Fraudulent Transactions Detection in Bitcoin Network using ML & DL Techniques

Asish Koyagura[1], Dr. S. Suhasini[2]

[1]M. Tech, Data Science, [2]Ph.D, Associate Professor, V R Siddhartha Engineering College Vijayawada, India

***Abstract*—With the advent of digital transactions, the use of crypto currency has become an important part in the domain of banking. This also led to the development of various methods to conduct fraudulent transactions which led to many scams. Bitcoin crypto currency is the first crypto currency to be introduced to the world. Even the bitcoin network built based upon blockchain technology is not free from these frauds. Hence, there is a need to take measures against these cries. This work researches the use of machine learning algorithms such as Logistic Regression (LR), Random Forests (RF), Multilayer Perceptron (MLP), XGBoost (XGB), Long-Short Term Memory (LSTM) & Convolution Neural Network (CNN) to detect these fraudulent transactions. It is based on data of the transactions details and wallet actors that are present in the Bitcoin network. It uses the transaction dataset & actor's dataset to train machine learning models based on different algorithms then compare their perfromance. This machine model and methodology can be expected to be useful for different cryptocurrencies such as ZCash etc.**

***Index Terms*—Transactions, Actors, Bitcoin, Fraud, Machine learning.**

I. INTRODUCTION

The increase in the amount of transactions happening in the Bitcoin network caused growth in the complexity and volume of data leading to various security issues. It became easier for the frauds to stay hidden and take action among the huge number of transactions taking place in the network. This leads to the victims not being able to detect the fraudulent transactions among the transactions in the network leading to huge losses. This project tries to resolve this problem using machine learning algorithms. By detecting fraudulent transactions, the model can contribute to enhancing the security of the Bitcoin network, protecting users from financial losses and preventing fraudulent activities such as theft and money laundering. Identifying fraudulent transactions in real-time can help prevent financial fraud and illegal activities associated with cryptocurrencies. This may help to preserve the integrity of the financial system and reducing the incidence of cybercrime. Machine learning models for detecting fraudulent transactions can assist regulatory authorities in enforcing adherence to know-your-customer (KYC) and countering money laundering (AML) laws throughout the bitcoin ecosystem. This can help combat illicit activities and ensure a safer and more transparent financial environment. Protecting consumers from fraudulent transactions is essential for maintaining trust and confidence in cryptocurrencies. Machine learning models can provide an additional layer of security and help users make informed decisions when conducting transactions in the Bitcoin network. Financial institutions and cryptocurrency exchanges can use machine learning models to assess the risk associated with transactions and identify potentially suspicious activities. This can help them implement appropriate risk management strategies and mitigate the impact of fraudulent transactions on their operations. Machine learning models for detecting fraudulent transactions can assist law enforcement agencies in investigating and prosecuting individuals involved in criminal works such as fraud, terrorist financing and money laundering. By providing actionable insights and evidence, these models can support law enforcement efforts to combat financial crime. Maintaining the integrity of the Bitcoin market is crucial for ensuring fair and efficient trading. Machine learning models can help identify fraudulent behaviors that could compromise market integrity, such as insider trading and market manipulation, thereby promoting a level playing field for all participants.

II. LITERATURE REVIEW

In this literature review research, it is found that the crime rate of cryptocurrency frauds is on an increasing curve [4]. Various ML algorithms such as Random Forest, KNN, Naïve Bayes, MLP, XGBoost etc., are very useful in detection of these frauds [1,2,3,5]. The research helps Identify that the wallet actor's details can also be used as a basis for the creation of detection model [5]. In this research, work is done which compares the various attributes in the model in terms of various used techniques and datasets [6]. Various trends of frauds taking place in crypto-transactions are studied [4]. Fraud detection using machine learning models in cryptocurrencies such as Ethereum is studied. As Ethereum is also a blockchain based cryptocurrency, it gives a basic overview of the working design of ML model for fraud detection [1,3]. Based upon the literature study, it is concluded that the use of machine learning for financial fraud detection in Bitcoin network is a very suitable measure for increasing transaction security.

