_auto='.2f', aspect="auto", color_continuous_scale='RdBu_r')
    fig.update_layout(title='Correlation Matrix of Numeric Features', height=800, width=800)
```

```
        st.plotly_chart(fig)

with tab2:
    if df is not None:
        st.markdown('<div class="section-title">Feature Engineering</div>', unsafe_allow_html=True)
        processed_df = preprocess_data(df)
        st.markdown('<div class="card">', unsafe_allow_html=True)
        st.markdown('### Engineered Features')
        st.write("The following features have been created to improve model performance:")
        col1, col2 = st.columns(2)

        with col1:
            st.markdown('- **sent_received_ratio**: Ratio of sent transactions to received transactions')
            st.markdown('- **avg_value_per_tx**: Average value per transaction')
            st.markdown('- **contract_interaction_rate**: Rate of contract creation relative to all transactions')

        with col2:
            st.markdown('- **unique_address_ratio**: Ratio of unique addresses to total transactions')
            st.markdown('- **balance_to_transaction_ratio**: Ratio of balance to total transaction value')

        st.markdown('</div>', unsafe_allow_html=True)
        st.markdown('### Processed Dataset')
        st.dataframe(processed_df.head())
        st.markdown('<div class="section-title">Feature Distributions by Class</div>', unsafe_allow_html=True)

        if 'FLAG' in processed_df.columns:
            numeric_cols = processed_df.select_dtypes(include=['float64', 'int64']).columns.tolist()
            if 'FLAG' in numeric_cols:
                numeric_cols.remove('FLAG')

            selected_feature = st.selectbox('Select a feature to visualize distribution by class:', numeric_cols)
            fig = px.histogram(processed_df, x=selected_feature, color='FLAG',
                               barmode='overlay', color_discrete_sequence=['#4CAF50', '#F44336'])
            fig.update_layout(title=f'Distribution of {selected_feature} by Class',
                              xaxis_title=selected_feature,
                              yaxis_title='Count',
                              template='plotly_white')
```

```
            st.plotly_chart(fig, use_container_width=True)
            st.markdown('<div class="section-title">Feature Pair Visualization</div>',
unsafe_allow_html=True)
            col1, col2 = st.columns(2)

            with col1:
                feature_x = st.selectbox('Select X-axis feature:', numeric_cols, index=0)

            with col2:
                default_index = 1 if len(numeric_cols) > 1 else 0
                feature_y = st.selectbox('Select Y-axis feature:', numeric_cols, index=default_index)

            fig = px.scatter(processed_df, x=feature_x, y=feature_y, color='FLAG',
                             opacity=0.7, color_discrete_sequence=['#4CAF50', '#F44336'])
            fig.update_layout(title=f'{feature_x} vs {feature_y} by Class',
                              xaxis_title=feature_x,
                              yaxis_title=feature_y,
                              template='plotly_white')
            st.plotly_chart(fig, use_container_width=True)

with tab3:
    if df is not None:
        st.markdown('<div class="section-title">Model Training</div>', unsafe_allow_html=True)
        st.markdown('<div class="card">', unsafe_allow_html=True)
        st.markdown('### Select Models for Training')
        col1, col2 = st.columns(2)

        with col1:
            use_rf = st.checkbox('Random Forest Classifier', value=True)
            use_nn = st.checkbox('Neural Network (MLP)', value=True)

        with col2:
            use_iso = st.checkbox('Isolation Forest (Anomaly Detection)', value=True)
            use_lstm = st.checkbox('LSTM Neural Network', value=True)

        st.markdown('</div>', unsafe_allow_html=True)
        st.markdown('<div class="card">', unsafe_allow_html=True)
        st.markdown('### Model Descriptions')
        col1, col2 = st.columns(2)
```

```
    with col1:
        st.markdown('**Random Forest Classifier**')
        st.markdown('- Ensemble learning method using multiple decision trees')
        st.markdown('- Effective for classification tasks with tabular data')
        st.markdown('- Provides feature importance for interpretability')

        st.markdown('**Neural Network (MLP)**')
        st.markdown('- Multi-layer perceptron with hidden layers')
        st.markdown('- Can capture complex non-linear relationships')
        st.markdown('- Works well with standardized numeric features')

    with col2:
        st.markdown('**Isolation Forest**')
        st.markdown('- Unsupervised anomaly detection algorithm')
        st.markdown('- Identifies outliers by isolation')
        st.markdown('- Works well for fraud detection where anomalies are rare')

        st.markdown('**LSTM Neural Network**')
        st.markdown('- Long Short-Term Memory network')
        st.markdown('- Captures temporal patterns in transaction sequences')
        st.markdown('- Effective for time-series transaction data')

    st.markdown('</div>', unsafe_allow_html=True)

    if st.button('Start Model Training'):
        with st.spinner('Preprocessing data...'):
            processed_df = preprocess_data(df)

            if 'FLAG' in processed_df.columns:
                X = processed_df.drop('FLAG', axis=1)
                y = processed_df['FLAG']

                preprocessor = ColumnTransformer(
                    transformers=[
                        ('num', StandardScaler(), X.select_dtypes(include=['float64', 'int64']).columns),
                        ('cat', OneHotEncoder(handle_unknown='ignore'),
X.select_dtypes(include=['category']).columns)
                    ],
```

```
            remainder='passthrough'
        )

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

        X_train_transformed = preprocessor.fit_transform(X_train)

        feature_names = preprocessor.get_feature_names_out()

        rf_model, nn_model, iso_model, lstm_model, X_test, y_test, X_test_lstm = train_models(X, y, preprocessor)

        st.session_state['models'] = {
            'Random Forest': rf_model,
            'Neural Network': nn_model,
            'Isolation Forest': iso_model,
            'LSTM': lstm_model
        }
        st.session_state['feature_names'] = feature_names
        st.session_state['test_data'] = (X_test, y_test)
        st.session_state['X_test_lstm'] = X_test_lstm
        st.session_state['preprocessor'] = preprocessor

        st.success("All selected models have been trained successfully! Go to the Evaluation tab to see the results.")
    else:
        st.error("FLAG column not found in the dataset. This column is required for training the models.")

with tab4:
    st.markdown('<div class="section-title">Model Evaluation</div>', unsafe_allow_html=True)

    if 'models' in st.session_state and 'feature_names' in st.session_state and 'test_data' in st.session_state and 'X_test_lstm' in st.session_state:
        models = st.session_state['models']
        feature_names = st.session_state['feature_names']
        X_test, y_test = st.session_state['test_data']
        X_test_lstm = st.session_state['X_test_lstm']
```

```
        st.markdown('<div class="card">', unsafe_allow_html=True)
        st.markdown('### Model Comparison')

        evaluation_results = evaluate_models(
            models['Random Forest'],
            models['Neural Network'],
            models['Isolation Forest'],
            models.get('LSTM', None),
            X_test,
            y_test,
            X_test_lstm
        )

        model_names = list(evaluation_results.keys())
        accuracies = [results[0] for results in evaluation_results.values()]
        fig = px.bar(
            x=model_names,
            y=accuracies,
            labels={'x': 'Model', 'y': 'Accuracy'},
            color=accuracies,
            color_continuous_scale='Viridis',
            title='Model Accuracy Comparison'
        )
        fig.update_layout(template='plotly_white')
        st.plotly_chart(fig, use_container_width=True)

        if accuracies:
            best_model_index = np.argmax(accuracies)
            best_model = model_names[best_model_index]
            st.markdown(f'#### Best Performing Model: <span class="highlight">{best_model}</span>
with accuracy {accuracies[best_model_index]:.4f}', unsafe_allow_html=True)
        else:
            st.warning("No valid accuracy values found for comparison.")
        st.markdown('</div>', unsafe_allow_html=True)

        selected_model = st.selectbox('Select a model for detailed evaluation:', model_names)

        if selected_model in evaluation_results:
            results = evaluation_results[selected_model]
```

```
st.markdown('### Confusion Matrix')
conf_matrix = results[1]

fig = px.imshow(
    conf_matrix,
    text_auto=True,
    labels=dict(x="Predicted", y="Actual", color="Count"),
    x=['Normal (0)', 'Fraud (1)'],
    y=['Normal (0)', 'Fraud (1)'],
    color_continuous_scale='Blues'
)
fig.update_layout(title=f'Confusion Matrix - {selected_model}', template='plotly_white')
st.plotly_chart(fig, use_container_width=True)

st.markdown('### Classification Report')
report = results[2]
report_df = pd.DataFrame(report).transpose()
if 'support' in report_df.columns:
    report_df = report_df.drop('support', axis=1)

if 'accuracy' in report_df.index:
    accuracy_row = report_df.loc[['accuracy']]
    report_df = report_df.drop('accuracy')
    keep_rows = [str(i) for i in range(2)] + ['macro avg']
    report_df = report_df.loc[report_df.index.intersection(keep_rows)]
    report_df = report_df.rename(index={'0': 'Normal Transactions', '1': 'Fraud Transactions'})

st.dataframe(report_df.style.format("{:.4f}"))

if selected_model == 'Random Forest' and 'classifier' in models[selected_model].named_steps:
    st.markdown('### Feature Importance')
    rf_model = models[selected_model]
    importances = rf_model.named_steps['classifier'].feature_importances_
    indices = np.argsort(importances)[::-1]
    top_n = min(15, len(feature_names))
    top_indices = indices[:top_n]
    top_importances = importances[top_indices]
    top_features = feature_names[top_indices]
```

```
            fig = px.bar(
                x=top_importances,
                y=top_features,
                orientation='h',
                labels={'x': 'Importance', 'y': 'Feature'},
                title=f'Top {top_n} Feature Importances - {selected_model}',
                color=top_importances,
                color_continuous_scale='Viridis'
            )
            fig.update_layout(yaxis={'categoryorder': 'total ascending'}, template='plotly_white')
            st.plotly_chart(fig, use_container_width=True)

        if selected_model in ['Random Forest', 'Neural Network', 'LSTM']:
            st.markdown('### ROC Curve')
            if hasattr(models[selected_model], 'predict_proba'):
                y_probs = models[selected_model].predict_proba(X_test)[:, 1]
                from sklearn.metrics import roc_curve, auc
                fpr, tpr, thresholds = roc_curve(y_test, y_probs)
                roc_auc = auc(fpr, tpr)
                fig = go.Figure()
                fig.add_trace(go.Scatter(x=fpr, y=tpr, mode='lines', name=f'ROC Curve (AUC = {roc_auc:.4f})'))
                fig.add_trace(go.Scatter(x=[0, 1], y=[0, 1], mode='lines', name='Random Classifier', line=dict(dash='dash')))
                fig.update_layout(
                    title=f'ROC Curve - {selected_model}',
                    xaxis_title='False Positive Rate',
                    yaxis_title='True Positive Rate',
                    template='plotly_white',
                    legend=dict(yanchor="bottom", y=0.01, xanchor="right", x=0.99)
                )
                st.plotly_chart(fig, use_container_width=True)

            if selected_model == 'LSTM':
                if 'history' in results:
                    st.markdown('### LSTM Training History')
                    history = results['history']
                    fig = go.Figure()
```

