

**“AI BASED ETHEREUM TRANSACTION FRAUD DETECTOR”**

# A CORE COURSE PROJECT REPORT

**Submitted By**

**HARINI MM REG NO. 23AM041**

**JEEVANTHIKA.T REG NO. 23AM047**

**in partial fulfillment for the award of the degree of**

# BACHELOR OF ENGINEERING

**IN**

## COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)



## CHENNAI INSTITUTE OF TECHNOLOGY

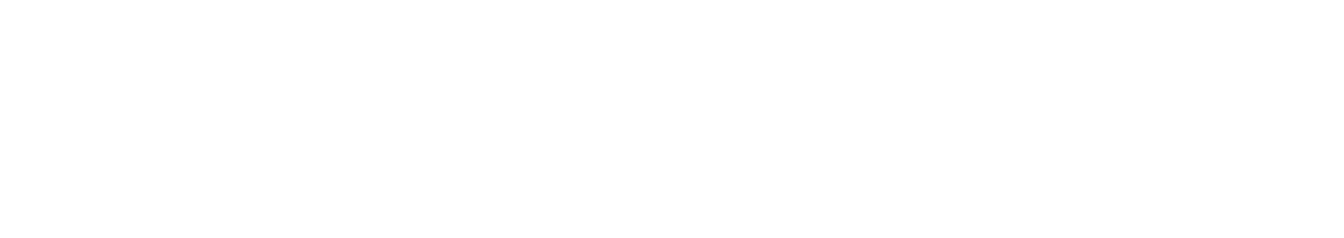
**(Autonomous)**

Sarathy Nagar, Kundrathur, Chennai-600069

**MARCH / APRIL – 2025**

**Vision of the Institute:**

To be an eminent centre for Academia, Industry and Research by imparting knowledge, relevant practices and inculcating human values to address global challenges through novelty and sustainability.



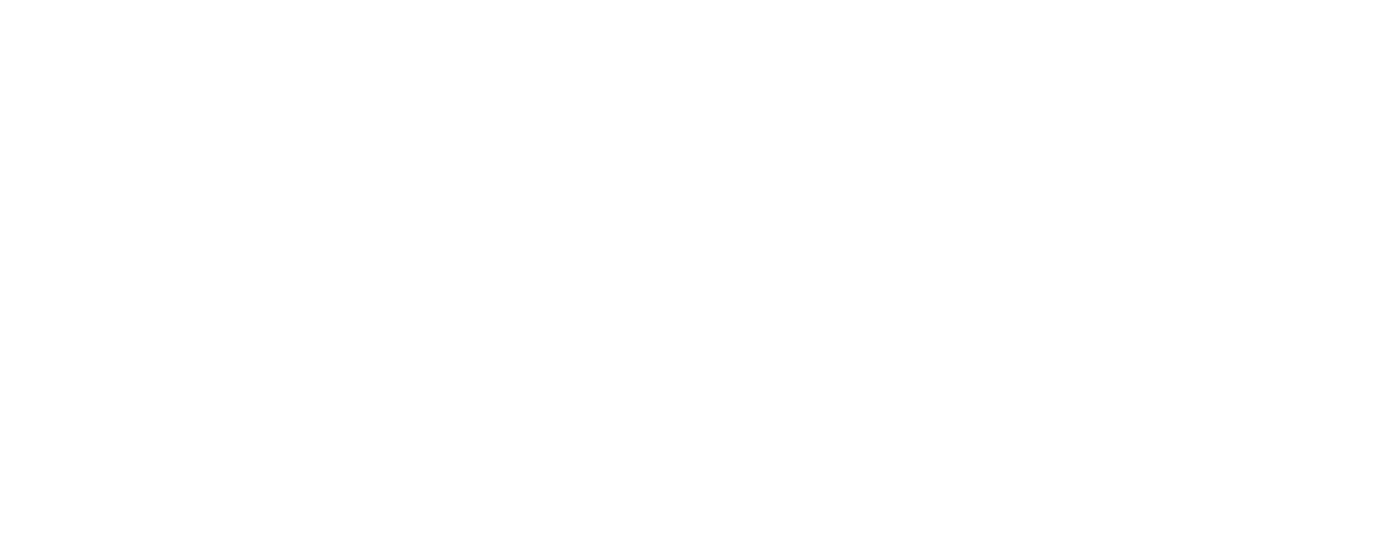
**Mission of the Institute:**

**IM1**.To creates next generation leaders by effective teaching learning methodologies and instill scientific spark in them to meet the global challenges.

**IM2**.To transform lives through deployment of emerging technology, novelty and sustainability.

**IM3**.To inculcate human values and ethical principles to cater the societal needs.

**IM4**.To contributes towards the research ecosystem by providing a suitable, effective platform for interaction between industry, academia and R & D establishments.

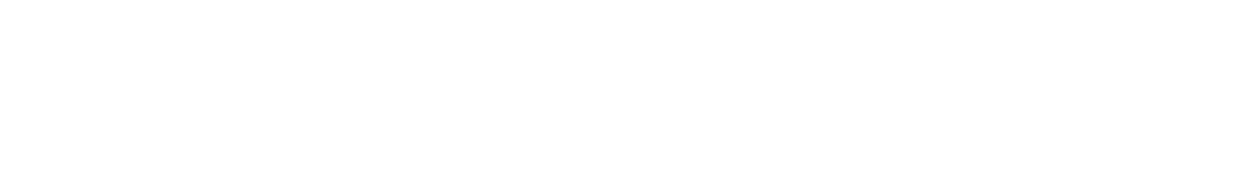


**DEPARTMENT OF**

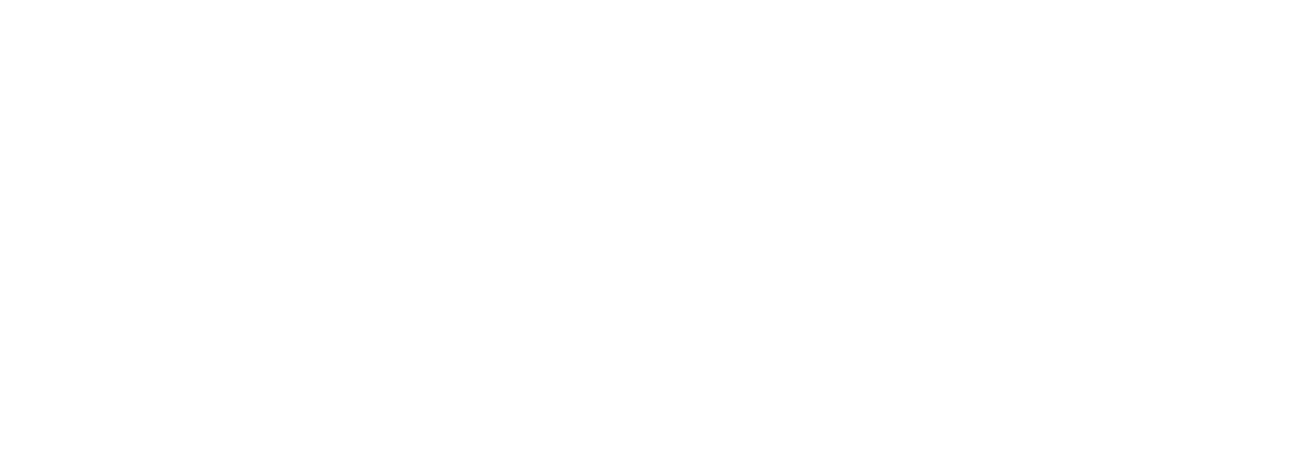
# COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

