Effect of Data Resampling on Feature Importance in Imbalanced Blockchain Data: Comparison Studies of Resampling Techniques

Ismail Alarab[1,1], Simant Prakoonwit[1]

[1] Bournemouth University, United Kingdom
{ialarab, sprakoonwit}@bournemouth.ac.uk

Abstract. Cryptocurrency blockchain data encounters a class-imbalance problem due to only a few known labels of illicit or fraudulent activities in the blockchain network. For this purpose, we seek to provide a comparison of various resampling methods applied to two highly imbalanced datasets derived from the blockchain of Bitcoin and Ethereum after further dimensionality reductions, unlike previous studies on these datasets. Firstly, we study the performance of various classical supervised learning methods to classify illicit transactions/accounts on Bitcoin/Ethereum datasets, respectively. Consequently, we apply a variety of resampling techniques to these datasets using the best performing learning algorithm on each of these datasets. Subsequently, we study the feature importance of the given models, wherein the resampled datasets have revealed a direct influence on the explainability of the model. Our main finding is that undersampling using the edited nearest-neighbour technique has attained an accuracy of more than 99% on the given datasets by removing the noisy data points from the whole dataset. Moreover, the best-performing learning algorithms have shown superior performance after feature reduction on these datasets in comparison to their original studies. The matchless contribution lies in discussing the effect of the data resampling on feature importance which is interconnected with explainable artificial intelligence techniques.

Keywords: Resampling Techniques, Cryptocurrency data, Bitcoin blockchain, Ethereum blockchain

1 Introduction

Imbalanced classification is a typical problem in machine learning, which can be encountered in many wide-ranging applications such as in financial services (Makki et al., 2019; Zhang & Trubey, 2019), healthcare (Akinnuwesi et al., 2021; Fan et al., 2021), biomedical (Oh et al., 2011) and blockchain (Harlev et al., 2018). In particular, blockchain technology has gained growing attention in the last few years whereby a machine learning approach is required to deal with the vast amount of data generated

[1] Please note that the LNCS Editorial assumes that all authors have used the western naming convention, with given names preceding surnames. This determines the structure of the names in the running heads and the author index.

by this technology. In (Weber et al., 2019) and (Farrugia et al., 2020), the machine learning approach has revealed promising outcomes to detect fraudulent activities (e.g., scams, money laundering) in the public blockchain data. The latter studies have contributed two real-world datasets derived from the Bitcoin and the Ethereum networks, respectively, to classify suspicious records of the public blockchain data. One of these datasets derived from Bitcoin, the so-called Elliptic dataset, is a highly imbalanced graph-structured data of Bitcoin transactions as nodes and edges as payments flow which is released by Elliptic company and studied in its original contribution by Weber et al. (2019). This dataset provides two different types of features called local features which belong to the transactions and global features which correspond to the topology of the graph network of the Elliptic data. In their original study, Weber et al. (2019) have benchmarked various classical supervised learning methods against graph convolutional networks to classify the licit (e.g., transactions belonging to miners) and illicit (e.g., transactions belonging to scams) transactions in the Elliptic data.

They have also examined the different combinations of local and global features on the classification results. As a result, the random forest has outperformed the graph convolutional network with an accuracy of 97.7% using the whole set of local and global features that count to 166 features. Another dataset is the Ethereum account data that has been introduced in (Farrugia et al., 2020) which inherits the class-imbalance problem. This study has performed classification using XGBoost to detect illicit accounts over the Ethereum blockchain. This study has achieved an accuracy of 96.3% as well as provided insights about the most important features.

Subsequently, these datasets have undergone a variety of studies to improve the classification or to study the model's uncertainty of illicit Bitcoin transactions as in (Alarab et al., 2020a, b, 2021; Alarab & Prakoonwit, 2021; Sun et al., 2022; Tasharrofi & Taheri, 2021; Bynagari & Ahmed, 2021) or illicit Ethereum accounts as in (Alarab & Prakoonwit, 2021; Bynagari & Ahmed, 2021; Ibrahim et al., 2021).

As a result, tree-based learning algorithms have performed the best on these cryptocurrency datasets. Regardless of the promising results provided by the preceding contributions, there is no comprehensive study that addresses the high class-imbalance embedded in these datasets. Dealing with such types of datasets is challenging due to their high dimensionality and class imbalance. On the other hand, resampling techniques have evolved starting from the generic Synthetic Minority Resampling Technique (SMOTE) and its variants to the recent adaptive oversampling techniques to tackle the class-imbalance problem (He & Garcia, 2009; Verbiest et al., 2014; Kovács, 2019b; Fernández et al., 2017). Addressing class imbalance using SMOTE and its variants has shown significant success in the literature due to its simplicity and outperformance. For this purpose, we aim to carry out a comprehensive study using a variety of resampling techniques using Bitcoin and Ethereum datasets to point out the impact of resampling techniques on feature importance which influences the explainability of the model. Firstly, we perform feature reductions then we apply various classical supervised learning methods to classify illicit transactions of Elliptic in the Bitcoin dataset and fraud accounts in the Ethereum dataset. We show that random forest and XGBoost have provided the best performance, respectively, on the data derived from Bitcoin and Ethereum. Moreover, we claim our achievement by applying data preprocessing and feature reductions which revealed significant outperformance in comparison to the models used in the original contributions of the given datasets. Using the latter algorithms on the relevant datasets, we address the class imbalance by applying a variety of oversampling and undersampling techniques, wherein we evaluate and compare the performance of the given models using accuracy, precision, recall, f_1-score receiver-operation-curve (ROC), and area-under-curve (AUC) scores. We discuss our best results using Edited Nearest Neighbours applied to the whole dataset that admits an accuracy greater than 99% in both datasets. On the other hand, feature importance in blockchain datasets is indispensable and plays a pivotal role in the explainability of the classification model. So, we point out the influence of resampling techniques on feature importance that have a significant impact on the explainability of the used models. We verify the preceded statement by performing the Wilcoxon signed-rank test – a non-parametric statistical hypothesis test – where we can provide evidence that the scores of the feature importance are different before and after applying a resampling technique.

The rest of this paper is organised as follows: Section 2 discusses the related work; Section 3 demonstrates the methods used in our experiments that are provided in Section 4. A discussion and conclusion are provided in Sections 5 and 6, respectively.

2 Related Work

Cryptocurrency blockchain has received a grown interest in the surveillance and analysis of its transactions flow to detect illicit activities in the blockchain (Liu et al. 2021). Since then, many studies have adopted visual analytics tools to trace the sources of illicit funds such as the case in the Bitcoin blockchain (Reid & Harrigan, 2013; Meiklejohn et al., 2013). However, the rapid increase of blockchain data has required machine learning models to handle the massive amount of data generated. The work in (Ostapowicz & Żbikowski, 2020) has presented different supervised methods to detect fraudulent activities in the blockchain. This work has focused on referring the malicious actors to applying well-known software or fake emails to steal money. The authors (Weber et al., 2019) have performed classification on data derived from Bitcoin, known as the Elliptic dataset, to detect illicit transactions, wherein random forest has shown superior performance on this dataset against all other learning algorithms such as logistic regression, multi-layer perceptrons, tree-based learning methods and graph convolutional networks in (Weber et al., 2019; Alarab et al., 2020a, b; Lorenz et al., 2020). Farrugia et al. (2020) have introduced Ethereum account data where XGBoost classifier has admittedly classified fraud accounts based on their transaction history with good performance. Other applications that adopted machine learning approach using datasets derived from the cryptocurrency blockchain also exist as Pham & Lee (2016) who performed a K-means clustering algorithm to detect the most suspicious users, Bartoletti et al. (2018) who used data mining for detecting PonziSchemes, Harlev et al. (2018) who performed classification of the non-identified entities on the Bitcoin using various classical supervised learning methods and Bhowmik et al. (2021) that conducted a comparative study of supervised learning algorithms to detect fraud in the blockchain.

2.1 Resampling of Blockchain Data

Despite the promising results provided by the preceded studies, only a few of them have considered resampling techniques to address the class imbalance problem in the given datasets. The classification results of the blockchain datasets in (Bartoletti et al., 2018) and (Harlev et al., 2018) have shown a further improvement with the random undersampling/oversampling techniques and SMOTE technique, respectively.

Also, Bynagari & Ahmed (2021) have applied data sampling techniques to the Elliptic data (Weber et al., 2019) and the Ethereum account data (Farrugia et al., 2020) where the classification of the resampled data has revealed effective results on the data derived from the blockchain.

However, the preceded studies lack a comprehensive discovery of the wide range of the existing resampling techniques such as SMOTE-variants on the data derived from the blockchain. The class-imbalance problem can be tackled through oversampling by adding new instances, undersampling by removing noisy instances, or hybrid sampling as the combination of oversampling and undersampling methods. The main idea of oversampling is to increase the number of instances in the positive class near the decision boundary that is already subject to vast class skews. SMOTE is a well-known technique that blindly interpolates positive instances to address class imbalance (Chawla et al., 2002). Other SMOTE-variants also exist that are more guided than SMOTE e.g., borderline-SMOTE (Han et al., 2005) in which these variants take into consideration the informative areas near the decision boundary to generate new data points.

