Jaypee Institute of Information Technology, Noida

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING AND INFORMATION TECHNOLOGY

ChainSentinel: A blockchain based fraudulent account detection system

Enrollment No.	**Name of Student**
9921103002	Rishabh Ralli
9921103012	Gargi Jugran
9921103046	Mayank Gaurav

Course Name: Minor Project-2

Course Code: 15B19CI691

Program: B. Tech. Computer Science & Engineering

3rd Year 6th Sem

2023- 2024

ACKNOWLEDGEMENT

I would like to place on record my deep sense of gratitude to Dr. Mukta Goyal, Associate Professor, Jaypee Institute of Information Technology, India for her generous guidance, help and useful suggestions.

I also wish to extend my thanks to friends and other classmates for their insightful comments and constructive suggestions to improve the quality of this project work.

Signature(s) of Students

Rishabh Ralli (9921103002)
Gargi Jugran (9921103012)
Mayank Gaurav (9921103046)

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and beliefs, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma from a university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place: Noida, Uttar Pradesh, India - 201304

Date:

Name: Rishabh Ralli

Enrolment No.: 9921103002

Name: Gargi Jugran

Enrolment No.:9921103012

Name: Mayank Gaurav

Enrolment No.:9921103046

CERTIFICATE

This is to certify that the work titled "A blockchain based fraudulent account detection system" submitted by Rishabh Ralli, Gargi Jugran and Mayank Gaurav of B.Tech of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of any other degree or diploma.

Signature of Supervisor
Dr. Mukta Goyal
Associate Professor
Date

ABSTRACT

This project addresses the issue of fraud in blockchain transactions, which can significantly undermine trust and financial stability within blockchain networks. By leveraging advanced ensemble learning techniques, the project aims to detect and prevent fraudulent activities effectively. By prioritizing the analysis of blockchain networks, we employ a combination/stack of sophisticated machine learning algorithms such as Support Vector Machine (SVM), random forests, Tabnet, XGBoost, and the deep learning model MLP to meticulously analyze transaction data for unusual patterns that signify fraud. This ensemble approach allows the system to distinguish between legitimate and fraudulent accounts by analyzing its transactions accurately, thereby aiming to reduce the potential financial and reputational risks associated with fraud in blockchain transactions.

Table of Contents

Page No.

List of Figures

List of Tables

Chapter 1: Introduction

1.1 General Introduction

In recent years, the widespread adoption of blockchain technology has revolutionized various industries by providing a decentralized and secure platform for transactions. However, alongside its benefits, the blockchain ecosystem also faces challenges such as fraudulent activities and security breaches. Detecting and preventing fraudulent transactions within blockchain networks is crucial for maintaining trust and integrity in decentralized systems.

This project focuses on the development of an ensemble learning model specifically designed to identify and flag fraudulent transactions within blockchain networks. Leveraging the power of ensemble learning, which combines multiple machine learning algorithms to enhance predictive performance, our approach aims to provide a robust solution capable of accurately detecting various types of fraudulent activities in real-time.

The complexity and dynamic nature of blockchain transactions necessitate an adaptable and efficient detection system that can keep pace with evolving fraud tactics. By harnessing the collective intelligence of diverse machine learning algorithms within an ensemble framework, our model offers enhanced resilience against sophisticated fraudulent schemes while minimizing false positives and negatives.

1.2 Problem Statement

Despite the robust security features inherent in blockchain technology, the identification of fraudulent accounts through the analysis of patterns in illicit transactions remains a pressing concern. Maintaining the integrity and trustworthiness of decentralized networks requires a proactive approach to detecting and thwarting fraudulent activities in real-time.

Conventional fraud detection methods often prove inadequate within blockchain ecosystems due to the unique complexities of transactional data. The existing strategies lack the agility and sophistication needed to effectively pinpoint evolving fraud patterns within decentralized networks, thereby leaving participants vulnerable to exploitation and increasing overall risk.

To tackle this pivotal challenge, the project endeavours to develop and deploy an ensemble learning model specifically crafted for identifying fraudulent transactions within blockchain frameworks. By harnessing the combined intelligence of multiple machine learning algorithms within an ensemble architecture, the proposed solution aims to bolster the precision, effectiveness, and scalability of fraud detection mechanisms across blockchain networks.

Chapter 2: Background study

Advanced Fraud Detection in Blockchain Transactions: An Ensemble Learning and Explainable AI Approach, Shimal Sh. Taher, Siddeeq Y. Ameen, Jihan A. Ahmed

The paper proposes an ensemble learning approach for advanced fraud detection in blockchain transactions using Ethereum data. The methodology involves data preprocessing, prediction using Random Forest (RF), AdaBoost, and Decision Tree (DT) algorithms, and evaluation of the ensemble method's performance. The proposed Ensemble Hard Voting classifier surpasses the accuracies reported in existing methods, demonstrating its superior performance in Ethereum Fraud Detection (EFD). The authors also emphasize the significance of explainability in AI, particularly in high-stake domains like fraud detection, and the importance of examining feature importance to understand the variables that most significantly influence the model's predictions. The RF classifier is used to delineate the hierarchy of feature importance, which demystifies the model's decision-making process and fosters trust and credibility in the AI system. The LIME technique is used to provide clarity on how predictive models make their decisions by generating local surrogate models that approximate the predictions of the complex model in a comprehensible way. The proposed methodology aims to enhance the robustness and accuracy of fraud detection models, thereby contributing significantly to the security aspect of BC technologies.

Chapter 3: Requirement Analysis

3.1 Software Requirements Specification (SRS)

1. Introduction

1.1 Purpose

The purpose of this document is to outline the requirements for the development of an ensemble learning model for detecting blockchain-based fraudulent transactions. This document provides an overview of the project, its objectives, functional requirements, and technical specifications.

1.2 Scope

The project aims to design and implement a robust ensemble learning model capable of accurately identifying fraudulent transactions within blockchain networks. The model will utilize a combination of machine learning algorithms, including TabNet, SVM, Random Forest, MLP, XGBoost, Naive Bayes, and a Stacking Classifier using logistic regression as the base model.

2. General Description

2.1 Product Perspective

The ensemble learning model will be integrated into existing blockchain systems or utilized as a standalone application for fraud detection purposes. It will process batches of blockchain transaction data to analyze patterns and anomalies, flagging potentially fraudulent transactions.