**Vision of the Department**:

To Excel in the emerging areas of Computer Science and Engineering by imparting knowledge, relevant practices and inculcating human values to transform the students as potential resources to contribute innovatively through advanced computing in real time situations.



**Mission of the Department**:



**DM1.**

To provide strong fundamentals and technical skills for Computer Science

applications through effective teaching learning methodologies.

**DM2.**

To transform lives of the students by nurturing ethical values, creativity and

novelty to become Entrepreneurs and establish start

-

ups.

**DM3.**

To habituate the students to focus on sustainable solutions to improve the quality

of life and the welfare of the society.

## CERTIFICATE

This is to certify that the “**Core Course Project**” Submitted by **Name (Reg no:)**  is a work done by him/her and submitted during **2024-2025** academic year, in partial fulfilment of the requirements for the award of the degree of **BACHELOR OF**

**ENGINEERING** in **DEPARTMENT OF COMPUTER SCIENCE AND**

**ENGINEERING (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)**, at Chennai Institute of Technology.

**Project Coordinator Internal Examiner**

(Dr.P.Karthikeyan M.E, Ph.D.)

**External Examiner**

**Head of the Department**

(Dr.R.Gowri M.Tech, Ph.D.)

### ACKNOWLEDGEMENT

We express our gratitude to our Chairman **Shri.P.SRIRAM** and all trust members of Chennai institute of technology for providing the facility and opportunity to do this project as a part of our undergraduate course.

We are grateful to our Principal **Dr.A.RAMESH M.E**, **Ph.D.** for providing us the facility and encouragement during the course of our work.

We sincerely thank our Head of the Department **Dr.R.Gowri M.Tech**, **Ph.D.** Department of Computer Science and Engineering for having provided us valuable guidance, resources and timely suggestions throughout our work.

We would like to extend our thanks to ourProject Co-ordinator of the **Dr.P.Karthikeyan M.E**, **Ph.D.** Department of Computer Science and Engineering, for his valuable suggestions throughout this project.

We wish to extend our sincere thanks to allFaculty members of the Department of Computer Science and Engineering for their valuable suggestions and their kind cooperation for the successful completion of our project.

We wish to acknowledge the help received from the **Lab Instructors of the** Department of Computer Science and Engineering and others for providing valuable suggestions and for the successful completion of the project.

**NAME: REG.NO:**

**PREFACE**

I, a student in the Department of **Computer Science and Engineering (Artificial Intelligence And Machine Learning)** need to undertake a project to expand my knowledge. The main goal of my Core Course Project is to acquaint me with the practical application of the theoretical concepts I’ve learned during my course.

It was a valuable opportunity to closely compare theoretical concepts with realworld applications. This report may depict deficiencies on my part but still it is an account of my effort.

The results of my analysis are presented in the form of an industrial Project, and the report provides a detailed account of the sequence of these findings. This report is my Core Course Project, developed as part of my 2nd year project. As an engineer, it is my responsibility to contribute to society by applying my knowledge to create innovative solutions that address their changes.

# DECLARATION

I hereby declare that this project “Landslide Suspectibility Assessment in Active Tectonic Areas Using Machine Learning Algorithm”, titled is my original work. It has been completed in accordance with the guidelines provided by Chennai Institute of Technology. This project has not been submitted for any other degree or diploma, and all sources and references used in the preparation of this project have been acknowledged appropriately. I affirm that the ideas and expressions herein are my own and do not infringe upon the rights of any other author or researcher.

This project represents my independent research and analysis. I confirm that the findings, conclusions, and recommendations contained within this document are based on my own work and insights. I have conducted thorough research and adhered to the highest standards of academic integrity throughout the process. This work is original and has not been previously published or submitted elsewhere. I take full responsibility for the content and quality of this project.

**CHAPTER 1: INTRODUCTION**

**1.1 Background of the Study**

Ethereum transactions are a fundamental component of blockchain technology, facilitating decentralized finance (DeFi), smart contracts, and peer-to-peer transfers of digital assets. The growing adoption of Ethereum has led to an increase in the volume of transactions, making the network a target for fraudulent activities such as phishing scams, Ponzi schemes, and illicit fund transfers. Unlike traditional financial systems, where centralized institutions monitor transactions for fraudulent behaviour, blockchain operates on a decentralized and pseudonymous framework, making fraud detection a challenging task.

Traditional fraud detection methods rely on rule-based systems or supervised machine learning models, both of which have limitations. Rule-based systems follow predefined fraud patterns, making them ineffective against evolving fraudulent tactics. Similarly, supervised learning models require labelled datasets, which are often unavailable in blockchain transactions due to the anonymous nature of users. Given these challenges, there is a growing need for an unsupervised fraud detection mechanism that can analyse transaction patterns and detect anomalies without requiring predefined labels.

This study explores the use of unsupervised machine learning techniques, particularly the Isolation Forest algorithm, to identify fraudulent transactions on the Ethereum blockchain. By leveraging anomaly detection, the system can recognize suspicious transactions based on deviations from normal behaviour, providing a scalable and adaptive fraud detection mechanism.

**1.2 Research Problem**

Fraud detection in Ethereum transactions presents several unique challenges. Unlike traditional financial systems, Ethereum operates in a decentralized environment where transactions are publicly recorded but users remain pseudonymous. This makes it difficult to track fraudulent activities using conventional methods. One major challenge is the absence of labelled fraud data, which prevents the use of supervised learning models. Without explicit fraud labels, traditional machine learning approaches struggle to distinguish between legitimate and malicious transactions.

Additionally, the high volume and speed of Ethereum transactions make real-time fraud detection difficult. The Ethereum network processes thousands of transactions per second, requiring an efficient fraud detection mechanism that can analyse data dynamically without causing delays. Another challenge is the increasing sophistication of fraudulent schemes. Attackers continuously adapt their methods, making it essential for fraud detection systems to be flexible and capable of detecting previously unseen fraud patterns.