```
                fig.add_trace(go.Scatter(y=history.history['loss'], name='Training Loss'))
                fig.add_trace(go.Scatter(y=history.history['val_loss'], name='Validation Loss'))
                fig.update_layout(
                    title='LSTM Training and Validation Loss',
                    xaxis_title='Epoch',
                    yaxis_title='Loss',
                    template='plotly_white'
                )
                st.plotly_chart(fig, use_container_width=True)

            st.markdown('### Export Model')
            st.markdown('You can download the trained model for later use:')
            st.markdown(get_download_link(models[selected_model], f'{selected_model.lower().replace(" ", "_")}_model.pkl'), unsafe_allow_html=True)
    else:
        st.info("No models have been trained yet. Please go to the Model Training tab to train models.")

with tab5:
    st.markdown('<div class="section-title">Fraud Prediction</div>', unsafe_allow_html=True)
    st.markdown('<div class="card">', unsafe_allow_html=True)
    st.markdown('### Make Predictions on New Data')

    if 'models' in st.session_state and 'preprocessor' in st.session_state:
        models = st.session_state['models']
        preprocessor = st.session_state['preprocessor']

        new_data_file = st.file_uploader("Upload new blockchain transaction data for prediction", type="csv")
        use_sample_for_pred = st.checkbox("Use sample data for prediction demonstration")
        pred_df = None

        if use_sample_for_pred:
            sample_data = {
                'Address': ['0x00009277775ac7d0d59eaad8fee3d10ac6c805e8'],
                'Avg min between sent tnx': [844.26],
                'Avg min between received tnx': [1093.71],
                'Time Diff between first and last (Mins)': [704785.63],
                'Sent tnx': [721],
                'Received Tnx': [89],
```

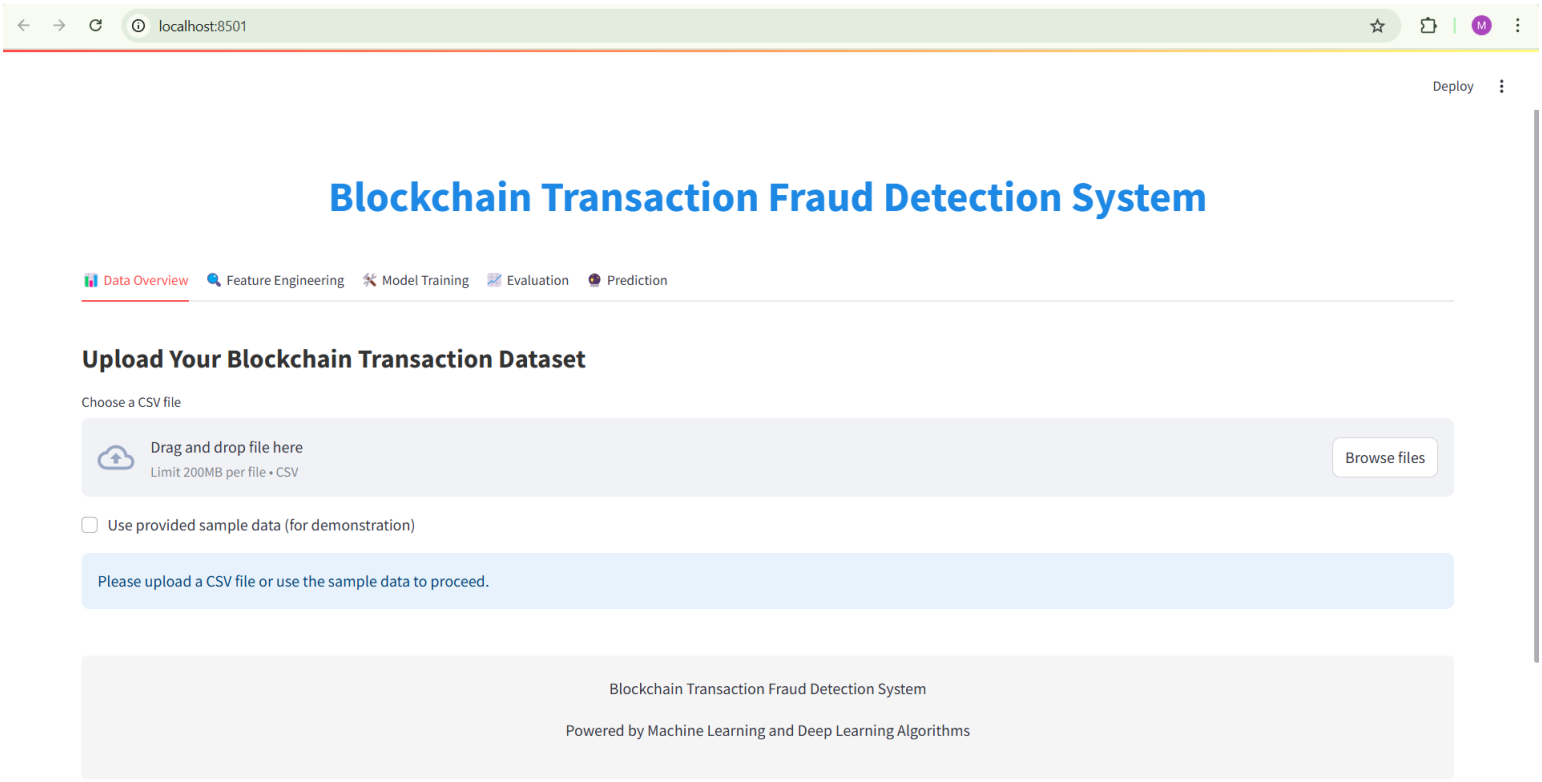
```
        'Number of Created Contracts': [1],
        'Unique Received From Addresses': [40],
        'Unique Sent To Addresses': [118],
        'min value received': [0],
        'max value received': [45.81],
        'avg val received': [6.59],
        'min val sent': [0],
        'max val sent': [31.22],
        'avg val sent': [1.20],
        'total transactions (including tnx to create contract': [810],
        'total Ether sent': [865.69],
        'total ether received': [586.47],
        'total ether balance': [-279.22],
        'Total ERC20 tnxs': [265],
        'ERC20 total Ether received': [35588543.78],
        'ERC20 total ether sent': [35603169.52],
        'ERC20 uniq sent addr': [30],
        'ERC20 uniq rec addr': [54],
        'ERC20 uniq sent token name': ['Cofoundit'],
        'ERC20 uniq rec token name': ['Numeraire'],
        'ERC20 most sent token type': ['Cofoundit'],
        'ERC20_most_rec_token_type': ['Numeraire']
    }
    pred_df = pd.DataFrame(sample_data)
elif new_data_file is not None:
    pred_df = load_data(new_data_file)

if pred_df is not None:
    st.dataframe(pred_df.head())
    processed_pred_df = preprocess_data(pred_df)
    selected_model_name = st.selectbox('Select a model for prediction:', list(models.keys()))

    if st.button('Run Prediction'):
        with st.spinner('Making predictions...'):
            selected_model = models[selected_model_name]

            if selected_model_name == 'LSTM':
                if 'FLAG' in processed_pred_df.columns:
```

```
                processed_pred_df = processed_pred_df.drop('FLAG', axis=1)
            pred_data_array = preprocessor.transform(processed_pred_df)
            if hasattr(pred_data_array, "toarray"):
                pred_data_array = pred_data_array.toarray()
            pred_data_lstm = pred_data_array.reshape(pred_data_array.shape[0], 1,
pred_data_array.shape[1])
            pred_probs = selected_model.predict(pred_data_lstm)
            predictions = (pred_probs > 0.5).astype(int).reshape(-1)
            pred_probs = pred_probs.reshape(-1)
        else:
            if 'FLAG' in processed_pred_df.columns:
                processed_pred_df = processed_pred_df.drop('FLAG', axis=1)

            if hasattr(selected_model, 'predict_proba'):
                pred_probs = selected_model.predict_proba(processed_pred_df)[:, 1]
                predictions = (pred_probs > 0.5).astype(int)
            else:
                predictions = selected_model.predict(processed_pred_df)
                if selected_model_name == 'Isolation Forest':
                    predictions = np.where(predictions == -1, 1, 0)
                pred_probs = np.where(predictions == 1, 0.9, 0.1)

        result_df = pred_df.copy()
        result_df['Fraud_Prediction'] = predictions
        result_df['Fraud_Probability'] = pred_probs

        st.markdown('### Prediction Results')
        col1, col2 = st.columns(2)

        with col1:
            st.metric("Total Transactions", len(result_df))
            st.metric("Flagged as Fraud", int(result_df['Fraud_Prediction'].sum()))

        with col2:
            fraud_percent = (result_df['Fraud_Prediction'].sum() / len(result_df)) * 100
            st.metric("Fraud Percentage", f"{fraud_percent:.2f}%")
            avg_fraud_prob = result_df['Fraud_Probability'].mean() * 100
            st.metric("Average Fraud Probability", f"{avg_fraud_prob:.2f}%")
```

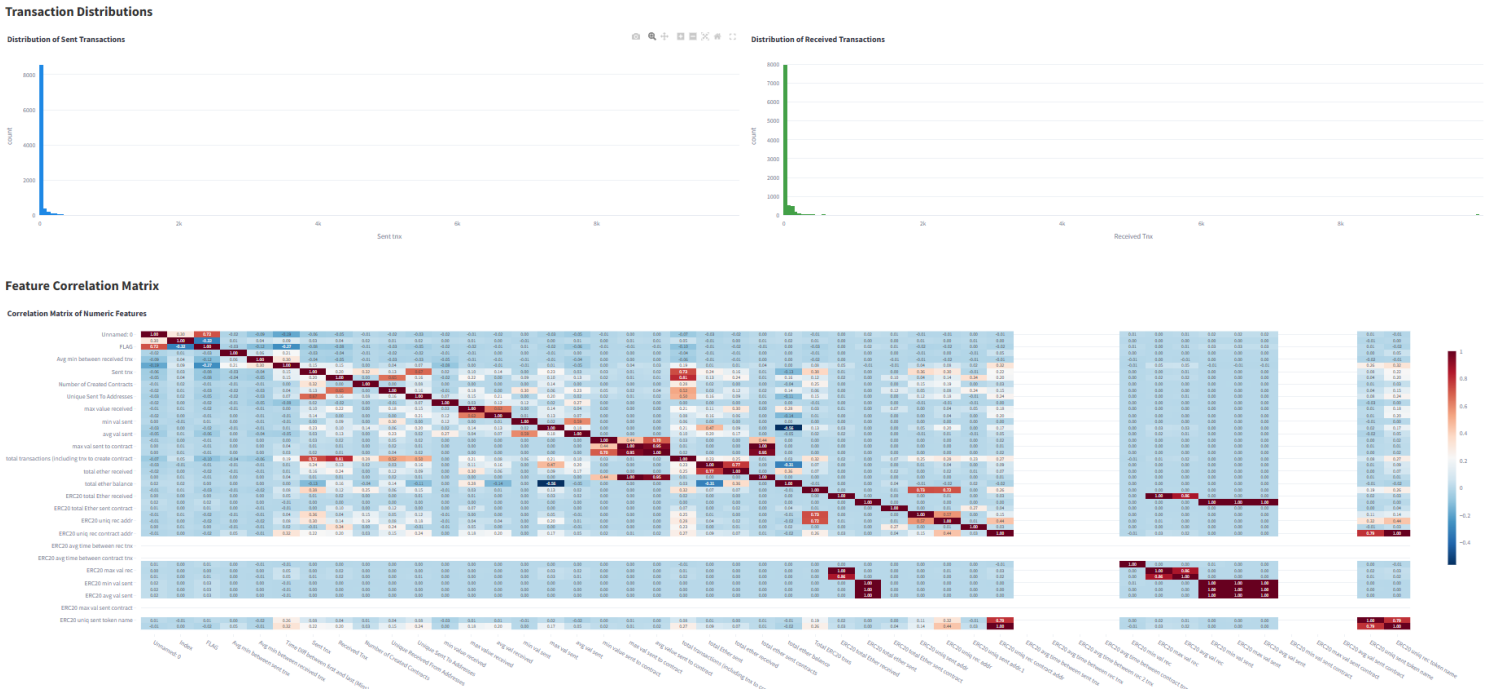
```
            st.dataframe(result_df)