III. METHODOLOGY

The model design for this machine learning project is constructed using Elliptic++ Transactions Dataset, a csv file format dataset consisting of nearly 203k bitcoin transactions and 183 feature labels and Actors dataset, a csv file format dataset consisting of nearly 822k wallet addresses and 56 feature labels. This dataset enables us to detect fraudulent transactions and classify them into 3 output labels – licit, illicit and unknown. The dataset used in our project is partitioned into training and testing data for model training. This partition is done based on the 'Timestep' feature present in the dataset. Timestep feature in this dataset is a measure of time period at which a transaction has taken place. Timestep are measured from 1-49 units in this dataset where each timestep (e. g. Timestep 1, Timestep 31) is 1 unit of time. In our project, data of 1-34 of the datasets is taken as training data and the units 35-49 are taken as testing data. The model is now trained using machine learning algorithms such as Logistic Regression (LR), Random Forest (RF), Multilayer Perceptron (MLP) and XGBoost (XGB), Long-Short Term Memory (LSTM) & CNN. The trained models are then tested and performance metrics such as accuracy, precision, f1-score, recall etc, are evaluated. The best model is taken as the final model and is used for prediction of classes for new transactions/actor input data. Then, we get the final result class for that new input i. e. one of classes licit, illicit and unknown. This is the design of our project construction.

A. System Architecture

A project workflow diagram is constructed based on out design methodology to understand various stages in which the project model is constructed. An illustration of the system architecture is depicted in figure 1.

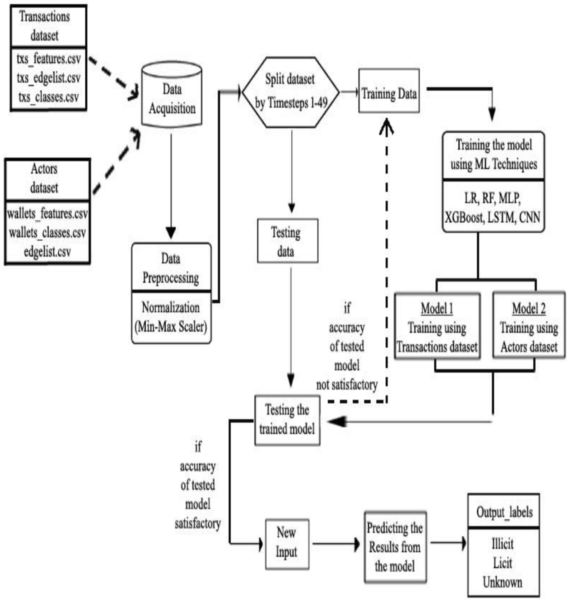
Transactions dataset
txs_features.csv
txs_edgelist.csv
txs_classes.csv
Actors dataset
wallets_features.csv
wallets_classes.csv
edgelist.csv
Data Acquisition
Data Preprocessing
Normalization (Min-Max Scaler)
Split dataset by Timesteps 1-49
Training Data
Testing data
Training the model using ML Techniques
LR, RF, MLP, XGBoost, LSTM, CNN
Model 1 Training using Transactions dataset
Model 2 Training using Actors dataset
if accuracy of tested model not satisfactory
Testing the trained model
if accuracy of tested model satisfactory
New Input
Predicting the Results from the model
Output_labels
Illicit
Licit
Unknown

Figure 1: System Architecture

B. Data Acquisition

Getting the data, you require to solve your machine learning challenge is the goal of this stage. Numerous sources, including databases, files, APIs, and manual collecting, may provide this data. Making sure the data is adequate, precise, and pertinent for training your model is crucial. In this paper, the data is acquired from github, the data is in csv file format with 203k bitcoin transactions data and 183 features and wallet actor's data with 822k actors with 56 features.

C. Data Pre-Processing

Once you have acquired the data, you need to preprocess it to make it suitable for training your model. This step involves tasks such as cleaning the data (removing missing or duplicate values), handling outliers, feature scaling, feature engineering (creating new features from existing ones), and encoding categorical variables.

MinMax Scaler is used in this project to preprocess the data. A well-liked method for preparing data for machine learning applications is MinMaxScaler, particularly when working with features that have different sizes. Each feature is scaled and translated separately such that it falls inside the training set's specified range, which is by default between zero and one.