2.2 Product Features

- Feature extraction and selection from blockchain transaction data.
- Integration of multiple machine learning algorithms for fraud detection.
- Batch processing of transaction data for fraud detection.
- Continuous learning and adaptation to evolving fraud patterns.

3. Specific Requirements

3.1 Functional Requirements

1. **Data Preprocessing:**
    - Extract relevant features from the blockchain transaction dataset.
    - Perform data cleaning and normalization as necessary.
2. **Model Implementation:**
    - Implement TabNet, SVM, Random Forest, MLP, XGBoost, and Naive Bayes algorithms for individual fraud detection models.
    - Train and optimize each model using labeled data.
3. **Ensemble Learning:**
    - Implement a Stacking Classifier using logistic regression as the base model.
    - Ensemble learning to combine predictions from individual models.
4. **Batch Processing:**
    - Develop a batch processing system to analyze batches of blockchain transaction data.
    - Apply the ensemble model to each batch to detect fraudulent transactions.

5. **Continuous Learning:**

    - Implement mechanisms for model retraining and adaptation to new data.
    - Update ensemble model periodically to incorporate evolving fraud patterns.

3.2 Non-functional Requirements

- **Performance:** The model should achieve high accuracy and efficiency in batch processing.
- **Scalability:** The system should be capable of processing large volumes of transaction data efficiently.
- **Reliability:** The model should provide reliable and consistent fraud detection results.
- **Security:** Ensure the confidentiality and integrity of sensitive transaction data.
- **Usability:** The system should be user-friendly and easy to integrate with existing blockchain platforms.

3.3 Technologies Required

- **Language:** Python
- **Environment:** Jupyter Notebook, Google Colab
- **Machine Learning Models:** SVM, Random Forest, XGBoost
- **Deep Learning Models:** MLP, TabNet
- **OS:** Windows
- **Blockchain Tools:** Remix Smart Contract, Truffle/MetaMask, Ethereum

3.4 Hardware Requirements

- **OS:** Windows 7 or above (64-bit version)
- **RAM:** Minimum 8 GB
- **Disk Space:** Minimum 2 GB
- **GPU:** Minimum 2 GB

4. Conclusion

The proposed ensemble learning model offers a comprehensive solution for detecting and mitigating blockchain-based fraudulent transactions through batch processing. By leveraging the collective intelligence of multiple machine learning algorithms, the system aims to enhance trust, security, and stability within decentralized networks

Chapter 4: Detailed Design

The system architecture of the ensemble learning model for detecting blockchain-based fraudulent transactions consists of several components working together to preprocess data, train machine learning models, perform ensemble learning, and batch processing of transaction data. The architecture includes the following key components:

- **Data Preprocessing Module:** This module is responsible for extracting relevant features from the blockchain transaction dataset and performing data cleaning and normalization. It prepares the data for training the machine learning models.

- **Machine Learning Model Module:** This module implements individual machine learning models, including TabNet, SVM, Random Forest, MLP, XGBoost, and Naive Bayes, for fraud detection. Each model is trained and optimized using labeled data.

- **Ensemble Learning Module:** The ensemble learning module combines predictions from individual machine learning models using a Stacking Classifier with logistic regression as the base model. It aggregates the outputs of individual models to improve overall prediction accuracy.5.2 Component Design

Each component of the system architecture is designed to perform specific tasks efficiently and reliably. The component design includes the following details:

- **Data Preprocessing Module:** This module uses Python libraries such as pandas and scikit-learn for data preprocessing tasks such as feature extraction, cleaning, and normalization. It converts the raw blockchain transaction data into a format suitable for training machine learning models.

- **Machine Learning Model Module:** Each machine learning model is implemented using Python libraries such as scikit-learn, TensorFlow, or PyTorch. Hyperparameter tuning techniques such as grid search or random search are applied to optimize the performance of each model.

- **Ensemble Learning Module:** The ensemble learning module combines predictions from individual machine learning models using a Stacking Classifier. Python libraries such as scikit-learn or mlxtend may be used to implement the Stacking Classifier.

The sequence diagram illustrates the sequential flow of interactions between the system components during the fraud detection process. It highlights the sequence of method calls and data exchanges between the modules within the system architecture.