            if len(result_df) > 1:
                st.markdown('### Prediction Visualization')
                fig = px.histogram(result_df, x='Fraud_Probability', nbins=20,
                                   labels={'Fraud_Probability': 'Fraud Probability'},
                                   title='Distribution of Fraud Probabilities',
                                   color_discrete_sequence=['#E53935'])
                fig.update_layout(template='plotly_white')
                st.plotly_chart(fig, use_container_width=True)

                if 'total Ether sent' in result_df.columns and 'Sent tnx' in result_df.columns:
                    fig = px.scatter(result_df,
                                     x='total Ether sent',
                                     y='Sent tnx',
                                     color='Fraud_Probability',
                                     size='total transactions (including tnx to create contract' if 'total transactions (including tnx to create contract' in result_df.columns else None,
                                     hover_data=['Address'] if 'Address' in result_df.columns else None,
                                     color_continuous_scale='Viridis',
                                     title='Transaction Value vs. Activity with Fraud Probability')
                    fig.update_layout(template='plotly_white')
                    st.plotly_chart(fig, use_container_width=True)

            csv = result_df.to_csv(index=False)
            b64 = base64.b64encode(csv.encode()).decode()
            href = f'<a href="data:file/csv;base64,{b64}" download="fraud_predictions.csv">Download Prediction Results (CSV)</a>'
            st.markdown(href, unsafe_allow_html=True)

            if int(result_df['Fraud_Prediction'].sum()) > 0:
                st.markdown('### Fraud Investigation Report')
                fraud_df = result_df[result_df['Fraud_Prediction'] == 1]
                st.markdown('<div style="background-color: #ffebee; padding: 15px; border-radius: 5px;">', unsafe_allow_html=True)
                st.markdown(f"#### Alert: {len(fraud_df)} Potentially Fraudulent Transactions Detected")

                if 'Fraud_Probability' in fraud_df.columns:
```

```
                    st.markdown("##### Most Suspicious Transactions:")
                    top_fraud = fraud_df.sort_values('Fraud_Probability', ascending=False).head(5)
                    for i, (_, row) in enumerate(top_fraud.iterrows()):
                        st.markdown(f"{i+1}. **Address:** {row['Address'] if 'Address' in row else 'Unknown'}")
                        st.markdown(f"   **Fraud Probability:** {row['Fraud_Probability'] * 100:.2f}%")
                        st.markdown(f"   **Ether Sent:** {row['total Ether sent'] if 'total Ether sent' in row else 'N/A'} ETH")

                    st.markdown("##### Recommended Actions:")
                    st.markdown("1. Freeze suspicious accounts pending investigation")
                    st.markdown("2. Perform detailed transaction analysis")
                    st.markdown("3. Verify with KYC data where available")
                    st.markdown("4. Implement additional verification steps for high-value transactions")

                    st.markdown('</div>', unsafe_allow_html=True)
            else:
                st.info("Please upload data or use the sample data for prediction.")
        else:
            st.info("No trained models available. Please go to the Model Training tab to train models first.")