For some machine learning algorithms, particularly those that are sensitive to feature scales (such as gradient descent-based techniques), this modification guarantees that every feature will have the same scale.

D. Split Dataset

It's standard procedure to divide the data into two or three subsets—a set for training, a set for validation, and a test set—after preprocessing. The test set is used to assess the performance of the finished model, the validation set is used to adjust hyperparameters and test several models during development, and the training set is used to train the model. Though this might vary according on the size of your dataset and other circumstances, the usual split ratio is 70-80\% for training, 10-15\% for validation, and 10-15\% for testing.

In this paper, the dataset is split based on 'timestep' feature present in the dataset. This feature is a unit of time at which the transaction has taken place. There are 1-49 timesteps in this dataset. For this project, 1-34 timesteps data is taken as training data and 35-49 timesteps are taken as testing data. In this way, the dataset is split into training and testing data for the training of the machine learning model in our task.

E. Training The Model

This step involves choosing a model architecture or machine learning technique and training it using the training data. In order to reduce the discrepancy between the actual and anticipated results, the model modifies its parameters after identifying patterns in the training data. During the training phase, the model is fed input data, the loss (error) is calculated, and the model's parameters are updated using optimization algorithms like gradient descent.

The model is trained for this study using the Random Forest, Multilayer Perceptron, XGBoost, and Logistic Regression techniques.

F. Testing The Model

You assess the model's performance on the test set after it has been trained. This entails feeding the trained model the test data and contrasting the predicted and actual results. Metrics like mean squared error (MSE) and R-squared are frequently used for regression tasks, whereas accuracy, precision, recall, F1 score, and ROC-AUC score are frequently used for classification tasks.

Precision, recall, f1-score, and micro average f1-score are the evaluation measures employed in this research.

G. Test Results

Finally, you analyze the results of your model evaluation to assess its efficacy and pinpoint areas in need of development. This may involve visualizing the model's predictions, inspecting misclassified or wrongly predicted instances, and comparing the model's performance with baseline models or other benchmarks.

In this research, we give a new transaction/wallet actor data input to classify the transaction/actor into licit, illicit and unknown transaction/actor based upon input data. The accurate classification of this new input shows the success of this machine learning model. The model used for this new input classification is the model with best evaluation metrics among the models trained using different machine learning algorithms used i.e., best model among models trained using LR, RF, MLP and XGB.

III .RESULTS ANALYSIS

Algorithms are used in the research for building the best model for fraud detection in bitcoin transactions are Logistic Regression (LR), Random Forest (RF), Multiple Layer Perceptron (MLP) and XGBoost (XGB).

The stepwise implementation process for training the model to get the results for this paper is given below.

Load the dataset.

- Remove any noise in the dataset by normalization & standardization
- Partition the data into training data (1-34 steps) and testing data (35-49 steps)
- Train the training data using the machine learning algorithms
- The following algorithms are used for the research:
- Logistic Regression
- Random Forest
- Multilayer Perceptron
- XGBoost
- Ensemble Models
- Test the model using testing data. If accuracy not good repeat training.
- If accuracy is good, use the model to classify new transaction/actor.
- Get the Results (Illicit/Licit/Unknown).

A. Transactions Dataset Observations

The following observations are obtained from the various analytics performed on the transaction's dataset and evaluation of the trained models for fraudulent transactions detection.

1. Count Of 3 Classes In The Dataset

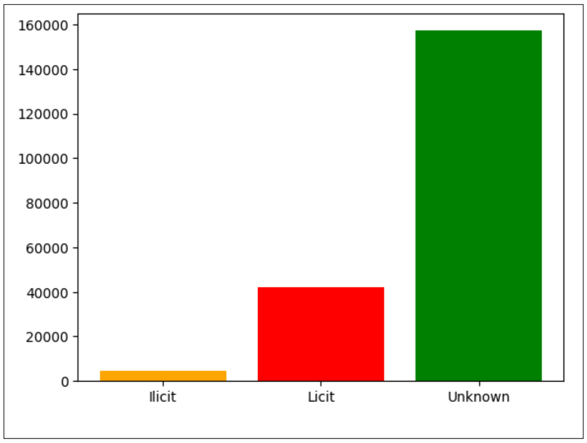

Figure 2: Count of 3 classes in dataset (Transactions Dataset)

Figure 2 shows the plot comparing count of different classes of transactions in the dataset. The plot shows that the count of unknown class transactions is very high compared to the other classes and the illicit class transactions are very low. Hence, we can say that the dataset has uneven observations.