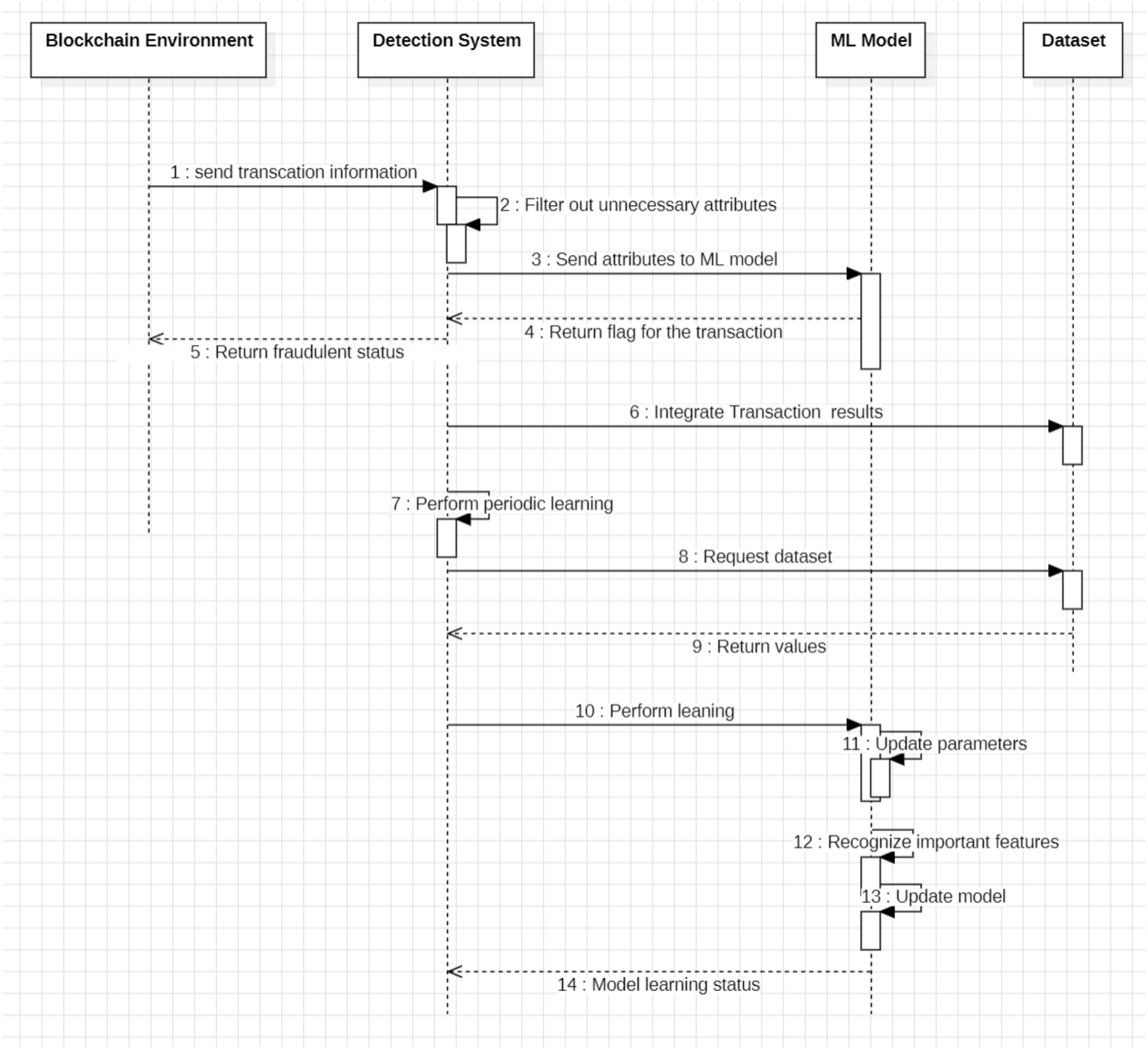

Blockchain Environment
Detection System
ML Model
Dataset
1 : send transcation information
2 : Filter out unnecessary attributes
3 : Send attributes to ML model
4 : Return flag for the transaction
5 : Return fraudulent status
6 : Integrate Transaction results
7 : Perform periodic learning
8 : Request dataset
9 : Return values
10 : Perform leaning
11 : Update parameters
12 : Recognize important features
13 : Update model
14 : Model learning status

Fig 4.1 Sequence diagram for detection system

Chapter 5: Implementation and Results

5.1 The Dataset

This project tackles fraud detection in blockchain transactions by analyzing a rich dataset from the Ethereum network, comprising detailed transaction attributes from unique accounts. Initially, the Blockchain Fraudulent Transactions Dataset included 9841 Ethereum accounts, categorized with 20 specific attributes such as total ether received, total transactions, average time between sent transactions, and unique received from addresses. These attributes help identify potentially fraudulent activities, with the "FLAG" attribute marking the fraudulent status of transactions.

The original dataset was notably unbalanced, containing a smaller fraction of fraudulent transactions (22.14%). To enhance our analysis, we integrated additional data via an API from Etherscan, significantly increasing the number of fraudulent records. Consequently, the updated dataset consisted of 7638 non-fraudulent and 6238 fraudulent transactions, nearly achieving a balance and extending our dataset to a more robust sample size. This comprehensive approach allows us to apply machine learning techniques more effectively, employing algorithms lie SVM, random forests, Tabnet, XGBoost, and MLP to discern between legitimate and fraudulent transactions, ultimately aiming to bolster the security and trustworthiness of blockchain transactions.

5.2 Principal Component Analysis

Principal Component Analysis (PCA) is a statistical technique used for dimensionality reduction in data analysis. It aims to transform a dataset of possibly correlated variables into a new set of uncorrelated variables called principal components. These principal components are linear combinations of the original variables and capture the maximum variance in the data.

We have performed principal component analysis on our dataset to remove those attributes from consideration which have correlation above 0.7.