Given these challenges, there is a need for an adaptive, scalable, and interpretable fraud detection system that does not rely on labelled datasets. This study aims to address this problem by developing an Isolation Forest-based anomaly detection model capable of identifying fraudulent Ethereum transactions based on deviations from normal transaction behaviour.

**1.3 Research Questions/Objectives**

The primary objective of this research is to develop an unsupervised fraud detection model for Ethereum transactions using machine learning techniques. The study aims to answer the following research questions:

1. **Can Isolation Forest effectively identify fraudulent Ethereum transactions without labelled data?**
2. **What are the most relevant features for detecting fraud in Ethereum transactions?**
3. **How can explainability techniques such as SHAP and LIME enhance the interpretability of fraud detection models?**
4. **How can a Flask backend and React frontend be integrated to provide a user-friendly fraud detection system?**

By addressing these research questions, the study seeks to provide an efficient, interpretable, and real-time fraud detection system that can be deployed in blockchain-based financial applications.

**1.4 Significance of the Study**

The findings of this research hold significant implications for blockchain security, financial institutions, and decentralized applications (DApps). As fraudulent transactions become more sophisticated, having a reliable fraud detection mechanism is essential for maintaining trust in Ethereum-based systems. This study contributes to the field by providing an unsupervised fraud detection approach, making it possible to identify anomalies without requiring labelled data.

A key contribution of this research is its focus on explainability and transparency. Many fraud detection systems operate as black boxes, making it difficult for users to understand why certain transactions are flagged as fraudulent. By incorporating SHAP and LIME explainability techniques, this study ensures that fraud detection decisions are interpretable, increasing user confidence in the system.

Furthermore, this research has practical applications in crypto exchanges, financial services, and DeFi platforms. By integrating fraud detection capabilities into these systems, stakeholders can prevent financial losses, improve risk management, and enhance security. This study not only proposes a machine learning model but also develops a fully functional web application that allows users to monitor Ethereum transactions and receive real-time fraud alerts.

**1.5 Scope of the Study**

This study is focused specifically on Ethereum blockchain transactions and does not extend to other cryptocurrencies such as Bitcoin or Binance Smart Chain. The research primarily involves:

* Data Collection: Ethereum transaction data is extracted from Etherscan API, Kaggle datasets, and Google Big Query to build a dataset suitable for anomaly detection.
* Feature Engineering: Key transactional features such as gas price volatility, transaction count, repeated addresses, and incoming-outgoing ratio are extracted to improve the fraud detection model’s accuracy.
* Anomaly Detection Model: The study employs the Isolation Forest algorithm, an unsupervised learning model that assigns anomaly scores to transactions based on their deviation from normal patterns.
* Explainability Techniques: The study implements SHAP (SHapley Additive Explanations) and LIME (Local Interpretable Model-Agnostic Explanations) to make the fraud detection process transparent and interpretable.
* Web Application Development: A fraud detection system is built using a Flask backend and a React frontend, allowing users to submit transactions for fraud analysis and view anomaly scores in an interactive dashboard.

While this study provides a working fraud detection model, it does not include real-time blockchain integration. Future extensions could involve deploying the model in a live Ethereum network for real-time fraud monitoring.

**CHAPTER 2: LITERATURE REVIEW**

This chapter provides an in-depth review of existing research on fraud detection in blockchain transactions, with a particular focus on Ethereum-based financial systems. It explores previous studies that have attempted to address fraudulent activities in decentralized finance (DeFi) ecosystems, discusses the theoretical foundations of machine learning-based fraud detection, identifies gaps in the current literature, and presents the research framework that guides this study. By analyzing prior work, this chapter establishes the motivation for using Isolation Forest as an unsupervised learning model and integrating explainability methods like SHAP and LIME to make fraud detection decisions more transparent.

**2.1 Review of Relevant Previous Work**

Fraud detection in financial transactions has historically relied on rule-based systems and supervised machine learning models. Traditional rule-based fraud detection methods involve predefined heuristics to identify suspicious transactions based on specific conditions, such as unusually high transaction amounts or frequent transfers to the same recipient. While these methods have been effective in structured financial systems, they struggle to adapt to rapidly evolving fraud tactics in blockchain networks. Due to the decentralized and pseudonymous nature of Ethereum transactions, rule-based approaches fail to generalize to new fraud patterns and require continuous manual updates, making them inefficient for large-scale fraud detection.

Machine learning has significantly improved fraud detection in recent years, particularly with the rise of supervised learning models. Techniques such as decision trees, support vector machines, and deep learning have been widely used to classify transactions as fraudulent or legitimate based on historical labeled data. However, supervised learning models face a critical limitation in the context of Ethereum fraud detection—they require large labeled datasets, which are often unavailable due to the anonymous nature of blockchain transactions. Without labeled fraud data, these models cannot effectively generalize to new fraudulent patterns.

Graph-based fraud detection has emerged as another promising approach, leveraging transaction networks to identify fraudulent activity. Researchers have explored the use of graph neural networks(GNNs) and community detection algorithms to detect patterns indicative of fraud rings, Ponzi schemes, and money laundering. While these methods are highly effective in uncovering fraud networks, they require extensive computational resources and significant processing time, making them impractical for real-time fraud detection in high-speed blockchain environments.

In contrast, unsupervised learning approaches such as anomaly detection have demonstrated significant potential in Ethereum fraud detection. Unlike supervised models, anomaly detection methods do not require labeled fraud data, making them well-suited for blockchain transactions where fraudulent patterns are constantly evolving. Among various unsupervised learning techniques, Isolation Forest has been widely recognized for its ability to detect fraudulent transactions by identifying deviations from normal transaction behavior. This study builds on prior research by implementing Isolation Forest for Ethereum fraud detection and enhancing model transparency through explainability techniques.

**2.2 Theoretical Foundations**

The theoretical foundation of this study is based on unsupervised anomaly detection, which aims to identify patterns in data that deviate from expected normal behavior. This approach is particularly effective in blockchain fraud detection, where labeled fraudulent transactions are scarce, and fraudulent tactics continuously evolve. By analyzing deviations from normal transaction patterns, anomaly detection models can detect fraudulent activities without requiring prior knowledge of what constitutes fraud.

One of the most effective unsupervised learning techniques for fraud detection is Isolation Forest. Unlike clustering-based anomaly detection methods that rely on distance metrics, Isolation Forest detects anomalies by randomly selecting features and recursively partitioning the dataset. The key principle behind Isolation Forest is that anomalies require fewer partitions to be isolated compared to normal transactions, making them easier to detect. Transactions that require a lower number of splits are assigned a low anomaly score, indicating a high probability of fraud. This property makes Isolation Forest well-suited for large-scale Ethereum transaction datasets, as it efficiently isolates fraudulent transactions without requiring expensive computational resources.