2.2 Feature Importance and Model's Explainability

The scarcity of the labelled datasets is a key challenge in the machine learning domain where the researchers have a piece of limited knowledge about the fraudulent accounts/transactions in the public blockchain and generally in the financial sector. On the other hand, explainable artificial intelligence (XAI) is an emergent research direction that assists the user in interpreting the predictions provided by the machine learning models (Kute et al., 2021).

The study by (Weber et al., 2019) has addressed the explainability of the machine learning predictions through visualisations to support anti-money laundering. Also, the study (Farrugia et al. 2020) has provided insights into the importance of all involved features to analyse the activity of the fraudulent accounts in the dataset derived from Ethereum.

Motivated by the preceded studies on the blockchain, we perform a comprehensive study using various oversampling (SMOTE and its variants) and undersampling techniques to address class imbalance on Bitcoin and Ethereum datasets that appeared in (Weber et al., 2019) and (Farrugia et al., 2020), respectively, as the largest labelled datasets in their relevant networks.

Since feature importance is one of the popular XAI techniques, we will study the effect of the resampled data on the feature importance which directly influences the explainability of the machine learning models.

3 Methods

In this section, we provide the necessary details of the experiments that are carried out using Bitcoin and Ethereum blockchain datasets. Firstly, we study the classification of these datasets after necessary feature reductions using various supervised learning algorithms as follows:

- Random Forest

- ExtraTrees
- Gradient Boosting
- XGBoost
- Logistic Regression
- Multi-Layer Perceptron (MLP)

Basically, a random forest chooses randomly a subset of features in order to construct a decision tree with the best split over its nodes, wherein multiple trees are formed to provide an ensemble of decision trees (Breiman, 2001). ExtraTrees algorithm is similar to the random forest that constructs a decision tree but with a random split over the nodes (Guerts et al., 2006). These bagging algorithms are known to reduce overfitting. XGBoost is an optimisation of gradient boosting algorithm (Chen & Guestrin, 2016). Gradient boosting is formed of a sequential number of trees as a weak classifier to obtain a strong classifier using gradient descent. Lastly, logistic regression and MLP are function approximations, where the former models a linear decision boundary to classify the data (Wright, 1995), whereas the latter handles non-linearly separated data (Gardner & Dorling, 1998). These learning methods have gained popularity in blockchain data, referring to (Weber et al., 2019; Farrugia et al., 2020; Alarab et al., 2020a). In what follows, we describe the necessary details of the datasets used to train the learning models, then we discuss the resampling techniques applied in our experiments. A schematic representation summarising the overall process in this paper is depicted in Figure 1.

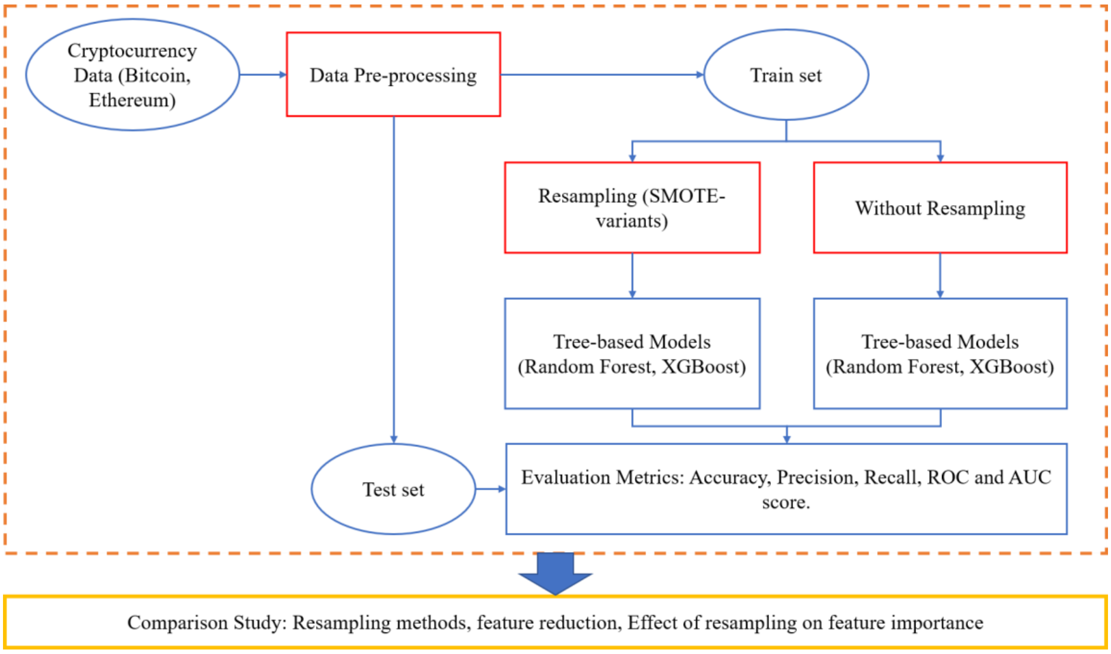
Cryptocurrency Data (Bitcoin, Ethereum)
Data Pre-processing
Train set
Resampling (SMOTE-variants)
Without Resampling
Tree-based Models (Random Forest, XGBoost)
Tree-based Models (Random Forest, XGBoost)
Test set
Evaluation Metrics: Accuracy, Precision, Recall, ROC and AUC score.
Comparison Study: Resampling methods, feature reduction, Effect of resampling on feature importance

Figure 1: Schematic representation of the method used in this study.

3.1 Data Preprocessing

3.1.1 Bitcoin Transaction Data

Elliptic data is one of the largest labelled available data derived from Bitcoin (Weber et al, 2019). Initially, this data is a subset of the Bitcoin transaction graph that comprises more than 203k nodes as transactions and 234k edges as the payments flow. This data acquires licit and illicit transaction labels as well as unknown labels. As we only consider the labelled transactions, the number of data points becomes 46,564 distributed as shown in Figure 2.

This data comprises 166-dimensional features that involve 94 local features belonging to Bitcoin transactions including timestamp (e.g., input degree, output degree...), and 72 aggregated features derived from the neighbouring nodes of the Bitcoin transaction graph. Moreover, there are 49 unique timestamps where each timestamp corresponds to a set of nodes belonging to a connected graph network that is extracted at a certain time from the blockchain. Since the features are anonymised, we refer to them as follows:

- First local feature: *timestamp*
- Remaining local features: *local_feat_2, local_feat_3, ..., local_feat_94*
- Aggregated features: *aggr_feat_1, aggre_feat_2, ..., aggre_feat_72*

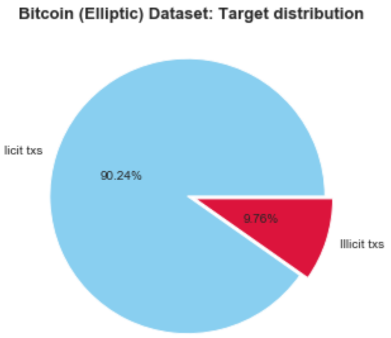

Figure 2: Target distribution of Bitcoin dataset.

We exclude the correlated features with a correlation coefficient greater than 0.9 chosen empirically, wherein the feature space is reduced to 91 features. An additional preprocessing step has been applied to the columns to remove the features with non-informative distributions. In other words, we empirically remove the features that have a number of unique values less than 10 as the case in local_feat_16 which acquires 6 unique values whereas most of the data points correspond to a single value as depicted in Figure 3. This eliminates further dimensions resulting in a dataset of 85 features. For further information about the used features, we refer the reader to view the correlation matrix after feature reduction that is represented in Figure 10 in Appendix A.

The dataset is then divided between train and test sets according to the temporal split in which the first 34 timestamps belong to the train set and the remaining 15 timestamps belong to the test set, to perform licit/illicit transaction classifications using supervised learning algorithms.

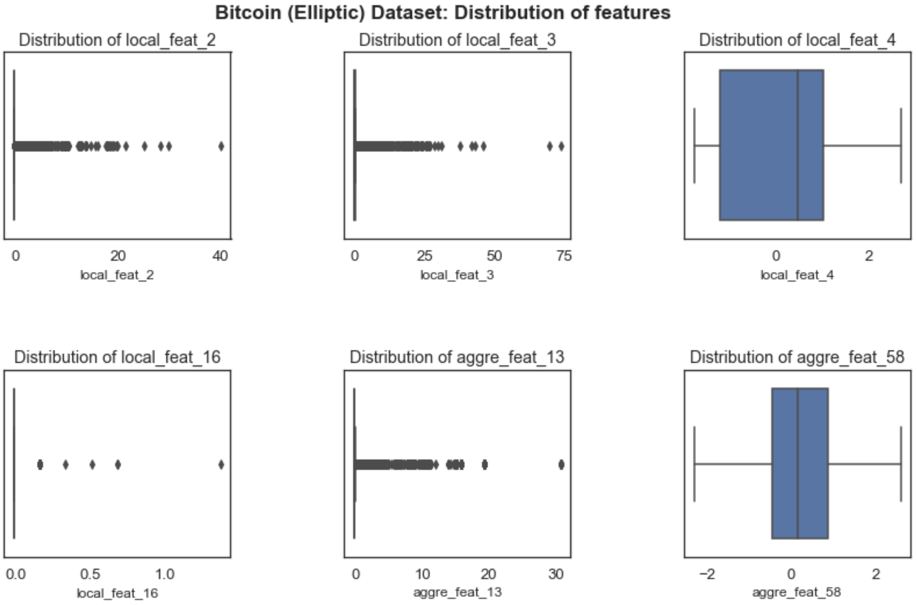

Figure 3: Boxplot of some features in Bitcoin dataset. Indication of how the features in the data are spread out.

