# Fraud Detection over Ethereum Blockchain

## CST4340 – COURSEWORK 2

Huma Zafar M00899071

# Contents

## 1. Introduction

As early as 2013, Vitalik Buterin introduced Ethereum with the possibility of its widespread use (Antonopoulos, 2018). There are several applications based on the Ethereum protocol, including ERC-20 tokens, which are Ethereum-based tokens that can be created and deployed on the Ethereum network. Over 400 million transactions have been made on Ethereum since its inception. There have been a number of illegal activities related to Ethereum, including smart-Ponzi schemes, phishing, money laundering, and fraud. Detecting and predicting such attacks over blockchain can be achieved through anomaly detection for blockchain.

The paper makes a number of contributions; first of all, it proposes a Random Forest Classifier, which is an effective method for detecting illicit accounts on the Ethereum network based on the testing of 8 models of four distinct types (Decision Tree, Random Forest, Gradient Boosting, and Extreme Gradient Boosting); secondly, it gives a multiple linear regression model for estimating total Ethereum transfers; and thirdly, it provides coherent and graphical representations of historical data. The software tool, KNIME is used to execute the statistical tasks. This tool uses visual nodes to do descriptive, predictive and prescriptive analytics (Berthold et al., 2009).

## 2. Research Objectives

This paper is designed to identify fraudulent transactions on the Ethereum Blockchain using the most effective classification model possible. It seeks to find illegitimate transactions using the best-fit model selected from four classification models: decision trees, random forests, gradient boost and extreme gradient boost models. Additionally, it investigates the possibility of calculating the total amount of Ether that a wallet has received and transferred. To determine which models are well-fitted, it performs statistical tests for model fitness. A visual display of token transaction history is another goal. Ultimately, it aims to assess the internal validity, external validity, and reliability of the finalised models.

## 3. Methodology

### 3.1 Dataset

Ethereum dataset was downloaded from Kaggle, a platform for data science. Wallet addresses used in transactions are classified as fraud by the Ethereum community for illicit behavior, such as imitating other contract addresses, operating fake lotteries, conducting fake initial coin offerings (ICOs), impersonating other users, operating Ponzi schemes, phishing, and mirroring websites. In this case, it is assumed that the dataset consists of supervised information and that no time-dependent effects exist. An overview of the dataset is provided in Table 1.

Table 1. Normal transaction fields of interest.

| Field of Interest | Description | Data Type |
|---|---|---|
| Flag | Determines whether account is flagged for fraudulent activity | Boolean<br>(1 = Fraud, 0= Fair) |
| Address | Wallet Address on Ethereum Blockchain | String |
| ERC20 most sent token type | Name of ERC20 token used for transfer of fund | String |
| ERC20_most_rec_token_type | Name of ERC20 token used for transfer of fund | String |
| Sent tnx | Number of Transaction taken placed for the transfer of fund | Integer |
| Received Tnx | Number of Transaction taken placed to receive the fund | Integer |
| total Ether sent | Total Amount Transferred (ether) | Float |
| total ether received | Total Amount Received (ether) | Float |

### 3.2 Data Pre-Processing

From the original dataset of 12,147 rows, 2,022 rows were randomly selected. The columns and rows of the data are manually selected as part of the data reduction process. There are 34 columns in the file, out of which 8 have been extracted. As shown in Table 1, the selected column and row type are described. Wallet addresses on the Ethereum Blockchain are generated as unique addresses (Antonopoulos, 2018), so duplicate addresses have been removed. A subsequent deletion was made of the columns containing 'none' and blank rows. The final row count was 1075 after filtering out non-unique accounts through the data cleansing process. Appendix A, Figure 2, illustrates the pre-processing technique workflow on KNIME.

The data mining process was applied to two different sets of data from the same dataset. In the first set, 272 unique tokens were transferred and 340 unique tokens were received in 1075 wallet addresses.  This dataset is called dataset A. Subsequently, on KNIME (see Figure 11, 12, 13 and 14 on Appendix A), the second analysis was conducted after the data transformation had been completed, while categorizing the aggregate 612 tokens into 8 categories: EOS, OmiseGO, Ether, Golem, TheDAO, DATAcoin, DMTS, and others; this one is named as dataset B. With 612 tokens, graphical explanations of data became difficult. Hence, the tokens are reclassified into 8 categories based on overall transaction volume, and Ethereum is included as well since it forms the first layer of all ERC20 tokens.

### 3.3 Data Visualisation

It is evident from the data sample selected that fraudulent activities are substantially more prevalent than fair transactions, as shown in Figure 1 and Figure 2. This indicates sampling bias in the extraction of data from the Ethereum Blockchain's general population. This section aims to provide an overview of the past trends in token transactions in terms of fair and fraudulent transactions. For this descriptive analysis, the histograms, donut pie charts, and linear correlations are used to visualize the data. Histogram is chosen because they provide a clear retrospective analysis of ERC20 token transactions. Pie charts illustrate the overall classification of flagged and fair transactions. These charts facilitate the identification of relationships and findings that are not readily apparent in raw data. All of these graphical representations are aimed at comprehending current data and how it can be used to generate new models. All the donut pies are in two colours: blue and orange. Blue is for fair transaction. Contrarily, orange is for fraud transaction.

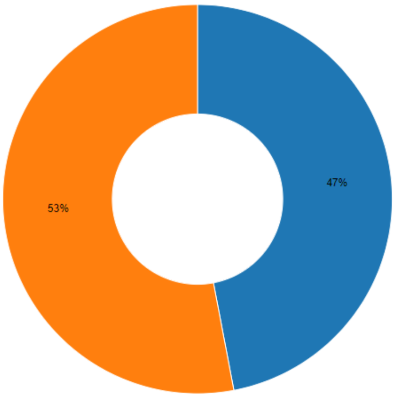


*Figure 1: Donut pie chart for Overall total ether transferred*

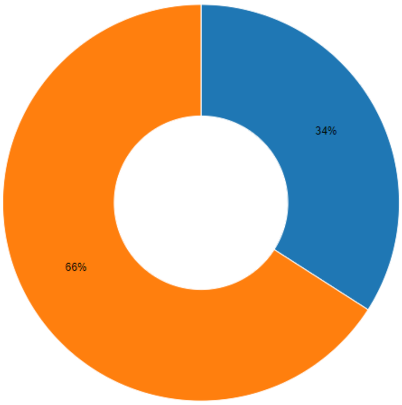


*Figure 2: Donut pie chart for Overall total ether received*

All the histograms are in two colours: blue and orange. Blue is representing total amount of ether transferred whereas orange is indicating total ether received. The histogram is having two bars on left hand side – embodies ‘fair’ transaction whilst the two bars on the right-hand side embodies ‘fraud’.

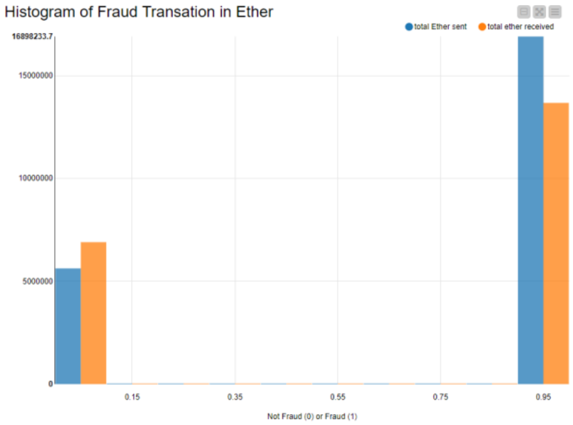


*Figure 3: Overall Flag analysis on Ethereum.*

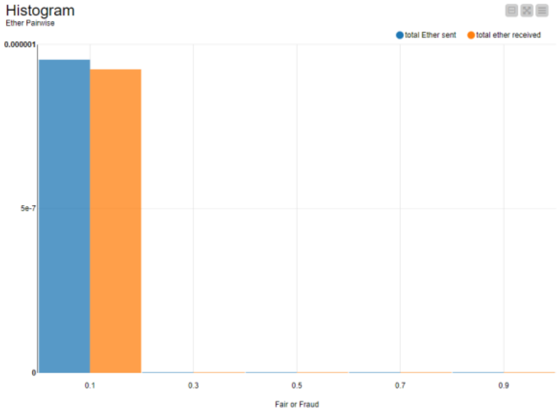


*Figure 4: Overall Ethereum Pair-wise Transaction Analysis*

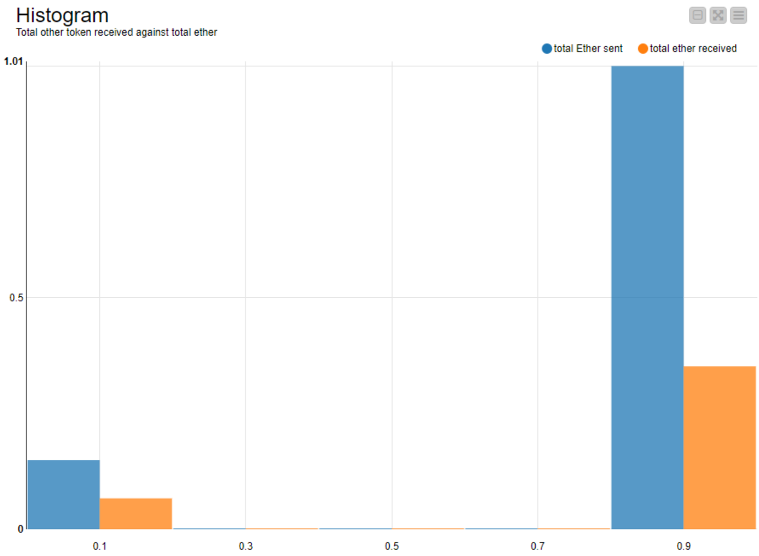


*Figure 5: Comparison of Transaction Analysis of Wallet involving transaction between Ethereum and other tokens.*

Based on the analysis of Figures 3, 4 and 5, it can be concluded that Ether as a token has been used by wallets that only use Ether as a token for fair purposes. In contrast, if Ether is paired up with other ERC-20 tokens, the addresses have been used for both illicit and legal transactions.

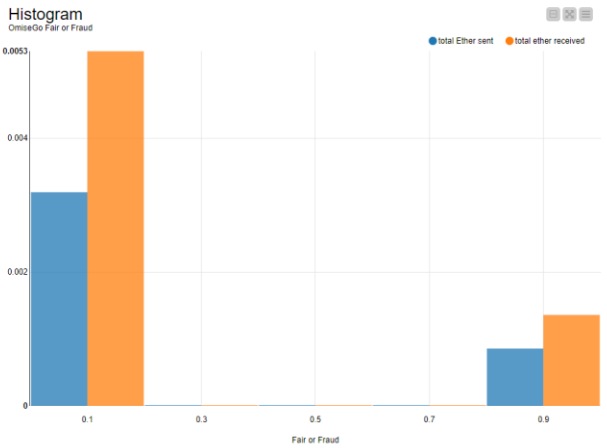


*Figure 6: Overall OmiseGo Pair-wise Transaction Analysis*

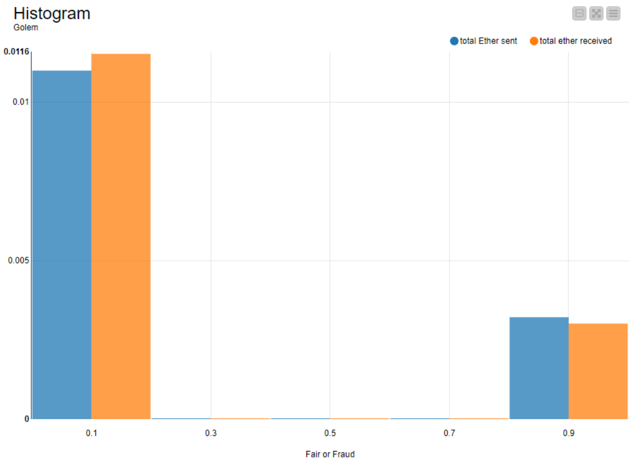


*Figure 7: Overall Golem Pair-wise Transaction Analysis*

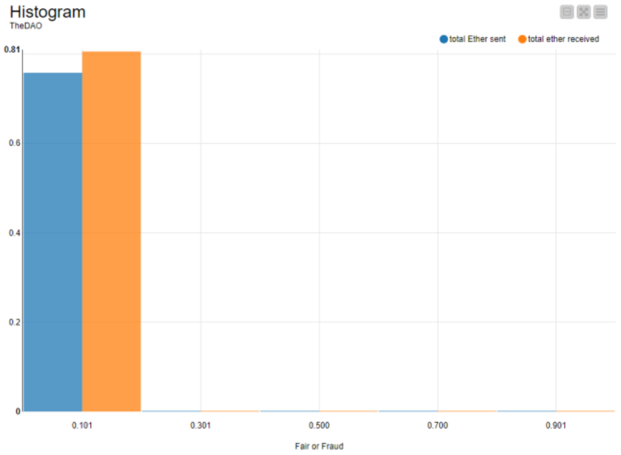


*Figure 8: Overall TheDAO Pair-wise Transaction Analysis*

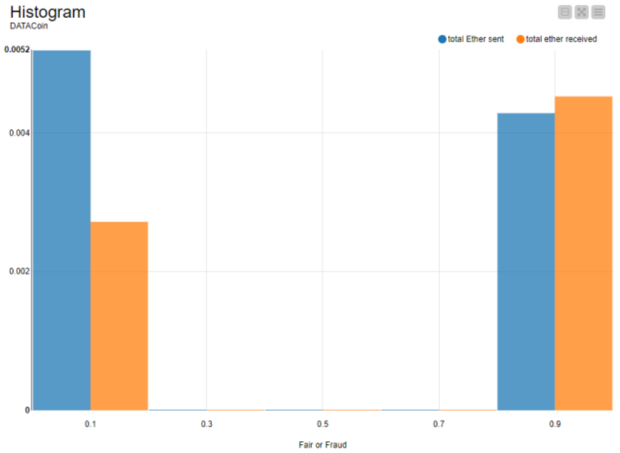


*Figure 9: Overall DATACoin Pair-wise Transaction Analysis*

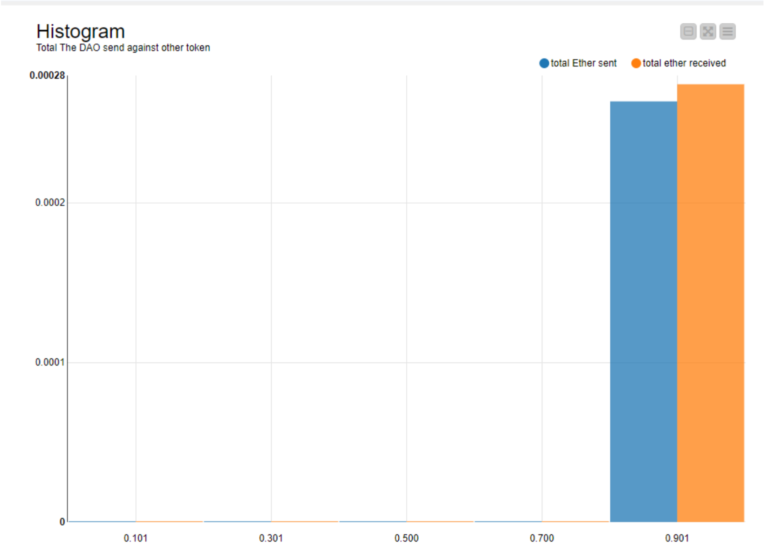
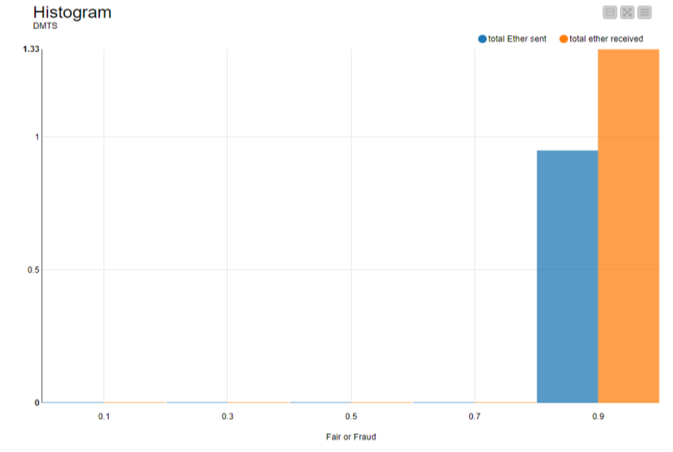