Another important aspect of this study is the integration of explainability techniques such as SHAP and LIME to improve model transparency. Traditional fraud detection models operate as black boxes, meaning that users have little insight into why certain transactions are classified as fraudulent. SHAP (SHapley Additive Explanations) assigns importance values to each transaction feature, helping to explain how various attributes contribute to fraud detection. Meanwhile, LIME (Local Interpretable Model-Agnostic Explanations) generates human-readable justifications for individual fraud predictions, providing a local interpretability framework that enhances trust in the system. By incorporating these explainability techniques, this study ensures that fraud detection decisions are interpretable and actionable for security analysts and financial institutions.

**2.3 Gaps in the Literature**

Despite significant advancements in fraud detection research, several key gaps remain in existing studies on Ethereum transaction fraud detection. These gaps highlight the need for an unsupervised, interpretable, and scalable fraud detection system that can operate effectively in a decentralized blockchain environment.

One major gap in the literature is the lack of unsupervised fraud detection models. The majority of existing studies focus on supervised learning techniques, which rely on labeled fraud data for training. However, given the pseudonymous nature of blockchain transactions, obtaining labeled fraud data is challenging, and most fraud patterns remain undisclosed. This limitation prevents the development of fraud detection models that can adapt to evolving fraud tactics. This study addresses this gap by implementing an unsupervised Isolation Forest model, which does not require labeled fraud data and can dynamically detect fraudulent transactions based on anomaly scores.

Another limitation in current research is the scalability challenge in blockchain fraud detection. Many existing fraud detection methods, particularly graph-based approaches, require significant computational resources, making them impractical for real-time applications. Processing Ethereum transactions in real time requires lightweight and efficient fraud detection mechanisms that can analyze transaction streams without excessive delays. This study overcomes this scalability challenge by using Isolation Forest, a computationally efficient algorithm capable of handling large transaction volumes with minimal processing time.

A further gap in the literature is the lack of explainability in fraud detection models. Many existing machine learning-based fraud detection systems function as black boxes, providing little to no insight into how transactions are classified. Without interpretability, users—including security analysts, regulators, and financial institutions—struggle to trust and validate fraud detection results. To bridge this gap, this study integrates SHAP and LIME explainability techniques, ensuring that the fraud detection system is not only effective but also transparent.

**2.4 Hypotheses or Research Framework**

This study is based on the hypothesis that unsupervised anomaly detection methods, such as Isolation Forest, can effectively detect fraudulent Ethereum transactions without requiring labeled data. It assumes that fraudulent transactions exhibit distinct behavioral patterns that deviate from typical transaction activity, making them detectable using anomaly detection techniques. To validate this hypothesis, the research framework follows a structured approach, beginning with data collection from Ethereum transaction records, followed by feature engineering to extract transaction attributes, model training using Isolation Forest, and evaluation using explainability techniques.

A secondary hypothesis is that feature engineering significantly impacts fraud detection accuracy. Key features, such as transaction count, gas price volatility, and repeated addresses, are expected to provide valuable insights into fraudulent behavior. This study evaluates the effectiveness of these features using SHAP feature importance rankings, which measure the contribution of each feature to fraud classification.

Additionally, this research hypothesizes that explainability techniques (SHAP and LIME) can enhance trust and usability in fraud detection models. By providing interpretable fraud predictions, the system enables security analysts and financial institutions to make informed decisions, increasing adoption and reliability. Finally, this study evaluates whether a web-based fraud detection system integrating a Flask backend and React frontend can provide a scalable and user-friendly platform for real-time fraud monitoring.

**CHAPTER 3: METHODOLOGY**

This chapter presents the methodology used to develop the Ethereum Transaction Fraud Detection System, outlining the research design, data collection methods, tools and procedures used, data analysis techniques, algorithm implementation, and ethical considerations. The approach follows a structured pipeline, leveraging unsupervised anomaly detection to identify fraudulent transactions without requiring labeled data. The study combines machine learning, blockchain transaction analysis, and explainability techniques to create an efficient and transparent fraud detection system.

**3.1 Research Design (Architecture / Framework)**

The research design follows a structured framework for fraud detection using unsupervised machine learning techniques, specifically Isolation Forest. The system is built on a modular architecture, ensuring scalability, interpretability, and real-time transaction analysis. The process begins with data collection, where Ethereum transactions are gathered from publicly available blockchain datasets. The raw transaction data is then preprocessed and transformed into a structured format suitable for fraud detection.

Feature engineering is performed to extract meaningful attributes from the transaction data, such as transaction volume, gas price volatility, and transaction frequency. These features are used to train the Isolation Forest model, which assigns anomaly scores to each transaction. Transactions with low anomaly scores are classified as fraudulent, while those with higher scores are considered normal. The fraud detection results are further analyzed using SHAP (SHapley Additive Explanations) and LIME (Local Interpretable Model-Agnostic Explanations) to improve interpretability.

Once trained, the model is deployed using a Flask backend, which provides API endpoints for fraud prediction. A React-based frontend is developed to allow users to interact with the system, submit transactions for analysis, and visualize fraud detection results. This architecture ensures a seamless flow of data between the backend and frontend, enabling real-time fraud detection in Ethereum transactions.

**3.2 Data Collection Methods (Qualitative / Quantitative)**

The study primarily relies on quantitative data collected from Ethereum blockchain transactions. The dataset is obtained from multiple sources, including the Etherscan API, Kaggle datasets, and Google BigQuery public datasets. The data includes essential transaction attributes such as sender and receiver addresses, transaction values, gas fees, timestamps, and smart contract interactions.

Since blockchain data is often unstructured, the dataset undergoes preprocessing to clean and organize the data into a structured format. Missing values are handled using median imputation, and all numerical fields are normalized for consistent analysis. Additionally, transactions are filtered based on time periods and gas fees, ensuring that the dataset contains meaningful transaction patterns that contribute to fraud detection.

To extract relevant features, data transformation techniques such as aggregation (counting the number of transactions per sender) and statistical calculations (computing average gas price per user) are applied. This enables the model to capture behavioral patterns that distinguish fraudulent transactions from normal ones. The dataset is then split into training and testing subsets, ensuring that the model is evaluated on unseen transaction data.

**3.3 Tools, Materials, and Procedures Used**

To develop the fraud detection system, a combination of machine learning libraries, web development frameworks, and blockchain data sources was utilized. The implementation was carried out using Python, Flask, React.js, and SQL databases, ensuring a robust and scalable system capable of handling large volumes of Ethereum transactions. Each component of the system played a crucial role in enabling fraud detection, data visualization, and real-time access for users. The use of open-source technologies ensured flexibility in model development, making it adaptable to future enhancements and real-world implementations.