3.2.2 Ethereum Account Data

This dataset comprises known fraud accounts and valid transaction history over the Ethereum blockchain extracted by a combination of two sources; a local Geth client and the Etherscamdb linked to the Ethereum network for normal and scam accounts respectively (Farrugia et al., 2020). The various accounts are labelled by the Ethereum community for illicit behaviour in several cases e.g., scams, Ponzi schemes and phishing. This dataset involves 9841 labelled accounts distributed as non-fraud/fraud in Figure 4 associated with 49 numerical and categorical features e.g., “total number of sent/received transactions” and “average value of ether ever sent”.

As this dataset includes some missing values in its features, these features such as categorical ones are disregarded in our experiments. Further feature reduction is applied by removing the correlated features whose correlations are greater than 0.9, chosen empirically, as well as features with zero variance. Moreover, another feature reduction is done by empirically removing the features with unique numerical values of less than 10 values as in the case of the feature "Distribution of max val sent to contract" depicted in Figure 5. Thus, the overall number of features of this dataset is reduced to 28. We refer the reader to Figure 11, in Appendix A, which summarises the used features in our experiment.

We split the dataset randomly after fixing the random seed to zero with a 70/30 split for train/test sets, respectively, to classify fraud accounts using supervised learning methods on the Ethereum account dataset.

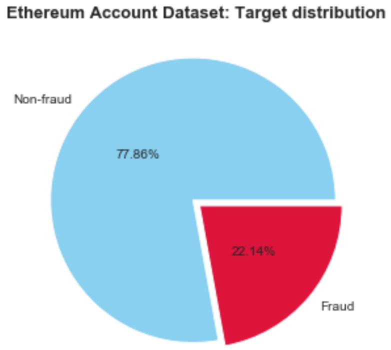

Figure 4: Target distribution of Ethereum account data.

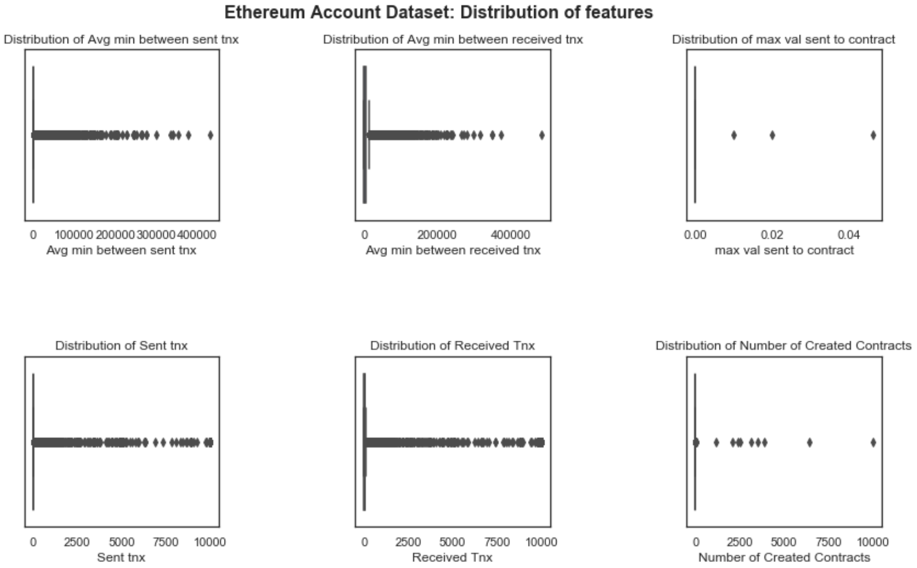

Figure 5: Boxplot of some features in Ethereum dataset. Indication of how the features in the dataset are spread out.

3.2 Resampling Methods

We have studied the effect of more than 80 resampling techniques on both datasets using SMOTE, its variants, and other recent resampling methods (see Appendix A); however, we pick the best performing techniques including SMOTE applied to both datasets as follows: Kmeans-SMOTE, AHC, Borderline-SMOTE1, Borderline-

SMOTE2, SOMO, SMOTE-TomekLinks, DEAGO, Safe-level-SMOTE, TRIM-SMOTE, CURE-SMOTE, LLE (refer to Kovács (2019a)) for a comprehensive overview) and the most recent techniques SMOTE-SF (SMOTE using subset features (Maldonado et al., 2019)) and OSCCD (Over Sampling-based Classification Contribution Degree (Jiang et al., 2021)). These techniques oversample new instances near the decision boundary in guided and more sophisticated ways than SMOTE. For instance, Kmeans-SMOTE is a combination of clustering algorithms and SMOTE, Borderline-SMOTE selects the most informative regions near the class boundary to oversample the minorities, and SMOTE-SF tackles high dimensional datasets by using SMOTE on a subset of features. Regarding undersampling, we include the ENN technique to remove noisy instances in overlapping distributions.

4 Experiments

4.1 Experimental Settings

In our experiments, we use sklearn (Pedregosa et al., 2011) and smote-variant packages (Kovács, 2019b) in Python programming language. Firstly, we train various supervised learning methods on Bitcoin and Ethereum datasets, wherein the hyper-parameters are empirically chosen in these models as summarised in Table 1. We evaluate supervised learning algorithms on these datasets using accuracy, f_1-score and AUC-score as provided in Table 2. Subsequently, we apply resampling methods to Bitcoin and Ethereum datasets, wherein we perform training and evaluations using the same supervised learning algorithm per dataset for a fair comparison.

Table 1: Hyper-parameters of the given models.

Dataset	Model	Hyperparameters
Bitcoin	Random Forest	Number of trees=50; max depth=50; max features=5
	ExtraTrees	Number of trees=50
	Gradient Boosting	Learning rate=0.1
	XGBoost	Number of trees=300; max depth = 50; learning rate=0.1
	Logistic Regression	C=10.0; epochs=50
	MLP	Optimiser='Adam'; hidden layer size=50, epochs=50
Ethereum	Random Forest	Number of trees=100
	ExtraTrees	Number of trees=100; max features=9
	Gradient Boosting	Number of trees=300; max depth=4; learning rate=0.1
	XGBoost	Number of trees=300; max depth=4; learning rate =0.1
	Logistic Regression	C=10; epochs=100
	MLP	Optimiser='Adam'; hidden layer size=50; epochs=100

Thus, we opt for the best performing algorithms which are random forest on the Bitcoin dataset to classify illicit transactions and XGBoost on Ethereum dataset to classify fraud addresses, referring to Table 2. Afterwards, we apply the abovementioned oversampling and undersampling methods to study the effect of the class-imbalance problem on these datasets.

To resample the datasets, we keep the default hyper-parameters for all oversampling methods except for the following methods which are empirically tuned:

- OSCCD: Number of clusters is set to 3.
- LLE: Number of components of the embedded feature space is set to 5.
- SMOTE-SF: Number of selected features is chosen 40 and 10 in Bitcoin and Ethereum datasets, respectively.

ENN is applied twice on both datasets. We distinguish between both ways by ENN and ENN-all. ENN corresponds to undersampling applied on the training set only, whereas ENN-all is applied to the whole dataset. The latter way allows us to provide more discussion regarding the noisy data points in the feature space. The experimental results using accuracy, precision, recall and f_1-score derived from different resampling techniques are tabulated in Tables 3 and 4 for Bitcoin and Ethereum datasets, respectively. Consequently, we plot ROC-AUC curves to analyse the goodness of classification with the resampled datasets as shown in Figures 6 and 7 for Bitcoin and Ethereum, respectively. We also compute the feature importance scores of the resampled datasets derived from ENN-all, SMOTE-SF and Kmeans-SMOTE resampling techniques that are arbitrarily chosen. We compare the most important features of the model derived from non-resampled dataset (i.e., represented by 'NoSMOTE') with other resampled datasets using the latter three resampling techniques. The feature importance scores are computed for the train and test sets of each of the Bitcoin and Ethereum datasets using the feature permutation method.

Table 2: Classification results of supervised learning models on Bitcoin and Ethereum datasets.