*Figure 10: Comparison of Transaction Analysis of Wallet involving transaction between TheDAO and other tokens.*



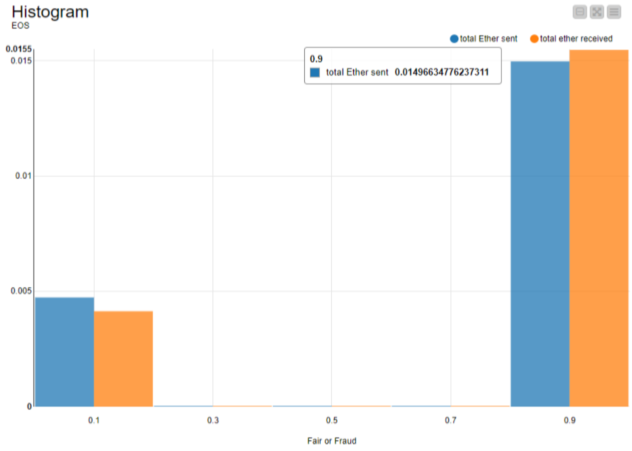


*Figure 11: Overall DMTS Pair-wise Transaction Analysis*

*Figure 12: Overall EOS Pair-wise Transaction Analysis*

As a result of the analysis of Figures 6 and 7, the tokens OmiseGo and Golem were relatively more frequently used in legal transactions than in illegal transactions. However, tokens such as DATAcoin, DMTS, and EOS are suspicious because they have historically been used in fraud transactions as demonstrated in Figures 9, 11, and 12. In Figure 8, it can be seen that pair-wise transactions between TheDAO were conducted legitimately; nevertheless, if there were transactions between TheDAO and others, every single transaction was flagged as fraudulent.

Four pie charts illustrate a graphical analysis of transaction activity among the eight categorised tokens based on transaction dataset B. The first two charts reflect the transactions conducted by receivers and the latter two illustrate the activities of senders. Figure 14 indicates that DMTS is the second most popular token among receivers (31%). Further, Figure 15 indicates that DMTS also dominates as the second-highest volume of transactions among receivers (21%). In contrast, Ether represents the second most popular token among senders. According to another pie chart, 79% of the total number of transactions involved sending other tokens.

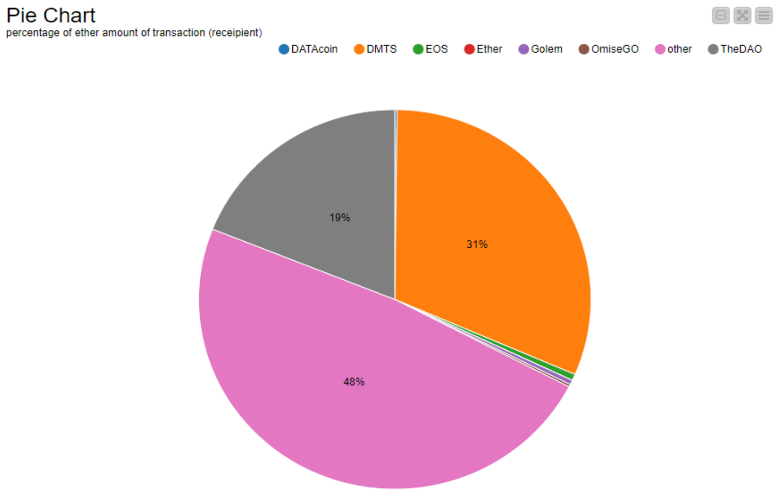


*Figure 13*

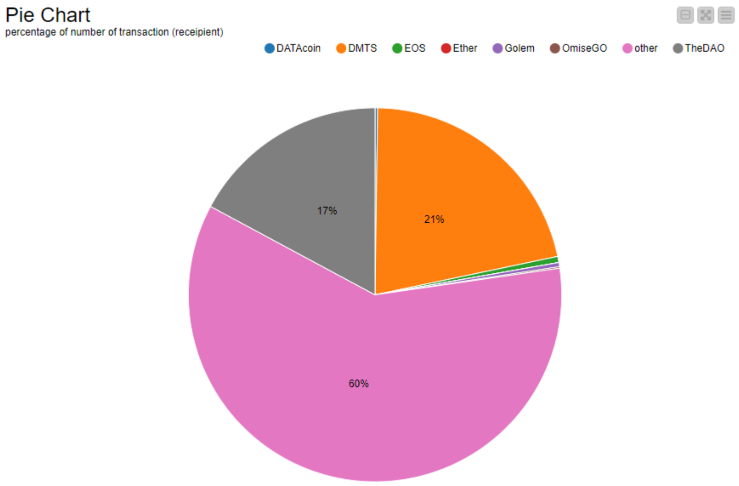


*Figure 14*

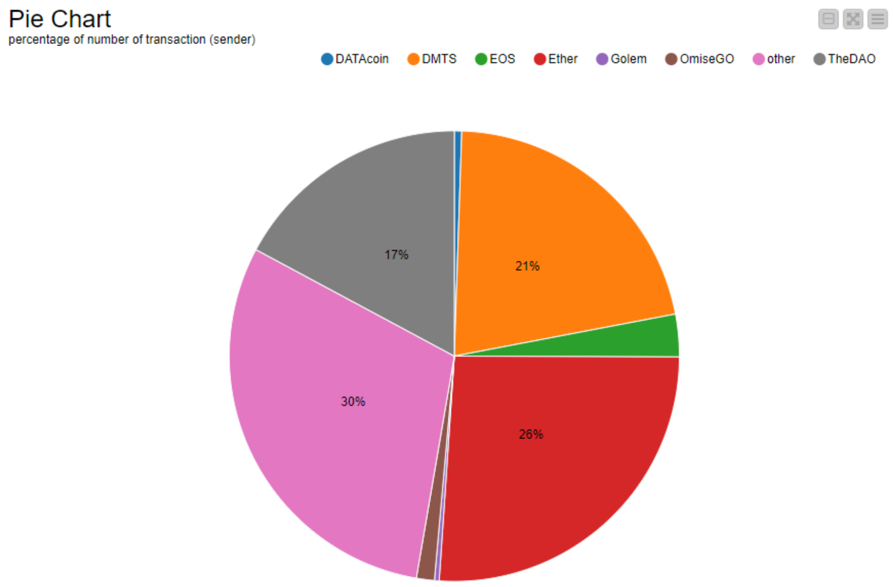


*Figure 15*

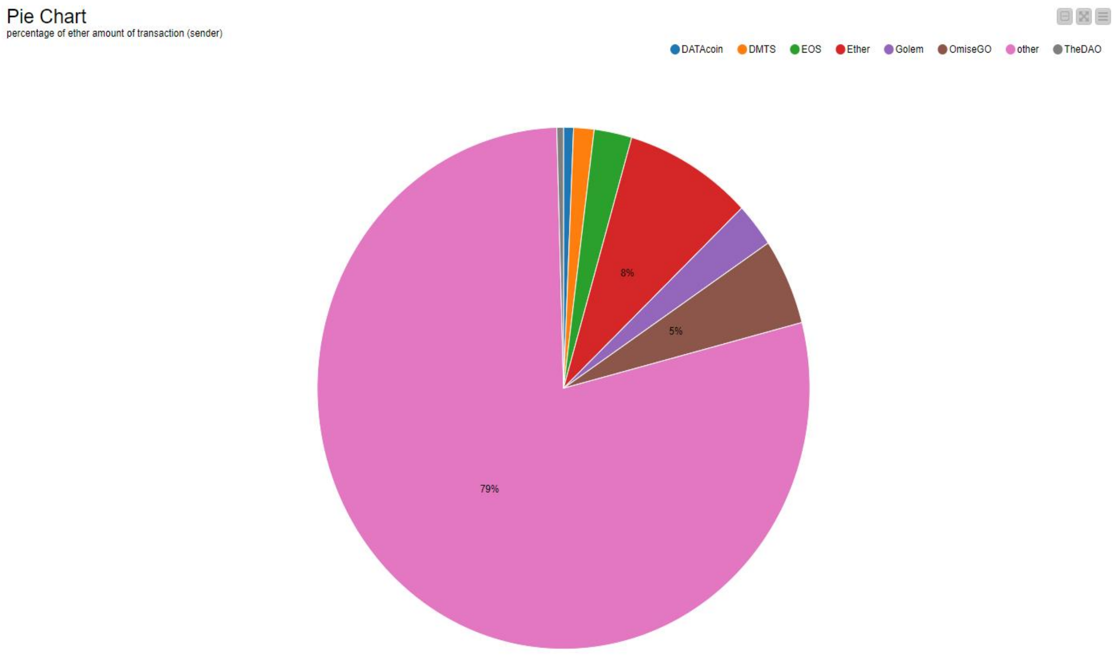


*Figure 16*

## 3.4 Data Mining Tasks

Under Data Mining mission, both descriptive and predictive analytical tasks are executed. For descriptive, clustering analysis is done using two different approaches: Principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE). Whilst, for predictive analysis, regression and classification methodologies are used.

### 3.4.1 Clustering Analysis

The clustering analysis is executed to identify pattern and outliers among the data (Barnett et al., 1978). It seeks to detect hidden patterns or groupings in a dataset using clustering algorithms. An anomaly detection (also called an outlier detection or novelty detection) refers to identifying rare items, events or observations that differ significantly from the majority of the data and do not conform to a well-defined concept of normal behavior (Lewis, 1978). In this section, two clustering analysis are done - PCA and t-SNE.

PCA is utilized to reduce the dimensionality of large data sets. By transforming a large collection of variables into a smaller collection, a significant amount of information from the large collection is retained. Additionally, PCA allows for easy exploration and visualization of data, as well as highlighting variations and patterns in a dataset as witnessed below in Figure 17.

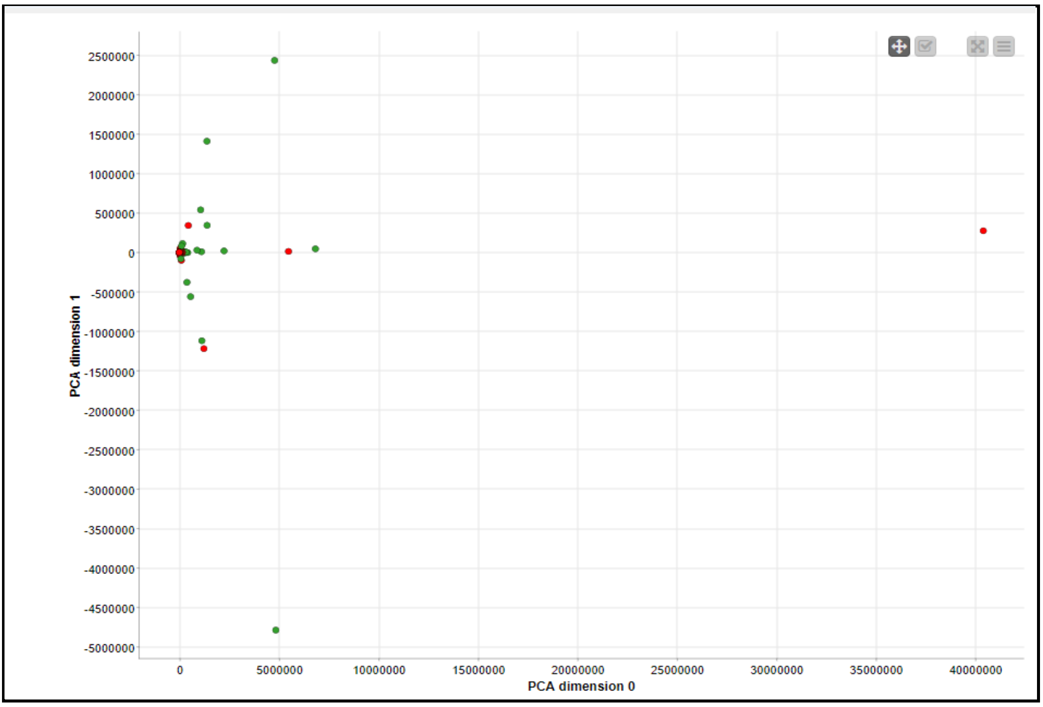


*Figure 17 PCA with fair outlier on the negative PCA dimension 1 while fraud outlier on the positive side.*

As a method of visualizing high-dimensional data, t-SNE assigns each data point a location on a two- or three-dimensional map. With stochastic neighbor embedding, a probability distribution is constructed for pairs of high-dimensional objects. Figure 18 displays five clusters : four mixture of fraud and fair transactions while one fraud transaction only. Conclusively, there is still a substantial overlap of different class data points present throughout.

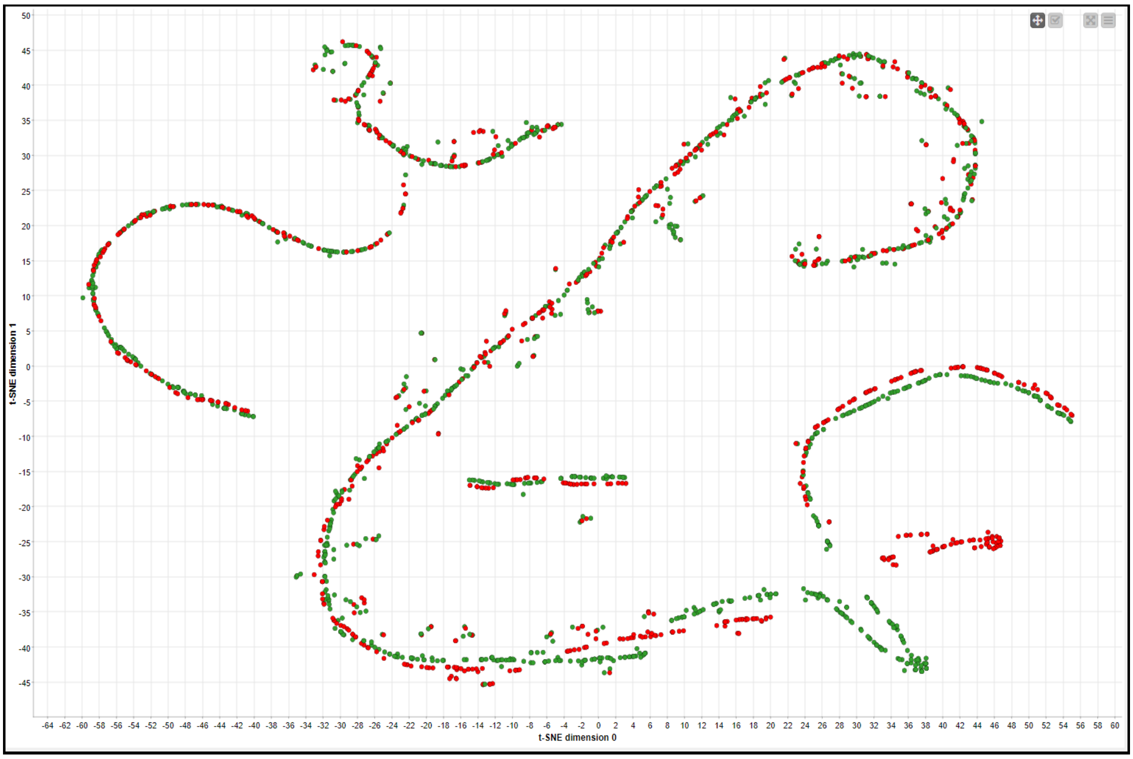


*Figure 18 depicts five clusters.*

It can be observed that there are arbitrary shaped  patterns in PCA and t-SNE. However, as studied by Hillemann et al. (2015), there is the high plausibility of cognitive bias in the form of clustering illusion as such patterns may not exist in real time data. Hence, to further find out the patterns, linear correlation is executed on KNIME.