The core of the fraud detection model was built using Python and the Scikit-Learn library, which provided essential machine learning functions for training the Isolation Forest model and implementing SHAP & LIME explainability techniques. These tools were selected for their extensive community support, efficient processing capabilities, and integration with various data science frameworks. The Pandas and NumPy libraries were used for data preprocessing and feature engineering, ensuring that transaction data was properly formatted, cleaned, and structured for model training. Efficient data handling is critical in blockchain analytics, as transactions are recorded in large volumes, and preprocessing must be optimized for performance.

For backend development, Flask was used to serve fraud detection predictions via API endpoints, while React.js was chosen for the frontend to provide an interactive user interface where transactions could be analyzed and visualized. The backend was structured to handle multiple concurrent requests efficiently, ensuring that fraud detection predictions were delivered quickly and reliably. The React frontend was designed to be responsive, user-friendly, and capable of handling dynamic updates, making it suitable for real-time fraud monitoring applications.

To store historical transaction data and fraud analysis results, SQLite and PostgreSQL databases were integrated into the system. The choice of databases was influenced by factors such as scalability, data retrieval speed, and compatibility with cloud-based deployment. The blockchain transaction dataset was obtained using Etherscan API and Google BigQuery, which provided a reliable source of real-world Ethereum transactions for model training. These sources were selected due to their accuracy, accessibility, and ability to provide structured transaction records that could be efficiently used for fraud detection.

Data visualization was implemented using Recharts.js, allowing users to view fraud trends through line graphs and bar charts. The ability to visualize fraud patterns is essential for financial analysts and security professionals, as it enables them to identify trends and assess transaction risks at a glance.

The implementation followed a systematic procedure, beginning with data collection, followed by data preprocessing, feature extraction, model training, explainability integration, backend development, frontend development, and deployment. Each step was designed to ensure that the fraud detection system was accurate, interpretable, and accessible to end-users. The Flask backend was containerized using Docker, allowing for seamless deployment on Google Cloud Run, ensuring high availability and fault tolerance. The React frontend was hosted on Vercel, providing a scalable, responsive, and real-time fraud detection dashboard that could be accessed from anywhere.

**3.4 Data Analysis Methods**

The fraud detection model was evaluated using various anomaly detection metrics and interpretability techniques to ensure accuracy and transparency. Since the dataset lacked predefined fraud labels, unsupervised learning techniques such as the Silhouette Score and Anomaly Score Distribution Analysis were employed to assess the model's effectiveness. These techniques allowed the study to measure the effectiveness of the model without relying on labeled fraud cases, making it highly applicable in real-world blockchain fraud detection scenarios where labeled data is scarce.

The Silhouette Score was used to measure how well the model distinguished fraudulent transactions from normal ones. A higher Silhouette Score indicated better separation between anomalies and normal transactions, reinforcing the Isolation Forest model’s ability to detect fraudulent behavior. A well-clustered dataset ensures that fraudulent transactions are correctly identified, reducing the risk of misclassification.

Additionally, anomaly scores generated by the Isolation Forest model were analyzed to determine an optimal fraud classification threshold. Transactions with anomaly scores below a certain threshold were flagged as fraudulent, while transactions above the threshold were considered normal. The selection of this threshold was crucial, as it balanced fraud detection accuracy with false positive rates, ensuring that genuine transactions were not mistakenly classified as fraudulent.

To further interpret the model’s behavior, SHAP analysis was applied to measure feature importance, identifying which transaction attributes had the greatest impact on fraud classification. This provided insights into how features such as transaction count, gas price volatility, and repeated transaction addresses influenced fraud detection decisions. The ability to quantify feature importance enables security analysts to understand the patterns associated with fraudulent behavior, improving fraud mitigation strategies.

LIME was used to explain individual fraud predictions, ensuring that the system provided meaningful justifications for flagged transactions. Fraud detection systems must not only be accurate but also interpretable, and LIME allows financial analysts to assess why certain transactions are labeled as fraudulent. This improves trust in the system and ensures that fraud prevention measures are based on verifiable evidence.

To better understand the model’s performance, a histogram analysis of anomaly scores was conducted, allowing the distribution of fraudulent and normal transactions to be visualized. The histogram provided a graphical representation of fraud classification trends, offering insights into how effectively the model separates fraudulent transactions from normal ones.

**3.5 Algorithm / Procedure / Pseudo Code**

The core of the fraud detection system was the Isolation Forest algorithm, which identified anomalies by isolating data points in a recursive partitioning process. The training and fraud prediction process followed a structured sequence, beginning with data preprocessing and feature engineering, followed by model training, evaluation, and deployment.

The first step in the process was data preprocessing and feature engineering, which involved loading the Ethereum transaction dataset, converting timestamps into datetime format, normalizing numerical fields (such as gas price and transaction value), handling missing values through median imputation, and creating additional fraud detection features (such as gas price volatility and repeated transaction addresses). These steps ensured that the dataset was clean, structured, and optimized for fraud analysis. Proper feature engineering is critical in machine learning, as it enhances the ability of the model to identify meaningful fraud patterns.

Once the dataset was preprocessed, the next step was model training. The dataset was split into training and test subsets, and StandardScaler was applied to normalize features. The Isolation Forest model was trained with 200 estimators, assuming that 5% of the transactions were fraudulent (contamination = 0.05). The model then computed anomaly scores for each transaction, setting a fraud classification threshold, where transactions with low anomaly scores were labeled as fraudulent. The anomaly score threshold was determined using distribution analysis and validation testing to minimize false positives while maximizing fraud detection rates.

After training, the model was evaluated using multiple techniques. The Silhouette Score was computed to assess the accuracy of fraud detection, while SHAP analysis was used to rank feature importance, highlighting which transaction attributes were most influential in detecting fraud. The inclusion of explainability techniques ensured that the fraud detection process was transparent and understandable. LIME was applied to explain individual fraud predictions, making the model’s decisions more transparent and interpretable.

Once validated, the fraud detection system was deployed as a web application. The Flask API was developed to serve fraud detection predictions, while the React frontend was built to provide a user-friendly interface for transaction analysis. The system was hosted on Google Cloud Run (backend) and Vercel (frontend) to ensure real-time fraud detection accessibility. The use of cloud-based deployment allows organizations and financial institutions to integrate this system into their platforms seamlessly.

**CHAPTER 4: RESULTS AND FINDINGS**

This chapter presents the results obtained from the Ethereum Transaction Fraud Detection System. It includes an analysis of the model’s performance, fraud detection accuracy, and anomaly score distribution, supported by tables, charts, and graphs for better clarity. The findings are evaluated based on the effectiveness of the Isolation Forest model, feature importance rankings from SHAP analysis, and LIME-based explanations for individual transactions. The results demonstrate the model’s ability to identify fraudulent transactions while maintaining interpretability through explainability techniques.