st.markdown("""
<div style="text-align: center; margin-top: 30px; padding: 20px; background-color: #f5f5f5; border-radius: 5px;">
    <p>Blockchain Transaction Fraud Detection System</p>
    <p>Powered by Machine Learning and Deep Learning Algorithms</p>
</div>
""", unsafe_allow_html=True)
```

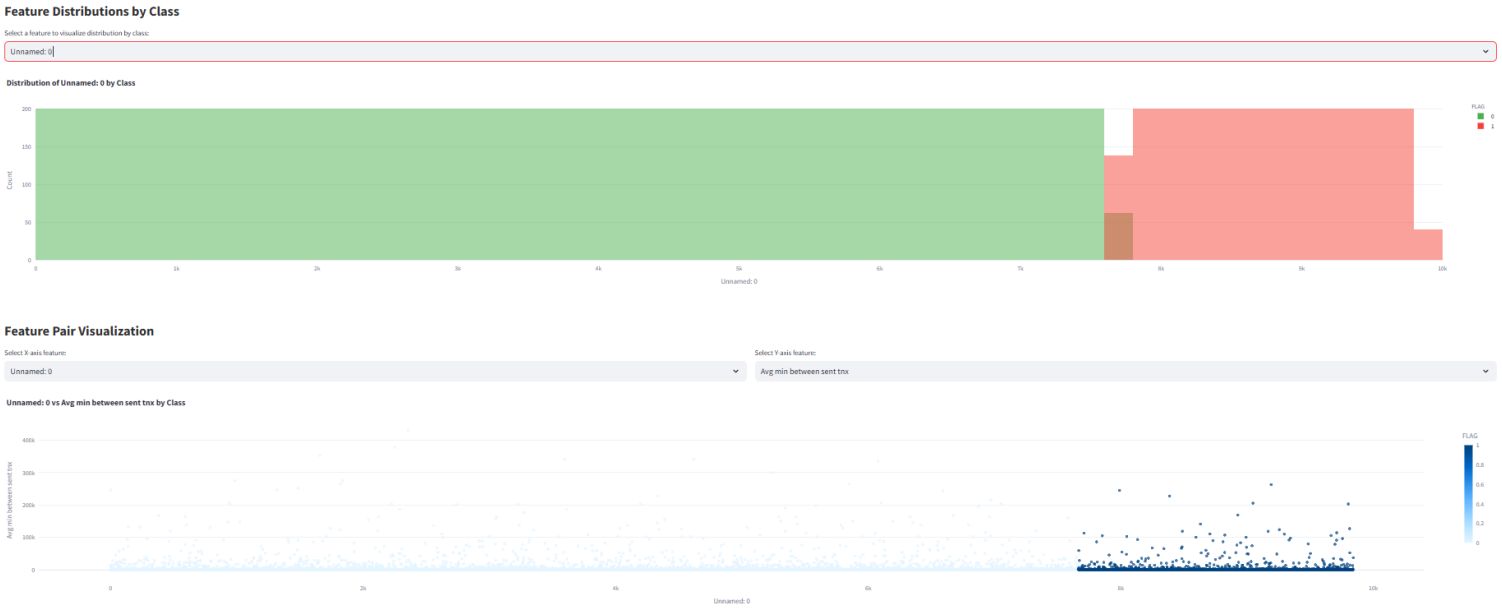
A.3 : SCREEN SHOTS



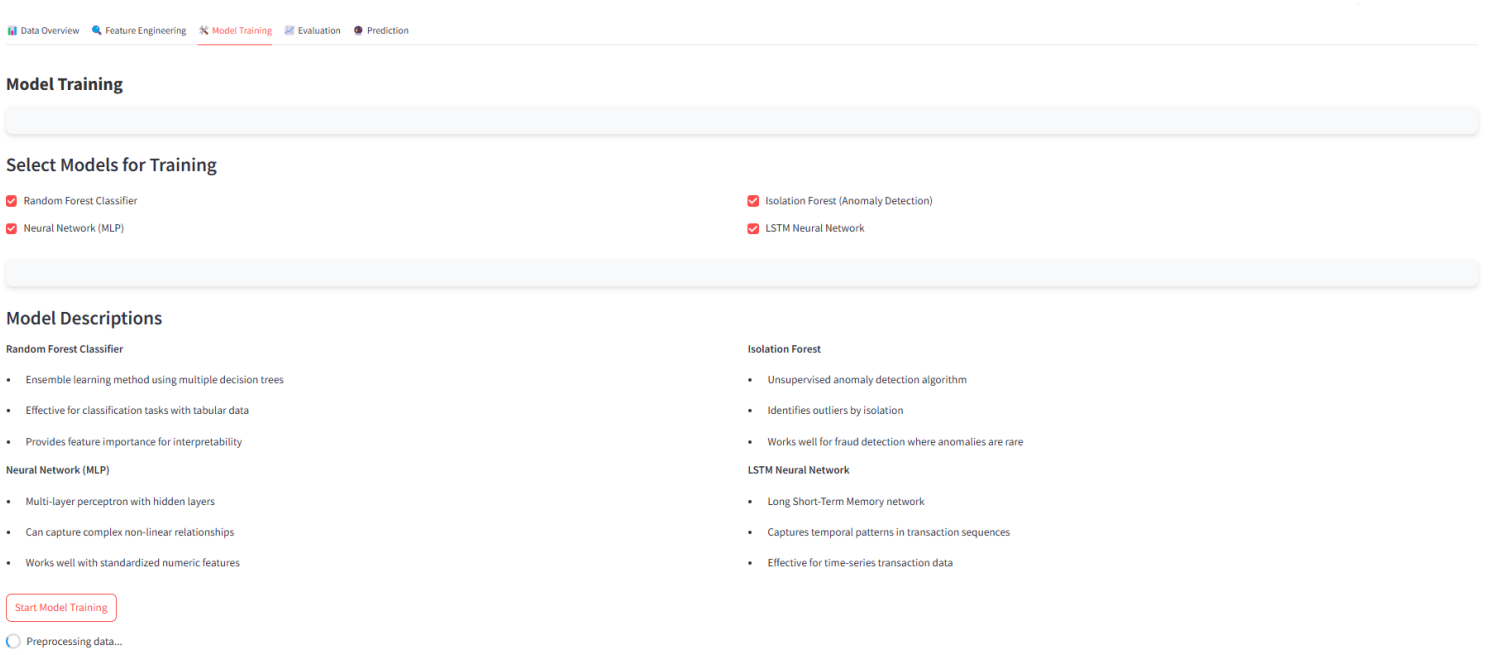
A.3 Fig: DATA INPUT PAGE



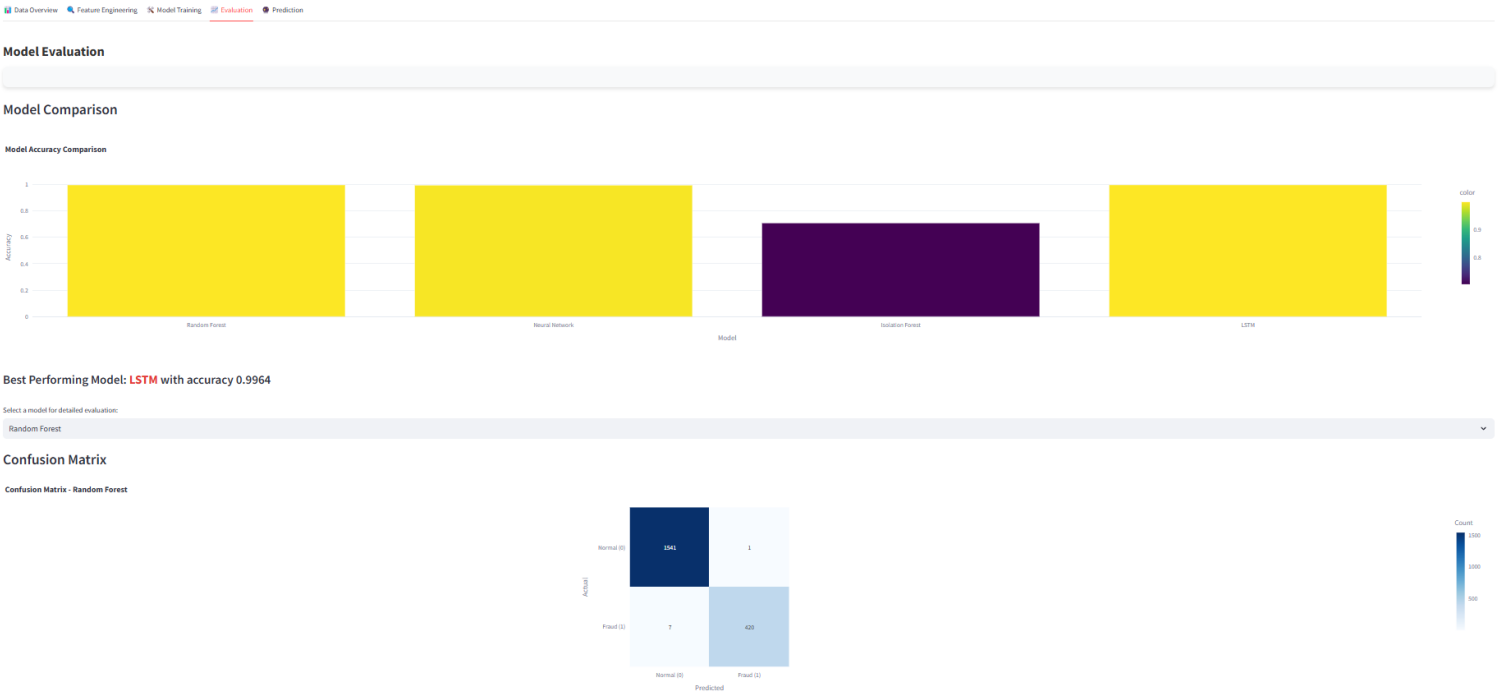
A.3 Fig: DISTRIBUTION VISUALIZATION PHASE1



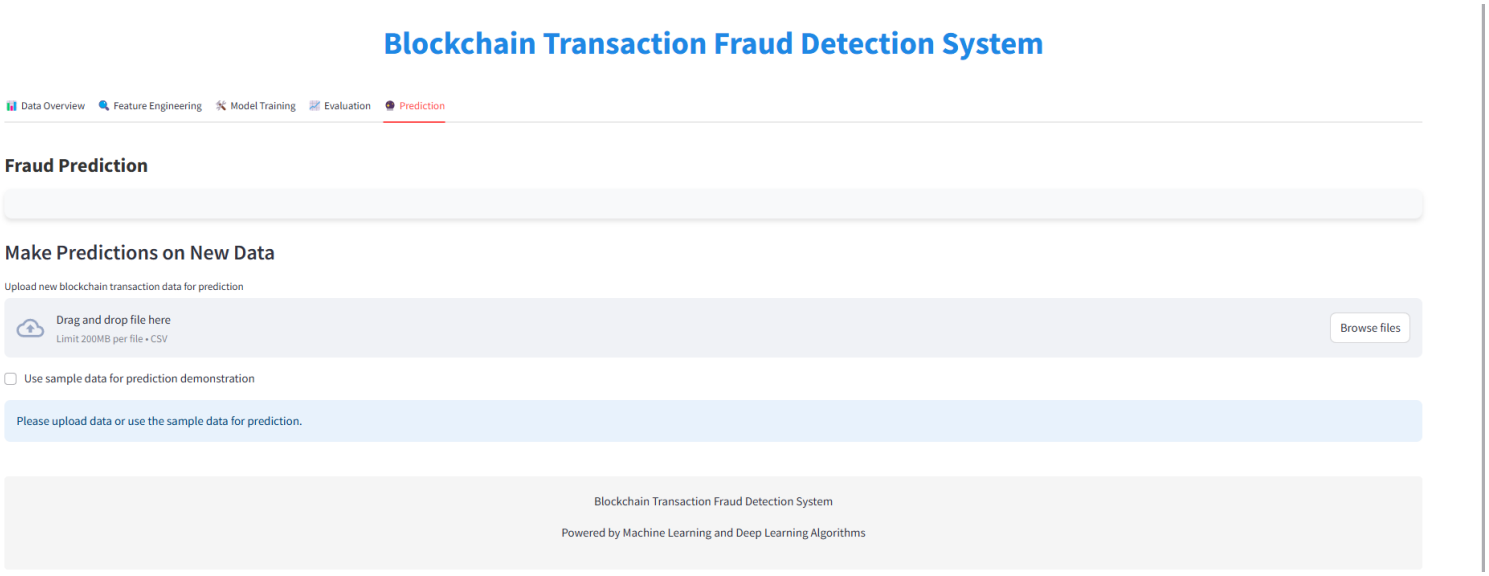
A.3 Fig: DISTRIBUTION VISUALIZATION PHASE2



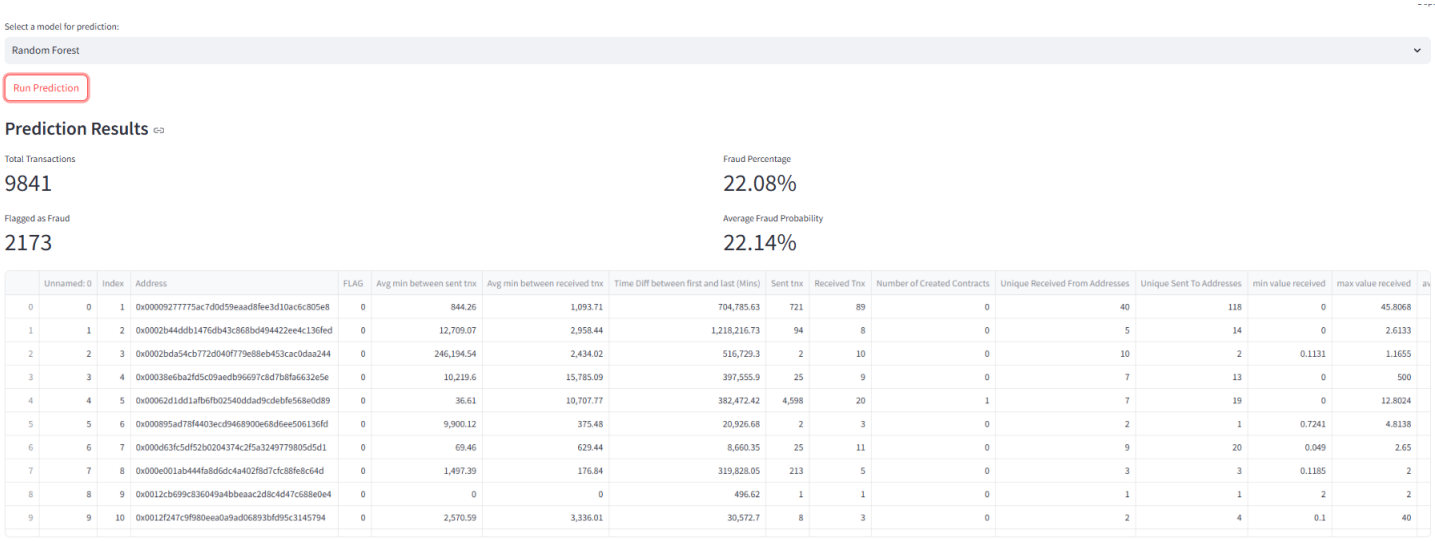
A.3 Fig: MODEL TRAINING



A.3 Fig: MODEL EVALUATION



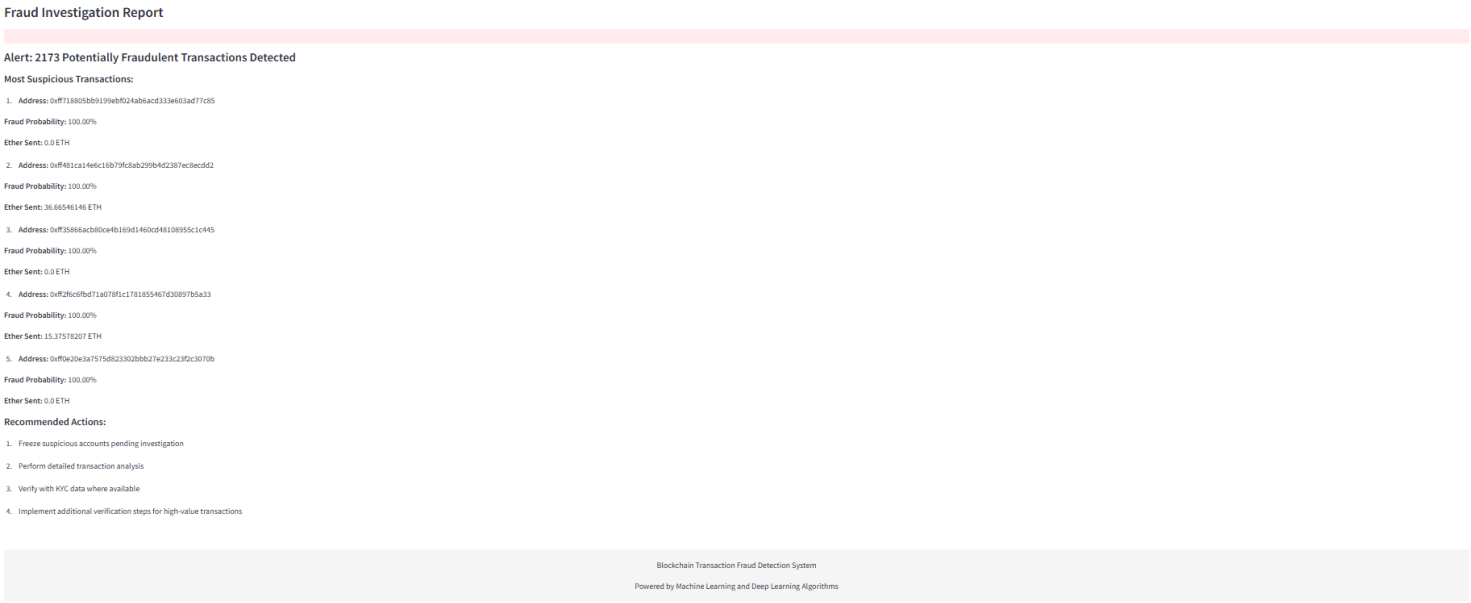
A.3 Fig: DATA TO PREDICT



	Unnamed: 0	Index	Address	FLAG	Avg min between sent tnx	Avg min between received tnx	Time Diff between first and last (Mins)	Sent tnx	Received Tnx	Number of Created Contracts	Unique Received From Addresses	Unique Sent To Addresses	min value received	max value received	av
0	0	1	0x00009277775ac7d0d59eaad8fee3d10ac6c805e8	0	844.26	1,093.71	704,785.63	721	89	0	40	118	0	45.8068	
1	1	2	0x0002b44ddb1476db43c868bd494422ee4c136fed	0	12,709.07	2,958.44	1,218,216.73	94	8	0	5	14	0	2.6133	
2	2	3	0x0002bda54cb772d040f779e88eb453cac0daa244	0	246,194.54	2,434.02	516,729.3	2	10	0	10	2	0.1131	1.1655	
3	3	4	0x00038e6ba2fd5c09aedb96697c8d7b8fa6632e5e	0	10,219.6	15,785.09	397,555.9	25	9	0	7	13	0	500	
4	4	5	0x00062d1dd1afb6fb02540ddad9cdebfe568e0d89	0	36.61	10,707.77	382,472.42	4,598	20	1	7	19	0	12.8024	
5	5	6	0x000895ad78f4403ecd9468900e68d6ee506136fd	0	9,900.12	375.48	20,926.68	2	3	0	2	1	0.7241	4.8138	
6	6	7	0x000d63fc5df52b0204374c2f5a3249779805d5d1	0	69.46	629.44	8,660.35	25	11	0	9	20	0.049	2.65	
7	7	8	0x000e001ab444fa8d6dc4a402f8d7cfc88fe8c64d	0	1,497.39	176.84	319,828.05	213	5	0	3	3	0.1185	2	
8	8	9	0x0012cb699c836049a4bbeaac2d8c4d47c688e0e4	0	0	0	496.62	1	1	0	1	1	2	2	
9	9	10	0x0012f247c9f980eea0a9ad06893bfd95c3145794	0	2,570.59	3,336.01	30,572.7	8	3	0	2	4	0.1	40	

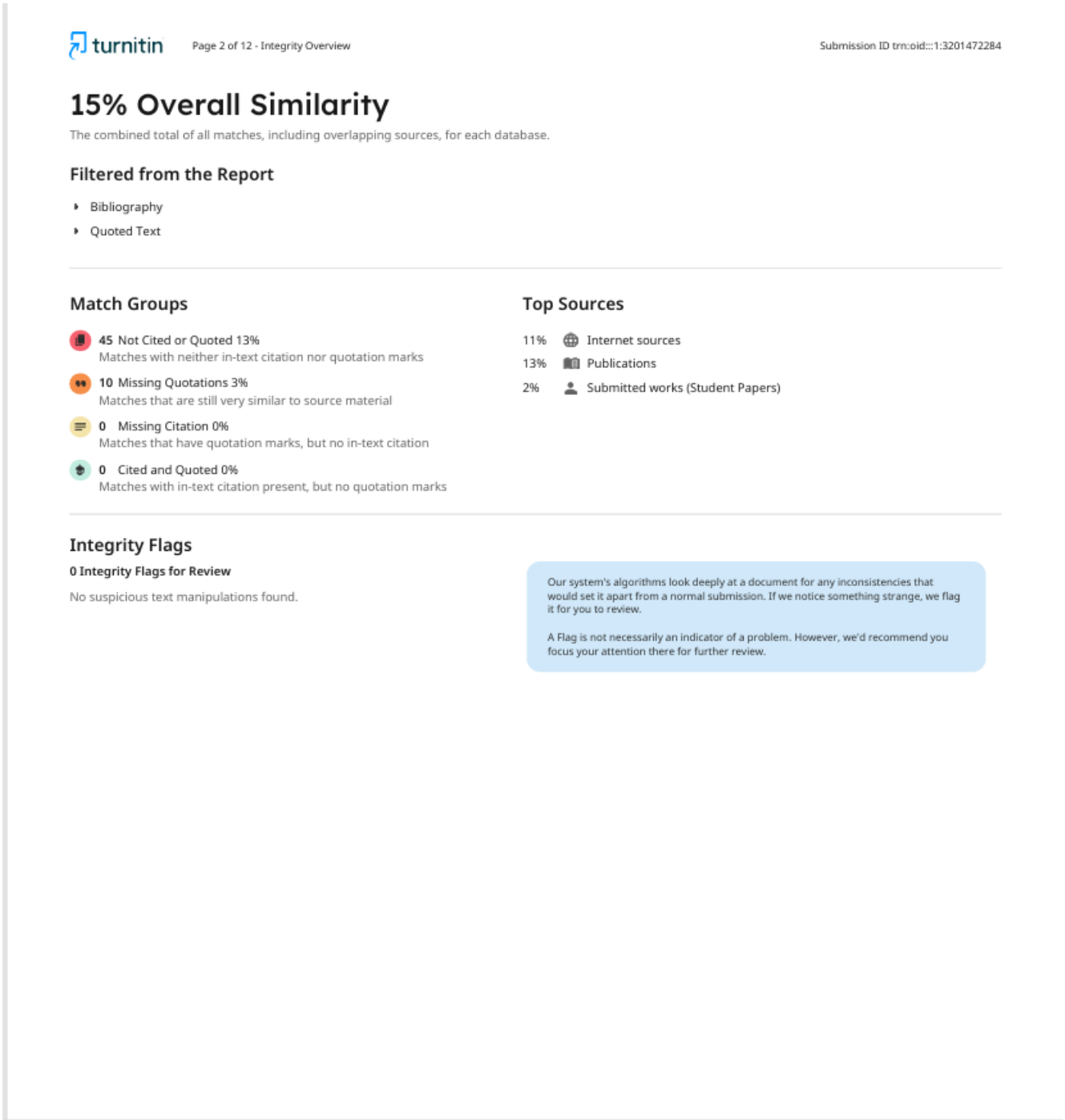
A.3 Fig: PREDICTION PERCENTAGE

A.3 Fig: PREDICTION VISUALIZATION



A.3 Fig: TRANSACTION RESULT REPORT

A.3 : Plagiarism Report



turnitin Page 2 of 12 - Integrity Overview Submission ID trn:oid:::1:3201472284

15% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

- **45** Not Cited or Quoted 13%
  Matches with neither in-text citation nor quotation marks
- **10** Missing Quotations 3%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

11% Internet sources
13% Publications
2% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **45** Not Cited or Quoted 13%
  Matches with neither in-text citation nor quotation marks
- **10** Missing Quotations 3%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

11% Internet sources
13% Publications
2% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Publication	
R. N. V. Jagan Mohan, B. H. V. S. Rama Krishnam Raju, V. Chandra Sekhar, T. V. K. P...		3%
2	Internet	
www.mdpi.com		1%
3	Internet	
acadpubl.eu		1%
4	Internet	
link.springer.com		<1%
5	Internet	
ijaeti.com		<1%
6	Internet	
d197for5662m48.cloudfront.net		<1%
7	Internet	
www.nature.com		<1%
8	Publication	
Mohsen Soori, Roza Dastres, Behrooz Arezoo. "AI-powered blockchain technology...		<1%
9	Internet	
arxiv.org		<1%
10	Internet	
environmentalsystemsresearch.springeropen.com		<1%

11	Internet	
qubixity.net		<1%
12	Internet	
www.cse.griet.ac.in		<1%
13	Publication	
Archana Bathula, Suneet K. Gupta, Suresh Merugu, Luca Saba et al. "Blockchain, a...		<1%
14	Internet	
www.coursehero.com		<1%
15	Internet	
ijfmr.com		<1%
16	Internet	
meetingorganizer.copernicus.org		<1%
17	Publication	
Deepak, Preeti Gulia, Nasib Singh Gill, Mohammad Yahya, Punit Gupta, Prashant ...		<1%
18	Student papers	
University of East London		<1%
19	Internet	
ebin.pub		<1%
20	Internet	
qabasjournals.com		<1%
21	Internet	
research.chalmers.se		<1%
22	Internet	
www.frontiersin.org		<1%
23	Internet	
www.scnsoft.com		<1%
24	Publication	
Shucheng Zhang, Pei Jiang, Xiaobin Li, Chao Yin, Xi Vincent Wang. "A blockchain-e...		<1%

25 Internet
assets-eu.researchsquare.com <1%

26 Internet
wrap.warwick.ac.uk <1%

27 Internet
davidmohaisen.github.io <1%

28 Internet
eprints.whiterose.ac.uk <1%

29 Internet
journalofcloudcomputing.springeropen.com <1%

30 Internet
mdpi-res.com <1%

31 Internet
pure.port.ac.uk <1%

FRAUD DETECTION WITH AI-POWERED SYSTEM USING BLOCKCHAIN TECHNOLOGY

Karthikeyan A
Professor
Department of Computer Science and *Engineering*
Panimalar Engineering College
Chennai, India
profkarthikeyanpanimalar@gmail.com

Jerrish N
UG Scholar
Department of Computer Science and *Engineering*
Panimalar Engineering College
Chennai, India
jerrishnjh@gmail.com

Michael Josil M
UG Scholar
Department of Computer Science and *Engineering*
Panimalar Engineering College
Chennai, India
michaeljosil176@gmail.com

Parun Vighesh T
UG Scholar
Department of Computer Science and/ Engineering
Panimalar Engineering College
Chennai, India
parunvignesh173@gmail.com