Dataset	Model	%Accuracy	%F_1-score	%AUC
Bitcoin	Random Forest	98.02	82.39	91.9
	ExtraTrees	97.84	80.34	92.4
	Gradient Boosting	96.79	74.3	89.9
	XGBoost	97.7	80.2	93.5
	Logistic Regression	88.33	41.72	87.6
	MLP	96.11	67.95	90.5
Ethereum	Random Forest	98.06	95.7	99.7
	ExtraTrees	97.76	94.99	99.7
	Gradient Boosting	98.47	96.63	99.8
	XGBoost	98.91	97.61	99.8
	Logistic Regression	79.58	20.96	70.5
	MLP	95.56	90.14	60.6

4.2 Evaluation and Comparison of Feature Importance

The feature permutation method (Breiman, 2001) shuffles the data of each feature to amass the prediction error with respect to a baseline model (i.e., the model with a non-shuffled dataset). The overall process is repeated several times to find the average of the importance of each feature. In our experiments, we perform feature permutation, using sklearn package (Pedregosa et al., 2011), with five repetitions to mitigate the biasedness caused by random shuffling. The feature importance on each of the train and test sets of the used datasets are depicted in Figures 8 and 9. As mentioned earlier, we arbitrarily choose four resampling techniques to study feature importance, however, the concept is viable with other resampling techniques and datasets. Moreover, we only visualise a set of six features (with the highest scores) since the visualisation of all features is quite large and non-informative. Moreover, we use the Wilcoxon signed-rank test (Wilcoxon, 1945) – a statistical method that tests the null hypothesis between two related paired samples derived from the same distribution. Using a paired sample test, the data can be expressed as: $\big(P(f_1), Q(f_1)\big)$, $\big(P(f_2), Q(f_2)\big), \ldots, \big(P(f_n), Q(f_n)\big)$, where n is the number of features, f_i is the feature at the i^{th}-dimension, $P(f_i)$ is the importance (i.e., scores) of the feature f_i on the resampled dataset using a certain resampling technique, and $Q(f_i)$ is the importance of the feature f_i using the original dataset which we refer to by 'NoSMOTE' as the baseline model.

The terms of the preceded expression can be replaced by the difference of scores as: $D_1, D_2, \ldots, D_n$, such that:

$$D_n = P(f_n) - Q(f_n) \tag{1}$$

Henceforth, the steps to perform the Wilcoxon test are listed as follows:

1. Find $|D_1|, |D_2|, \ldots, |D_n|$, where $|.|$ is the absolute value notation.
2. Arrange $|D_1|, |D_2|, \ldots, |D_n|$ in the increasing order.
3. Assign ranks to the sorted values in the step 2 as $R_1, R_2, \ldots, R_n$, where R_i is the rank corresponding to $|D_i|$ at feature i. The ranks are assigned such that the smallest $|D_i|$ correspond to rank 1, and the second smallest to rank 2 and so on.
4. Find the test statistic of the signed rank sum T as:
   $$T = \sum_{i=1}^{n} sign(D_i) R_i \ , \tag{2}$$
   Where sign(.) denotes the sign function that returns 1 when the input value is positive and -1 otherwise.
5. Find the p-value – probability value given that null hypothesis is true—by comparing the test statistic T to Student's t-distribution.

To test if the feature importance scores have changed after applying the resampling technique, we can formulate the hypothesis test as follows:

- Null Hypothesis (H_0): Feature importance scores (before resampling) = Feature importance scores (after resampling).
- Alternative Hypothesis (H_1): The importance of features is influenced by the resampling technique.

We then choose the values of α – the significance level – to be equal to 0.05. This value is an area in the t-distribution where we can reject the null hypothesis with confidence level of 95%. Consequently, the p-value smaller than the significance level α means that we have strong evidence against the null hypothesis, and we can accept the alternative one which states that the importance of features is influenced by the resampled technique. For instance, we refer to the Wilcoxon test of the feature importance after using SMOTE resampling technique as follows:

$$Wilcoxon(\text{SMOTE}, NoSMOTE),$$

Where $P(f)$ is derived from the feature importance after using SMOTE and $Q(f)$ is derived from the feature importance using the original dataset under the same model. We perform the Wilcoxon test for the resampling technique shown in Figure 8 and 9 for the Bitcoin and Ethereum datasets, respectively. The p-values of the Wilcoxon test are computed for the three resampling techniques in each dataset as tabulated in Table 6.

Table 3: Comparison between resampling techniques applied to Bitcoin dataset using random forest.

Bitcoin Dataset	%Accuracy	% Precision	% Recall	%F_1-score
ENN-all	99.42	99.31	89.10	93.93
NoSMOTE	98.02	97.96	71.09	82.39
Kmeans-SMOTE	98.02	97.96	71.09	82.39
LLE-SMOTE	98.02	97.96	71.09	82.39
DEAGO	98.02	97.96	71.00	82.33
ENN	98.01	99.34	69.89	82.05
AHC	97.96	96.38	71.37	82.01
CURE-SMOTE	97.95	96.96	70.72	81.79
Safe-Level-SMOTE	97.96	97.93	70.17	81.76
OSCCD	97.82	95.34	69.89	80.66
SMOTE-SF	97.65	90.13	71.74	79.89
TRIM-SMOTE	97.66	90.72	71.37	79.89
SMOTE	97.57	88.87	71.56	79.28
SOMO	97.66	97.01	66.02	78.57
SMOTE-TomekLinks	97.41	86.14	71.74	78.28
Borderline-SMOTE1	97.25	84.00	71.28	77.12
Borderline-SMOTE2	97.35	88.12	68.51	77.09

Table 4: Comparison between resampling techniques applied to Ethereum dataset using XGBoost.

Ethereum Dataset	%Accuracy	% Precision	% Recall	% F_1-score
ENN-all	99.38	98.75	97.93	98.34
NoSMOTE	98.91	99.09	96.17	97.61
SMOTE-SF	98.71	98.05	96.32	97.18
Kmeans-SMOTE	98.71	98.63	95.73	97.16
SMOTE	98.67	97.34	96.91	97.12
LLE-SMOTE	98.67	98.19	96.02	97.10
OSCCD	98.61	98.04	95.88	96.95
SOMO	98.57	98.33	95.44	96.86
DEAGO	98.54	98.33	95.29	96.78
AHC	98.54	98.47	95.14	96.78
CURE-SMOTE	98.51	97.74	95.73	96.73
Borderline-SMOTE1	98.40	97.02	96.02	96.52
SMOTE-TomekLinks	98.34	96.87	95.88	96.37
Borderline-SMOTE2	98.34	96.87	95.88	96.37
Safe-Level-SMOTE	98.06	96.01	95.58	95.79
TRIM-SMOTE	97.05	92.78	94.55	93.66
ENN	96.61	96.47	88.52	92.33

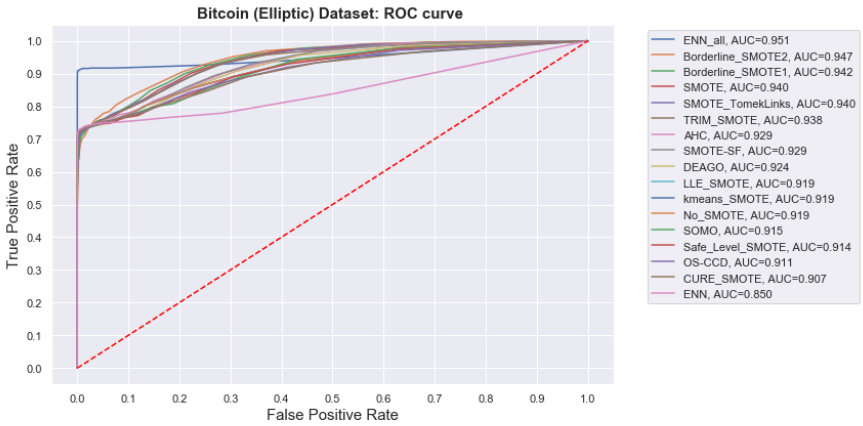

Figure 6: ROC-curve analysis of random forest with various data resampling methods in Bitcoin.

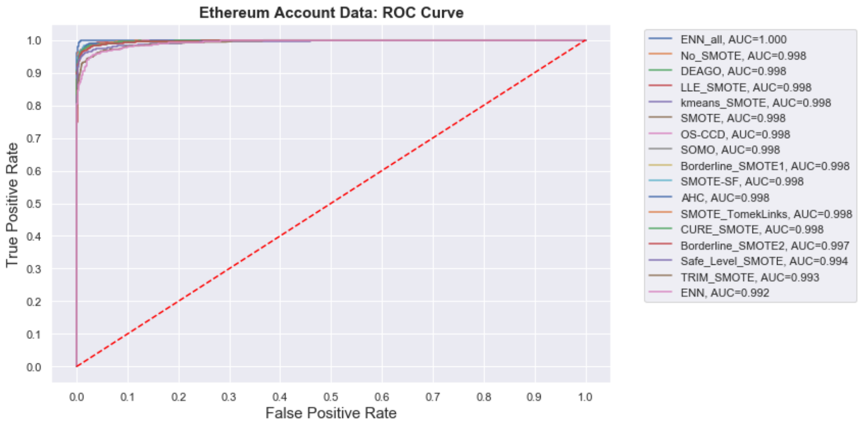

Figure 7: ROC-curve analysis of XGBoost with various data resampling methods in Ethereum.

Table 5: Comparison between our experiments and the original contribution of Bitcoin and Ethereum datasets. These tables highlight the effect of data preprocessing in our experiments.