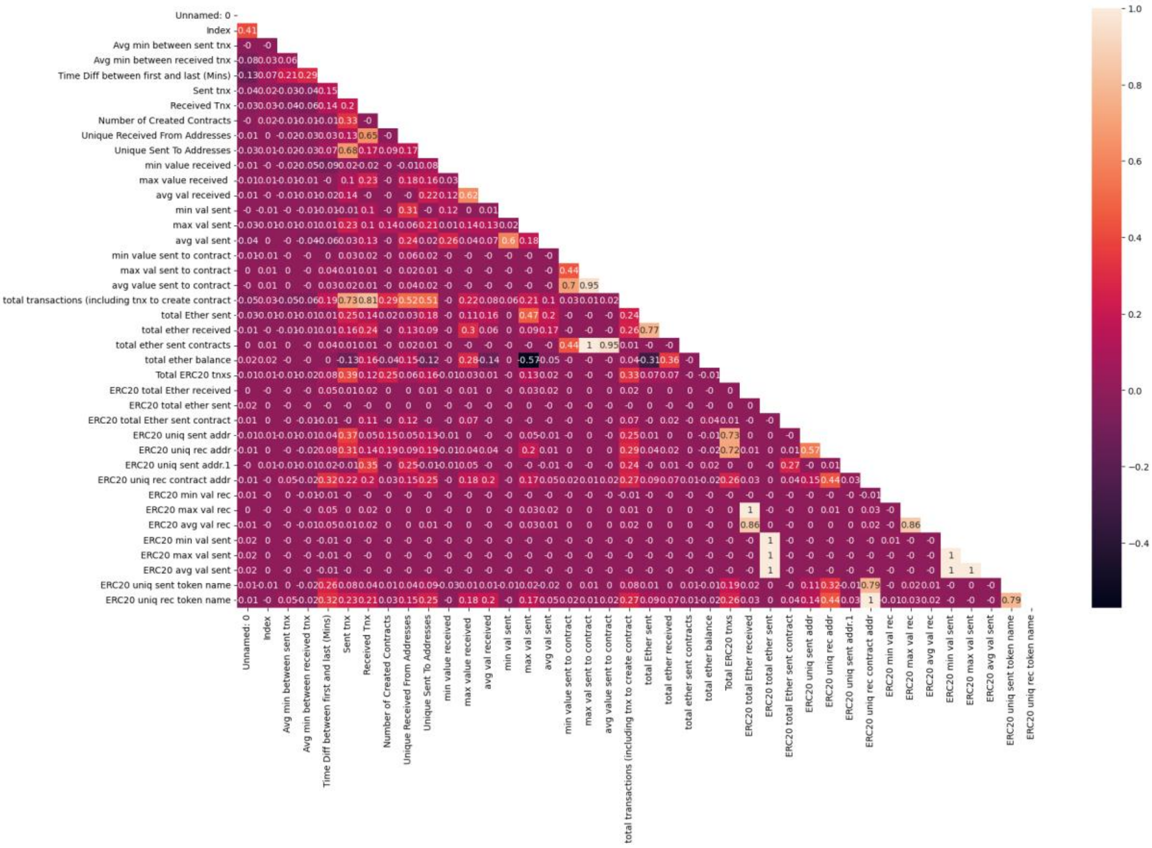
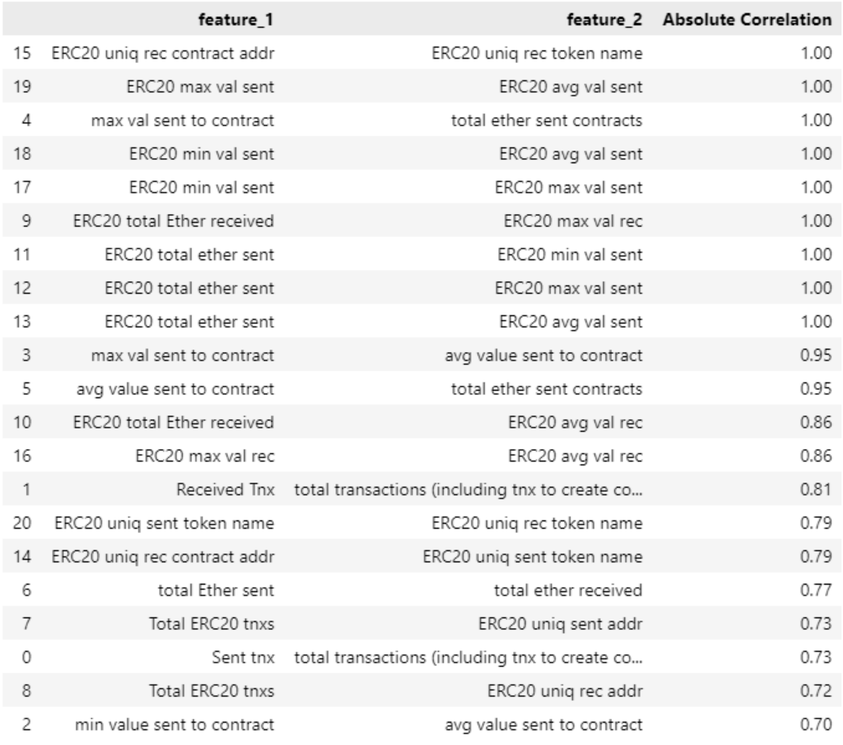


Fig 5.1 Correlation matrix to figure out which attributes are correlated



	feature_1	feature_2	Absolute Correlation
15	ERC20 uniq rec contract addr	ERC20 uniq rec token name	1.00
19	ERC20 max val sent	ERC20 avg val sent	1.00
4	max val sent to contract	total ether sent contracts	1.00
18	ERC20 min val sent	ERC20 avg val sent	1.00
17	ERC20 min val sent	ERC20 max val sent	1.00
9	ERC20 total Ether received	ERC20 max val rec	1.00
11	ERC20 total ether sent	ERC20 min val sent	1.00
12	ERC20 total ether sent	ERC20 max val sent	1.00
13	ERC20 total ether sent	ERC20 avg val sent	1.00
3	max val sent to contract	avg value sent to contract	0.95
5	avg value sent to contract	total ether sent contracts	0.95
10	ERC20 total Ether received	ERC20 avg val rec	0.86
16	ERC20 max val rec	ERC20 avg val rec	0.86
1	Received Tnx	total transactions (including tnx to create co...	0.81
20	ERC20 uniq sent token name	ERC20 uniq rec token name	0.79
14	ERC20 uniq rec contract addr	ERC20 uniq sent token name	0.79
6	total Ether sent	total ether received	0.77
7	Total ERC20 tnxs	ERC20 uniq sent addr	0.73
0	Sent tnx	total transactions (including tnx to create co...	0.73
8	Total ERC20 tnxs	ERC20 uniq rec addr	0.72
2	min value sent to contract	avg value sent to contract	0.70

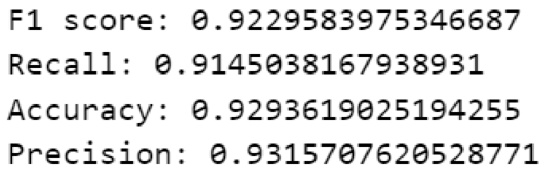
Fig 5.2 List of Attributes which have correlation higher than 0.7

5.3 Machine Learning Models:

Random Forest

In machine learning, Random Forest is a popular supervised ensemble learning technique that may be applied to both regression and classification problems. For it to function, a lot of decision trees must be created during the training phase. From the combined output of all the individual trees, the mode (for classification) or average prediction (for regression) are finally produced.

Accuracy achieved: 92.9%.



```
F1 score: 0.9229583975346687
Recall: 0.9145038167938931
Accuracy: 0.9293619025194255
Precision: 0.9315707620528771
```