### 3.4.2 Linear Correlation

Any statistical relationship between two random variables or bivariate data is referred to as correlation or dependency, and it can reveal a predictive relationship that can be used practically. Correlation does not necessarily indicate cause, though. The linear correlation in Figure 19 reflects that there are the positive relationships among the variables: Sent tnx, Received Tnx, total Ether sent and total ether received. This implies that their linear relationship can be formulated among these variables. These will be split into two distinct models: Total Amount of Ether received and Total Amount of Ether transferred.

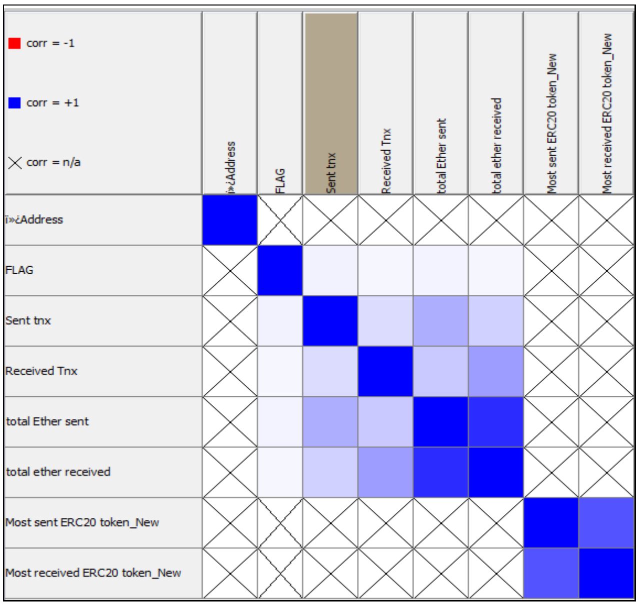
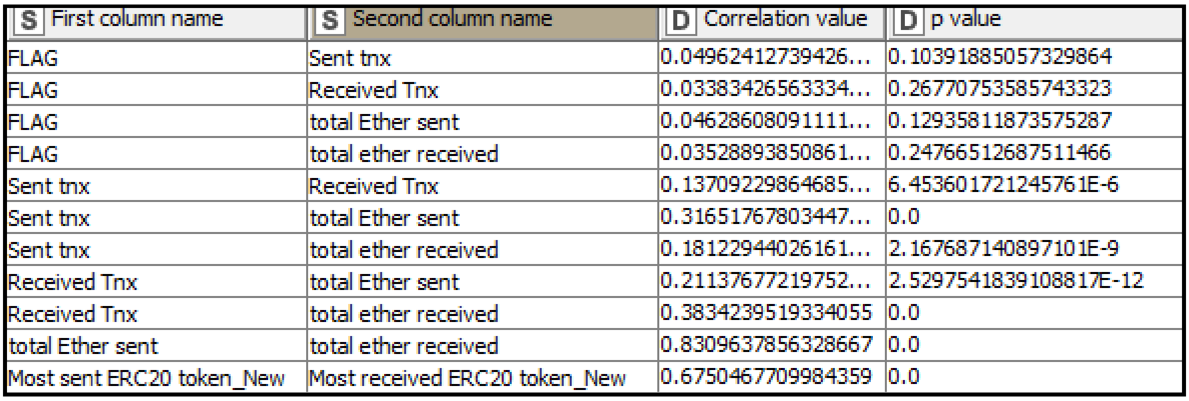


*Figure 19 represents the relationship among variables. The darker blue color indicates the stronger is the relationship.*



| S First column name | S Second column name | D Correlation value | D p value |
|---|---|---|---|
| FLAG | Sent tnx | 0.04962412739426... | 0.10391885057329864 |
| FLAG | Received Tnx | 0.03383426563334... | 0.26770753585743323 |
| FLAG | total Ether sent | 0.04628608091111... | 0.12935811873575287 |
| FLAG | total ether received | 0.03528893850861... | 0.24766512687511466 |
| Sent tnx | Received Tnx | 0.13709229864685... | 6.453601721245761E-6 |
| Sent tnx | total Ether sent | 0.31651767803447... | 0.0 |
| Sent tnx | total ether received | 0.18122944026161... | 2.167687140897101E-9 |
| Received Tnx | total Ether sent | 0.21137677219752... | 2.5297541839108817E-12 |
| Received Tnx | total ether received | 0.3834239519334055 | 0.0 |
| total Ether sent | total ether received | 0.8309637856328667 | 0.0 |
| Most sent ERC20 token_New | Most received ERC20 token_New | 0.6750467709984359 | 0.0 |

*Figure 20 is the table extracted from KNIME.*

This table in Figure 20 outlines that the relationships between the following variables are positive, moderate and statistically significant (because p-value is less than 0.05):

- Sent tnx and Received Tnx
- Sent tnx and total Ether sent
- Sent Tnx and total ether received
- Received Tnx and Total Ether sent
- Received Tnx and total ether received
- total Ether sent and total ether received

- Most sent ERC20 token_New and Most received ERC20 token_New

Below two diagrams (Figure 21 and Figure 22) are dispersion analysis in the form of the box plots extracted from KNIME for the variables containing continuous numeric data.

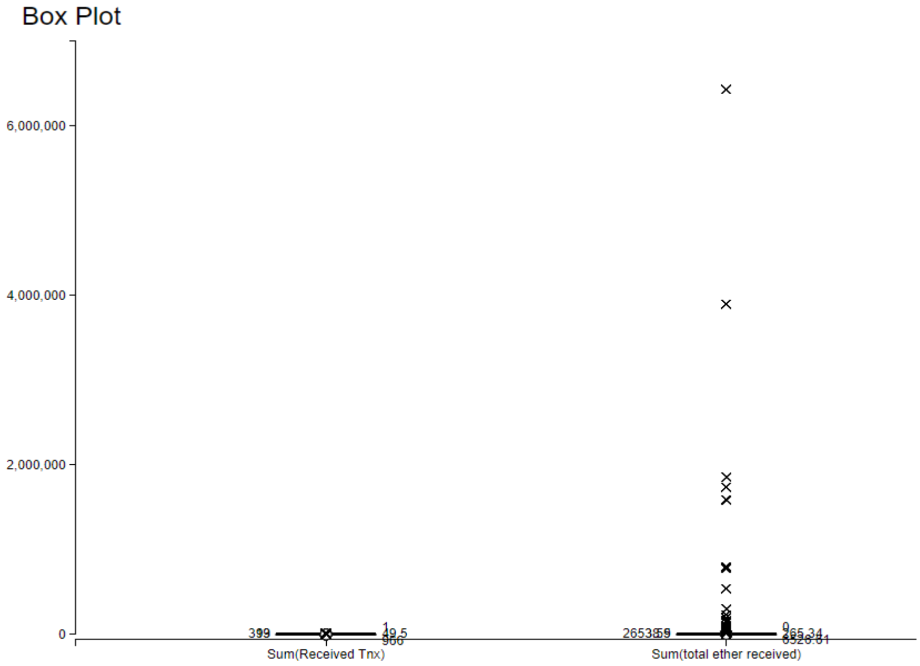


*Figure 21 shows that the boxplots for both variables- 'Received Tnx' and ' total ether received'. In the data of variable, total ether received, there are significant number of outliers.*

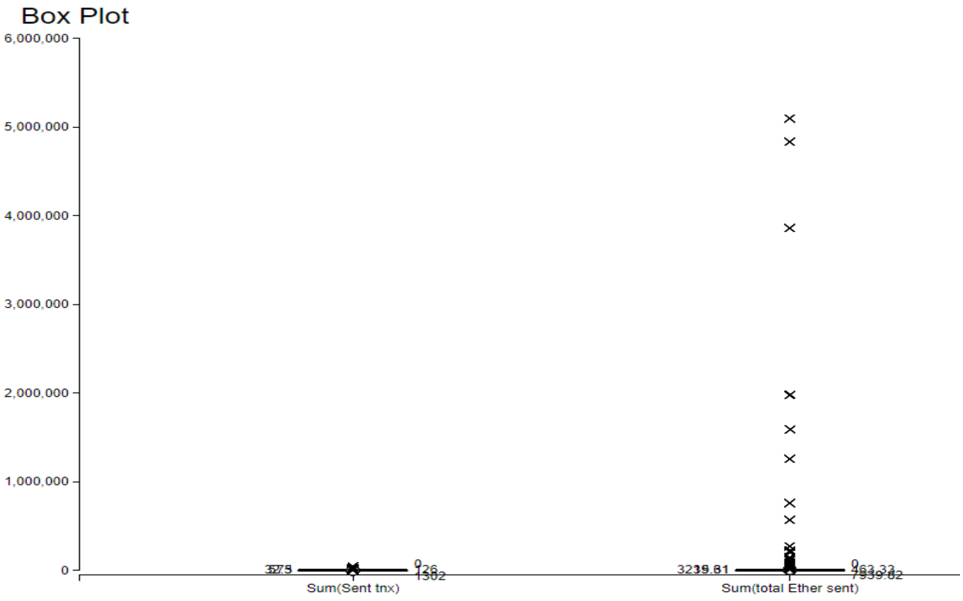


*Figure 22 shows that the boxplots for both variables - 'Sent Tnx' and 'total ether sent'. In the data of variable, total ether sent, there are significant number of outliers.*

The results of numerous data description studies indicate that the Ethereum dataset may be used to develop predictive models. However, in light of the fact that there are outliers, the models formulated should be tested for their fitness. The next section will provide an overview of ten models, along with a fitness test for each of them.

### 3.4.2 Multiple Linear Regression Modelling

For predictive analytical tasks, linear regression and classification methodologies are applied to project the future events after training the past data. To generate the multiple linear regression among variables, the heteroscedasticity of the residuals, and the normality of the residuals, KNIME (Appendix A, figure 8) is used. The following Table 2 represents the variables assigned to the Field of Interest defined as in table 1.

| Field of Interest | Variable |
|---|---|
| FLAG | F |
| Sent tnx | A |
| Received Tnx | B |
| total ether received | C |
| total Ether sent | D |

*Table 2: Variable Name for Field of Interest*

Multilinear Regression Model for 'Total Amount of Ether Transferred' = D :

The new simple fit model for D is as follows:

$D = 0.001F + 0.093A – 0.057B + 0.963C – 0.001$

Although Figure 23 depicts there is a residual have weak heteroskedastic, D model's R-squared value is 0.7 and normal distribution of standard error (Figure 22) further justify that the following equation can be run to predict the total amount of ether transferred.  So, as per D's equation , for every flagged transaction, there is an average increase of 0.001 ether in the average total amount of total ether transferred. Likewise, for every one unit of token transfer, there is an average increase of 0.0093 ether in the total amount of ether transferred. Similarly, for every one unit of ether received, there is an average 0.963 ether increase in the average amount of total ether transferred. However, for every one unit of token received, there is an average of 0.057 ether decrease in total amount of ether transferred.

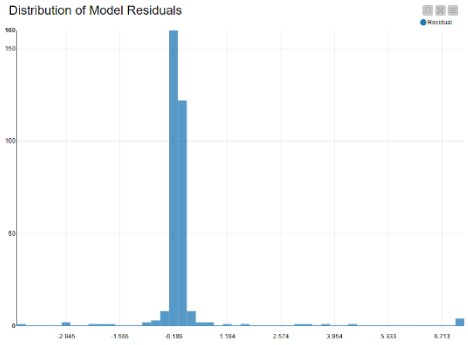


*Figure 23: Normally Distributed Histogram of standard errors*

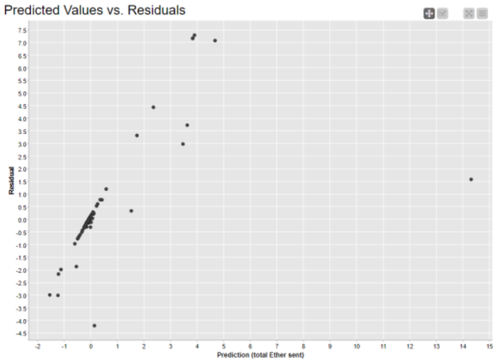


*Figure 24: Residuals' heteroskedasticity test*

Multilinear Regression Model for 'Total Amount of Ether Received' = C :

The new simple linear model for C is as follows:

C = - 0.037A + 0.048B+ 0.738 D

Visually, Figure 26 indicates the presence of heteroskedasticity and Figure 25 implies that the residuals are not normally distributed. Together with negative R-squared value of -0.632, random scatterplot and not achieving normal distribution for the residual, the model for C is mis specified and misfit for future usage.

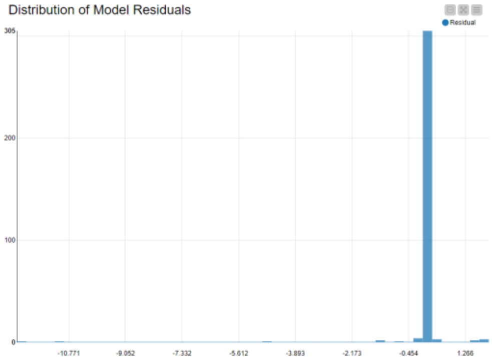


*Figure 25: Not Normally Distributed Histogram of standard errors*

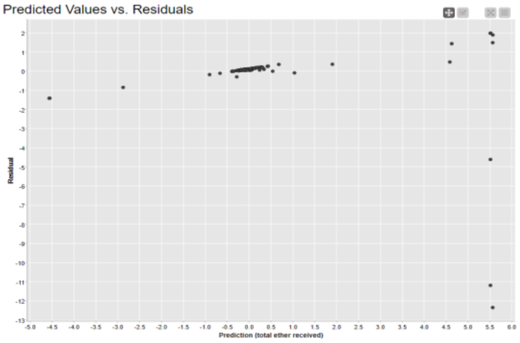


*Figure 26: Not Normally Distributed Histogram of standard errors*

### 3.4.3 Classification Analysis

This study uses KNIME to generate eight classification models (see Appendix A, Figures 9, 10, 11 and 12 for additional information). A decision tree classifier, a random forest classifier, a gradient booster model, and an extreme gradient booster model are separately developed using two datasets A and B. To ensure that both fraud and fair transactions are represented in the training and test sets, stratified sampling is employed, ensuring that both classes are proportionately represented in the final subset, based on the original distribution of classes. This analysis utilizes the SMOTE algorithm presented by Chawla et al. (2002) in order to produce a better classification model. By adjusting the distribution of classes, it improves supervised learning algorithms' classification performance. In Cohen (1988), Kappa results are interpreted as follows: 0 indicates no agreement, 0.01-0.20 indicates little or no agreement, 0.21-1.40 indicates fair agreement, 0.41- 0.60 indicate moderate agreement, 0.61-0.80 indicates substantial agreement, and 0.81-1.00 indicates almost perfect agreement. In addition to the area under curve (AUC), receiver operating characteristic curves (ROC curves) provide additional indicators of model fitness (Bruce et al., 2017).