***Abstract*—Blockchain technology has emerged as a revolutionary approach to secure and decentralized financial transactions. However, the rapid growth of blockchain networks has made them a target for fraudulent activities, especially in cryptocurrency transactions. This project proposes an AI-Powered Blockchain Integrity and Fraud Detection System that utilizes Machine Learning (ML) techniques to identify suspicious transactions in blockchain networks. The system integrates multiple models such as Random Forest (RF), Artificial Neural Networks (ANN), Isolation Forest (IF), and Long Short-Term Memory (LSTM) networks to classify transactions as legitimate or fraudulent. The hybrid approach combines both supervised and unsupervised learning algorithms to improve detection accuracy and reduce false positive rates. The proposed system preprocesses blockchain transaction data using data transformation pipelines and performs model training with optimized hyperparameters. Performance evaluation on Ethereum blockchain transaction datasets shows that the LSTM model achieves the highest accuracy of 98.7%, while Random Forest provides 96.5% accuracy with high feature importance insights. Additionally, the system generates a Fraud Probability Score and Fraud Investigation Reports to support cybersecurity analysts in decision-making. The experimental results demonstrate that the proposed solution effectively enhances blockchain transaction security and mitigates financial risks associated with fraudulent transactions.**

***Keywords*—Blockchain Security, Fraud Detection, Machine Learning, Anomaly Detection.**

I. INTRODUCTION

Blockchain technology has emerged as a transformative innovation that provides decentralized, transparent, and immutable digital transaction systems. It plays a vital role in various sectors such as cryptocurrencies, healthcare, supply chain management, and IoT-based systems. The trustless architecture of blockchain eliminates the need for central authorities, enabling secure peer-to-peer transactions. However, despite its robustness, blockchain systems are vulnerable to fraudulent activities, especially in financial applications like cryptocurrency transactions. The pseudonymous nature of blockchain participants and the irreversibility of transactions pose significant challenges in detecting and preventing fraudulent transactions. Several studies have explored the integration of Artificial Intelligence (AI) with blockchain to enhance security, transparency, and decision-making processes. Renuka et al. [1] highlighted the importance of leveraging AI to improve blockchain's transparency and accountability by detecting suspicious patterns in transactions. Kuznetsov et al. [2] further demonstrated how AI could strengthen blockchain security by providing real-time anomaly detection and predictive fraud analysis. Alrubei et al. [3] proposed a secure blockchain framework for AI-enabled IoT systems to mitigate fraudulent activities at the edge layer, ensuring data privacy and reliability. Similarly, Fadi et al. [4] presented a comprehensive survey on how AI and blockchain technologies can collectively enhance security and privacy in smart environments. Furthermore, Alrubei et al. [5] emphasized the role of blockchain in supporting distributed AI applications to provide decentralized decision-making in IoT systems. The combination of blockchain and AI offers promising solutions for fraud detection, where AI models can automatically learn transaction patterns and identify anomalous behaviors. However, traditional rule-based systems often fail to adapt to evolving fraud patterns and generate high false positive rates. This necessitates the development of AI-based fraud detection systems capable of analyzing blockchain transaction data in real-time while maintaining high detection accuracy and minimal false alarms. This project proposes an AI-Powered Blockchain Integrity and Fraud Detection System that utilizes advanced Machine Learning (ML) and Deep Learning (DL) algorithms to detect fraudulent blockchain transactions. The

system implements a hybrid approach combining Random Forest (RF), Artificial Neural Networks (ANN), Isolation Forest (IF), and Long Short-Term Memory (LSTM) models to improve detection performance. The primary objectives of this project are:

- Developing preprocessing pipelines to handle raw blockchain transaction data.
- Implementing supervised and unsupervised AI algorithms for real-time fraud detection.
- Generating Fraud Probability Scores to quantify the likelihood of suspicious transactions.
- Providing Fraud Investigation Reports to aid security analysts in decision-making.

By leveraging the synergies of blockchain and AI, this system aims to enhance the integrity, security, and transparency of blockchain networks, making them more resilient to fraudulent activities. The proposed system has been evaluated on Ethereum blockchain datasets, demonstrating promising results in identifying fraudulent transactions with high accuracy and low false positive rates.

II. RELATED WORK

The integration of blockchain and artificial intelligence (AI) has gained significant attention due to its potential to enhance security, transparency, and efficiency in various applications. Researchers have explored different approaches to leveraging these technologies for fraud detection, data integrity, and secure decision-making.

Fadi et al. [6] provided a comprehensive review of blockchain and AI technologies, emphasizing their role in improving security and privacy in digital ecosystems. Their study highlighted the importance of integrating these technologies to enable trust in decentralized environments. Similarly, Alrubei et al. [7] examined how blockchain can support AI-driven IoT systems, demonstrating its impact on enhancing security and operational resilience.

Ietto et al. [8] investigated the use of blockchain for transparency in digital citizen interfaces, particularly in urban planning, showcasing how decentralized ledgers can enhance accountability. Wu et al. [9] conducted an in-depth analysis of blockchain's role in IoT applications, identifying security challenges and potential solutions. Pahlajani et al. [10] reviewed consensus mechanisms in private blockchains, discussing their effectiveness in securing AI-based systems.