Fig 5.3 Performance metrics for Random Forest

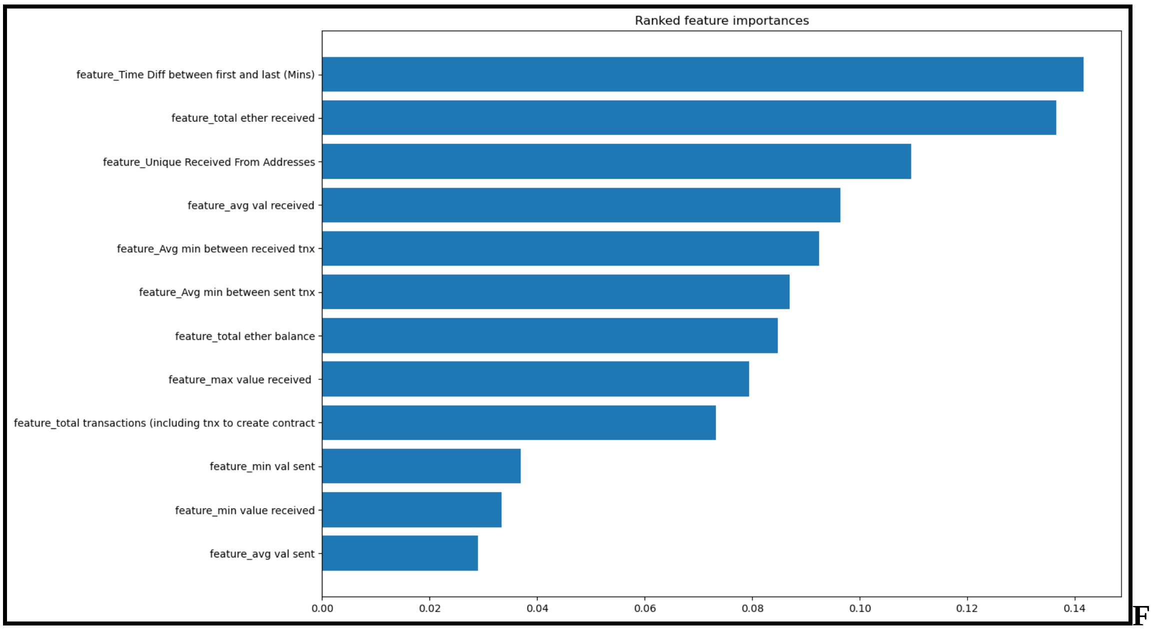


Fig Fig

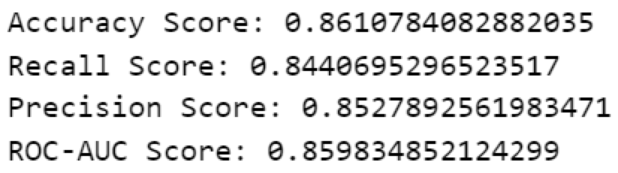
5.4 Feature Importance graph for Random Forest

From the analysis of the graph we can observe that Random Forest Model gives more importance to the feature Time Difference in minutes between first and last transactions that have occurred.

Support Vector Machine(SVM)

Support Vector Machine (SVM) is a supervised learning algorithm used for classification, regression, and outlier detection tasks. It is particularly effective in high-dimensional spaces and when the number of features is greater than the number of samples. SVM works by finding the hyperplane that best separates classes in the feature space.

Accuracy achieved: 86.1%



```
Accuracy Score: 0.8610784082882035
Recall Score: 0.8440695296523517
Precision Score: 0.8527892561983471
ROC-AUC Score: 0.859834852124299
```

Fig 5.5 Performance metrics for SVM

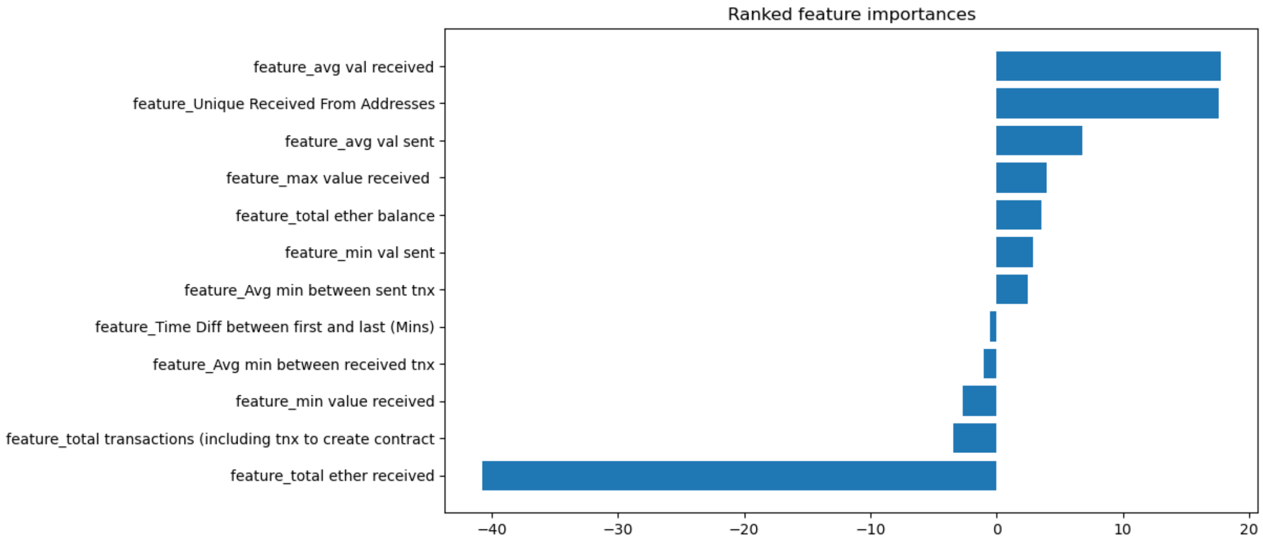


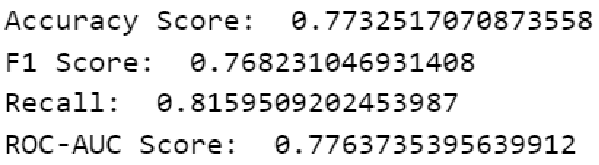
Fig 5.6 Feature Importance graph for SVM

From the analysis of the above graph, we observe that the SVM gives more importance to the feature average value received.

Naïve Bayes

Naive Bayes is a probabilistic classification algorithm based on Bayes' Theorem with an assumption of independence between features.