2. Count Comparison Between The Class Labels In Timesteps(1-34)

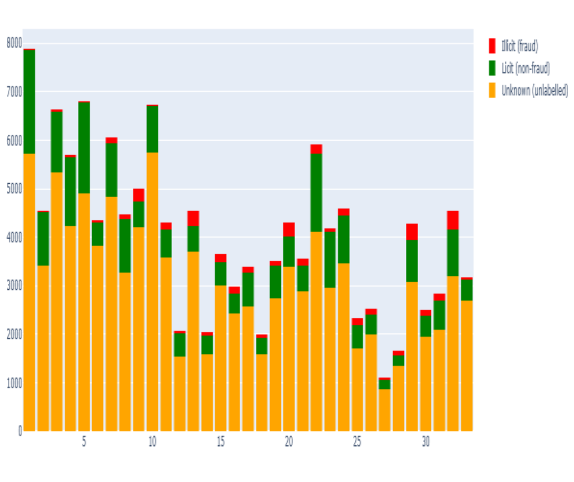

Figure 3: Transactions of 3 classes from timestep 1-34 (Training Data)

The plot in figure 3 shows the 3 classes of transactions present in the training dataset i.e. 1-34 timesteps. From the graph, it is observed that in timesteps 8,12,19,28,31, a large number of illicit transactions are detected and confirmed. Also, it is observed that not a single timestep is free of illicit transactions in the dataset. This shows how critical rate at which the frauds are occurring in the bitcoin network.

3. Top 20 Features And Bottom 20 Features

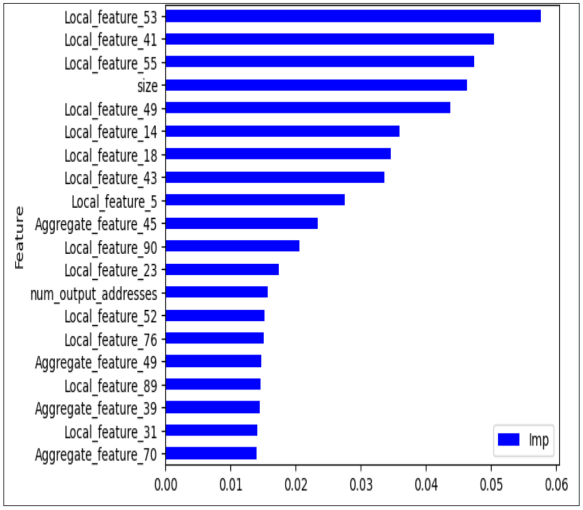

Figure 4: Top 20 features (Transactions Dataset)

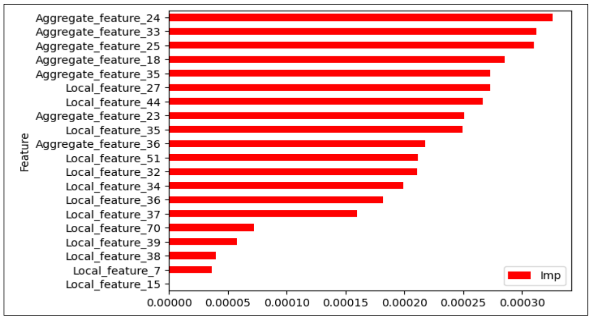

Figure 5: Bottom 20 features (Transactions Dataset)

The figures 4 and 5 show the 20 features with highest feature importance score and 20 feature with lowest feature importance score. This score is a feature analysis which shows the extent of effect of a feature on the resultant output label/transaction class in the transaction dataset.