### 3.4.3.1 Decision Tree Modelling

It is a non-parametric supervised learning method used for classification and regression. It aims to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. Table 3 represents the confusion matrices for decision tree classifier outcomes based on both dataset A and B. Based on Cohen’s criteria, with dataset A, the model is moderate for classification whereas with dataset B, the model is evaluated at fair level.

*Table 3: Confusion Matrices for Decision Tree Models*

| Decision Tree model Confusion Matrix (dataset A) | | |
|---|---|---|
| | **Fair (Predicted)** | **Fraud (Predicted)** |
| **Fair (Actual)** | 240 | 106 |
| **Fraud (Actual)** | 74 | 225 |
| Correctly classified: 465 | Wrongly classified: 180 | |
| Accuracy: 72.09% | Error: 27.91% | |
| Cohen’s kappa (K): 0.443% | | |
| Decision Tree model Confusion Matrix (dataset B) | | |
| | **Fair (Predicted)** | **Fraud (Predicted)** |
| **Fair (Actual)** | 256 | 90 |
| **Fraud (Actual)** | 104 | 195 |
| Correctly classified : 451 | Wrongly classified: 194 | |
| Accuracy: 71.11% | Error: 30.08% | |
| Cohen’s kappa (K): 0.393% | | |

### 3.4.3.2 Random Forest Modelling

Random forest is a technique used in behaviour predictions and behaviour analysis and is built on decision trees. It contains many decision trees that represent a distinct instance of the classification of data input into the random forest. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes model’s prediction. Table 4 presents the confusion matrices for Random Forest classifier outcomes for datasets A and B. According to Cohen's criteria, with both datasets, the model is sufficiently robust to be deployed in the future, since their kappa value is between 0.61% and 0.80%.

*Table 4: Confusion Matrices for Random Forest Models*

| Random Forest Modelling Confusion Matrix (Dataset A) | | |
|---|---|---|
| | **Fair (Predicted)** | **Fraud (Predicted)** |
| **Fair (Actual)** | 309 | 37 |
| **Fraud (Actual)** | 31 | 268 |
| Correctly classified: 577 | | Wrongly classified: 68 |
| Accuracy: 89.46% | | Error: 10.54% |
| Cohen’s kappa (K): 0.788 % | | |
| Random Forest Modelling Confusion Matrix (Dataset B) | | |
| | **Fair (Predicted)** | **Fraud (Predicted)** |
| **Fair (Actual)** | 310 | 36 |
| **Fraud (Actual)** | 44 | 255 |
| **Correctly classified :** 565 | | **Wrongly classified:** 80 |
| **Accuracy:** 87.6% | | **Error:** 12.4% |
| **Cohen’s kappa (K):** 0.750% | | |

### 3.4.3.3 Gradient Boost Modelling

Gradient boosting is a machine learning technique used in regression analysis and classification. The model provides a prediction in the form of an ensemble of weak prediction models, typically decision trees. When a decision tree is the weak learner, a gradient-boosted tree is produced and usually outperforms a random forest algorithm (Piryonesi, et al., 2020). As measured by Cohen's criteria, the Gradient Boosting Model has moderate effectiveness, as its kappa value ranges from 0.41% to 0.60% as displayed in table 5.

*Table 5: Confusion Matrices for Gradient Boosting Models*

| Confusion Matrix: Gradient Boosting Modelling (Dataset A) | | |
|---|---|---|
| | **Fair (Predicted)** | **Fraud (Predicted)** |
| **Fair (Actual)** | 280 | 66 |
| **Fraud (Actual)** | 81 | 218 |
| **Correctly classified :** 496 | **Wrongly classified:** 147 | |
| **Accuracy:** 77.21 % | **Error:** 22.79 % | |
| **Cohen's kappa (K):** 0.540% | | |
| Confusion Matrix: Gradient Boosting Regression (Dataset B) | | |
| | **Fair (Predicted)** | **Fraud (Predicted)** |
| **Fair (Actual)** | 284 | 62 |
| **Fraud (Actual)** | 101 | 198 |
| **Correctly classified :**482 | **Wrongly classified:** 163 | |
| **Accuracy:** 74.73 % | **Error:** 25.27 % | |
| **Cohen's kappa (K):** 0.487% | | |

### 3.4.3.4 Extreme Gradient Boost Modelling

Kaggle's machine learning competition in 2015 featured over half of the winning solutions that used XGBoost (Zhang et al., 2018). The most commonly used method for identifying patterns of anomalous behavior is Extreme Gradient Boosting (XGBoost). The XGBoost algorithm is a decision tree ensemble Machine Learning algorithm that utilizes a gradient boosting architecture. In contrast to gradient boosting machines based on the same architecture, it is 20 times faster and can scale to billions of instances while requiring sparse resources. In this model, three main importance metrics are provided to quantify the importance of all features, namely weight, gain and cover. It is a matrix with the first column listing the features and the other columns defining the importance value in relation to the importance metric. In accordance with Cohen's criteria, the Extreme Gradient Boosting Model is substantially significant, as its kappa value ranges from 0.61% to 0.80% (table 6).

*Table 6: Confusion Matrices for Extreme Gradient Boosting Models*

| XGBoost Modelling (Dataset A) | | |
|---|---|---|
| | **Fair (Predicted)** | **Fraud (Predicted)** |
| **Fair (Actual)** | 284 | 52 |
| **Fraud (Actual)** | 43 | 256 |
| **Correctly classified :** 550 | **Wrongly classified:** 95 | |
| **Accuracy:** 85.27 % | **Error:** 14.73 % | |
| **Cohen's kappa (K):** 0.704 % | | |
| **Table 2:** XGBoost Modelling (Dataset B) | | |
| | **Fair (Predicted)** | **Fraud** |
| **Fair (Actual)** | 305 | 41 |
| **Fraud (Actual)** | 51 | 248 |
| **Correctly classified:** 553 | **Wrongly classified:** 92 | |
| **Accuracy:** 85.74% | **Error:** 14.26% | |
| **Cohen's kappa (K):** 0.713 % | | |

### 3.4.3.5 Classification Model Results

After assessing the eight classification models along with their results in the previous section, this section will justify its selection of Random Forest Classifier for the detection of fraudulent activities on Ethereum Blockchain. As an indicator of model fitness, ROC and AUC are commonly used (Bruce et al., 2017). These two criteria were used to formulate the ranking algorithm, as demonstrated in Figures 9 and 10 of Appendix A. This analysis has led to the selection of the Random Forest Model as the best-fit model for dataset A (see Figure 26 and Figure 27) having Cohen's Kappa(%) of 0.788, an average accuracy of 0.8946, and an average AUC of 0.939; and dataset B (see Figure 28 and Figure 29) having Cohen's Kappa(%) of 0.750, an average accuracy of 0.876 and an average AUC of 0.949, respectively. Conversely, the least fit classifier is Decision Tree with dataset A (see Figure

27 and Figure 28) having Cohen's kappa(%) of 0.443 and an average accuracy of 0.72 as well as an average AUC of 0.767, and dataset B (see Figure 29 and Figure 30) having Cohen's kappa(%) of 0.393, an average accuracy of 0.6992 and an average AUC of 0.785, respectively.

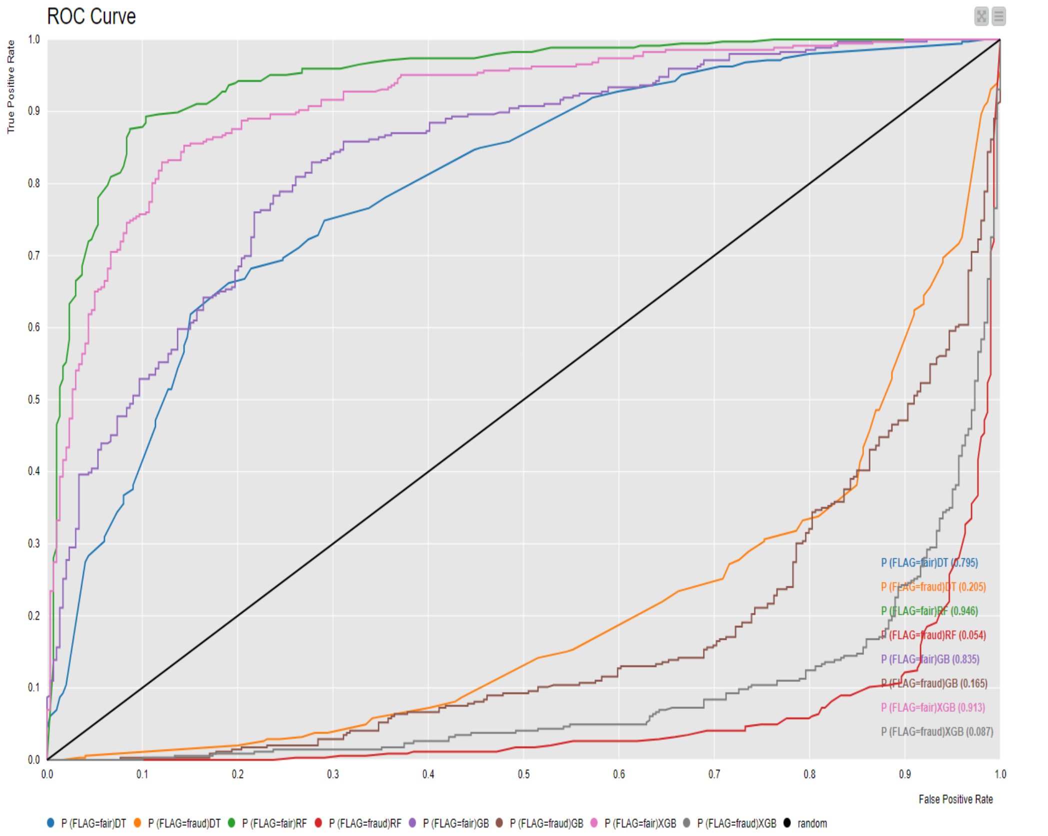
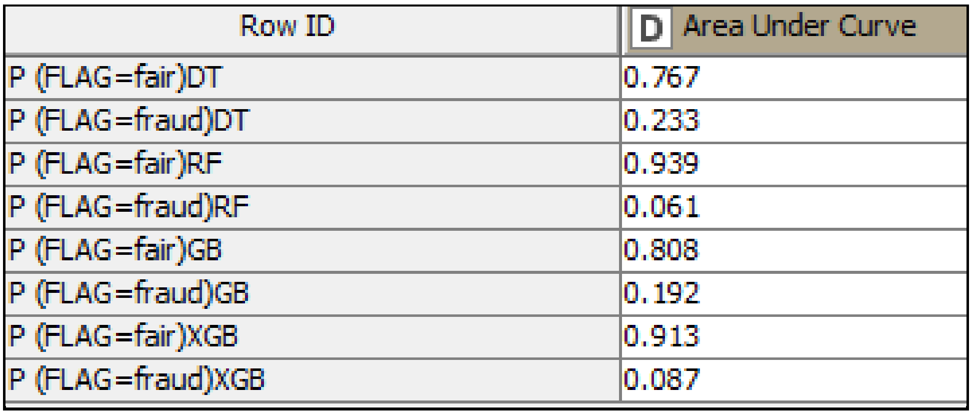


*Figure 27: ROC for four classified models based on dataset A*



| Row ID | D Area Under Curve |
|---|---|
| P (FLAG=fair)DT | 0.767 |
| P (FLAG=fraud)DT | 0.233 |
| P (FLAG=fair)RF | 0.939 |
| P (FLAG=fraud)RF | 0.061 |
| P (FLAG=fair)GB | 0.808 |
| P (FLAG=fraud)GB | 0.192 |
| P (FLAG=fair)XGB | 0.913 |
| P (FLAG=fraud)XGB | 0.087 |

*Figure 28: Area Under Curve value for models with dataset A.*

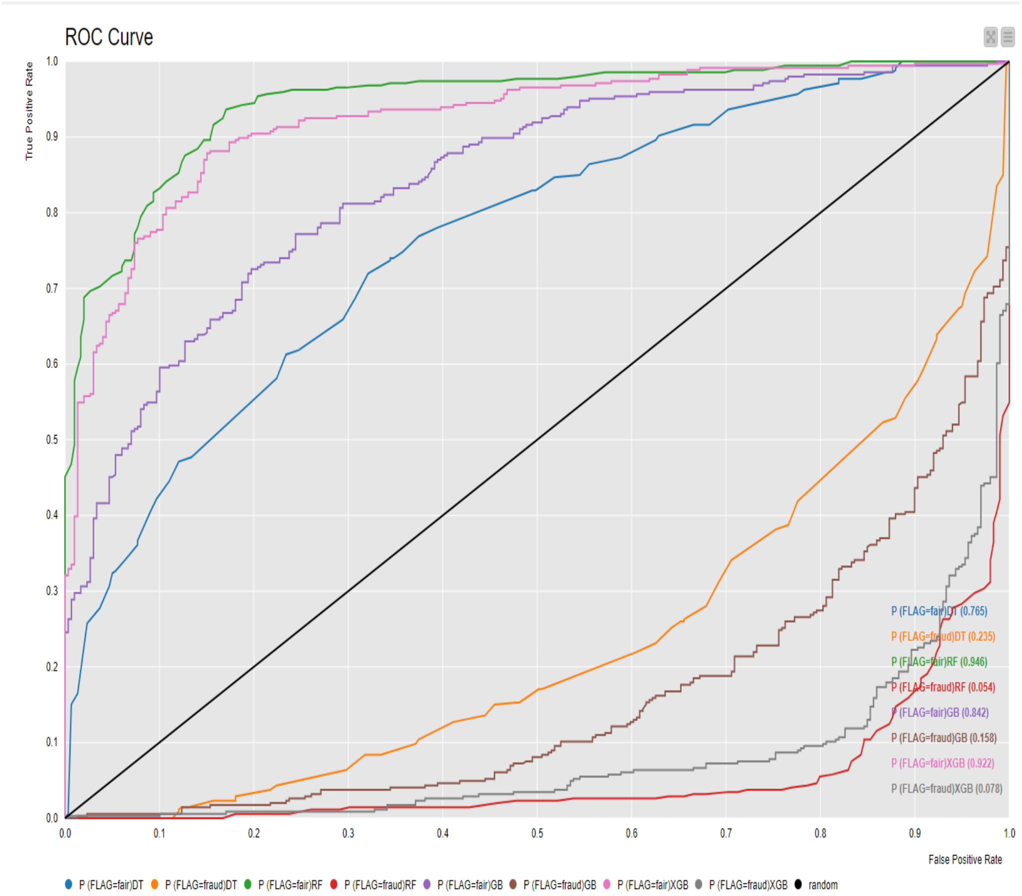
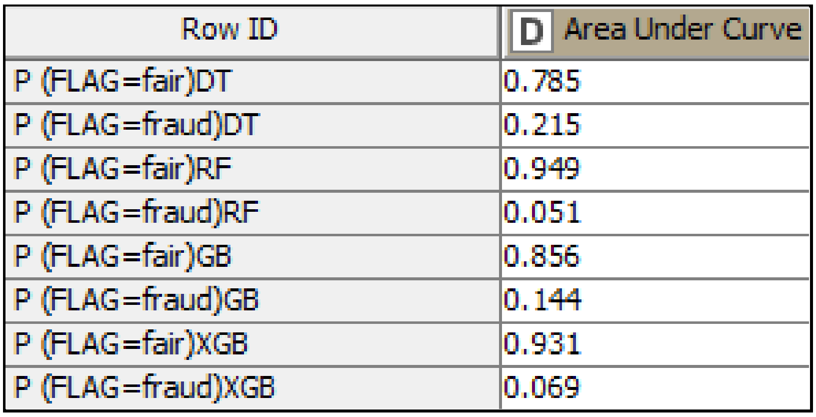


*Figure 29: ROC for four classification models based on dataset B.*



| Row ID | D Area Under Curve |
|---|---|
| P (FLAG=fair)DT | 0.785 |
| P (FLAG=fraud)DT | 0.215 |
| P (FLAG=fair)RF | 0.949 |
| P (FLAG=fraud)RF | 0.051 |
| P (FLAG=fair)GB | 0.856 |
| P (FLAG=fraud)GB | 0.144 |
| P (FLAG=fair)XGB | 0.931 |
| P (FLAG=fraud)XGB | 0.069 |

*Figure 30: Area Under Curve value for models with dataset B.*

## 4. Conclusion

Considering the results of the data mining, it is evident that the Random Forest Classification method is an extremely effective means of detecting illegitimate Ethereum accounts. Despite this, there are internal validity issues with the models since the SMOTE and normalizer nodes were used during the execution process. Without these nodes, the classification models' performances are substandard. Moreover, the box plots and clustering analysis were found to contain outliers. In addition, after reviewing the descriptive histogram of EOS and Ether and cross-checking them with news (Reynolds, 2022), it appears that the charts are misleading and indicate sampling bias. To predict the future, it will therefore be premature to finalize the implementation of the Random Forest Model and the multiple linear regression model (D's equation). Therefore, neither model can be generalized. For this study to be externally valid and reliable, higher quality and larger volumes of data must be trained (Bruce et al., 2017). Further research and data should be collected to validate the results.  Moreover, other techniques should be examined in order to be applied in a real-life situation.