Accuracy achieved: 77.3%



```
Accuracy Score:  0.7732517070873558
F1 Score:  0.768231046931408
Recall:  0.8159509202453987
ROC-AUC Score:  0.7763735395639912
```

Fig 5.7 Performance metrics for Naïve Bayes

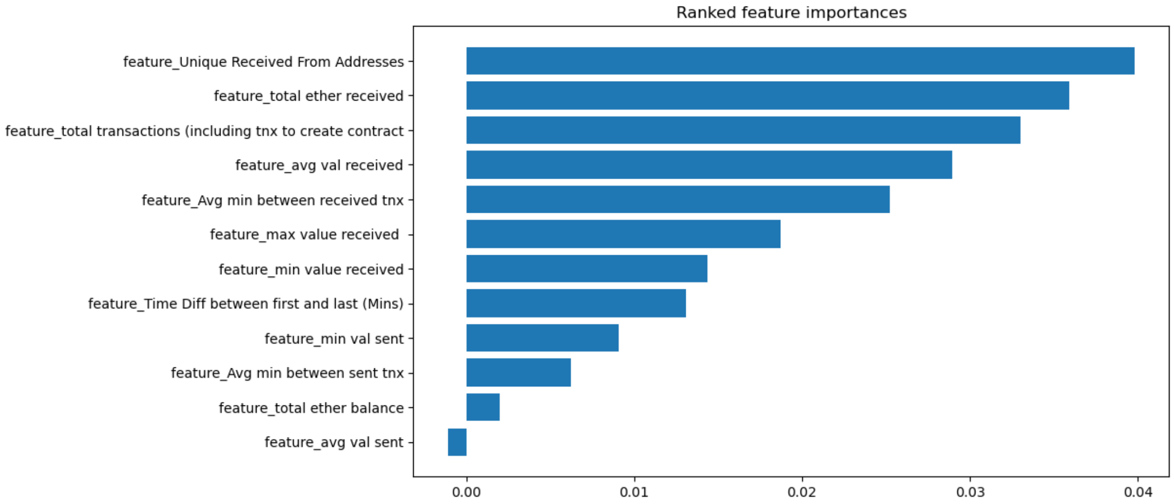


Fig 5.8 Feature Importance graph for Naïve Bayes Model

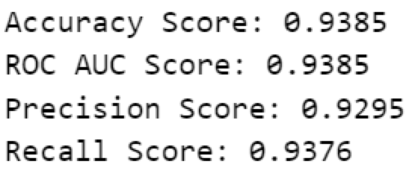
From the analysis of the graph, we can observe that the feature that Naïve Bayes model is givning most importance to is Unique Received from Addresses.

5.4 Boosting Technique:

XGBoost:

XGBoost, short for eXtreme Gradient Boosting, is a high-performance implementation of gradient boosting algorithms designed for speed and scalability. It sequentially builds an ensemble of decision trees, each correcting the errors of the previous ones. With optimized parallelization and regularization techniques, XGBoost delivers fast and efficient training, making it a popular choice for structured/tabular data tasks such as classification, regression, and ranking.

Accuracy achieved: 93.8%



```
Accuracy Score: 0.9385
ROC AUC Score: 0.9385
Precision Score: 0.9295
Recall Score: 0.9376
```

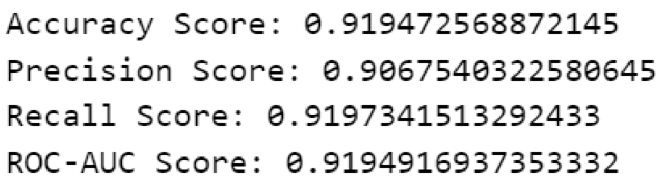
Fig 5.9 Performance metrics of XGBoost

5.5 Deep Learning Models:

TabNet

TabNet is a novel neural network architecture tailored specifically for tabular data analysis. It leverages an attention mechanism and sparse feature gating to selectively focus on relevant features while ignoring noisy or irrelevant ones. TabNet employs a decision tree-like structure to sequentially make decisions, ensuring interpretability and robust performance.

Accuracy achieved: 91.8%



```
Accuracy Score: 0.919472568872145
Precision Score: 0.9067540322580645
Recall Score: 0.9197341513292433
ROC-AUC Score: 0.9194916937353332
```