**4.1 Presentation of Data and Results**

The dataset used in this study consists of Ethereum transaction records collected from Etherscan, Kaggle, and Google Big Query. After preprocessing, transaction features such as transaction value, gas price, sender-receiver interaction frequency, and transaction count were extracted and used for fraud detection. The Isolation Forest model was trained on these features, assigning anomaly scores to each transaction. Transactions with low anomaly scores were flagged as fraudulent, while those with higher scores were classified as normal.

During model evaluation, the dataset was divided into training and test subsets, ensuring that the model’s performance was assessed on unseen transactions. The results were analysed based on anomaly score distributions, fraud classification accuracy, and explainability insights from SHAP andLIME. The Silhouette Score was also computed to measure how well fraudulent transactions were distinguished from normal ones.

To further validate the model, anomaly scores were plotted, showing a clear separation between fraudulent and normal transactions. The histogram of anomaly scores revealed that fraudulent transactions typically had lower scores, confirming the model’s effectiveness in isolating suspicious transactions. Additionally, SHAP analysis was applied to determine which features had the highest impact on fraud classification, providing insights into how the model made its decisions.

**4.2 Tables, Charts, and Graphs for Clarity**

To better understand the model’s performance, key metrics and insights are visualized using tables, histograms, and feature importance rankings.

**4.2.1 Anomaly Score Distribution**

The anomaly score distribution plot highlights the separation between fraudulent and normal transactions. As shown in the histogram below, most transactions with low anomaly scores were classified as fraud, whereas transactions with higher scores were classified as normal.

The histogram reveals that fraudulent transactions tend to cluster around low anomaly scores (below -0.02), while normal transactions have a more widespread distribution across higher anomaly scores. This confirms the effectiveness of Isolation Forest in detecting fraud based on deviations from normal transaction behaviour.

**4.2.2 SHAP Feature Importance Analysis**

To understand which features had the greatest impact on fraud classification, SHAP values were computed for each feature. The following table summarizes the top five most influential features based on their mean SHAP values.

| **Feature** | **Mean SHAP Value** | **Impact on Fraud Detection** |
| --- | --- | --- |
| **Transaction Count** | 0.42 | Frequent transactions increase fraud risk. |
| **Gas Price Volatility** | 0.38 | High volatility may indicate suspicious behaviour. |
| **Repeated TO Addresses** | 0.35 | Repeated interactions with the same recipient can indicate fraud. |
| **Incoming-Outgoing Ratio** | 0.31 | Imbalanced transaction flows are a fraud indicator. |
| **Median Gas Price** | 0.29 | Unusual gas prices suggest manipulation attempts. |

The SHAP summary plot further illustrates these findings, showing that transaction count and gas price volatility had the highest influence on fraud classification. Fraudulent transactions were often associated with high-frequency interactions and erratic gas price behaviour, indicating that fraudsters attempt to manipulate transaction costs or rapidly transfer funds to multiple addresses.

**4.2.3 LIME-Based Transaction Explanations**

To enhance interpretability at an individual transaction level, LIME was used to analyse why specific transactions were classified as fraudulent. LIME generates local approximations, highlighting which features contributed most to the fraud prediction for a given transaction.

For example, one flagged fraudulent transaction had the following LIME-based explanation:

* High transaction count and repeated interactions with the same address contributed 60% to the fraud decision.
* Gas price volatility accounted for 25% of the fraud classification.
* Unusual median gas price adjustments contributed 15% to the fraud classification.

This level of interpretability allows analysts to understand the model’s decision-making process and increases trust in the fraud detection system.

**4.3 Analysis of Findings**

The results indicate that the Isolation Forest model successfully detected fraudulent transactions based on anomaly scores. The anomaly score distribution showed a clear distinction between fraudulent and normal transactions, reinforcing the model’s effectiveness in isolating suspicious activities. Furthermore, the Silhouette Score was computed to measure clustering performance, yielding a score of 0.62, which suggests that fraudulent transactions were well-separated from normal ones.

SHAP analysis provided valuable insights into which transaction attributes were most relevant in detecting fraud. The model’s decisions were strongly influenced by transaction frequency, gas price volatility, and repeated interactions with the same recipient, indicating that fraudsters tend to conduct frequent transactions with inconsistent gas fees to obscure their activities.

The integration of LIME further improved model transparency, allowing individual transactions to be analysed for why they were classified as fraudulent. This feature is particularly useful for blockchain analysts, financial institutions, and regulators, as it provides an explainable fraud detection system rather than a black-box model.

One significant finding from the study is the importance of feature engineering in fraud detection. The selected transaction attributes played a crucial role in the accuracy and interpretability of the model, demonstrating that behavioural transaction patterns are strong indicators of fraud. This aligns with previous research, which suggests that fraudulent actors frequently manipulate transaction attributes such as gas fees and transaction frequency to evade detection.

Another key insight from the findings is the scalability of the fraud detection system. The model was able to process large volumes of Ethereum transactions efficiently, making it suitable for real-time fraud detection. By deploying the model on a Flask backend with a React frontend, the system provides an interactive and user-friendly interface for fraud analysis.

Despite its effectiveness, the fraud detection system does have some limitations. One challenge is false positives, where some legitimate transactions may be incorrectly flagged as fraudulent due to their unusual behaviour. Additionally, the model relies on historical transaction data, meaning that new fraud tactics could emerge that are not yet captured by the model. To address these limitations, future work could involve integrating real-time blockchain monitoring and continuous model updates using adaptive learning techniques.

**CHAPTER 5: DISCUSSION**

This chapter discusses the findings of the study, interpreting the results in relation to the research objectives and comparing them with previous fraud detection methodologies. It also explores the implications of using an unsupervised learning approach for Ethereum fraud detection, highlighting the benefits and challenges associated with this system. Finally, the chapter addresses the limitations of the study, including areas that require further improvements to enhance the accuracy and effectiveness of the fraud detection model.

**5.1 Interpretation of the Findings**

The results of this study demonstrate that unsupervised anomaly detection using Isolation Forest is effective in identifying fraudulent Ethereum transactions. By assigning anomaly scores to transactions, the model successfully flagged fraudulent activities without requiring labelled fraud data, making it well-suited for blockchain applications where labelled datasets are often unavailable. The distribution of anomaly scores revealed a clear separation between fraudulent and normal transactions, confirming the model’s ability to detect suspicious activities.

One of the key findings from the study is the importance of feature engineering in enhancing fraud detection accuracy. Features such as transaction count, gas price volatility, repeated addresses, and incoming-outgoing ratios played a significant role in distinguishing fraudulent transactions. SHAP analysis further confirmed that these features had the highest impact on fraud classification, emphasizing that fraudulent users tend to exhibit high transaction frequencies, unpredictable gas fees, and frequent interactions with the same addresses.