# 5. Appendix A – Workflow Diagram

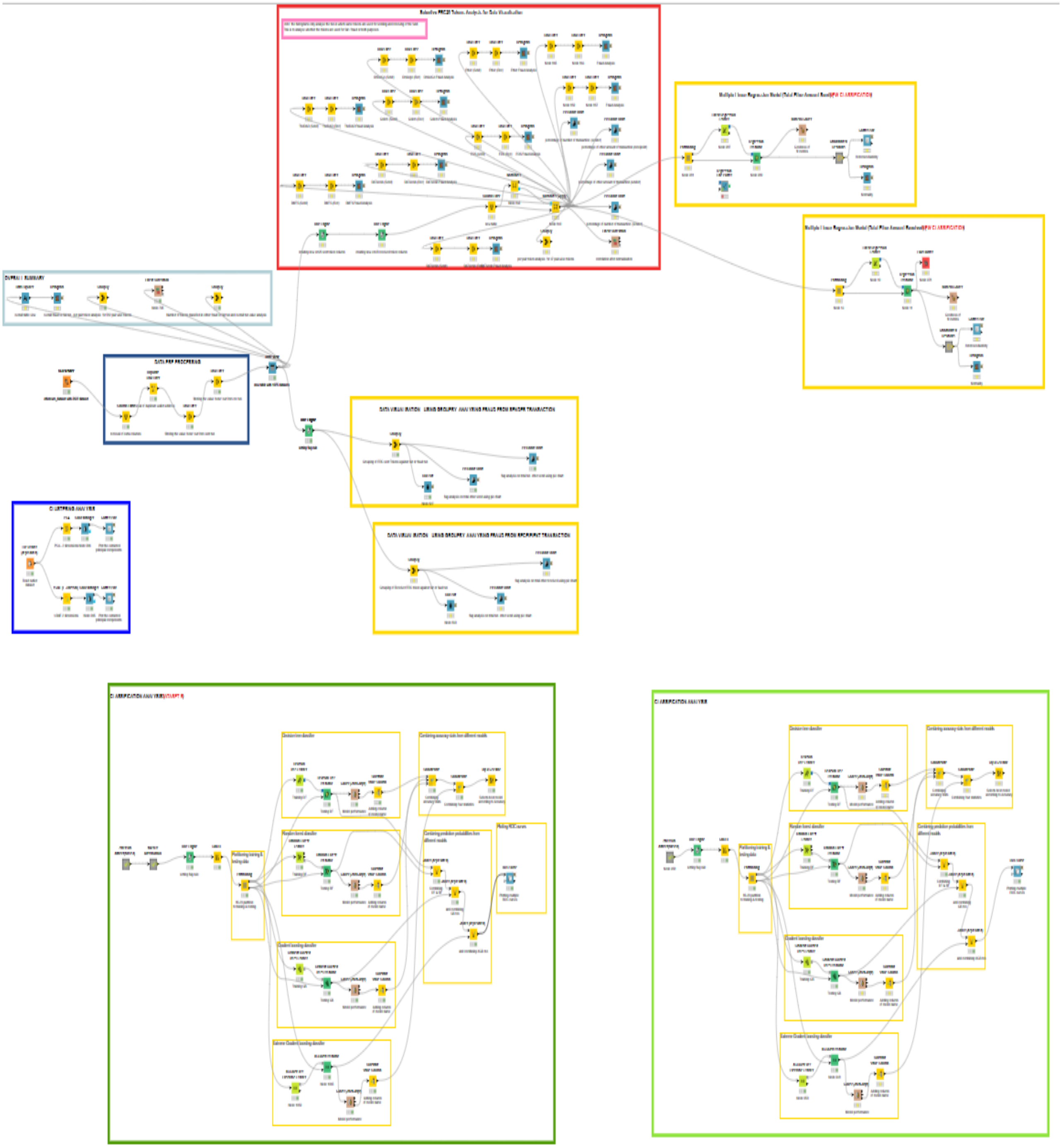


Figure 1: Overview of the workflow

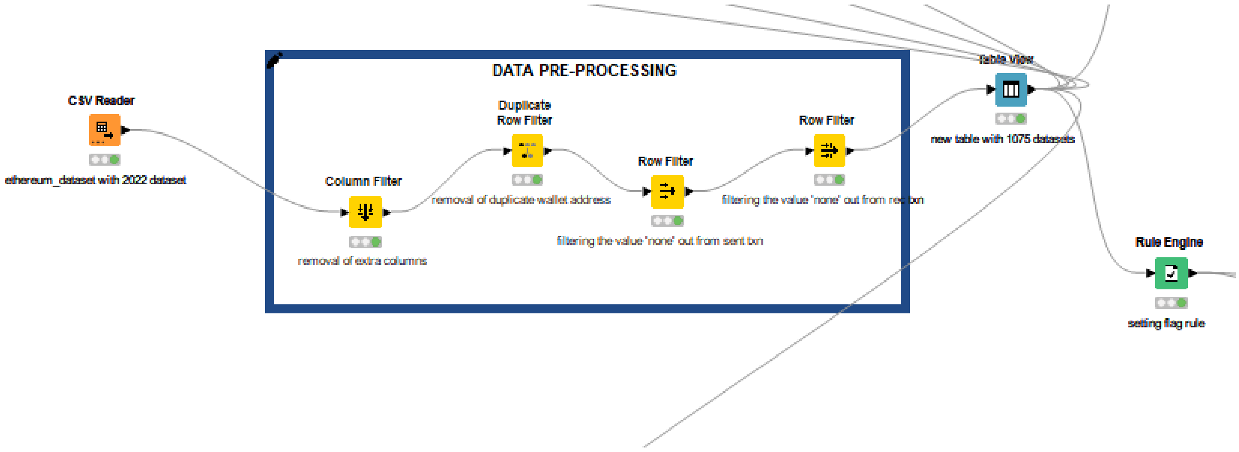

DATA PRE-PROCESSING
CSV Reader
ethereum_dataset with 2022 dataset
Column Filter
removal of extra columns
Duplicate Row Filter
removal of duplicate wallet address
Row Filter
filtering the value 'none' out from sent txn
Row Filter
filtering the value 'none' out from rec txn
Table View
new table with 1075 datasets
Rule Engine
setting flag rule

*Figure 2 : Data Pre-Processing Task*

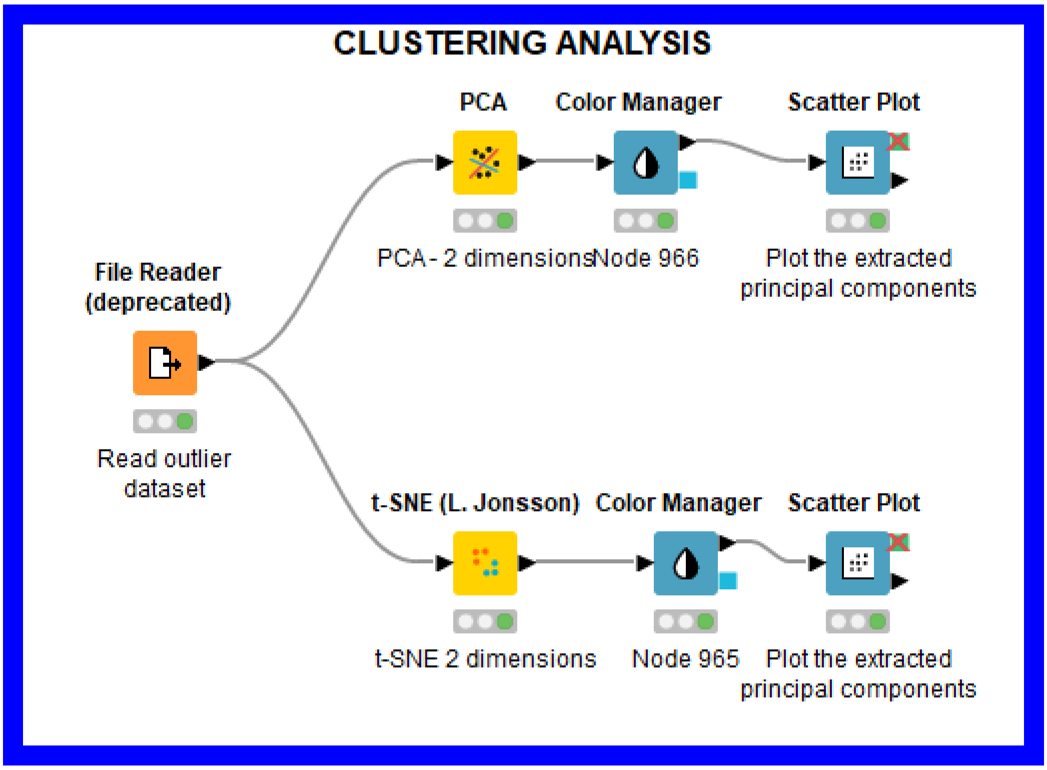


*Figure 3: Clustering Analysis*

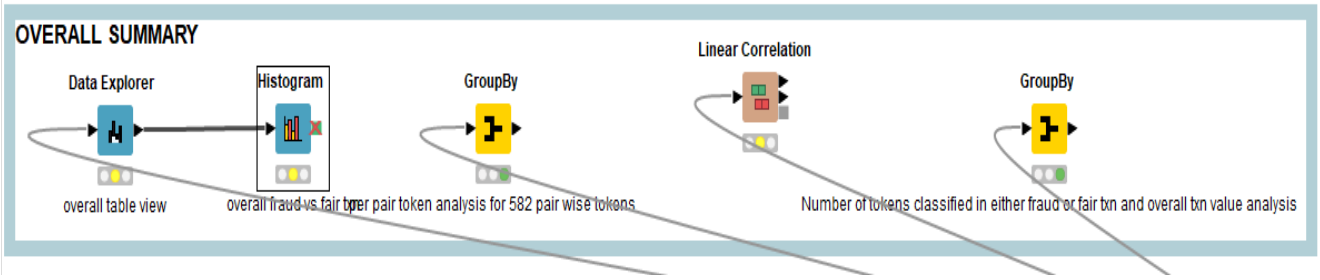

OVERALL SUMMARY
Data Explorer
overall table view
Histogram
overall fraud vs fair txn
GroupBy
per pair token analysis for 582 pair wise tokens
Linear Correlation
GroupBy
Number of tokens classified in either fraud or fair txn and overall txn value analysis