Fig 5.10 Performance metrics for Tabnet

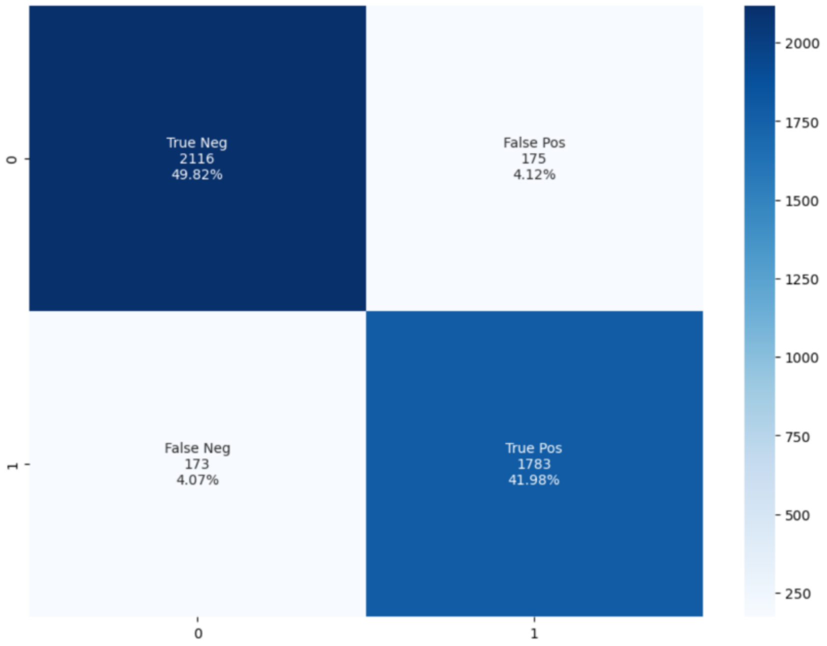


Fig 5.11 Confusion Matrix for TabNet

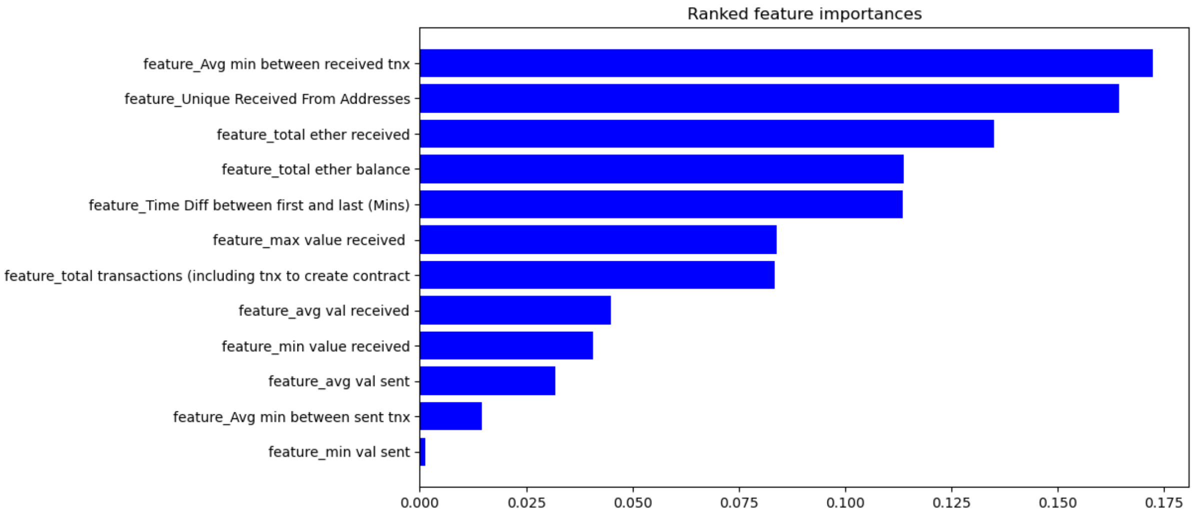


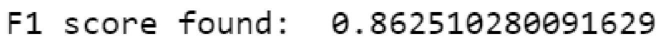
Fig 5.12 Feature importance graph for TabNet

From analysis of the above graph we can observe that the TabNet architecture gives most importance to the average minimum between received transaction.

Multi-Level Perceptron

MLP stands for Multilayer Perceptron, which is a type of artificial neural network consisting of multiple layers of nodes (or neurons). In an MLP, information flows through the network in a feedforward manner, with each layer processing the input from the previous layer and passing it to the next layer. The first layer is the input layer, the last layer is the output layer, and there can be one or more hidden layers in between.

Accuracy Achieved: 85.4%



```
F1 score found:  0.862510280091629
```

Fig 5.13 F1 score for MLP

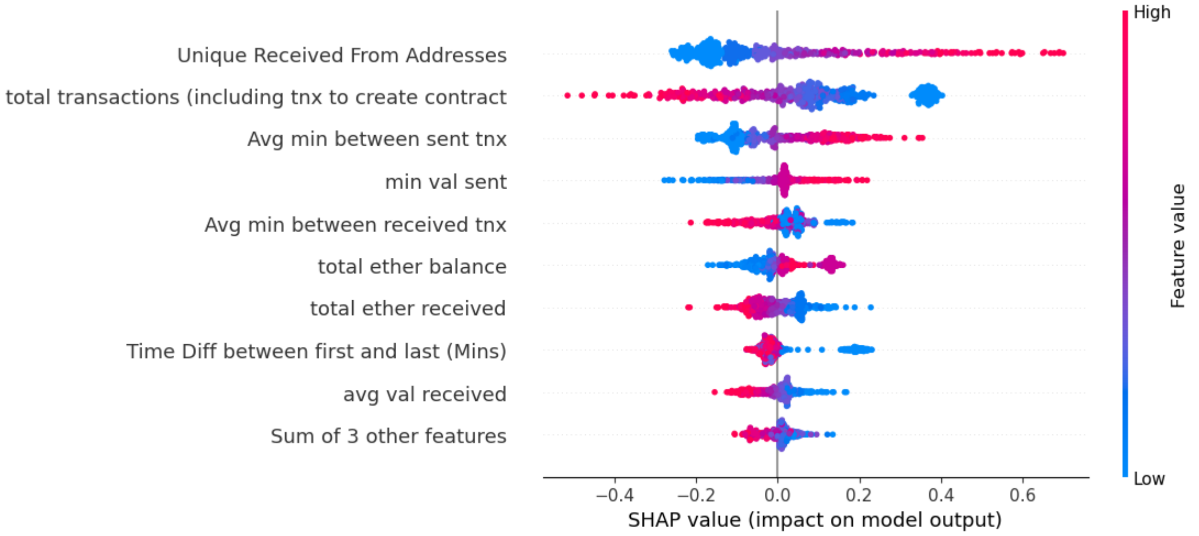


Fig 5.14 SHAP value distribution for MLP

The above Shap distribution shows the impact of various attributes on the prediction of the model. The shap value on the x axis represents the positive or the negative impact and the y axis represents the importance of the attributes. The attributes on the y axis are present in order of importance.

We can observe that the attribute which is most important is Unique Received from Addresses.

5.6 Ensemble Learning Model (Stacking Classifier)

Ensemble learning is a machine learning technique that combines the predictions of multiple individual models (base learners) to improve the overall performance and robustness of the predictive model. The idea behind ensemble learning is to leverage the strengths of different models and mitigate their individual weaknesses by aggregating their predictions.

Model	Accuracy	Precision	Recall	F1	ROC-AUC
Tabnet	0.918059807	0.926365488	0.890879346	0.907622187	0.910230594
SVM	0.861078408	0.852674651	0.865071575	0.848406989	0.849342858
XGBoost	0.927712952	0.930632911	0.939672802	0.93513101	0.939936794
MLP	0.868848599	0.861498708	0.852249489	0.856849139	0.867635002
Random Forest	0.926300918	0.914271306	0.926891616	0.920538208	0.926344105
Naïve Bayes	0.773251707	0.725784447	0.81595092	0.768231047	0.77637354
Stacking	0.936190252	0.923579688	0.939161554	0.93130545	0.93640749