Dataset	Methods	%Accuracy	%F_1-score
Bitcoin Dataset	Random Forest (Weber et al., 2019)	97.7	78.8
	Preprocessing + Random Forest (Ours)	**98.02**	**82.39**
Ethereum Dataset	Preprocessing+XGBoost (Farrugia et al., 2020)	96.3	96
	Preprocessing+XGBoost (Ours)	**98.91**	**97.6**

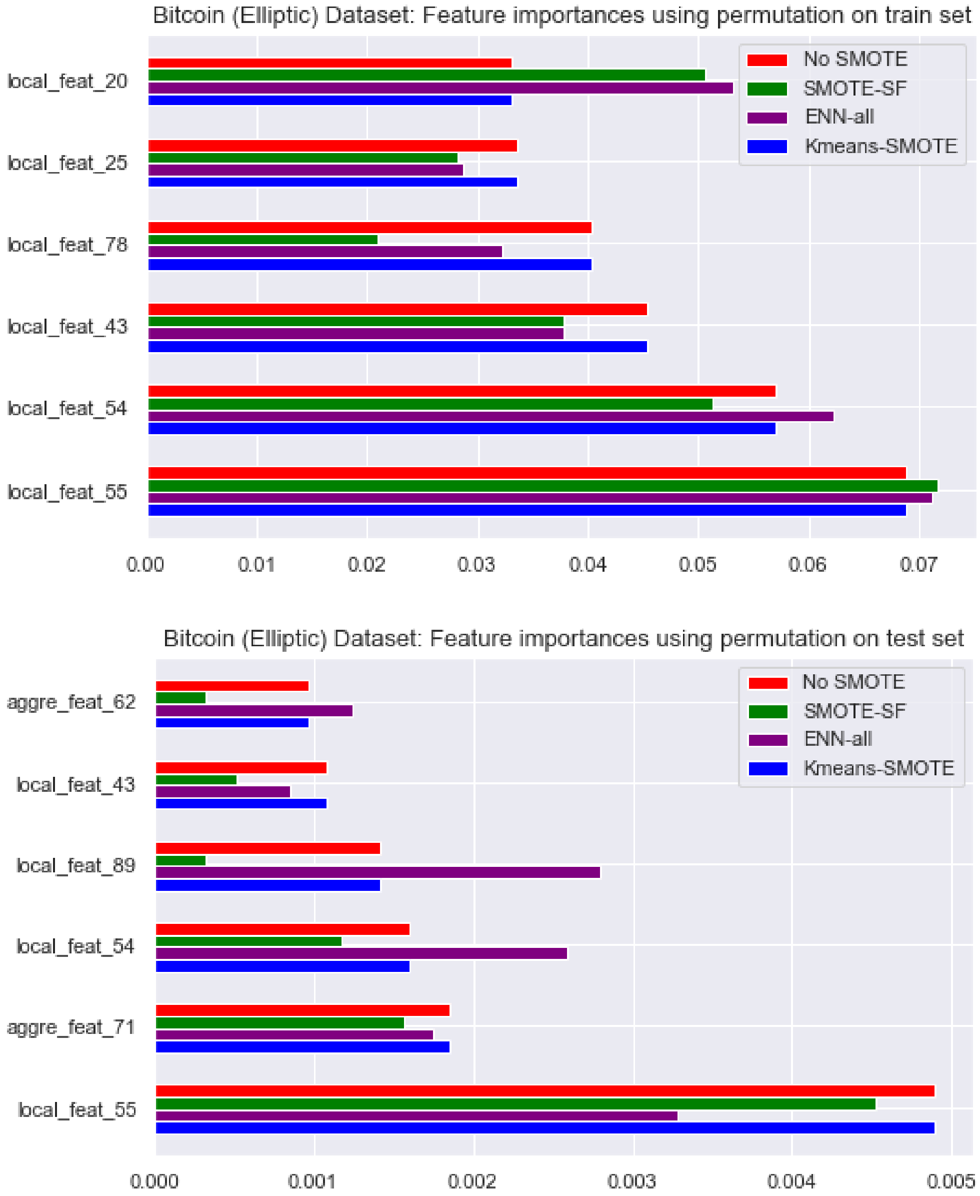

Figure 8: Effect of resampling techniques on feature importance in Bitcoin.

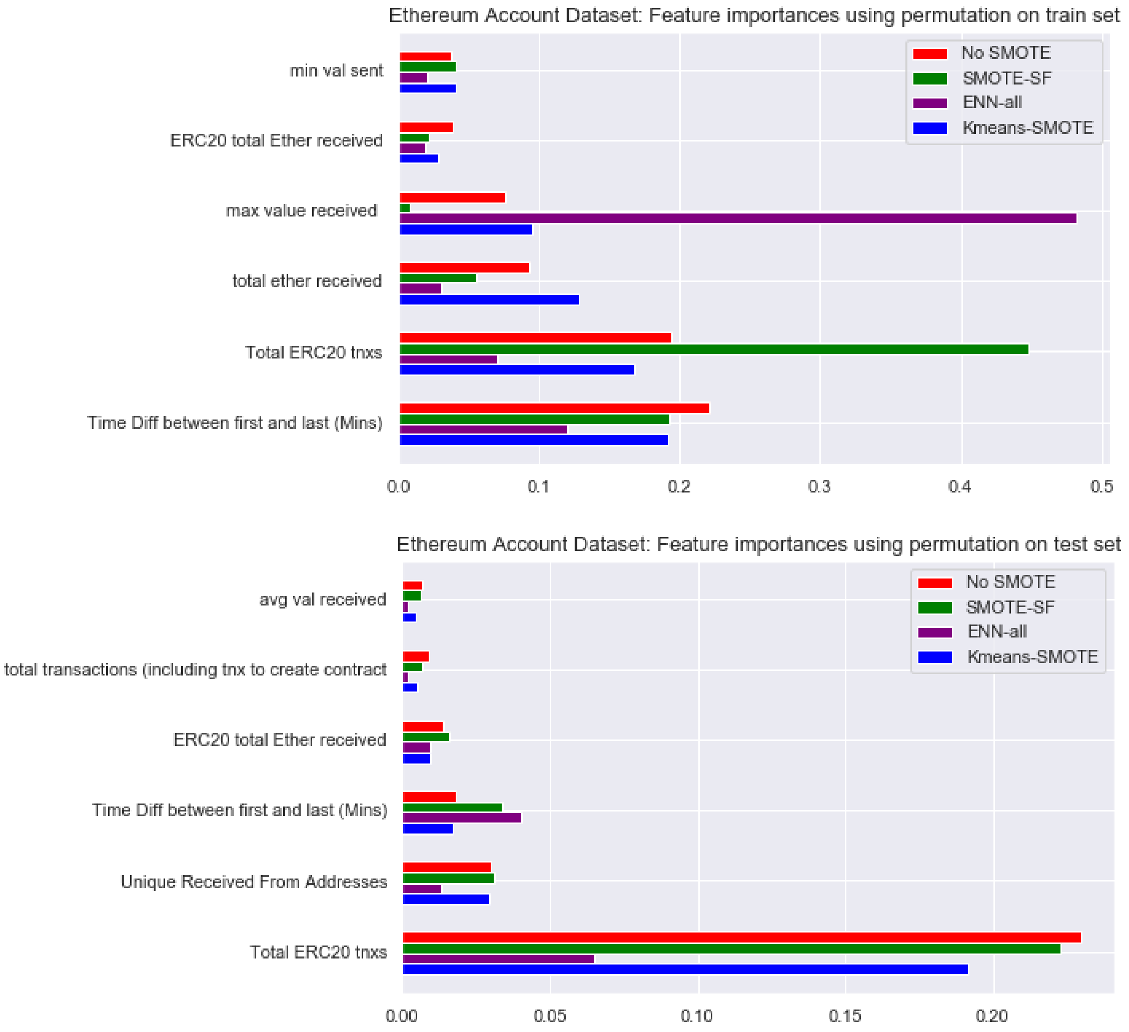

Figure 9: Effect of resampling techniques on feature importance in Ethereum.

5 Discussion

5.1 Results of Classifications and Resampling Techniques

Referring to Tables 3 and 4, ENN-all, undersampling method, has outperformed all other resampling techniques as well as the non-sampled data (NoSMOTE) on Bitcoin and Ethereum datasets. Regarding the Bitcoin dataset, the experimental results with ENN-all have shown a remarkable increase in the accuracy and f_1-score, respectively, from 98.02% to 99.42% and 82.39% to 93.93% in comparison to NoSMOTE. This increase explains the high number of noisy instances that are removed by ENN-all to provide a good decision boundary that works for the whole data. The remaining resampling techniques have revealed a trade-off between the number of false positives and false negatives, either in improving precision or recall referring to Table 3. In comparison to NoSMOTE, ENN has boosted the precision from 97.96% to 99.34% but this comes at the cost of decreasing recall from 71.09% to 69.89%. This happens because ENN has removed noisy instances that are derived from the illicit transactions on the train set resulting in fewer false positives. Oversampling methods have played a remarkable role in improving recall such as in the SMOTE-SF technique that attained a recall of value 71.74% wherein the train set is randomly oversampled on a particular subset of features. Regarding the Ethereum dataset, we highlight the slight increase from 98.91% to 99.38% for accuracy and from 97.61% to 98.34% for f_1-score using ENN-all undersampling technique as provided in Table 4. This slight increase in the model's performance illustrates the few noisy instances that already exist in the Ethereum dataset, in contrast to the Bitcoin dataset. Consequently, all resampling techniques on the Ethereum dataset have revealed good decision making due to a smaller number of noisy instances. SMOTE has recorded the highest recall on this data of value 96.91%. However, this reduces the precision from 99.09% to 97.34%. Accordingly, oversampling is not able to reduce the misclassified instances, while still able to provide a better classification rule by improving AUC scores by different oversampling techniques as depicted in ROC-curve analysis in Figures 6 and 7. Normally, oversampling influences the model's performance when the generated data lies near the decision boundary of the used model.