4. Correlation of Features Classification

The above project used for building a machine learning model for detection of fraudulent transactions an also be used to evaluation of various metrics related to Bitcoin transaction dataset. This analysis helps us in understanding the relationships between the various features in the dataset and how they affect each other. For example, we can create a plot showing the good/bad correlation for classification based on a feature.

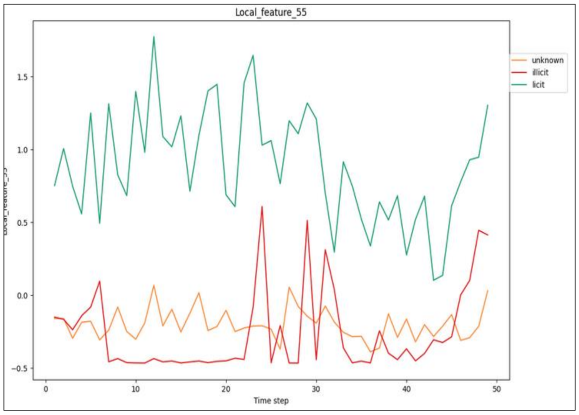

Figure 6: Local Feature 53 showing good correlation for classification

The figure 6 shows local feature 53 with its corresponding line plot, showing the correlation with classification and using different colors for each class. In this graph, the classification line of illicit and licit classes is distinct and does not intersect with each other which shows that the local feature 53 is a good feature for classification of classes.

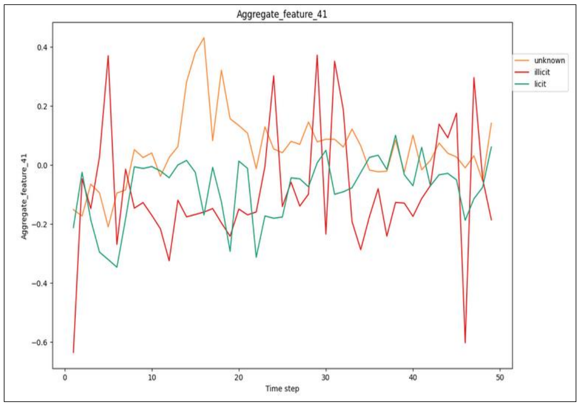

Figure 7: Aggregate Feature 41 showing bad correlation for classification

The figure 7 shows aggregate feature 41 with its corresponding line plot, showing the correlation with classification and using different colors for each class. In this graph, the classification line of illicit and licit classes is frequently intersecting with each other making it hard for the classification to be done accurately. This shows that the aggregate feature 41 is a bad feature for classification of classes.

B. Actors Dataset Observations

The following observations are obtained from the various analytics performed on the actor's dataset and evaluation of the trained models for fraudulent transactions detection.

1. Count Of 3 Classes in Dataset

The figure 8 shows the plot comparing count of different classes of actors in the dataset. The plot shows that the count of unknown class wallet actors is very high compared to the other classes and the illicit class t are very low. Hence, we can say that the dataset has uneven observations.

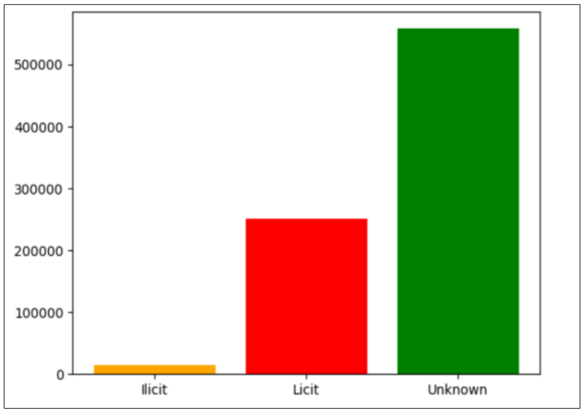

Figure 8: Count of 3 classes in Actors dataset

2. Count Comparison Between the Class Labels in Timesteps

The plot in figure 9 shows the 3 classes of actors present in the actor's dataset. From the graph, it is observed that in timesteps 1,10,36,42 a large number of actors are detected and confirmed. Also, it is observed that not a single timestep is free of illicit transactions in the dataset. This shows how critical rate at which the frauds are occurring in the bitcoin network.