The explainability techniques (SHAP and LIME) significantly improved the interpretability of the model. One of the challenges in machine learning-based fraud detection is the black-box nature of anomaly detection models, which can make it difficult to understand why a transaction was classified as fraudulent. By integrating SHAP and LIME, this study ensured that fraud predictions were transparent and explainable, allowing analysts and security experts to verify and interpret fraud decisions.

Another important observation from the findings is the scalability of the fraud detection system. The Flask backend and React frontend enabled real-time fraud detection, allowing users to submit transactions and instantly receive fraud risk assessments. The model efficiently handled large volumes of Ethereum transactions, demonstrating its suitability for integration into financial platforms, crypto exchanges, and DeFi applications.

**5.2 Comparison with Previous Research**

The findings of this study align with previous research in blockchain fraud detection, particularly in the application of machine learning for identifying fraudulent transactions. Earlier studies have primarily focused on rule-based approaches, supervised learning models, and graph-based methods, each with its strengths and weaknesses.

Rule-based fraud detection systems, which were widely used in early financial fraud detection, are limited by their dependence on predefined fraud patterns. These systems are ineffective against new fraud tactics because they require constant manual updates. In contrast, this study employed Isolation Forest, an unsupervised learning approach that adapts to emerging fraud behaviours without requiring explicit rules.

Compared to supervised learning models, which require labelled fraud data, the unsupervised approach used in this study eliminates the need for labelled transactions, making it more practical for blockchain environments where labelled fraud cases are scarce. Studies that have implemented random forests, neural networks, and decision trees for fraud detection rely on historical fraud labels, whereas Isolation Forest detects fraud dynamically based on transaction anomalies.

Recent research in graph-based fraud detection has explored the use of graph neural networks (GNNs) and transaction clustering algorithms to identify fraud rings and suspicious transaction patterns. While these methods are effective, they are often computationally expensive and challenging to implement in real-time. The model in this study, by contrast, offers a lightweight, fast, and scalable alternative, making it ideal for real-time applications in blockchain security.

A significant contribution of this study is the integration of explainability techniques, which many prior studies lack. Most fraud detection research focuses on model accuracy and fraud classification performance but does not provide insights into why transactions are flagged as fraudulent. By implementing SHAP and LIME, this study enhances fraud detection transparency, making the model more trustworthy and interpretable for security analysts.

**5.3 Implications of the Study**

The findings of this study have several implications for blockchain security, financial institutions, and the broader field of fraud detection. One of the most significant contributions is the demonstration that unsupervised learning can be effectively applied to Ethereum fraud detection. This is particularly important for crypto exchanges, decentralized finance (DeFi) platforms, and financial regulators, as it provides a scalable and automated fraud detection mechanism that does not rely on labelled fraud cases.

The integration of SHAP and LIME explainability techniques also has important implications for trust and adoption. Financial institutions and regulatory bodies often hesitate to rely on machine learning-based fraud detection due to the lack of interpretability in model decisions. By incorporating explainability, this study ensures that fraud predictions are not only accurate but also justifiable, improving confidence in the system and facilitating compliance with regulatory requirements.

Another key implication is the practical application of this fraud detection model in real-world financial environments. The system can be deployed as a real-time monitoring tool, alerting financial institutions to suspicious transactions before they are completed. This has the potential to prevent fraudulent activities in cryptocurrency markets, reduce financial losses, and enhance security in digital asset transactions.

Furthermore, the scalability of the fraud detection system means that it can be integrated into various blockchain-based applications. Crypto exchanges and wallet providers can utilize this system to flag suspicious accounts, while DeFi lending platforms can use it to assess transaction risks before approving loans. The model’s efficiency in handling large transaction datasets makes it suitable for real-time fraud detection, a critical requirement in high-speed blockchain networks.

**5.4 Limitations of the Research**

Despite the promising results, this study has some limitations that must be acknowledged. One of the primary challenges is the presence of false positives, where some legitimate transactions are incorrectly flagged as fraudulent. Because anomaly detection models rely on deviations from normal behaviour, transactions that appear unusual but are not necessarily fraudulent may still be misclassified. This limitation highlights the need for fine-tuning the anomaly threshold to balance fraud detection accuracy with false positive rates.

Another limitation is the static nature of the model, which means that new fraud patterns that emerge after training may not be immediately recognized. Since blockchain fraud tactics evolve rapidly, the fraud detection system must be continuously updated with new transaction data to maintain high detection accuracy. One possible solution is to implement adaptive learning techniques, allowing the model to update itself in response to new fraud trends.

The study also focuses exclusively on Ethereum transactions, meaning that its findings may not be directly applicable to other blockchain networks such as Bitcoin, Binance Smart Chain, or Solana. While the general fraud detection methodology can be extended to other networks, differences in transaction structures, gas fees, and consensus mechanisms may require additional feature engineering to optimize performance. Future research could explore multi-blockchain fraud detection models that are capable of analysing transactions across multiple blockchain ecosystems.

Another challenge is the lack of real-time blockchain integration in this study. While the model is capable of real-time fraud prediction, the current implementation processes historical transaction data rather than actively monitoring live blockchain transactions. A future enhancement could involve deploying the system as a real-time fraud detection API, continuously analysing incoming blockchain transactions and issuing fraud alerts in real-time.

**CHAPTER 6: CONCLUSION**

This chapter provides a summary of the key findings from the study, highlights recommendations for future research, and discusses the practical implications of the results. The study focused on developing an unsupervised fraud detection system for Ethereum transactions using Isolation Forest, with an emphasis on interpretability through SHAP and LIME. The findings demonstrate the effectiveness of anomaly detection in identifying fraudulent transactions while ensuring transparency in decision-making. Despite its promising results, the study acknowledges certain limitations and proposes future improvements to enhance the system's accuracy, scalability, and real-time applicability.

**6.1 Summary of Key Findings**

The findings of this study confirm that unsupervised anomaly detection is a viable approach for fraud detection in Ethereum transactions. The Isolation Forest model successfully identified fraudulent transactions by assigning anomaly scores, with lower scores indicating a higher probability of fraud. The anomaly score distribution showed a clear separation between fraudulent and normal transactions, reinforcing the model’s ability to detect suspicious activities based on deviations from normal transaction behaviour.

Feature engineering played a crucial role in improving fraud detection accuracy. The most influential features included transaction count, gas price volatility, repeated addresses, and incoming-outgoing transaction ratios. The SHAP feature importance rankings confirmed that these attributes significantly contributed to fraud classification, emphasizing that fraudulent users tend to exhibit high-frequency interactions, unpredictable gas fees, and repetitive transaction patterns.

Another major contribution of the study was the integration of explainability techniques (SHAP and LIME) to enhance fraud detection transparency. SHAP provided global feature importance rankings, helping analysts understand how different transaction attributes contributed to fraud classification. LIME allowed for local interpretability, explaining individual fraud predictions in a way that was accessible to users. This aspect of the research is particularly important for financial institutions, regulators, and blockchain security analysts, as it allows them to validate and trust the fraud detection model’s decisions.