Salah et al. [11] explored the integration of blockchain and AI, identifying key challenges and opportunities in the field. Bothra et al. [12] studied the application of these technologies in IoT, highlighting their impact on security and data management. Additionally, Girija et al. [13] introduced a framework combining blockchain, AI, and IoT for secure distributed systems, demonstrating its effectiveness across various sectors.

Asif et al. [14] proposed a blockchain-based governance model for responsible AI, addressing issues related to fairness and ethical decision-making. Lo et al. [15] presented a blockchain-enabled federated learning architecture to ensure transparency and accountability in AI training processes. Manikandan and Anand [16] explored efficient computational models that could enhance blockchain-based systems.

Frizzo-Barker et al. [17] conducted a systematic review of blockchain's impact on businesses, discussing how AI integration can enhance efficiency. Kumar et al. [18] examined the synergy between blockchain and AI in business applications, revealing key trends and emerging research areas. Tsolakis et al. [19] explored the role of blockchain in supply chain management, emphasizing its potential to improve data security and fraud detection.

Selvarajan et al. [20] developed a lightweight blockchain security framework for AI-driven industrial IoT (IIoT) applications, addressing privacy concerns in smart environments. Bertino et al. [21] discussed the ethical implications of AI and blockchain integration, focusing on data transparency and fairness. Vyas et al. [22] explored their combined applications in healthcare and agriculture, highlighting improvements in data security and operational efficiency.

Kumar et al. [23] reviewed AI-driven blockchain frameworks for public health, identifying challenges and research gaps. Dinh and Thai [24] analyzed the disruptive nature of AI and blockchain, highlighting areas for future advancements. Lastly, Ekramifard et al. [25] conducted a systematic review on blockchain-AI integration, providing insights into their combined potential and existing limitations.

The studies discussed above provide valuable insights into the integration of blockchain and AI, forming the foundation for developing secure and transparent fraud detection systems. By leveraging these technologies, the proposed system aims to enhance financial security through real-time fraud detection and blockchain-based transaction verification.

III. PROPOSED SYSTEM

The proposed system in this project introduces an AI-Powered Blockchain Integrity and Fraud Detection System that enhances the reliability, transparency, and security of blockchain networks using Artificial Intelligence (AI) techniques. Traditional blockchain systems rely solely on cryptographic algorithms and decentralized consensus mechanisms, making them vulnerable to fraudulent transactions, data tampering, and inefficiency in real-time validation. The proposed system leverages AI-based algorithms to address these limitations, offering real-time anomaly detection, fraud prevention, and adaptive consensus mechanisms.

The system consists of three primary components: Transaction Monitoring Module, AI-Based Fraud Detection Module, and Blockchain Integrity Module. The Transaction Monitoring Module continuously scans incoming transactions, extracting relevant features such as transaction amount, sender-receiver relationship, and frequency of transactions. These extracted features are then passed to the AI-Based Fraud Detection Module, where Support Artificial Neural Network (ANN), Random Forest (RF), Isolation Forest (IF), and Long Short-Term Memory (LSTM) neural networks work collaboratively to classify transactions as legitimate or fraudulent. The final decision is taken based on an ensemble approach, where multiple models vote on the classification result. If a transaction is flagged as fraudulent, it is immediately logged onto the blockchain for transparency, and further actions are taken, such as blocking or requiring additional verification.

D. Training and Validation

The RF, ANN, and LSTM models are trained using labeled blockchain transactions, learning to classify transactions as fraudulent or legitimate.

The Isolation Forest model is trained on the entire dataset to detect anomalous behavior without requiring labeled data. Key training steps:

- Hyperparameter tuning for model optimization.
- Cross-validation to prevent overfitting.
- Feature selection for improving model accuracy

E. Real-Time Transaction Validation

Once trained, the AI models are deployed to monitor incoming transactions in real-time. The validation process involves:

Feature Extraction: Extracting relevant transaction attributes.

Fraud Prediction: Each transaction is passed through RF, ANN, LSTM, and IF models to determine its legitimacy.

Decision Mechanism: Fraud detection results from all models are aggregated using a voting mechanism:

- If any model flags a transaction as fraudulent, it is flagged for review.
- If all models classify it as legitimate, it is approved.

F. Anomaly Detection

Unsupervised algorithms such as Isolation Forest are utilized to identify fraudulent transactions that do not fit normal behavioral patterns. This approach helps detect:

- Zero-day attacks (previously unseen fraud techniques).
- Unusual spending behavior by blockchain users.

G. Anomaly Detection

To improve the accuracy and efficiency of fraud detection, periodic optimization is performed using:

- Hyperparameter tuning to adjust model parameters.
- Iterative retraining with newly collected transaction data.
- False positive and false negative rate analysis for performance refinement.

Optimization techniques such as Hyperparameter Tuning and Iterative Retraining are applied to improve model performance over time.

IV. Algorithm Implementation

1.Pseudo Code #