On the other hand, we highlight the outperformance of the supervised learning algorithms on Bitcoin and Ethereum datasets, respectively, after data preprocessing in comparison to their original works in (Weber et al., 2019) and (Farrugia et al., 2020). The random forest has attained an accuracy of 98.02% instead of 97.7% to classify the Bitcoin dataset using 85 features instead of 166 features. Similarly, data preprocessing on the Ethereum dataset has provided an increased performance with accuracy and f_1-score of 98.91% and 97.6%, respectively, referring to Table 5.

5.2 Feature Importance

We discuss the influence of resampling techniques on the feature importance using the given supervised learning models. Mainly, feature permutation method provides the highest scores for the most important features used by the classifier to perform predictions. Particularly, feature permutation method with the test set is tied with the explainability of the model's predictions.

For the Bitcoin dataset, random forest reveals different feature importance on train and test sets with different resampling methods referring to Figure 8. However, the feature "local_feat_55" has revealed the highest importance score which means that the latter feature plays an important role in providing decisions on the test set. We also notice that the local features on the Bitcoin dataset have appeared with higher importance than the aggregated ones.

For the Ethereum dataset, "Total ERC20 tnxs" feature has played an important role in the whole dataset using SMOTE-SF as shown in Figure 9, whereby this resampling technique oversamples a subset of features that are selected with the highest Fisher-score. Meanwhile, the feature "Time Diff between first and last (Mins)" reflects the total duration of account usage in Ethereum which reveals a high impact on the classification of fraud accounts.

5.3 Influence of Resampling Techniques on Feature Importance

As explainability of the models in this field is highly desirable, the change in feature importance caused by resampling methods affects the explainability of the model as it is highly tied with the feature importance. We verify this statement by performing the Wilcoxon test for the feature importance between the resampled and non-sampled datasets. This statistical method provides the p-values as revealed in Table 6. The p-values with less than 0.05 shows strong evidence to reject the null hypothesis and eventually accept the alternative one. For Bitcoin test set, the Wilcoxon test for the resampling technique SMOTE-SF has revealed a p-value equals to 0.001 which means that the test was statistically significant. For Ethereum train set, the hypothesis test for SMOTE-SF and ENN-all is statistically significant where we have evidence to reject the null hypothesis. For the Bitcoin train set and the Ethereum test set, there is no evidence to highlight the effect of the given resampling techniques on the feature importance referring to Table 6. In general, the difference of feature importance means that the data distribution is changed after applying the resampling methods using the same classification model. Furthermore, the model with the highest performance should produce more accurate feature importance and hence better explainability. This is reasonable because an explainable machine learning method seeks to interpret the predictions of a given model.

Table 6: Wilcoxon test for the feature importance between resampled and non-sampled datasets of Bitcoin and Ethereum using different resampling techniques.

Model Used	Dataset	Wilcoxon Test	P-values
Random Forest	Bitcoin Train Set	$Wilcoxon$(SMOTE-SF, No SMOTE)	0.995
		$Wilcoxon$(ENN-all, No SMOTE)	0.354
		$Wilcoxon$(Kmeans-SMOTE, No SMOTE)	0.999
	Bitcoin Test Set	$Wilcoxon$(SMOTE-SF, No SMOTE)	0.001
		$Wilcoxon$(ENN-all, No SMOTE)	0.227
		$Wilcoxon$(Kmeans-SMOTE, No SMOTE)	0.999
XGBoost	Ethereum Train Set	$Wilcoxon$(SMOTE-SF, No SMOTE)	0.013
		$Wilcoxon$(ENN-all, No SMOTE)	0.003
		$Wilcoxon$(Kmeans-SMOTE, No SMOTE)	0.564
	Ethereum Test Set	$Wilcoxon$(SMOTE-SF, No SMOTE)	0.061
		$Wilcoxon$(ENN-all, No SMOTE)	0.866
		$Wilcoxon$(Kmeans-SMOTE, No SMOTE)	0.259

6 Conclusion

Based on our conducted experiments, we have shown that random forest has performed the best in detecting illicit transactions in the Bitcoin dataset, whereas XGBoost has shown superior success in capturing fraud accounts in the Ethereum dataset. We have then studied the class-imbalance problem on these datasets by applying various resampling techniques (oversampling, undersampling and hybrid resampling). ENN-all, an undersampling technique, has provided the best performances on these datasets with an accuracy greater than 99%. Moreover, we have also provided the experimental results of other resampling techniques using accuracy, precision, recall, f_1-score and ROC-AUC score. As a result, oversampling techniques have improved the model's recall at the cost of its precision and vice-versa. Meanwhile, most oversampling methods have revealed a remarkable increase in AUC scores on the given datasets. We also claim the outperformance of the used models on Bitcoin and Ethereum datasets after data pre-processing in comparison to the results in their original contributions. On the other hand, we have also studied the effect of data resampling on feature importance. For that, we have used the feature permutation method to compute the feature importance of each of the used models on the train and test sets using Bitcoin and Ethereum datasets. The provided results have depicted changes in feature importance among different resampling techniques which influence the explainability of the model, where the model's explainability is more reliable with the high performing models. To show that resampling methods affect the feature importance, we have performed the Wilcoxon statistical method to test the statistical evidence to reject the null hypothesis which states that the feature importance scores remain the same before and after data sampling. For some resampling techniques, the test was statistically significant to reject the null hypothesis with confidence level 95%. This means that we have enough evidence to say that the feature importance scores are influenced by the resampling techniques under the given null hypothesis.

In this study, none of the oversampled data has shown better performance in terms of the model's accuracy. In future work, we will explore generative algorithms for data oversampling using artificial neural networks as well as study the model's explainability using other XAI techniques (e.g., local surrogate models) rather than the feature permutation method.

Funding

No funding to declare.

Conflict of Interest

The authors declare that they have no conflict of interest.

Ethics Approval

This paper does not contain any studies with human participants or animals performed by any of the authors

Consent to Participate

Not Applicable.

Consent for Publication

Informed consent was obtained from all individual participants included in the study.

Availability of Code and Material

Not Applicable.

Author's Contributions

Both authors have provided the conception and design of the study, acquisition of data, analysis and interpretation of data, drafting the article, revised it critically for important intellectual content, and final approval of the version to be submitted.

References

Akinnuwesi, B. A., Fashoto, S. G., Mbunge, E., Odumabo, A., Metfula, A. S., Mashwama, P., Uzoka, F.-M., Owolabi, O., Okpeku, M., & Amusa, O. O. (2021). Application of intelligence-based computational techniques for classification and early differential diagnosis of covid-19 disease. Data Science and Management, 4, 10–18. URL: https://www.sciencedirect.com/science/article/pii/S2666764921000473. doi:https://doi.org/10.1016/j.dsm. 2021.12.001.

Alarab, I., & Prakoonwit, S. (2021). Adversarial attack for uncertainty estimation: Identifying critical regions in neural networks. Neural Processing Letters, URL: https://doi.org/10.1007/s11063-021-10707-3. doi:10.1007/s11063-021-10707-3.

Alarab, I., Prakoonwit, S., & Nacer, M. I. (2020a). Comparative analysis using supervised learning methods for anti-money laundering in bitcoin. In Proceedings of the 2020 5th International Conference on Machine Learning Technologies ICMLT 2020 (p. 11–17). New

York, NY, USA: Association for Computing Machinery. URL: https://doi.org/10.1145/3409073.3409078. doi:10. 1145/3409073.3409078.

Alarab, I., Prakoonwit, S., & Nacer, M. I. (2020b). Competence of graph convolutional networks for anti-money laundering in bitcoin blockchain. In Proceedings of the 2020 5th International Conference on Machine Learning Technologies ICMLT 2020 (p. 23–27). New York, NY, USA: Association for Computing Machinery. URL: https://doi.org/10.1145/3409073.3409080. doi:10. 1145/3409073.3409080.

Alarab, I., Prakoonwit, S., & Nacer, M. I. (2021). Illustrative discussion of mc-dropout in general dataset: Uncertainty estimation in bitcoin. Neural Processing Letters, 53 , 1001–1011. URL: https://doi.org/10.1007/s11063-021-10424-x. doi:10.1007/s11063-021-10424-x.

Bartoletti, M., Pes, B., & Serusi, S. (2018). Data mining for detecting bitcoin ponzi schemes. In 2018 Crypto Valley Conference on Blockchain Technology (CVCBT) (pp. 75–84). doi:10.1109/CVCBT.2018.00014.