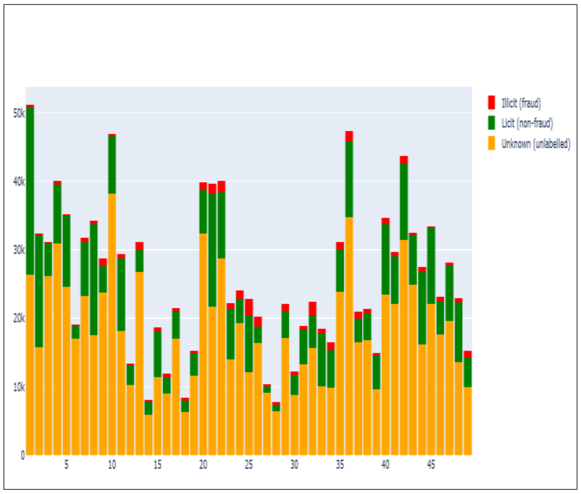

Figure 9: Actors of 3 classes from timestep in each timestep

3. Top 20 Features & Bottom 20 Features

The figures 10 and 11 show the 20 features with highest feature importance score and 20 feature with lowest feature importance score. This score is a feature analysis which shows the extent of effect of a feature on the resultant output label class in the actor's dataset.

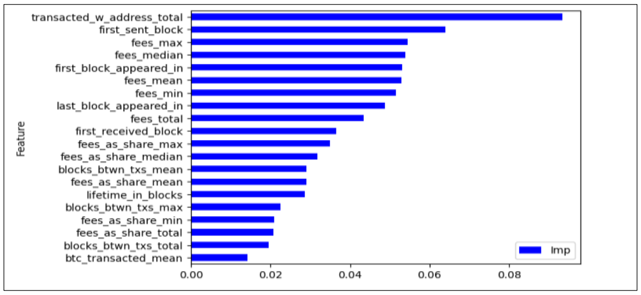

Figure 10: Top 20 features affecting output label (actors' dataset)

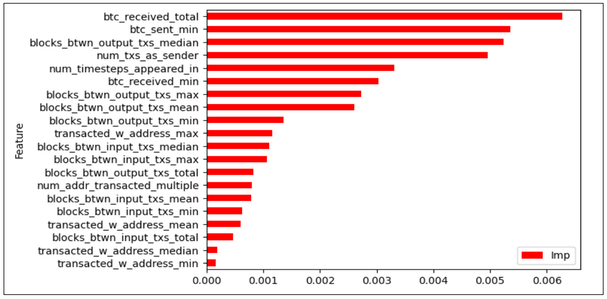

Figure 11: Bottom 20 features affecting output label (actors' dataset)

4. Correlation of Features Classification

The above project used for building a machine learning model for detection of fraudulent actors an also be used to evaluation of various metrics related to Bitcoin actor's dataset. This analysis helps us in understanding the relationships between the various features in the dataset and how they affect each other. For example, we can create a plot showing the good/bad correlation for classification based on a feature.

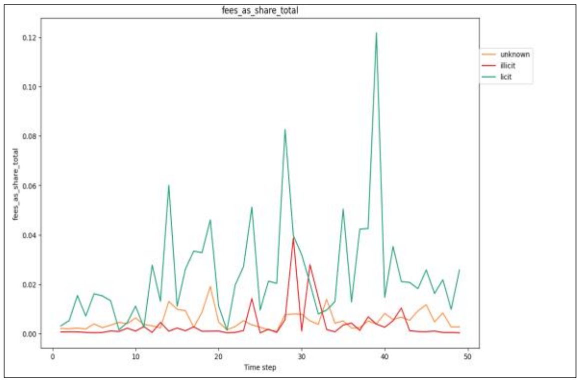

Figure 12: fees_as_share_total feature showing good correlation for classification

The figure 12 shows fees_as_share_total feature with its corresponding line plot, showing the correlation with classification and using different colors for each class. In this graph, the classification line of illicit and licit classes is distinct and does not intersect with each other which shows that the fees_as_share_total feature is a good feature for classification of classes.