*Figure 4: Overall Data Description*

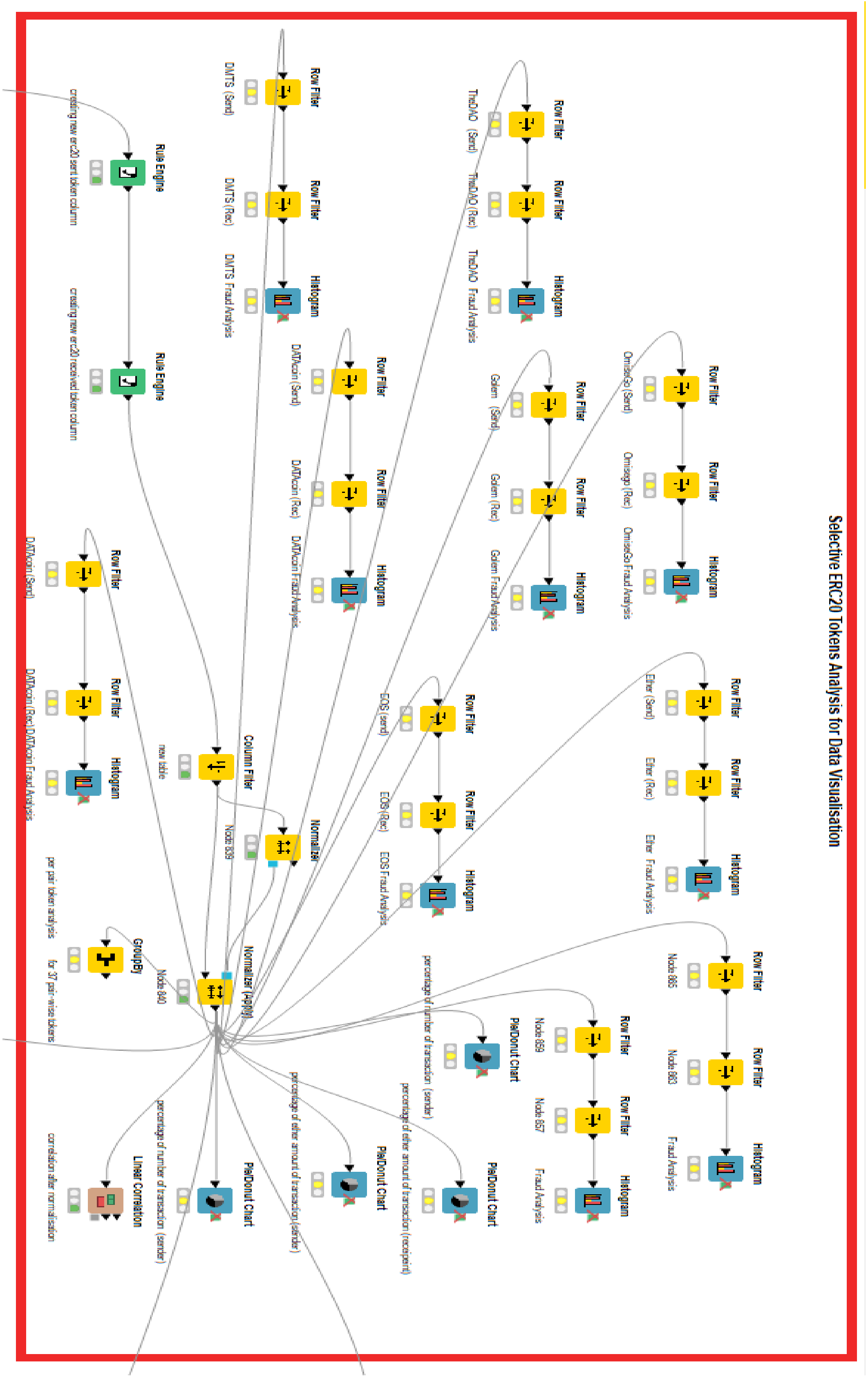


*Figure 2: Data Visualisation for selective tokens*

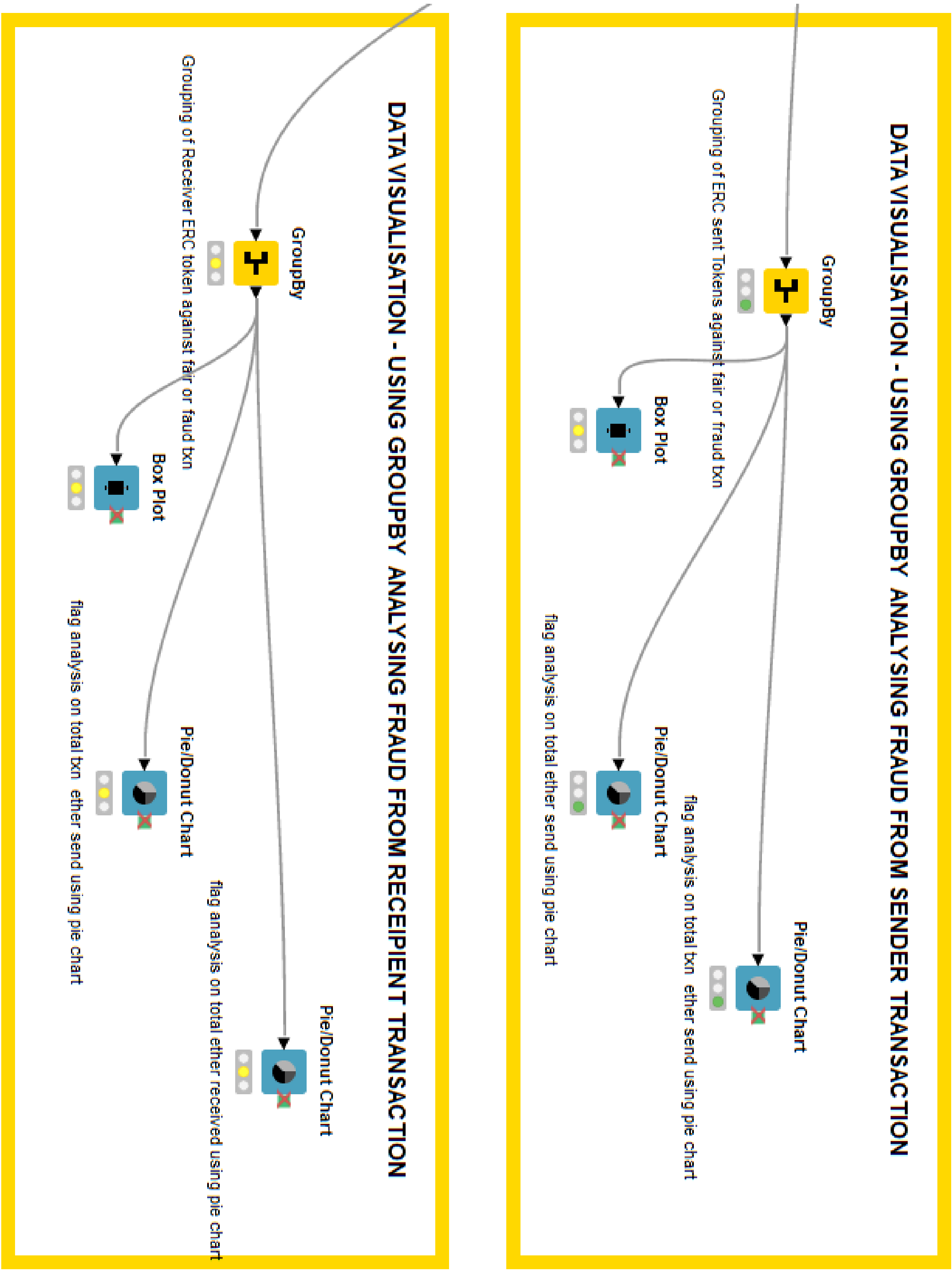

DATA VISUALISATION - USING GROUPBY ANALYSING FRAUD FROM SENDER TRANSACTION
GroupBy
Grouping of ERC sent Tokens against fair or fraud txn
Box Plot
Pie/Donut Chart
flag analysis on total txn ether send using pie chart
Pie/Donut Chart
flag analysis on total ether send using pie chart
DATA VISUALISATION - USING GROUPBY ANALYSING FRAUD FROM RECEIPIENT TRANSACTION
GroupBy
Grouping of Receiver ERC token against fair or faud txn
Box Plot
Pie/Donut Chart
flag analysis on total ether received using pie chart
Pie/Donut Chart
flag analysis on total txn ether send using pie chart

*Figure 3: Data visualisation for numeric data*

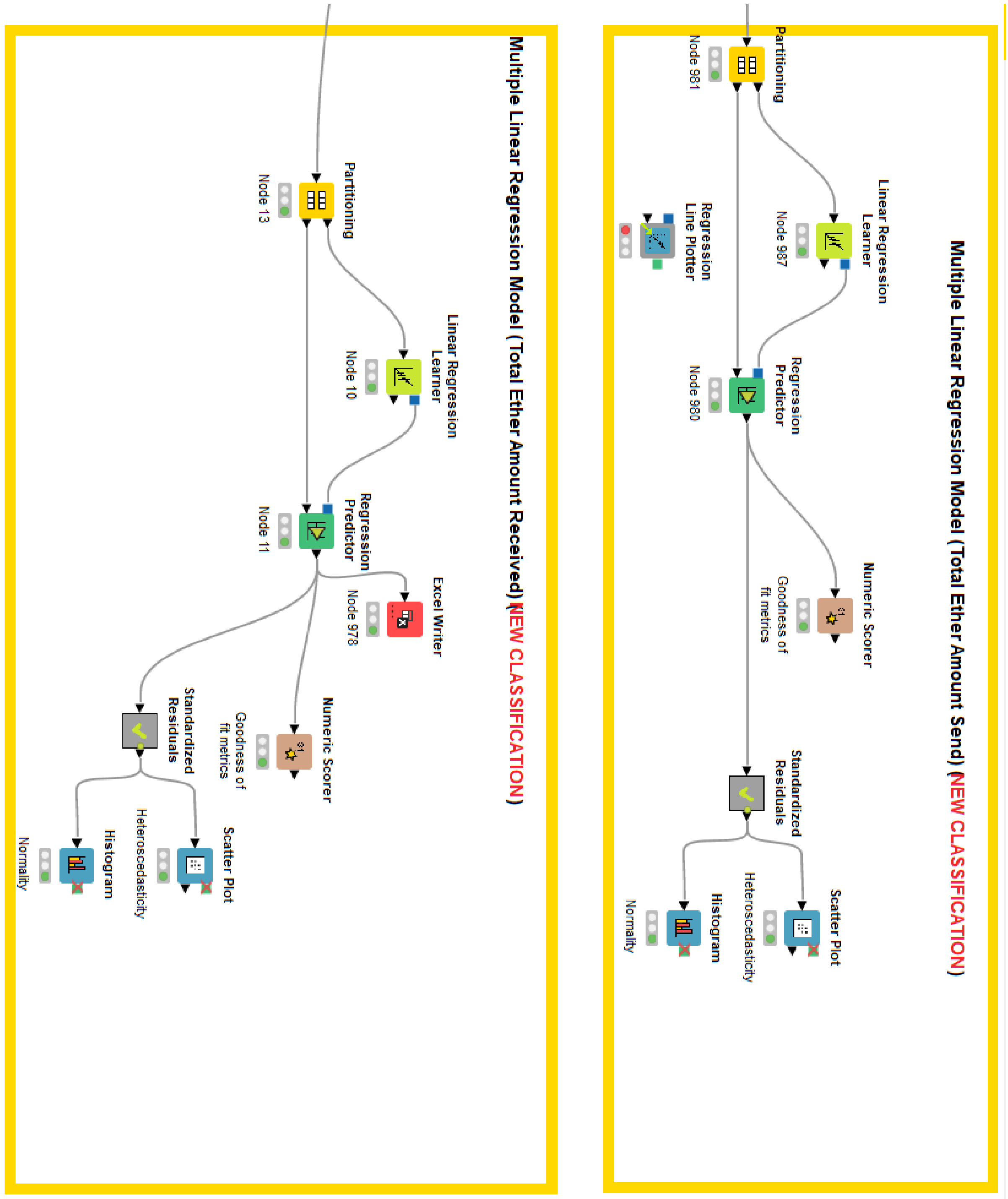


*Figure 4 : Multiple Linear Regression Models based on newly categorised tokens.*

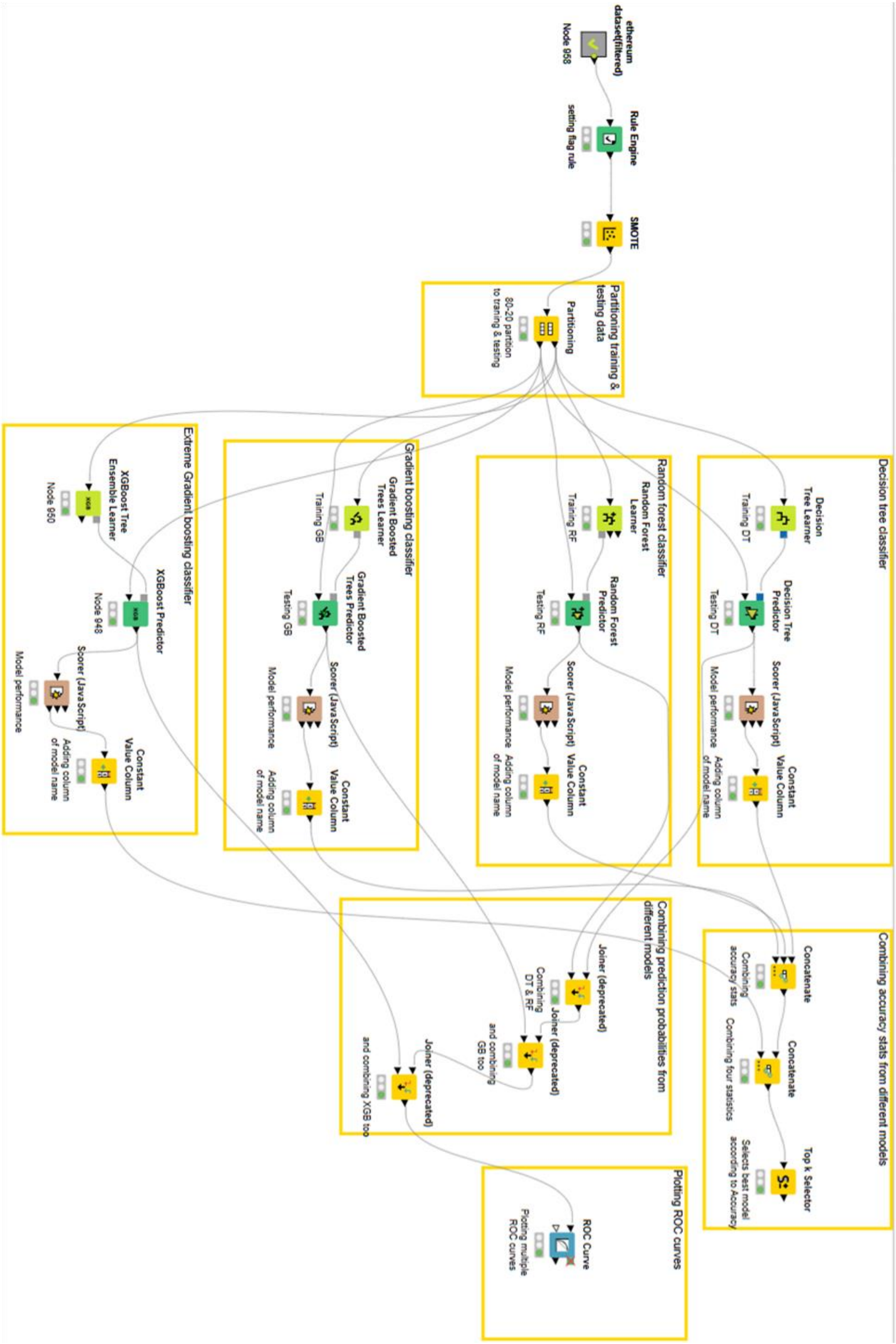


*Figure 5: Classification Models based on dataset A.*

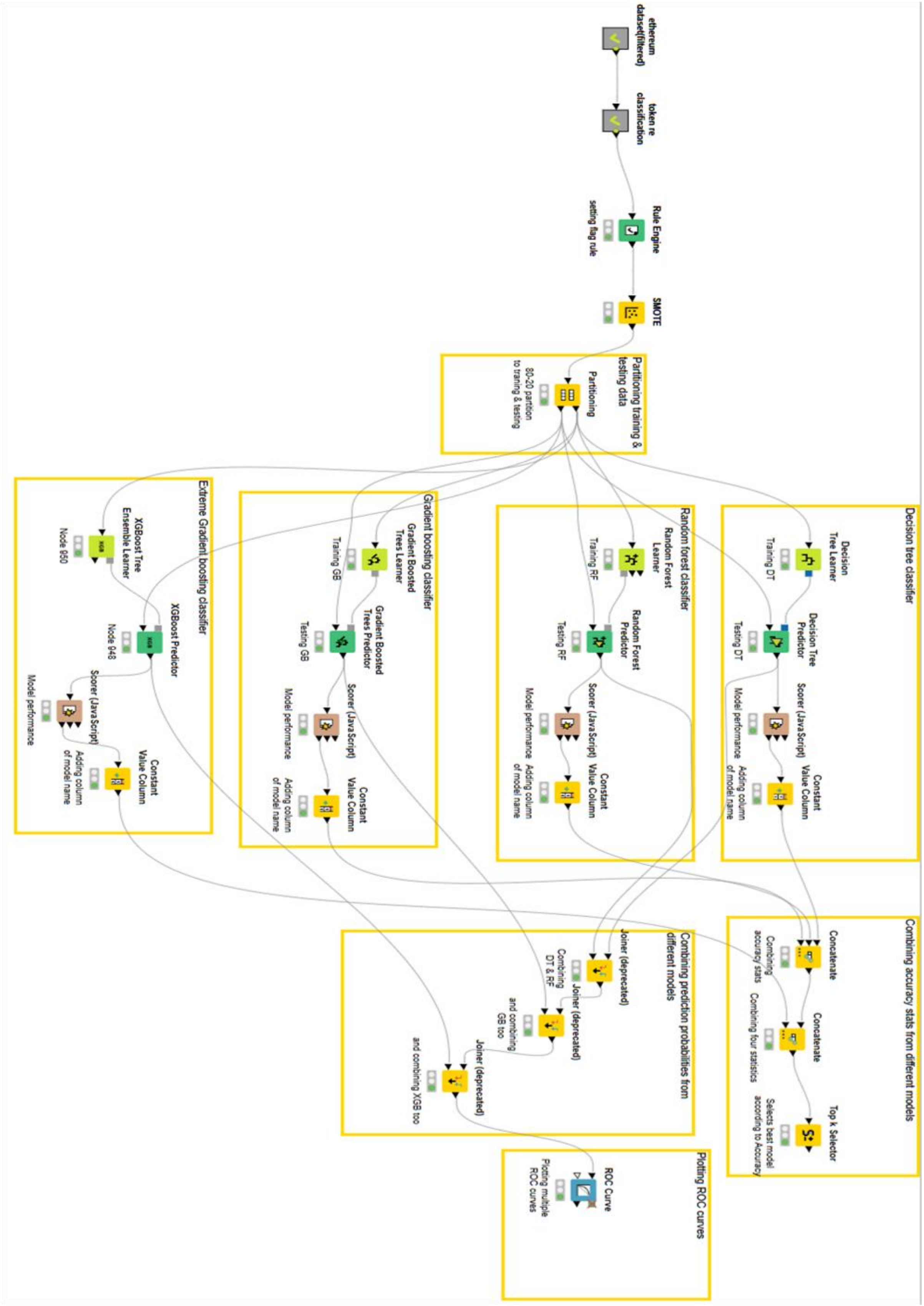
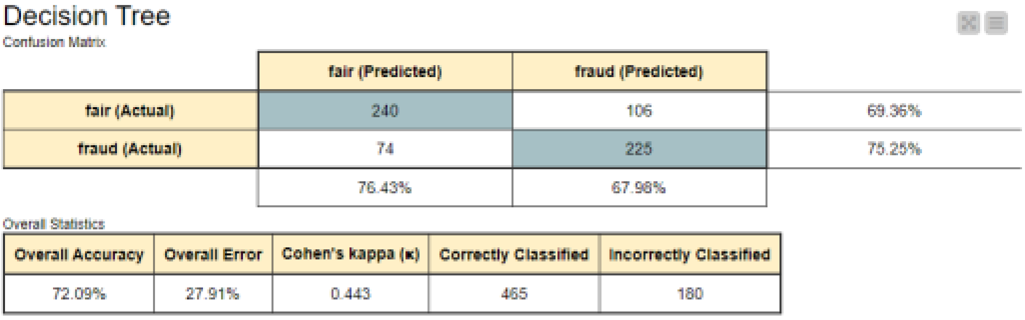