Bhowmik, M., Sai Siri Chandana, T., & Rudra, B. (2021). Comparative study of machine learning algorithms for fraud detection in blockchain. In 2021 5th International Conference on Computing Methodologies and Communication (ICCMC) (pp. 539–541). doi:10.1109/ICCMC51019.2021.9418470.

Breiman, L. (2001). Random forests. Machine Learning, 45 , 5–32. URL: https: //doi.org/10.1023/A:1010933404324. doi:10.1023/A:1010933404324.

Bynagari, N. B., & Ahmed, A. A. (2021). Anti-money laundering recognition through the gradient boosting classifier. Academy of Accounting and Financial Studies Journal , 25 , 1–11. URL: https://www.abacademies.org/articles/antimoney-laundering-recognition-through-the-gradient-boosting-classifier-12577.html.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16 , 321–357. URL: https://doi.org/10.1613/jair.953.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD ’16 (p. 785–794). New York, NY, USA: Association for Computing Machinery. URL: https://doi.org/10. 1145/2939672.2939785. doi:10.1145/2939672.2939785.

Fan, G., Deng, Z., Ye, Q., & Wang, B. (2021). Machine learning-based prediction models for patients no-show in online outpatient appointments. Data Science and Management, 2 , 45–52. URL: https: //www.sciencedirect.com/science/article/pii/S2666764921000175. doi:https://doi.org/10.1016/j.dsm.2021.06.002.

Farrugia, S., Ellul, J., & Azzopardi, G. (2020). Detection of illicit accounts over the ethereum blockchain. Expert Systems with Applications, 150, 113318. URL: https://www.sciencedirect.com/science/article/pii/S0957417420301433. doi:https://doi.org/10.1016/j.eswa.2020.113318.

Fernández, A., del R´ıo, S., Chawla, N. V., & Herrera, F. (2017). An insight into imbalanced big data classification: outcomes and challenges. Complex & Intelligent Systems, 3 , 105–120. URL: https://doi.org/10.1007/s40747-017-0037-9. doi:10.1007/s40747-017-0037-9.

Gardner, M., & Dorling, S. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric Environment, 32, 2627–636. URL: https: //www.sciencedirect.com/science/article/pii/S1352231097004470.

doi:https://doi.org/10.1016/S1352-2310(97)00447-0.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. Machine Learning, 63, 3–42. URL: https://doi.org/10.1007/s10994-006-6226-1. doi:10.1007/s10994-006-6226-1.

Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-smote: a new over-sampling method in imbalanced data sets learning. In International conference on intelligent computing (pp. 878–887). Springer. URL: https: //doi.org/10.1007/11538059_91.

Harlev, M., Sun Yin, H., Langenheldt, K., Mukkamala, R., & Vatrapu, R. (2018). Breaking bad: De-anonymising entity types on the bitcoin blockchain using supervised machine learning. In Proceedings of the 51st Hawaii International Conference on System Sciences 2018 Proceedings of the Annual Hawaii International Conference on System Sciences (pp. 3497–3506). United States: Hawaii International Conference on System Sciences (HICSS). URL: http://www.urbanccd.org/events/2018/1/3/hawaii-international-conference-on-system-sciences-hicss-51 the 51st Hawaii International Conference on System Sciences. HICSS 2018, HICSS2018 ; Conference date: 03-01-2018 Through 06-01-2018.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21 , 1263–1284. doi:10.1109/TKDE.2008.239.

Ibrahim, R. F., Mohammad Elian, A., & Ababneh, M. (2021). Illicit account detection in the ethereum blockchain using machine learning. In 2021 International Conference on Information Technology (ICIT) (pp. 488–493). doi:10.1109/ICIT52682.2021.9491653.

Jiang, Z., Pan, T., Zhang, C., & Yang, J. (2021). A new oversampling method based on the classification contribution degree. Symmetry, 13 . URL: https: //www.mdpi.com/2073-8994/13/2/194. doi:10.3390/sym13020194.

Kovács, G. (2019a). An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets. Applied Soft Computing, 83 , 105662. doi:10.1016/j.asoc.2019.105662. (IF-2019=4.873).

Kovács, G. (2019b). Smote-variants: A python implementation of 85 minority oversampling techniques. Neurocomputing, 366 , 352–354. URL: https: //www.sciencedirect.com/science/article/pii/S0925231219311622. doi:https://doi.org/10.1016/j.neucom.2019.06.100.

Kute, D. V., Pradhan, B., Shukla, N., & Alamri, A. (2021). Deep learning and explainable artificial intelligence techniques applied for detecting money laundering–a critical review. IEEE Access, 9 , 82300–82317. doi:10.1109/ACCESS.2021.3086230.

Liu, X. F., Jiang, X.-J., Liu, S.-H., & Tse, C. K. (2021). Knowledge discovery in cryptocurrency transactions: A survey. IEEE Access, 9 , 37229–37254. doi:10.1109/ACCESS.2021.3062652.

Lorenz, J., Silva, M. I., Apar´ıcio, D., Ascens˜ao, J. a. T., & Bizarro, P. (2020). Machine learning methods to detect money laundering in the bitcoin blockchain in the presence of label scarcity. In Proceedings of the First ACM International Conference on AI in Finance ICAIF ’20. New York, NY, USA: Association for Computing Machinery. URL: https://doi.org/10. 1145/3383455.3422549. doi:10.1145/3383455.3422549.

Makki, S., Assaghir, Z., Taher, Y., Haque, R., Hacid, M.-S., & Zeineddine, H. (2019). An experimental study with imbalanced classification approaches for credit card fraud detection. IEEE Access, 7 , 93010–93022. doi:10.1109/ACCESS.2019.2927266.

Maldonado, S., L ́opez, J., & Vairetti, C. (2019). An alternative smote oversampling strategy for high-dimensional datasets. Applied Soft Computing, 76, 380–389. URL: https://www.sciencedirect.com/science/article/pii/S1568494618307130. doi:https://doi.org/10.1016/j.asoc.2018.12.024.

Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G. M., & Savage, S. (2013). A fistful of bitcoins: Characterizing payments among men with no names. In Proceedings of the 2013 Conference on Internet Measurement Conference IMC '13 (p. 127–140). New York, NY, USA: Association for Computing Machinery. URL: https://doi.org/10.1145/2504730.2504747. doi:10.1145/2504730.2504747.

Oh, S., Lee, M. S., & Zhang, B.-T. (2011). Ensemble learning with active example selection for imbalanced biomedical data classification. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 8 , 316–325. doi:10.1109/TCBB.2010.96.

Ostapowicz, M., & ̇Zbikowski, K. (2020). Detecting fraudulent accounts on blockchain: A supervised approach. In International Conference on Web Information Systems Engineering (pp. 18–31). Springer. URL: https://doi.org/10.1007/978-3-030-34223-4_2.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12 , 2825–2830.

Pham, T., & Lee, S. (2016). Anomaly detection in bitcoin network using unsupervised learning methods. arXiv preprint arXiv:1611.03941.

Reid, F., & Harrigan, M. (2013). An analysis of anonymity in the bitcoin system. In Security and privacy in social networks (pp. 197–223). Springer. URL: https://doi.org/10.1007/978-1-4614-4139-7_10.

Sun, X., Yang, T., & Hu, B. (2022). Lstm-tc: Bitcoin coin mixing detection method with a high recall. Applied Intelligence, 52 , 780–793. URL: https://doi.org/10.1007/s10489-021-02453-9. doi:10.1007/s10489-021-02453-9.

Tasharrofi, S., & Taheri, H. (2021). De-gcn: Differential evolution as an optimization algorithm for graph convolutional networks. In 2021 26th International Computer Conference, Computer Society of Iran (CSICC) (pp. 1–6). doi:10.1109/CSICC52343.2021.9420542.

Verbiest, N., Ramentol, E., Cornelis, C., & Herrera, F. (2014). Preprocessing noisy imbalanced datasets using smote enhanced with fuzzy rough prototype selection. Applied Soft Computing, 22, 511–517. URL: https://www.sciencedirect.com/science/article/pii/S1568494614002506. doi:https://doi.org/10.1016/j.asoc.2014.05.023.

Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Bellei, C., Robinson, T., & Leiserson, C. E. (2019). Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. arXiv preprint arXiv:1908.02591 , .

Wilcoxon, F. (1945). Individual comparisons by ranking methods. biometrics bulletin 1, 6 (1945), 80–83. URL http://www. jstor. org/stable/3001968 , .

Wright, R. E. (1995). Logistic regression. (1995). In L. G. Grimm & P. R. Yarnold (Eds.), Reading and understanding multivariate statistics (pp. 217–244). American Psychological Association. URL: https://psycnet.apa.org/record/1995-97110-007.

Zhang, Y., & Trubey, P. (2019). Machine learning and sampling scheme: An empirical study of money laundering detection. Computational Economics, 54, 1043–1063. URL: https://doi.org/10.1007/s10614-018-9864-z. doi:10.1007/s10614-018-9864-z.