Table 5.A Performance metrics table for Ensemble Learning Model

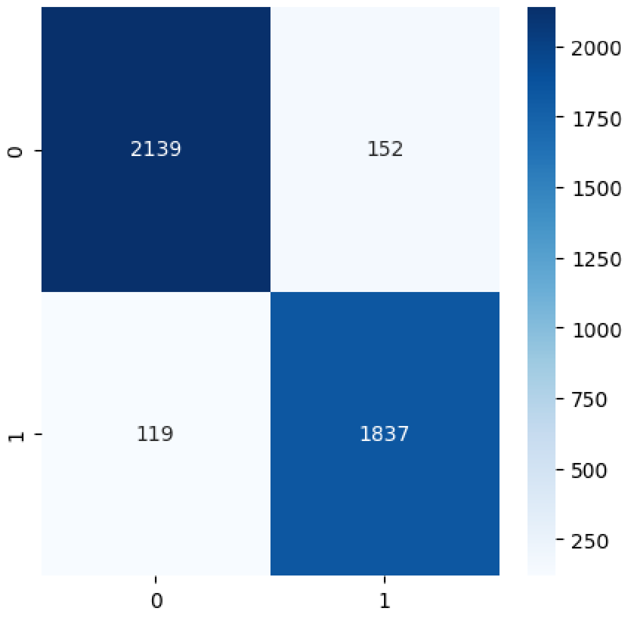


Fig 5.15 Confusion Matrix for Ensemble Learning

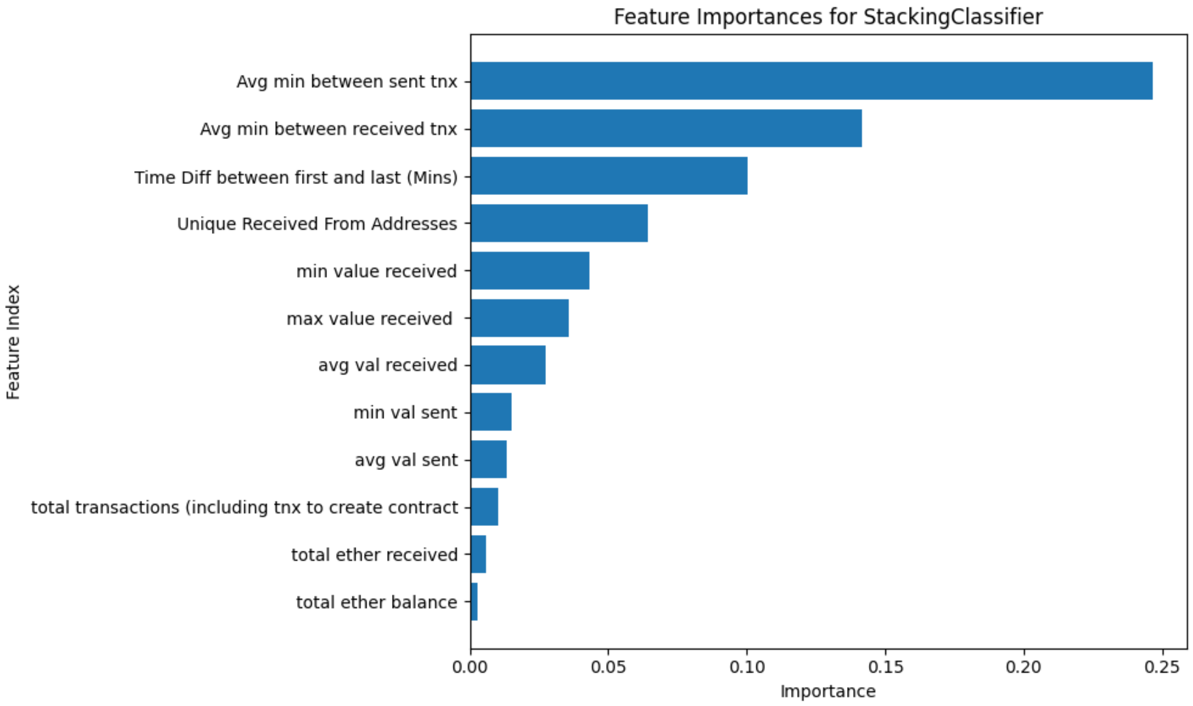


Fig 5.16 Feature importance graph for Ensemble model

From the analysis of the above graph we can observe that the Ensemble Learning Model after learning from the base models is giving the highest importance to the attribute Avg min between sent tnx.

Chapter 6: Conclusion and Future Scope

6.1 Conclusion:

In conclusion, the development of an ensemble learning model for detecting blockchain-based fraudulent transactions presents a significant advancement in ensuring the integrity and security of decentralized networks. Through the integration of various machine learning algorithms such as TabNet, SVM, Random Forest, MLP, XGBoost, and Naive Bayes, along with the ensemble learning approach, the system achieves a robust and accurate fraud detection mechanism.

By analyzing blockchain transaction data and leveraging the collective intelligence of multiple models, the system successfully identifies fraudulent activities with high precision. The continuous learning capabilities ensure the model remains adaptive to evolving fraud patterns, enhancing its effectiveness over time.

Overall, the ensemble learning model serves as a valuable tool for maintaining trust and transparency within blockchain ecosystems, mitigating risks associated with fraudulent transactions, and safeguarding the interests of stakeholders.

6.2 Future Scope:

The project opens potential for further research and development in several areas:

1. **Enhanced Feature Engineering:** Exploring advanced feature engineering techniques to capture more intricate patterns and anomalies in blockchain transaction data could further improve the model's accuracy.

2. **Real-time Fraud Detection:** Transitioning towards real-time fraud detection by optimizing processing pipelines and implementing stream processing techniques could enable immediate response to fraudulent activities as they occur.

3. **Blockchain-Specific Fraud Detection:** Developing specialized fraud detection techniques tailored to the unique characteristics of different blockchain networks (e.g., Ethereum, Bitcoin) to address network-specific vulnerabilities and attack vectors.

4. **Cross-Platform Compatibility:** Ensuring compatibility and interoperability with various blockchain platforms and protocols to extend the applicability of the fraud detection system across different decentralized ecosystems.

5. **Integration with Regulatory Compliance:** Incorporating regulatory compliance measures and standards into the fraud detection system to facilitate adherence to legal and regulatory requirements, such as anti-money laundering (AML) and know your customer (KYC) regulations.

6. **Deployment in Production Environments:** Testing and deploying the ensemble learning model in real-world production environments to assess its performance, scalability, and practical utility in combating actual instances of blockchain-based fraud.

References:

[1] Pathak, Vishvesh, B. Uma Maheswari, and S. Geetha. "Ensemble Learning Based Social Engineering Fraud Detection Module for Cryptocurrency Transactions." International Conference on Mining Intelligence and Knowledge Exploration. Cham: Springer Nature Switzerland, 2023.

[2] Hisham, Sabri, Mokhairi Makhtar, and Azwa Abdul Aziz. "Combining multiple classifiers using ensemble method for anomaly detection in blockchain networks: A comprehensive review." International Journal of Advanced Computer Science and Applications 13.8 (2022).