*Figure 6: Classification Models based on dataset with dataset B*



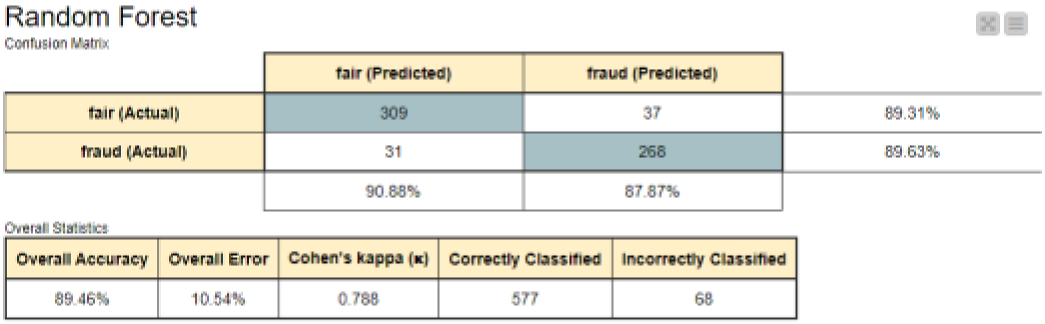
Decision Tree

Confusion Matrix

| | fair (Predicted) | fraud (Predicted) | |
|---|---|---|---|
| fair (Actual) | 240 | 106 | 69.36% |
| fraud (Actual) | 74 | 225 | 75.25% |
| | 76.43% | 67.98% | |

Overall Statistics

| Overall Accuracy | Overall Error | Cohen's kappa (κ) | Correctly Classified | Incorrectly Classified |
|---|---|---|---|---|
| 72.09% | 27.91% | 0.443 | 465 | 180 |



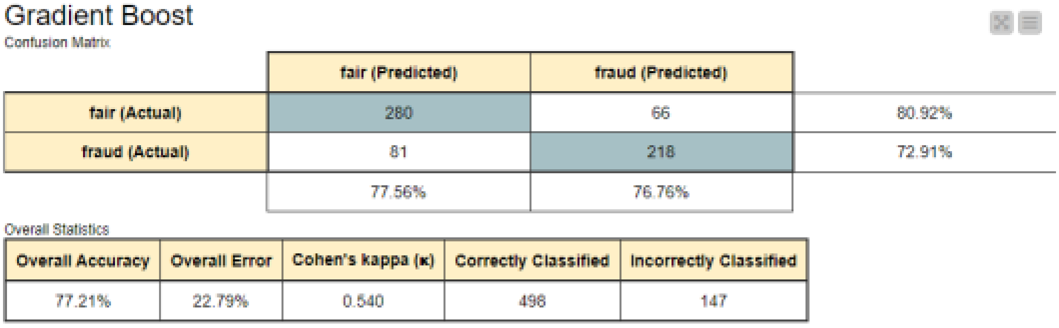
Random Forest

Confusion Matrix

| | fair (Predicted) | fraud (Predicted) | |
|---|---|---|---|
| fair (Actual) | 309 | 37 | 89.31% |
| fraud (Actual) | 31 | 268 | 89.63% |
| | 90.88% | 87.87% | |

Overall Statistics

| Overall Accuracy | Overall Error | Cohen's kappa (κ) | Correctly Classified | Incorrectly Classified |
|---|---|---|---|---|
| 89.46% | 10.54% | 0.788 | 577 | 68 |



Gradient Boost

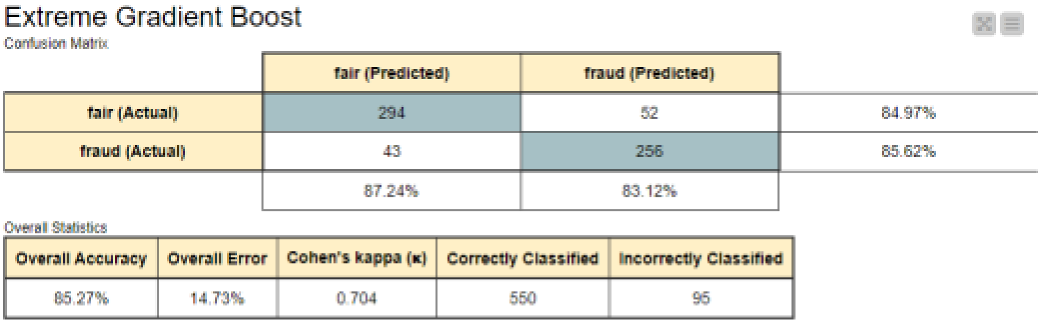
Confusion Matrix

| | fair (Predicted) | fraud (Predicted) | |
|---|---|---|---|
| fair (Actual) | 280 | 66 | 80.92% |
| fraud (Actual) | 81 | 218 | 72.91% |
| | 77.56% | 76.76% | |

Overall Statistics

| Overall Accuracy | Overall Error | Cohen's kappa (κ) | Correctly Classified | Incorrectly Classified |
|---|---|---|---|---|
| 77.21% | 22.79% | 0.540 | 498 | 147 |

-



Extreme Gradient Boost

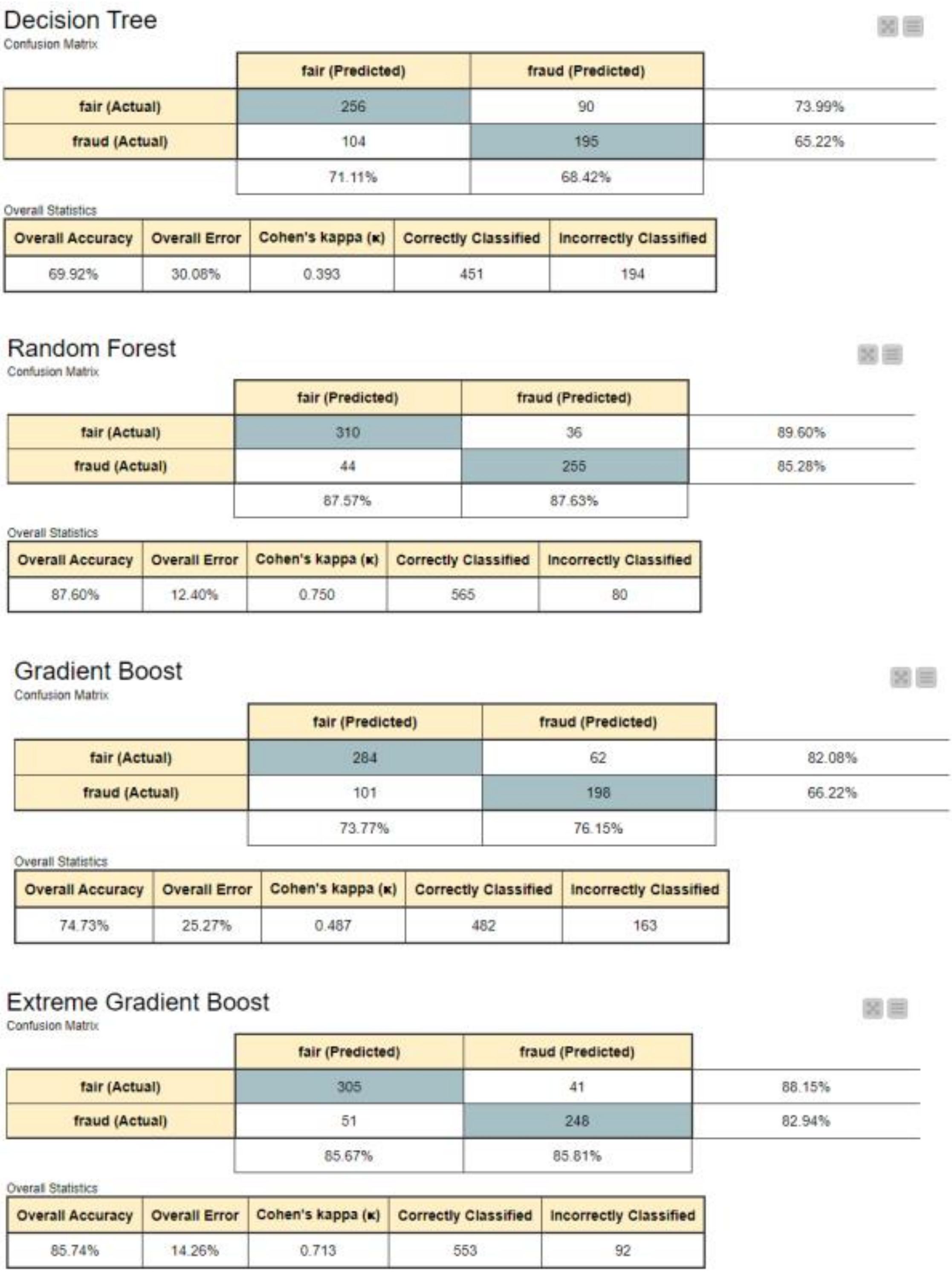
Confusion Matrix

| | fair (Predicted) | fraud (Predicted) | |
|---|---|---|---|
| fair (Actual) | 294 | 52 | 84.97% |
| fraud (Actual) | 43 | 256 | 85.62% |
| | 87.24% | 83.12% | |

Overall Statistics

| Overall Accuracy | Overall Error | Cohen's kappa (κ) | Correctly Classified | Incorrectly Classified |
|---|---|---|---|---|
| 85.27% | 14.73% | 0.704 | 550 | 95 |

*Figure 7: KNIME's output of Confusion Matrices of four distinct classification models based on dataset A.*



## Decision Tree

Confusion Matrix

| | fair (Predicted) | fraud (Predicted) | |
|---|---|---|---|
| fair (Actual) | 256 | 90 | 73.99% |
| fraud (Actual) | 104 | 195 | 65.22% |
| | 71.11% | 68.42% | |

Overall Statistics

| Overall Accuracy | Overall Error | Cohen's kappa (κ) | Correctly Classified | Incorrectly Classified |
|---|---|---|---|---|
| 69.92% | 30.08% | 0.393 | 451 | 194 |

## Random Forest

Confusion Matrix

| | fair (Predicted) | fraud (Predicted) | |
|---|---|---|---|
| fair (Actual) | 310 | 36 | 89.60% |
| fraud (Actual) | 44 | 255 | 85.28% |
| | 87.57% | 87.63% | |

Overall Statistics

| Overall Accuracy | Overall Error | Cohen's kappa (κ) | Correctly Classified | Incorrectly Classified |
|---|---|---|---|---|
| 87.60% | 12.40% | 0.750 | 565 | 80 |

## Gradient Boost

Confusion Matrix

| | fair (Predicted) | fraud (Predicted) | |
|---|---|---|---|
| fair (Actual) | 284 | 62 | 82.08% |
| fraud (Actual) | 101 | 198 | 66.22% |
| | 73.77% | 76.15% | |

Overall Statistics

| Overall Accuracy | Overall Error | Cohen's kappa (κ) | Correctly Classified | Incorrectly Classified |
|---|---|---|---|---|
| 74.73% | 25.27% | 0.487 | 482 | 163 |

## Extreme Gradient Boost

Confusion Matrix

| | fair (Predicted) | fraud (Predicted) | |
|---|---|---|---|
| fair (Actual) | 305 | 41 | 88.15% |
| fraud (Actual) | 51 | 248 | 82.94% |
| | 85.67% | 85.81% | |

Overall Statistics

| Overall Accuracy | Overall Error | Cohen's kappa (κ) | Correctly Classified | Incorrectly Classified |
|---|---|---|---|---|
| 85.74% | 14.26% | 0.713 | 553 | 92 |

*Figure 8: KNIME's output of Confusion Matrices of four distinct classification models based on dataset B*

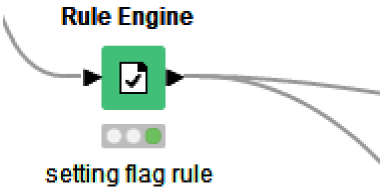
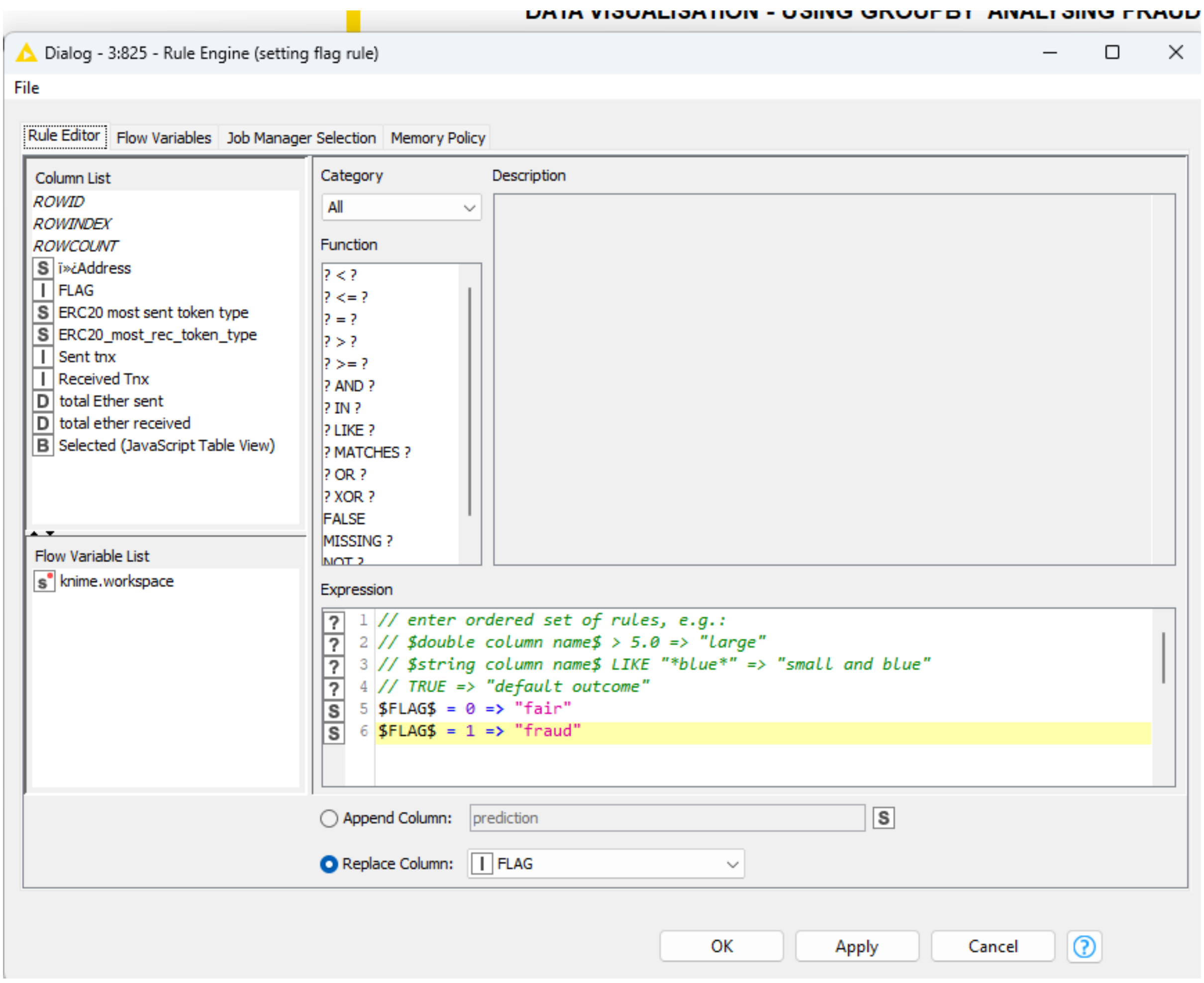