Appendix A

Many resampling methods have been tested on the given datasets derived from Bitcoin and Ethereum blockchain. We apply various collection of oversampling, undersampling and hybrid sampling techniques on these datasets. The studied techniques are mostly implemented in smote-variants package in Python programming language as follows:

- Oversampling techniques: SMOTE, Borderline-SMOTE1, Borderline-SMOTE2, ADASYN, AHC, LLE-SMOTE, distance-SMOTE, SMMO, polynom-fit-SMOTE, Stefanowski, ADOMS, Safe-Level-SMOTE , MSMOTE , DE-oversampling, SMOBD, SUNDO, MSYN, SVM-balance, TRIM-SMOTE , SMOTE-RSB , ProWSyn, SL-graph-SMOTE, NRSBoundary-SMOTE , LVQ-SMOTE , SOI-CJ , ROSE, SMOTE-OUT, SMOTE-Cosine, Selected-SMOTE, LN-SMOTE, MWMOTE, PDFOS, IPADE-ID, RWO-sampling , NEATER, DEAGO, Gazzah, MCT, ADG, SMOTE-IPF, KernelADASYN, MOT2LD, V-SYNTH, OUPS, SMOTE-D, SMOTE-PSO, CURE-SMOTE, SOMO, ISOMAP-Hybrid, CE-SMOTE, Edge-Det-SMOTE, CBSO, E-SMOTE, DBSMOTE, ASMOBD, Assembled-SMOTE, SDSMOTE, DSMOTE, G-SMOTE, NT-SMOTE, Lee, SPY , SMOTE-PSOBAT, MDO, Random-SMOTE, ISMOTE , VIS-RST, GASMOTE , A-SUWO, SMOTE-FRST-2T, AND-SMOTE, NRAS, AMSCO, SSO, NDO-sampling , DSRBF, Gaussian-SMOTE, Kmeans-SMOTE, Supervised-SMOTE, SN-SMOTE, CCR, ANS, and cluster-SMOTE. In addition to all these oversampling techniques, OSCCD and SMOTE-SF are also used that are not implemented in the given package.
- Undersampling techniques: TomekLinkRemoval, CondensedNearestNeighbors, OneSidedSelection, CNNTomekLinks, NeighborhoodCleaningRule and EditedNearestNeighbors.
- Hybrid sampling techniques: SMOTE-TomekLinks and SMOTE-ENN.

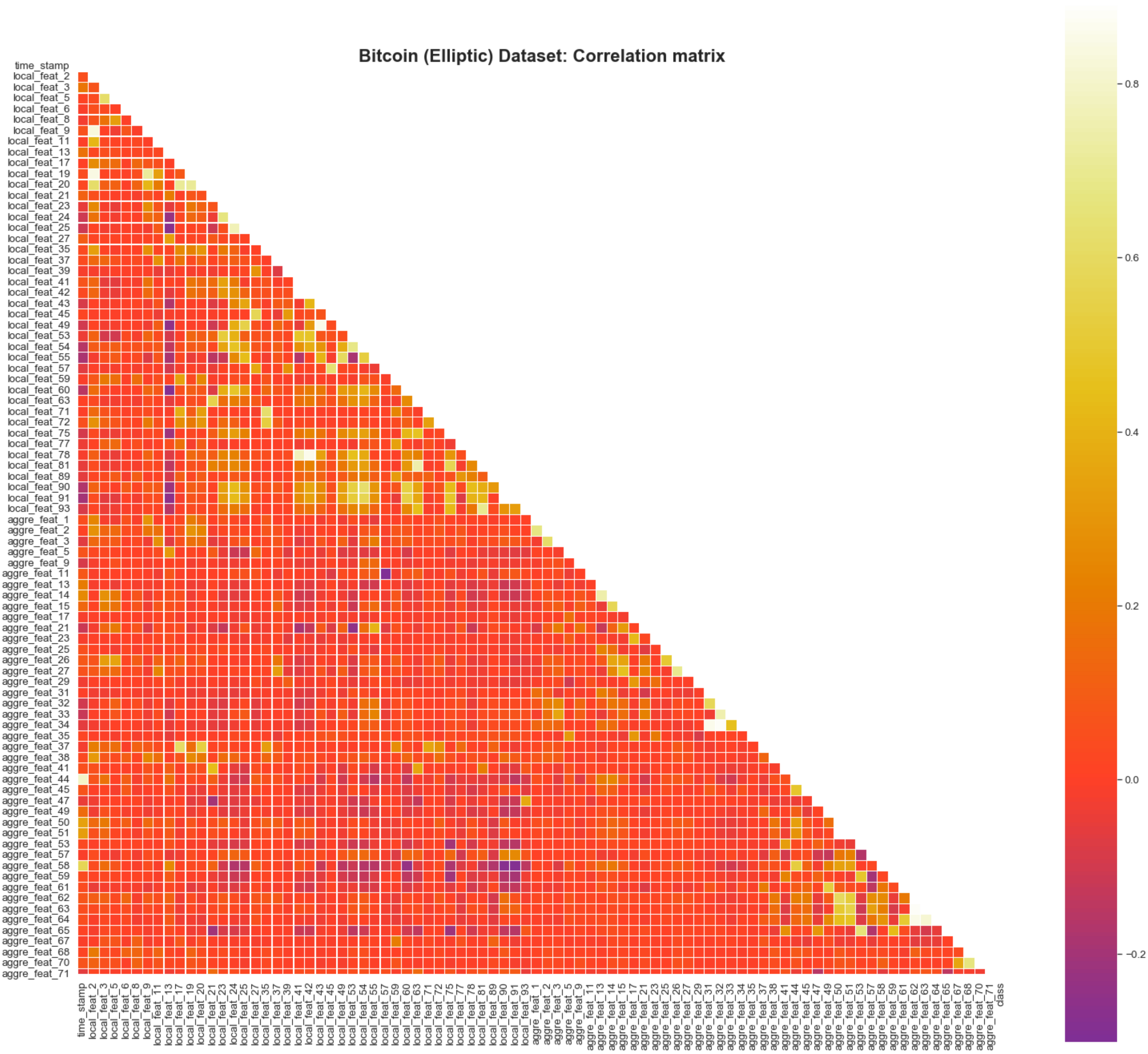

Figure 10: Correlation matrix of preprocessed features of Bitcoin dataset.

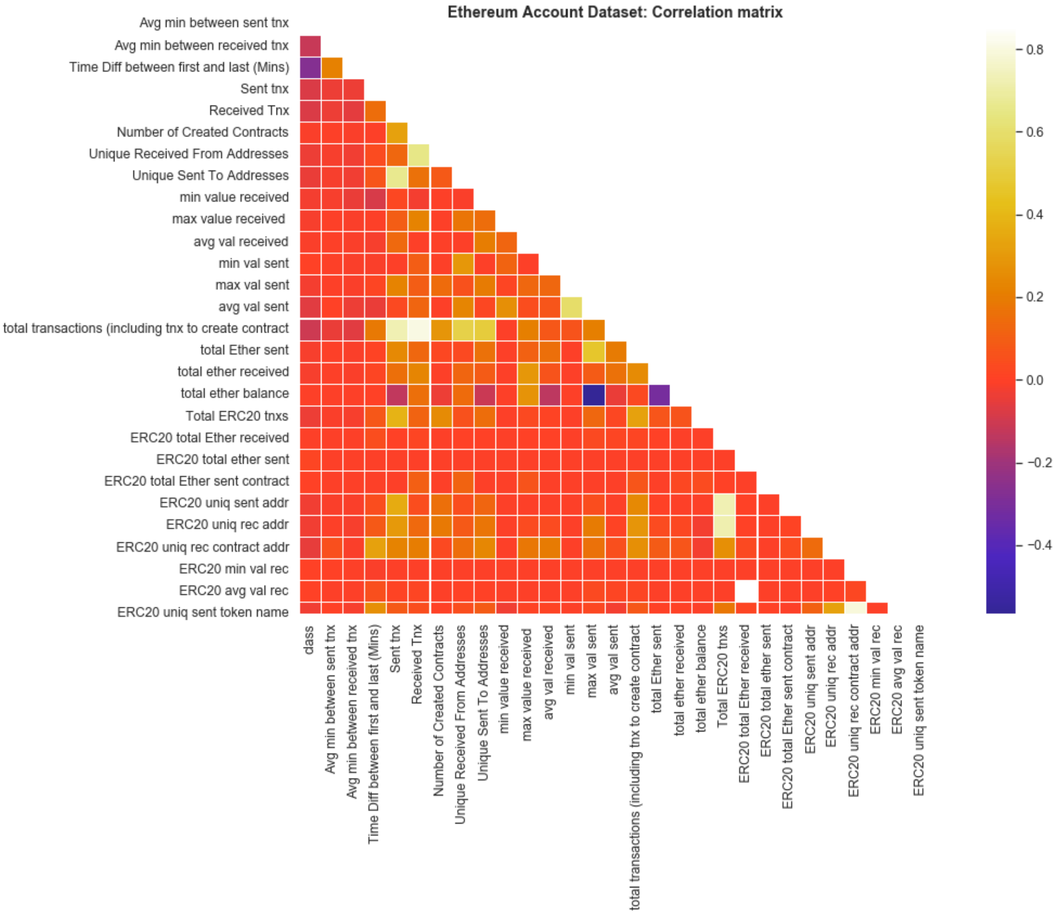

Figure 11: Correlation matrix of preprocessed features of Ethereum dataset.