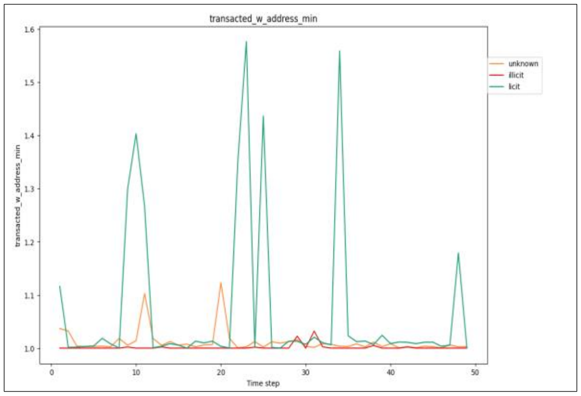

Figure 13: transacted_w_address_min feature showing bad correlation for classification

The figure 13 shows transacted_w_address_min feature with its corresponding line plot, showing the correlation with classification and using different colors for each class. In this graph, the classification line of illicit and licit classes are frequently intersecting with each other making it hard for the classification to be done accurately. This shows that the transacted_w_address_min feature is a bad feature for classification of classes.

C. Results Of Transactions Model And Actors Model

1. Fraud Transaction Detection Model

Table-1: Evaluation Metrics of all algorithms (transactions model)

	Precision	Recall	F1-Score	Micro-Average F1-Score	Accura-cy
LR	0.327	0.707	0.448	0.884	0.884
RF	0.958	0.725	0.825	0.98	0.98
MLP	0.607	0.618	0.612	0.948	0.948
XGBOOST	0.906	0.733	0.811	0.977	0.977
XGB + RF	0.977	0.721	0.83	0.98	0.98
MLP + XGB	0.979	0.645	0.777	0.976	0.976
RF + MLP	0.991	0.625	0.767	0.975	0.975
XGB+ RF + MLP	0.963	0.73	0.831	0.98	0.98
LSTM	0.296	0.631	0.403	0.876	0.876
CNN	0.603	0.669	0.634	0.949	0.949

The table 1 presents all the evaluation metrics of algorithms based on the metrics precision, recall, f1-score, micro average f1-score and accuracy. From the table, it is observed that the model trained using shows the best performance among all the models trained for fraud detection in bitcoin transactions.

Figure 14 shows confusion matrix for Random Forest model. From the observation of evaluation metrics, it can be observed that Random forest algorithms built works best for transactions dataset model in detecting fraudulent bitcoin transactions.

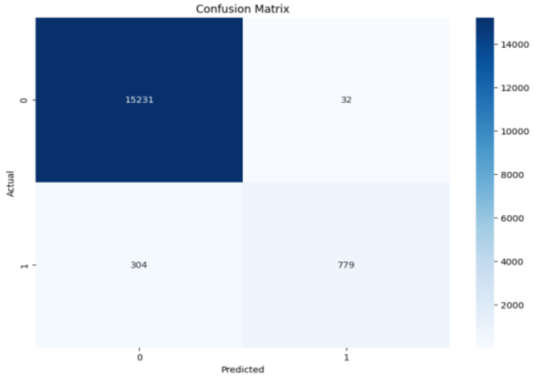

Figure 14: Confusion Matrix (RF – Transactions model)

2. Fraud Actors Detection Model

Table 4.2: Evaluation Metrics of all algorithms (actors model)

	Precision	Recall	F1-Score	Micro-Average F1-Score	Accura-cy
LR	0.478	0.046	0.083	0.964	0.964
RF	0.913	0.789	0.846	0.99	0.99
MLP	0.682	0.519	0.589	0.974	0.974
XGBOOST	0.893	0.808	0.848	0.989	0.989
XGB + RF	0.938	0.766	0.843	0.99	0.99
MLP + XGB	0.982	0.407	0.575	0.978	0.978
RF + MLP	0.965	0.413	0.578	0.978	0.978
XGB+ RF + MLP	0.921	0.77	0.839	0.989	0.989
LSTM	0.694	0.525	0.598	0.974	0.974
CNN	0.728	0.678	0.702	0.979	0.979

The table 2 presents all the evaluation metrics of algorithms based on the metrics precision, recall, f1-score, micro average f1-score and accuracy. From the table, it is observed that the model trained using Random forest shows the best performance among all the models trained for fraud detection in bitcoin actors.