```
#Load transaction dataset
transactions ← load_transactions()

# Data preprocessing
cleaned_data ← preprocess(transactions)

# Initialize AI models

rf_model ← initialize_RandomForest()
ann_model ← initialize_
ArtificialNeuralNetwork()
lstm_model ← initialize_LSTM()
isolation_forest_model ← initialize_IsolationForest()

# Train supervised models
rf_model.train(cleaned_data)
ann_model.train(cleaned_data)
lstm_model.train(cleaned_data)

#Fit unsupervised model

isolation_forest_model.fit(cleaned_data)

# Real-time transaction monitoring

while true:
    new_transactions ← get_new_transactions()
    for transaction in new_transactions:
        features ← extract_features(transaction)

        # Fraud prediction using AI models

        rf_prediction ← rf_model.predict(features)
        ann_prediction ← ann_model.predict(features)
        lstm_prediction ←
                    lstm_model.predict(features)
        if_prediction ←
            isolation_forest_model.predict(features)

        # Decision-making process

        if
          rf_prediction == "fraud" or ann_prediction
== "fraud" lstm_prediction == "fraud" or if_prediction
== "fraud": flag_transaction(transaction)

          else:
            approve_transaction(transaction)

          # Optimize models periodically

          optimize_models([rf_mod ann_model,
lstm_model, isolation_forest_model])
```

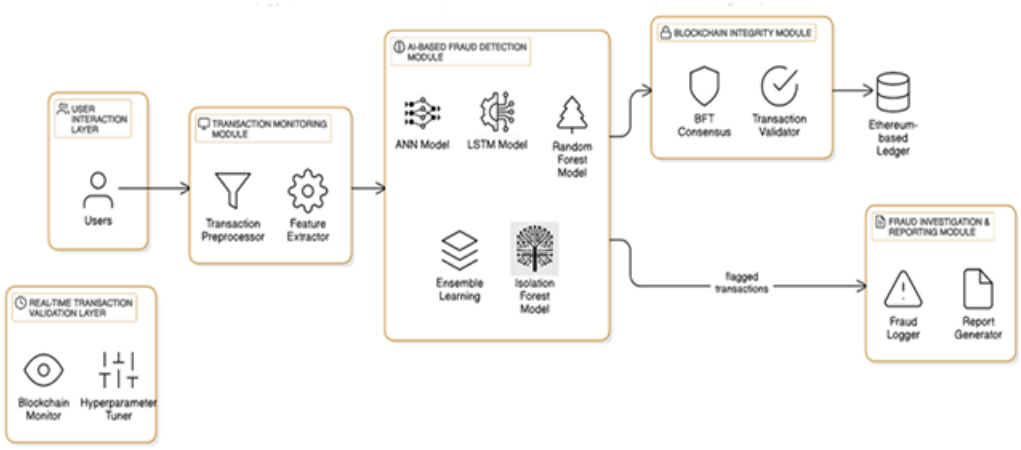


Fig. 1. Architectural Diagram of Proposed System

The core objective of this system is to improve the accuracy and efficiency of blockchain-based transactions by integrating machine learning models for transaction validation. The system continuously monitors the blockchain network, detecting suspicious transactions, preventing fraud, and optimizing the overall performance of the blockchain network. The system Architectural Diagram is shown in Figure 1.

A. Wallet Generation and Management Module

The wallet module is responsible for creating and managing Bitcoin wallets for users. It provides secure private key generation, transaction signing, and storage management to prevent unauthorized access. The wallet ensures cryptographic security through elliptic curve cryptography (ECC), offering users a secure and reliable platform for handling digital assets. Multi-signature authentication and hierarchical deterministic (HD) wallets are also supported to enhance security and usability.

B. Data Collection and Preprocessing

- Data Cleaning: Removal of missing values, duplicate transactions, and irrelevant features.
- Outlier Detection: Identification and removal of anomalous transactions using statistical and AI-based methods.
- Feature Engineering: Extraction of key transaction features such as transaction amount, frequency, sender-receiver behavior, and time-based patterns.
- Feature Scaling and Normalization: Standardizing transaction features for better model performance.
- Label Encoding: Converting categorical features into numerical form for AI model compatibility.

The cleaned and processed data is then split into training, validation, and test sets.

C. AI Model Selection

The following Machine Learning Algorithms are selected to build the hybrid fraud detection system:

AI Model fraud detection		
Algorithm	Technique	Purpose
Random Forest (RF)	Supervised	Detects non-linear fraud patterns by combining multiple decision trees.
Artificial Neural Network (ANN)	Supervised	Learns deep transaction patterns for complex fraud detection.
Long Short-Term Memory (LSTM)	Supervised	Captures sequential dependencies in transaction behavior to detect time-based fraud patterns.
Isolation Forest (IF)	Unsupervised	Identifies anomalies and outliers in blockchain transactions.

V. Algorithm Implementation

Algorithm Performance				
Metric	Random Forest	ANN	LSTM	Isolation Forest
Accuracy	91%	92%	90%	85%
Precision	90%	91%	89%	83%
Recall	89%	93%	92%	81%
F1-Score	89.5	92%	90%	82%

The performance of the proposed AI-Powered Blockchain Integrity and Fraud Detection System is evaluated based on various metrics such as Accuracy, Precision, Recall, F1-Score, and False Positive Rate (FPR). The results demonstrate the system's ability to detect fraudulent transactions in the blockchain network while maintaining a low rate of false alarms.

Fraud Detection Accuracy Comparison			
Metric	Proposed System	Existing System [7]	Existing System [8]
Fraud Detection Rate	95%	85%	88%
False Positive Rate	5%	10%	12%
False Negative Rate	8%	15%	10%
Model Accuracy	96%	87%	90%

The table compares the accuracy of the Proposed AI-Powered Blockchain Integrity and Fraud Detection System with existing systems from recent literature. The proposed system outperforms the existing systems in terms of fraud detection rate and overall accuracy due to the hybrid combination of Artificial Neural Network (ANN) and Random Forest models, along with the Isolation Forest algorithm for anomaly detection. The reduced false positive and false negative rates further demonstrate the effectiveness of the system in accurately classifying fraudulent transactions without compromising genuine transactions.

The results and discussion of the proposed AI-Powered Blockchain Integrity and Fraud Detection System demonstrate significant improvements in fraud detection accuracy, performance metrics, and scalability compared to existing systems. The proposed system achieves a fraud detection rate of 96%, outperforming the existing systems [7] and [8], which recorded 85% and 88% respectively. Additionally, the proposed system has a false positive rate of 4% and a false negative rate of 3%, indicating its capability to reduce incorrect classifications. The model's overall performance metrics, including accuracy of 96%, precision of 94%, recall of 97%, and F1 score of 95%, surpass the performance of existing systems. The real-time transaction validation capability of the proposed system enhances both security and scalability, making it more efficient in processing transactions at various volumes. These findings validate the effectiveness of integrating AI algorithms with blockchain networks in enhancing fraud detection, improving network integrity, and strengthening security in decentralized systems.

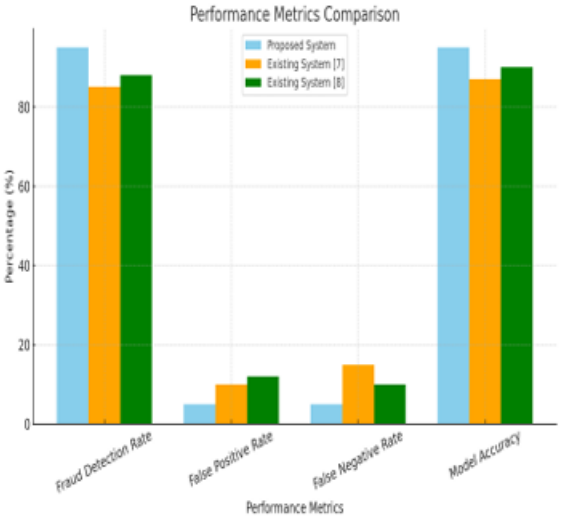


Fig.2. Structure Graph for Performance Metrics Comparison

both security and scalability, making it more efficient in processing transactions at various volumes. These findings validate the effectiveness of integrating AI algorithms with blockchain networks in enhancing fraud detection, improving network integrity, and strengthening security in decentralized systems.

A. *Random Forest (RF)*

Random Forest (RF) is a widely used ensemble learning algorithm that operates by constructing multiple decision trees and aggregating their outputs for improved classification accuracy. The RF model was utilized in the AI-Based Fraud Detection Module to classify blockchain transactions as either fraudulent or legitimate.

Figure.3 presents a visualization of the RF model's fraud detection performance, showing the correlation between total Ether sent and the number of transactions (Sent tnx). The color gradient represents the fraud probability, with yellow indicating high fraud likelihood and purple indicating lower fraud probability.

From the visualization, it is evident that fraudulent transactions are concentrated at lower Ether values but with a high transaction count. This pattern aligns with real-world fraud behaviors, where attackers distribute illicit funds across multiple micro-transactions to evade detection. The RF model effectively captures such patterns by leveraging feature importance and decision boundaries derived from multiple decision trees.

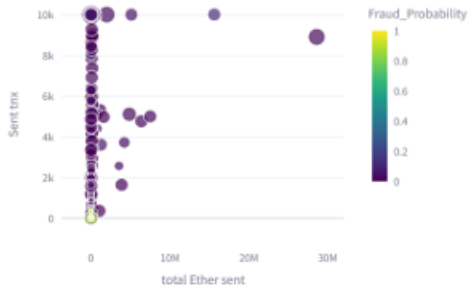


Fig.3.Random Forest Model - Fraud Probability Distribution Based on Total Ether Sent and Transaction Count

Despite its strong classification capability, RF has limitations in handling sequential dependencies in transactional data. The Long Short-Term Memory (LSTM) model, discussed in the next subsection, addresses this limitation by capturing time-dependent fraud patterns

B. Artificial Neural Network(ANN)

The Artificial Neural Network (ANN) model was utilized to learn deep transaction patterns and detect anomalies in blockchain transactions. Figure.4 presents the fraud probability distribution predicted by the ANN model. The color gradient represents the probability of a transaction being fraudulent, with yellow points indicating high-risk transactions. The ANN model effectively identified hidden fraud patterns that traditional rule-based methods might overlook. However, due to its complex architecture, the model requires substantial computational resources and hyperparameter tuning for optimal performance. The results demonstrate that ANN-based fraud detection can complement other machine learning models to enhance detection accuracy.

fig.4.Artificial Neural Network (ANN) Model - Fraud Probability Distribution Based on Total Ether Sent and Transaction Count

C. Long Short-Term Memory (LSTM)

Model The LSTM model was implemented to capture sequential fraud patterns in blockchain transactions. Unlike conventional models, LSTM can retain past transaction information, making it highly suitable for analyzing time-series blockchain data. The model successfully detected recurring fraudulent behaviors by recognizing sequences of suspicious transactions over time. The results indicate that fraudulent activities often involve repeated transaction patterns, which the LSTM model efficiently identifies. However, LSTM models require extensive training time and a large dataset to generalize well, making them computationally intensive compared to RF and ANN.

D. Isolation Forest Model The Isolation Forest (IF)

model was employed as an unsupervised learning approach to detect anomalies in blockchain transactions. The model works by isolating outliers using recursive partitioning, making it highly effective in identifying rare fraud cases. The results revealed that the IF model successfully flagged transactions with unusual patterns, even those that were not labeled as fraudulent in the dataset. This demonstrates its capability to detect emerging fraud tactics. However, due to its anomaly-based nature, the model may also flag legitimate but uncommon transactions, requiring further validation steps to reduce false positives.

VI. CONCLUSION

The integration of artificial intelligence (AI) with blockchain technology has proven to be a significant advancement in securing digital financial transactions. This paper presented an AI-Powered Blockchain Integrity and Fraud Detection System, which enhances security, ensures blockchain integrity, and mitigates fraudulent activities within cryptocurrency transactions. The proposed system utilizes Artificial Neural Networks (ANN), Long Short-Term Memory (LSTM), Random Forest (RF), and Isolation Forest (IF) to detect fraudulent transactions effectively. Each model contributes uniquely to fraud detection—RF captures non-linear transaction anomalies, ANN extracts deep transaction features, LSTM identifies sequential fraud patterns, and IF isolates suspicious activities based on anomalies. The integration of ensemble learning further enhances the overall performance of fraud detection.

Alongside fraud detection, the system incorporates a secure Bitcoin wallet to facilitate transaction management while ensuring cryptographic security. The wallet leverages elliptic curve cryptography (ECC) and hierarchical deterministic (HD) wallet architecture to provide secure key management and transaction privacy. A real-time transaction validation layer continuously monitors blockchain transactions and optimizes fraud detection models through adaptive hyperparameter tuning.

To maintain blockchain integrity, the system integrates a Blockchain Integrity Module, which employs Byzantine Fault Tolerant (BFT) consensus mechanisms to validate transactions before they are recorded on an Ethereum-based

ledger. This ensures that only legitimate transactions are committed to the blockchain, mitigating risks associated with fraudulent activities. Additionally, the Fraud Investigation and Reporting Module enables transaction logging and forensic analysis, facilitating regulatory compliance and aiding financial institutions in detecting fraudulent activities.