*Figure 12: Rule engine for Flag Classification as either fraud or fair.*



Column List
ROWID
ROWINDEX
ROWCOUNT
S ï»¿Address
I FLAG
S ERC20 most sent token type
S ERC20_most_rec_token_type
I Sent tnx
I Received Tnx
D total Ether sent
D total ether received
B Selected (JavaScript Table View)

Flow Variable List
s knime.workspace

Category: All

Function:
? < ?
? <= ?
? = ?
? > ?
? >= ?
? AND ?
? IN ?
? LIKE ?
? MATCHES ?
? OR ?
? XOR ?
FALSE
MISSING ?
NOT ?

```
// enter ordered set of rules, e.g.:
// $double column name$ > 5.0 => "large"
// $string column name$ LIKE "*blue*" => "small and blue"
// TRUE => "default outcome"
$FLAG$ = 0 => "fair"
$FLAG$ = 1 => "fraud"
```

○ Append Column: prediction

● Replace Column: I FLAG

*Figure 12: Rule engine's Expression for Flag Classification as either fraud or fair.*

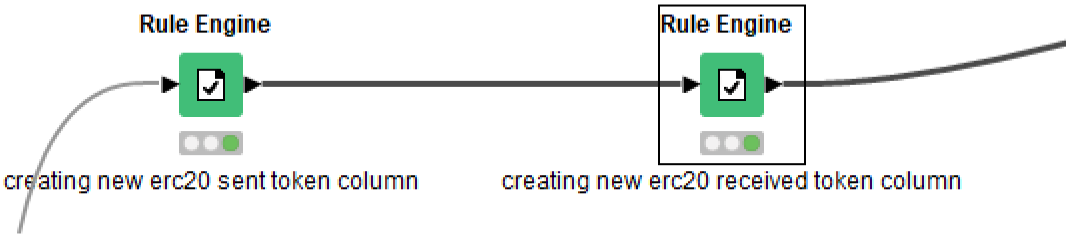
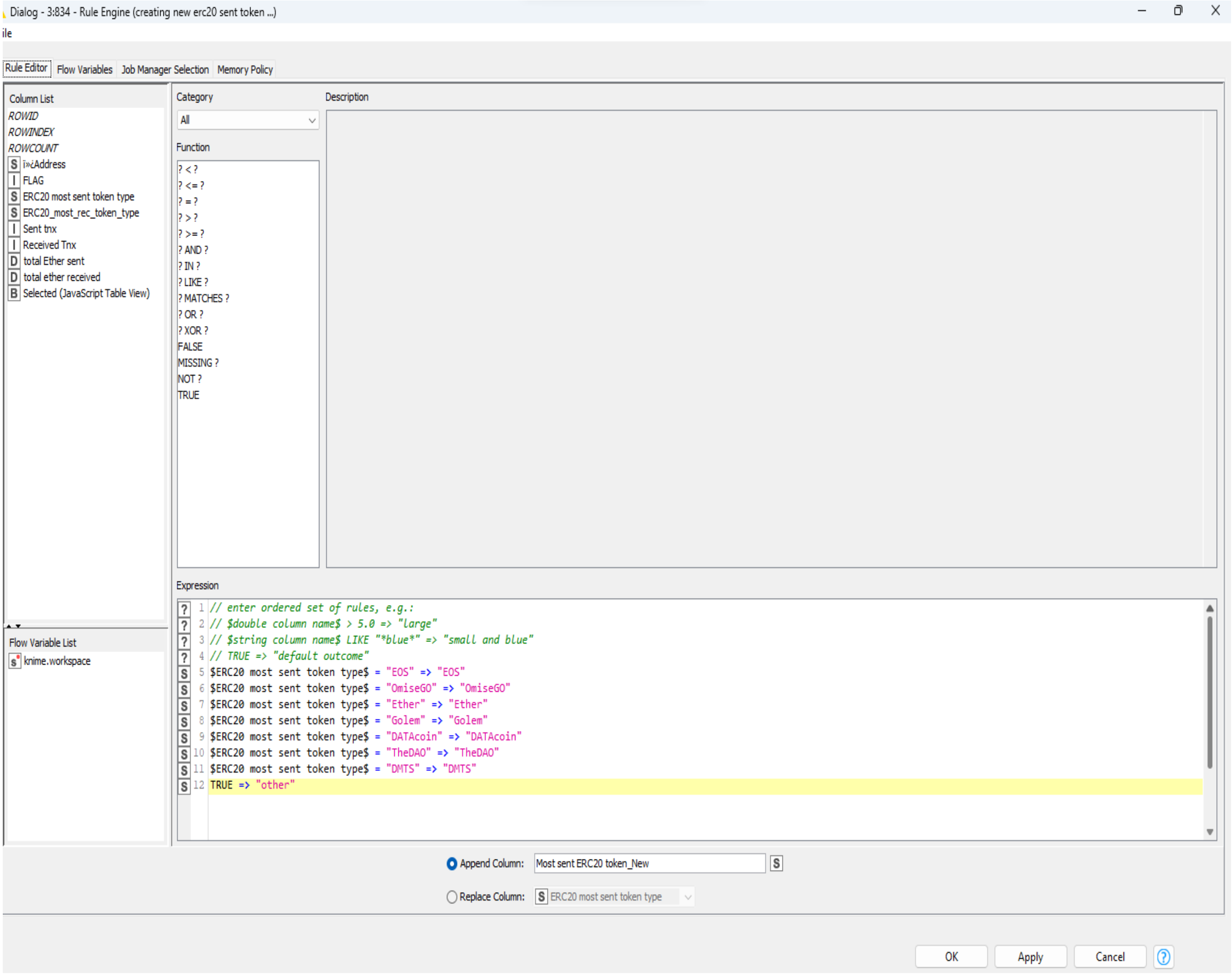

Rule Engine
creating new erc20 sent token column
Rule Engine
creating new erc20 received token column

*Figure 13: Rule engines for Classification of 612 tokens into 8 categories.*

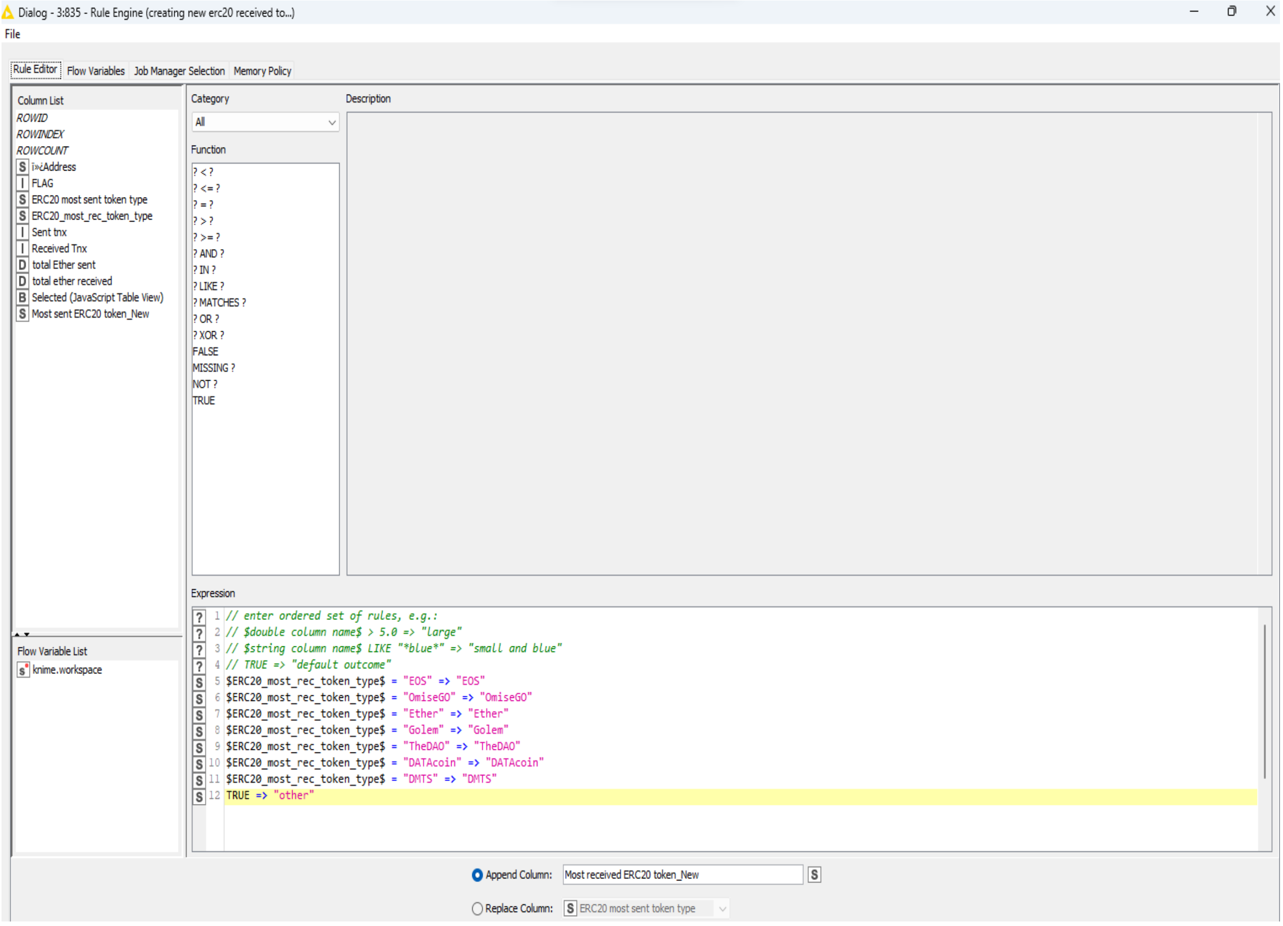


```
// enter ordered set of rules, e.g.:
// $double column name$ > 5.0 => "large"
// $string column name$ LIKE "*blue*" => "small and blue"
// TRUE => "default outcome"
$ERC20 most sent token type$ = "EOS" => "EOS"
$ERC20 most sent token type$ = "OmiseGO" => "OmiseGO"
$ERC20 most sent token type$ = "Ether" => "Ether"
$ERC20 most sent token type$ = "Golem" => "Golem"
$ERC20 most sent token type$ = "DATAcoin" => "DATAcoin"
$ERC20 most sent token type$ = "TheDAO" => "TheDAO"
$ERC20 most sent token type$ = "DMTS" => "DMTS"
TRUE => "other"
```

● Append Column: Most sent ERC20 token_New

○ Replace Column: ERC20 most sent token type

*Figure 14: Rule engine's expression for ERC-20 sent tokens' classification for dataset B.*



Category: All

```
// enter ordered set of rules, e.g.:
// $double column name$ > 5.0 => "large"
// $string column name$ LIKE "*blue*" => "small and blue"
// TRUE => "default outcome"
$ERC20_most_rec_token_type$ = "EOS" => "EOS"
$ERC20_most_rec_token_type$ = "OmiseGO" => "OmiseGO"
$ERC20_most_rec_token_type$ = "Ether" => "Ether"
$ERC20_most_rec_token_type$ = "Golem" => "Golem"
$ERC20_most_rec_token_type$ = "TheDAO" => "TheDAO"
$ERC20_most_rec_token_type$ = "DATAcoin" => "DATAcoin"
$ERC20_most_rec_token_type$ = "DMTS" => "DMTS"
TRUE => "other"
```

● Append Column: Most received ERC20 token_New S

○ Replace Column: S ERC20 most sent token type

*Figure 15: Rule engine's expression for ERC-20 received tokens' classification for dataset B.*

## 6. References

Antonopoulos, A.M. and Gavin Wood Ph.D (2018). *Mastering Ethereum*. O'Reilly Media.

Barnett, Vic; Lewis, Lewis (1978). *Outliers in statistical data*. John Wiley & Sons Ltd.

Berthold, M.R., Cebron, N., Dill, F., Gabriel, T.R., Kötter, T., Meinl, T., Ohl, P., Thiel, K. and Wiswedel, B. (2009). KNIME - the Konstanz information miner. *ACM SIGKDD Explorations Newsletter*, 11(1), p.26. doi:https://doi.org/10.1145/1656274.1656280.

Bruce, P. and Bruce, A. (2017). *Practical Statistics for Data Scientists*. 'O'Reilly Media, Inc.'

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16(16), pp.321–357. doi:https://doi.org/10.1613/jair.953.

Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences*. 2nd ed. Routledge. https://doi.org/10.4324/9780203771587.

Farrugia, S., Ellul, J. and Azzopardi, G. (2020). Detection of illicit accounts over the Ethereum blockchain. *Expert Systems with Applications*, p.113318. https://doi.org/10.1016/j.eswa.2020.113318.

Han, J., Kamber, M. and Pei, J. (2012). *Data mining: concepts and techniques*. Burlington, Ma: Elsevier.

Hillemann, E.-C., Nussbaumer, A. and Albert, D. (2015). The Role of Cognitive Biases in Criminal Intelligence Analysis and Approaches for their Mitigation. *2015 European Intelligence and Security Informatics Conference*. https://doi.org/10.1109/eisic.2015.9.

Reynolds, S. (2022). *Most Crypto Scams on BNB Chain, Solidus Labs Says*. [online] www.coindesk.com. Available at: https://www.coindesk.com/business/2022/10/27/most-crypto-scams-on-bnb-chain-solidus-labs-says/ [Accessed 15 Mar. 2023].

Piryonesi, S.M. and El-Diraby, T.E. (2020). Data Analytics in Asset Management: Cost-Effective Prediction of the Pavement Condition Index. *Journal of Infrastructure Systems*, 26(1), p.04019036. https://doi.org/10.1061/(asce)is.1943-555x.0000512.

Zhang, D., Qian, L., Mao, B., Huang, C., Huang, B. and Si, Y. (2018). A Data-Driven Design for Fault Detection of Wind Turbines Using Random Forests and XGboost. *IEEE Access*, 6, pp.21020–21031. https://doi.org/10.1109/access.2018.2818678.