Figure 15 shows confusion matrix for Random Forest model. From the observation of evaluation metrics, it can be observed that Random forest algorithms built works best for Actors dataset model in detecting fraudulent bitcoin transaction actors.

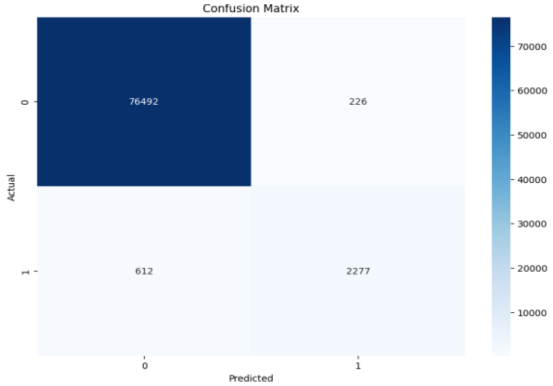

Figure 15: Confusion Matrix (RF – Actors Model)

3. Overall Observations

From the model training for fraud detection, the observations are made –

- The dataset is uneven with a too many unknown labels and few illicit labels.
- The logistic regression (LR) algorithm is not suitable for the dataset in this project as a large number of features decreased the model performance.
- The Random forest algorithm model worked best among all the models trained without applying any boosting methods.
- After boosting, the models trained Random forest & MLP algorithms works best for this project.

IV. CONCLUSION & FUTURE WORK

From the above observations, it can be concluded that the models trained using Random Forest algorithm work best for this research. This model can also be used for other cryptocurrencies such as Zcash, Ethereum etc. The dataset used for the project is found to have a huge number of unknown observations which is expected to have affected the prediction of other labels such as illicit, licit decreasing model credibility. In the future, the dataset can be improved by getting uniform no. of labels for the outcome labels. A research using this project model to detect fraudulent transactions in other crypto currency networks and compare the situation of different crypto currency networks can be done.

REFERNCES

[1] Aziz, Rabia Musheer, Rajul Mahto, Kartik Goel, Aryan Das, Pavan Kumar, and Akash Saxena. "Modified genetic algorithm with deep learning for fraud transactions of ethereum smart contract." Applied Sciences 13, no. 2 (2023): 697.

[2] Dutta, Anurag, Liton Chandra Voumik, Athilingam Ramamoorthy, Samrat Ray, and Asif Raihan. "Predicting Cryptocurrency Fraud Using ChaosNet: The Ethereum Manifestation." Journal of Risk and Financial Management 16, no. 4 (2023): 216.

[3] Aziz, Rabia Musheer, Mohammed Farhan Baluch, Sarthak Patel, and Pavan Kumar. "A machine learning based approach to detect the Ethereum fraud transactions with limited attributes." Karbala International Journal of Modern Science 8, no. 2 (2022): 139-151.

[4] Kutera, Małgorzata. "Cryptocurrencies as a subject of financial fraud." Journal of Entrepreneurship, Management and Innovation 18, no. 4 (2022): 45-77.

[5] Mittal, Ruchi, and Mahinder Pal Singh Bhatia. "Detection of suspicious or un-trusted users in crypto-currency financial trading applications." International Journal of Digital Crime and Forensics (IJDCF) 13, no. 1 (2021): 79-93.

[6] Sabry, Farida, Wadha Labda, Aiman Erbad, and Qutaibah Malluhi. "Cryptocurrencies and artificial intelligence: Challenges and opportunities." IEEE Access 8 (2020): 175840-175858.

[7] Elmougy, Youssef, and Ling Liu. "Demystifying Fraudulent Transactions and Illicit Nodes in the Bitcoin Network for Financial Forensics." arXiv preprint arXiv:2306.06108 (2023).

[8] Das, Abhik. "Logistic regression." In Encyclopedia of Quality of Life and Well-Being Research, pp. 1-2. Cham: Springer International Publishing, 2021.