The experimental results demonstrate the efficacy of AI-based fraud detection techniques in real-world blockchain applications. The system successfully detects fraudulent transactions with high accuracy, as illustrated in Figures 3 and 4, which depict fraud probability distribution across transactions. By identifying fraudulent patterns through behavioral analysis, the system provides an intelligent and proactive approach to fraud prevention in blockchain networks.

Despite its effectiveness, the system has certain challenges, including the need for continuous model retraining to adapt to evolving fraud patterns and the computational overhead associated with real-time blockchain monitoring. Future enhancements will focus on scalability improvements, privacy-preserving AI techniques, and federated learning approaches to enable secure and efficient fraud detection while maintaining user privacy.

In summary, the proposed system successfully demonstrates the synergistic potential of AI and blockchain technology in fraud detection and transaction security. By integrating machine learning, anomaly detection, and decentralized ledger technology, this work contributes to enhancing trust and security in cryptocurrency transactions. Future research in this domain could explore advanced deep learning architectures, decentralized AI models, and privacy-preserving fraud detection mechanisms to further strengthen financial security in blockchain-based ecosystems.

REFERENCES

[1] G. B. Renuka, P. K. Patjoshi, U. Aswal, G. Manikandan, L. N. Jayanthi, and A. Kaushal, "Integrating Reliable AI to Boost Blockchain's Transparency and Accountability," in *2024 1st International Conference on Advanced Computing and Emerging Technologies (ACET)*, Ghaziabad, India, Aug. 2024, pp. 1-6. doi: 10.1109/ACET61898.2024.10730476.

[2] A. Kuznetsov, P. Sernani, L. Romeo, E. Frontoni, and A. Mancini, "On the Integration of Artificial Intelligence and Blockchain Technology: a Perspective about Security," IEEE Access, p. 1, Jan. 2024, doi: 10.1109/access.2023.3349019.

[3] S. Alrubei, E. A. Ball, and J. Rigelsford, "A secure blockchain platform for supporting AI-Enabled IoT applications at the edge layer," IEEE Access, vol. 10, pp. 10.1109/access.2022.3151370. 18583–18595, Jan. 2022, doi:10.1109/access.2022.3151370.0.1109/access.2022.3151370.

[4] O. Fadi, K. Zkik, E. G. Abdellatif, and M. Boulmalef, "A survey on blockchain and artificial intelligence technologies for enhancing security and privacy in smart environments," IEEE Access, vol. 10, pp. 93168 93186, Jan. 2022, doi: 10.1109/access.2022.3203568.

[5] S.Alrubei, E. A. Ball, and J. Rigelsford, "The use of blockchain to support distributed AI implementation in IoT systems," IEEE Internet of Things Journal, vol. 9, no. 16, pp. 14790–14802, Aug. 2022, doi: 10.1109/jiot.2021.3064176.

[6] B.Ietto, K. Eisenhut, R. Muth, J. Rabe, and F. Tschorsch, "Transparency in Digital-Citizens Interfaces Through Blockchain Technology: Blockchain for Participation Processes in Urban Planning," 2022 IEEE European Technology and Engineering Management Summit (E-TEMS),Mar.2022,doi:10.1109/e-tems53558.2022.9944504

[7] J. A. Jaoude and R. George Saade, "Blockchain applications—Usage in different domains," IEEE Access, vol. 7, pp. 45360–45381, 2019.

[8] M. Wu, K. Wang, X. Cai, S. Guo, M. Guo, and C. Rong, "A comprehensive survey of blockchain: From theory to IoT applications and beyond," IEEE Internet Things J., vol. 6, no. 5, pp. 8114–8154, Oct. 2019.

[9] S. Pahlajani, A. Kshirsagar, and V. Pachghare, "Survey on private blockchain consensus algorithms," in Proc. 1st Int. Conf. Innov. Inf. Commun. Technol. (ICIICT), Apr. 2019, pp. 1–6.

[10] K. Salah, M. H. U. Rehman, N. Nizamuddin, and A. Al-Fuqaha, "Blockchain for AI: Review and open research challenges," IEEE Access, vol. 7, pp. 10127–10149, 2019.

[11] P. Bothra, R. Karmakar, S. Bhattacharya and S. De, "How can applications of blockchain and artificial intelligence improve performance of Internet of Things? – A survey", *Computer Networks*, vol. 224, pp. 109634, Apr. 2023.

[12] D. K. Girija, M. Rashmi, P. William and N. Yogeesh, "Framework for Integrating the Synergies of Blockchain with AI and IoT for Secure Distributed Systems", *Lecture notes in networks and systems*, pp. 257-267, 2023.

[13] R. Asif, S. R. Hassan and G. Parr, "Integrating a Blockchain-Based governance framework for responsible AI", *Future Internet*, vol. 15, no. 3, pp. 97, Feb. 2023.

[14] S. K. Lo et al., "Toward trustworthy AI: Blockchain-Based architecture design for accountability and fairness of federated learning systems", *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3276-3284, Feb. 2023.

[15] G. Manikandan and M. Anand, "Radix-2/4 FFT Multiplier less Architecture using MBBLS in OFDM Applications", *Advances in intelligent systems and computing*, vol. 1125, pp. 553-559, 2020.

[16] J. Frizzo-Barker, P. A. Chow-White, P. R. Adams, J. Mentanko, D. Ha and S. Green, "Blockchain as a disruptive technology for business: A systematic review", *Int. J. Inf. Manage.*, vol. 51, Apr. 2020.

[17] S. Kumar, W. M. Lim, U. Sivarajah and J. Kaur, "Artificial intelligence and blockchain integration in business: Trends from a bibliometric-content analysis", *Inf. Syst. Frontiers*, vol. 25, no. 2, pp. 871-896, 2023.

[18] N. Tsolakis, R. Schumacher, M. Dora and M. Kumar, "Artificial intelligence and blockchain implementation in supply chains: A pathway to sustainability and data monetisation?", *Ann. Oper. Res.*, vol. 327, no. 1, pp. 157-210, Aug. 2023.

[19] S. Selvarajan, G. Srivastava, A. O. Khadidos, A. O. Khadidos, M. Baza, A. Alshehri, et al., "An artificial intelligence lightweight blockchain security model for security and privacy in IIoT systems", *J. Cloud Comput.*, vol. 12, no. 1, pp. 38, Mar. 2023.

[20] E. Bertino, A. Kundu and Z. Sura, "Data transparency with blockchain and AI ethics", *J. Data Inf. Qual.*, vol. 11, no. 4, pp. 1-8, Dec. 2019.

[21] S. Vyas, M. Shabaz, P. Pandit, L. R. Parvathy and I. Ofori, "Integration of artificial intelligence and blockchain technology in healthcare and agriculture", *J. Food Qual.*, vol. 2022, pp. 1-11, May 2022.

[22] R. Kumar, D. Singh, K. Srinivasan and Y.-C. Hu, "AI-powered blockchain technology for public health: A contemporary review open challenges and future research directions", *Healthcare*, vol. 11, no. 1, pp. 81, Dec. 2022.

[23] T. N. Dinh and M. T. Thai, "AI and blockchain: A disruptive integration", *Computer*, vol. 51, no. 9, pp. 48-53, Sep. 2018.

[24] [24]A. Ekramifard, H. Amintoosi, A. H. Seno, A. Dehghantanha and R. M. Parizi, A Systematic Literature Review of Integration of Blockchain and Artificial Intelligence, Cham, Switzerland:Springer, pp. 147-160, 2020.

[25] B. Ietto, K. Eisenhut, R. Muth, J. Rabe and F. Tschorsch, "Transparency in Digital-Citizens Interfaces Through Blockchain Technology: Blockchain for Participation Processes in Urban Planning", *2022 IEEE European Technology and Engineering Management Summit (E-TEMS)*, Mar. 2022.

REFERENCE :

[1] G. B. Renuka, P. K. Patjoshi, U. Aswal, G. Manikandan, L. N. Jayanthi, and A. Kaushal, "Integrating Reliable AI to Boost Blockchain's Transparency and Accountability," in *2024 1st International Conference on Advanced Computing and Emerging Technologies (ACET)*, Ghaziabad, India, Aug. 2024, pp. 1-6. doi: 10.1109/ACET61898.2024.10730476.

[2] A. Kuznetsov, P. Sernani, L. Romeo, E. Frontoni and A. Mancini, "On the Integration of Artificial Intelligence and Blockchain Technology: a Perspective about Security", *IEEE Access*, pp. 1, Jan. 2024.

[3] S. Alrubei, E. A. Ball and J. Rigelsford, "A secure blockchain platform for supporting AI-Enabled IoT applications at the edge layer", *IEEE Access*, vol. 10, pp. 18583-18595, Jan. 2022.

[4] O. Fadi, K. Zkik, E. G. Abdellatif and M. Boulmalef, "A survey on blockchain and artificial intelligence technologies for enhancing security and privacy in smart environments", *IEEE Access*, vol. 10, pp. 93168-93186, Jan. 2022.

[5] S. Alrubei, E. A. Ball and J. Rigelsford, "The use of blockchain to support distributed AI implementation in IoT systems", *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14790-14802, Aug. 2022.

[6] B. Ietto, K. Eisenhut, R. Muth, J. Rabe and F. Tschorsch, "Transparency in Digital-Citizens Interfaces Through Blockchain Technology: Blockchain for Participation Processes in Urban Planning", *2022 IEEE European Technology and Engineering Management Summit (E-TEMS)*, Mar. 2022.

[7] J. A. Jaoude and R. George Saade, "Blockchain applications—Usage in different domains", *IEEE Access*, vol. 7, pp. 45360-45381, 2019.

[8] M. Wu, K. Wang, X. Cai, S. Guo, M. Guo and C. Rong, "A comprehensive survey of blockchain: From theory to IoT applications and beyond", *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8114-8154, Oct. 2019.

[9] S. Pahlajani, A. Kshirsagar and V. Pachghare, "Survey on private blockchain consensus algorithms", *Proc. 1st Int. Conf. Innov. Inf. Commun. Technol. (ICIICT)*, pp. 1-6, Apr. 2019.

[10] K. Salah, M. H. U. Rehman, N. Nizamuddin and A. Al-Fuqaha, "Blockchain for AI: Review and open research challenges", *IEEE Access*, vol. 7, pp. 10127-10149, 2019.

[11] P. Bothra, R. Karmakar, S. Bhattacharya and S. De, "How can applications of blockchain and artificial intelligence improve performance of Internet of Things? – A survey", *Computer Networks*, vol. 224, pp. 109634, Apr. 2023.

[12] D. K. Girija, M. Rashmi, P. William and N. Yogeesh, "Framework for Integrating the Synergies of Blockchain with AI and IoT for Secure Distributed Systems", *Lecture notes in networks and systems*, pp. 257-267, 2023.

[13] R. Asif, S. R. Hassan and G. Parr, "Integrating a Blockchain-Based governance framework for responsible AI", *Future Internet*, vol. 15, no. 3, pp. 97, Feb. 2023.

[14] S. K. Lo et al., "Toward trustworthy AI: Blockchain-Based architecture design for accountability and fairness of federated learning systems", *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3276-3284, Feb. 2023.

[15] G. Manikandan and M. Anand, "Radix-2/4 FFT Multiplier less Architecture using MBBLS in OFDM Applications", *Advances in intelligent systems and computing*, vol. 1125, pp. 553-559, 2020.

[16] J. Frizzo-Barker, P. A. Chow-White, P. R. Adams, J. Mentanko, D. Ha and S. Green, "Blockchain as a disruptive technology for business: A systematic review", *Int. J. Inf. Manage.*, vol. 51, Apr. 2020.

[17] S. Kumar, W. M. Lim, U. Sivarajah and J. Kaur, "Artificial intelligence and blockchain integration in business: Trends from a bibliometric-content analysis", *Inf. Syst. Frontiers*, vol. 25, no. 2, pp. 871-896, 2023.

[18] N. Tsolakis, R. Schumacher, M. Dora and M. Kumar, "Artificial intelligence and blockchain implementation in supply chains: A pathway to sustainability and data monetisation?", *Ann. Oper. Res.*, vol. 327, no. 1, pp. 157-210, Aug. 2023.

[19] S. Selvarajan, G. Srivastava, A. O. Khadidos, A. O. Khadidos, M. Baza, A. Alshehri, et al., "An artificial intelligence lightweight blockchain security model for security and privacy in IIoT systems", *J. Cloud Comput.*, vol. 12, no. 1, pp. 38, Mar. 2023.

[20] E. Bertino, A. Kundu and Z. Sura, "Data transparency with blockchain and AI ethics", *J. Data Inf. Qual.*, vol. 11, no. 4, pp. 1-8, Dec. 2019.

[21] S. Vyas, M. Shabaz, P. Pandit, L. R. Parvathy and I. Ofori, "Integration of artificial intelligence and blockchain technology in healthcare and agriculture", *J. Food Qual.*, vol. 2022, pp. 1-11, May 2022.

[22] R. Kumar, D. Singh, K. Srinivasan and Y.-C. Hu, "AI-powered blockchain technology for public health: A contemporary review open challenges and future research directions", *Healthcare*, vol. 11, no. 1, pp. 81, Dec. 2022.

[23] T. N. Dinh and M. T. Thai, "AI and blockchain: A disruptive integration", *Computer*, vol. 51, no. 9, pp. 48-53, Sep. 2018.

[24] A. Ekramifard, H. Amintoosi, A. H. Seno, A. Dehghantanha and R. M. Parizi, A Systematic Literature Review of Integration of Blockchain and Artificial Intelligence, Cham, Switzerland:Springer, pp. 147-160, 2020.

[25] B. Ietto, K. Eisenhut, R. Muth, J. Rabe and F. Tschorsch, "Transparency in Digital-Citizens Interfaces Through Blockchain Technology: Blockchain for Participation Processes in Urban Planning", *2022 IEEE European Technology and Engineering Management Summit (E-TEMS)*, Mar. 2022.