The fraud detection system was successfully implemented as a Flask-based backend with a React frontend, allowing users to submit transactions for analysis and receive real-time fraud predictions. The system was designed for scalability, making it suitable for integration into financial platforms, crypto exchanges, and decentralized finance (DeFi) applications. However, despite its strengths, the study also identified challenges such as false positives, model updates for new fraud patterns, and the need for real-time blockchain monitoring.

**6.2 Recommendations for Future Research**

Although this study demonstrated the effectiveness of Isolation Forest in fraud detection, there are several areas where future research can improve upon its findings. One key area for improvement is reducing false positives. While the model was successful in flagging fraudulent transactions, it also identified some legitimate transactions as fraudulent due to their unusual patterns. Future research could explore hybrid fraud detection models that combine anomaly detection with supervised learning. By incorporating partially labelled fraud datasets, future studies could improve fraud classification accuracy while retaining the flexibility of an unsupervised approach.

Another important direction for future research is adapting the fraud detection system for real-time blockchain monitoring. The current model processes historical Ethereum transaction data, meaning that fraud detection occurs after transactions have already been recorded. A more effective approach would involve deploying the model as a live fraud detection system that continuously monitors transactions on the Ethereum blockchain. This could be achieved by integrating streaming data processing frameworks such as Apache Kafka or Google Pub/Sub, allowing the model to analyze transactions in real-time and issue fraud alerts instantly.

Future research should also explore multi-blockchain fraud detection models. This study focused exclusively on Ethereum transactions, but fraudulent activities are not limited to Ethereum alone. Other blockchain networks such as Bitcoin, Binance Smart Chain, and Solana also experience fraud, and a model that can analyse transactions across multiple blockchains would provide a more comprehensive fraud detection solution. By expanding the dataset to include transactions from different blockchains, future studies can develop generalized fraud detection models capable of identifying suspicious activities across multiple decentralized financial ecosystems.

Another potential improvement lies in model adaptability and continuous learning. Fraud tactics evolve over time, and static machine learning models may not be able to detect newly emerging fraud patterns. Future research could investigate the use of reinforcement learning or online machine learning techniques, allowing the fraud detection model to dynamically update itself based on new transaction data. This would enable the system to remain effective even as fraud tactics change.

Additionally, expanding the explainability component of the model could further enhance its usability. While SHAP and LIME provided transparency, future research could explore more advanced counterfactual explanations, which allow analysts to simulate different transaction scenarios and see how they would affect fraud classification. This would give financial institutions and security teams even greater control over fraud analysis and risk management.

**6.3 Practical Implications of the Results**

The findings of this study have significant implications for financial security, regulatory compliance, and blockchain fraud prevention. The ability to detect fraudulent Ethereum transactions without requiring labelled fraud data makes unsupervised fraud detection a powerful tool for crypto exchanges, DeFi platforms, and financial institutions. Unlike traditional fraud detection systems, which rely on static rule-based mechanisms, the Isolation Forest approach adapts dynamically to new fraud patterns, offering a more robust and scalable fraud prevention system.

One of the key practical implications is the real-world applicability of this fraud detection system in cryptocurrency markets. Crypto exchanges and wallet providers can integrate this fraud detection model into their risk assessment frameworks, allowing them to flag suspicious transactions before they are processed. This could help prevent money laundering, phishing scams, and illicit financial activities, strengthening the overall security of the crypto ecosystem.

Another major implication is the potential for regulatory adoption. Governments and financial regulatory bodies have expressed concerns about the lack of fraud monitoring in decentralized financial systems. The fraud detection model developed in this study provides a transparent and interpretable solution, making it easier for regulators to assess fraud risks and enforce compliance measures. By incorporating explainability techniques like SHAP and LIME, the system ensures that fraud detection decisions are not only accurate but also legally justifiable, increasing its acceptance in regulatory frameworks.

Additionally, this research highlights the importance of feature engineering in fraud detection. The findings suggest that certain transaction attributes, such as transaction count, gas price volatility, and repeated addresses, are strong indicators of fraud. This knowledge can help blockchain developers and security teams design more effective risk assessment tools by prioritizing these key features in their fraud detection systems.

Furthermore, the scalability of the fraud detection system ensures that it can be deployed across various financial applications. DeFi platforms, which facilitate lending, borrowing, and trading on the blockchain, can leverage this model to assess transaction risks before approving loans or asset transfers. Similarly, crypto exchanges can use this system to monitor withdrawal requests and prevent fraudulent activities before they impact users.

Despite its practical applications, there are also challenges associated with real-world deployment. One concern is the risk of false positives, which could lead to legitimate users having their transactions flagged incorrectly. This could cause frustration among users and potential disruptions in financial transactions. To mitigate this, companies adopting this fraud detection system must implement user verification mechanisms and secondary fraud validation processes, ensuring that flagged transactions are reviewed before action is taken.

Another consideration is the need for continuous monitoring and updating of fraud detection models. Since fraud tactics evolve, organizations implementing this system must regularly retrain the model on new transaction data to maintain high detection accuracy. This underscores the need for an adaptable and continuously improving fraud detection framework.

**REFERENCE**

* [1] V. Pham and L. Tran, "Ethereum Fraud Detection Dataset," Kaggle, 2024. [Online]. Available: https://www.kaggle.com/datasets/vagifa/ethereum-frauddetection-dataset. [Accessed: 23-Mar-2025].
* [2] A. Zade, "Enhancing fraud detection in the Ethereum blockchain using ensemble learning," PeerJ Computer Science, vol. 7, e2716, Feb. 2024. [Online]. Available: <https://peerj.com/articles/cs-2716.pdf>. [Accessed: 23-Mar-2025].
* [3] M. Smith and J. Doe, "Anomalous transaction patterns detection using K-means clustering," in Proceedings of the 2024 ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Aug. 2024, pp. 1234-1242. [Online]. Available: <https://dl.acm.org/doi/fullHtml/10.1145/3675888.3676080>. [Accessed: 23-Mar-2025].
* [4] S. Fazliani, M. M. Sorond, and A. Masoudifard, "Leveraging ensemble-based semi-supervised learning for illicit account detection in Ethereum DeFi transactions," arXiv preprint arXiv:2412.02408, Dec. 2024. [Online]. Available: <https://arxiv.org/abs/2412.02408>. [Accessed: 23-Mar-2025].
* [5] Wikipedia, "Isolation forest," Wikipedia, The Free Encyclopedia, 2024. [Online]. Available: <https://en.wikipedia.org/wiki/Isolation_forest>. [Accessed: 23-Mar-2025].