Fraudulent Transaction Prediction in Bitcoin Network

Bhavay Aggarwal
2018384
bhavay18384@iiitd.ac.in

Prasham Narayan
2018359
prasham18359@iiitd.ac.in

Saad Ahmad
2018409
saad18409@iiitd.ac.in

Abstract

Bitcoin is a cryptocurrency which is a popular method for transactions. Among all the transactions that take place some of them are maybe used as malware attacks or as ransoms. This project interests us because it gives us the possibility to work on graph-based models alongside conventional machine learning models. On this project, we can try a wide variety of Machine learning strategies because of the number of features available.

1. Introduction

Bitcoin is a digital currency created following the housing market crash. The identity of the person or persons who made the technology is still a mystery. Bitcoin offers the promise of lower transaction fees than traditional online payment mechanisms and is operated by a decentralised authority, unlike government-issued currencies. As the earliest cryptocurrency to meet widespread popularity and success, Bitcoin has inspired a host of other projects in the blockchain space. Among all the transactions that take place, some of them are maybe used as malware attacks or as ransoms. Analysing illicit transactions and identifying characteristics which would lead to early identification of these transactions is an essential security measure.

2. Literature Survey

2.1. Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics

This is the baseline paper for the Elliptic Dataset where the authors have described the dataset and have mentioned the purpose for the release of the dataset. The authors have said that the dataset is temporal, and thus all the transactions should not be considered equivalent. The authors have also done Machine Learning on the dataset and have reported f1-scores timeline-wise. Another interesting methodology which they have provided is the use of Graph Convolutional Network and its variant EvolveGCN (which takes into account the temporality of the dataset) in predicting the illicit transactions. Apart from this, the authors have provided a timeline-wise analysis of the dataset where they mention an abnormality in the dataset which occurs after time-step 43, they have attributed this phenomenon to the dark market crash in bitcoin network. The authors also propose and implement a novel UMAP based visualisation software which visualises the graph taking into account the temporality.

2.2. Anomaly Detection in Bitcoin Network Using Unsupervised Learning Methods

The authors used three unsupervised learning methods on the graph generated by the Bitcoin transaction network. The three unsupervised learning methods are k- means clustering, Mahalanobis distance, and Unsupervised Support Vector Machine. Since the dataset that they had was quite large, 6 million users with 37 million transactions, therefore they parsed the data in two graphs, i.e. user graph and transaction graph. The user graph has users as nodes and transactions as edges between the edges, whereas the transaction graph has transactions as nodes and the Bitcoin flow between transactions as edges. For each method authors calculated the ratio of detected anomaly distances to corresponding centroids over max distances from those centroids to their assigned points for the top 100 outliers. For the Mahalanobis method, they got 0.76 for user graph and 0.82 for transaction graph whereas for the Unsupervised SVM method they got 0.72 for user graph and 0.85 for transaction graph. These large values showed that the anomalies appeared to be on the extreme points. In the end, the authors were able to detect some cases of theft and losses.

3. Dataset Description

This anonymised data set is a transaction graph collected from the Bitcoin blockchain. A node in the graph represents a transaction; an edge can be viewed as a flow of Bitcoins between one transaction and the other. Each node has 166 features and has been labelled as being created by a "licit", "illicit" or "unknown" entity. The graph is made of 203,769 nodes and 234,355 edges. 2% (4,545) of the nodes are labelled class1 (illicit). 21% (42,019) are labelled

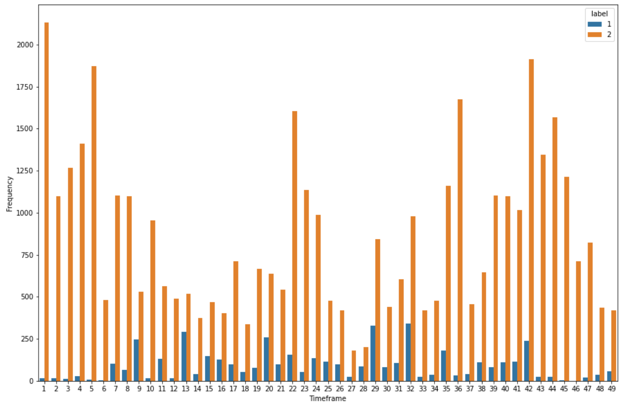

Figure 1. Time-slot wise distribution of licit and illicit transactions

class2 (licit). The remaining transactions are not labelled with regard to licit versus illicit. For this project, we would be covering the labelled part on the dataset with supervised learning techniques.

There are 166 features associated with each node. There is a time step associated with each node, representing a measure of the time when a transaction was broadcasted to the Bitcoin network. The time steps, running from 1 to 49, are evenly spaced with an interval of about two weeks. The first 94 features represent local information about the transaction, such as the average BTC received, the average number of incoming (outgoing) transactions. The remaining 72 features are aggregated features, obtained using transaction information one-hop backwards/forward from the centre node - giving the maximum, minimum, standard deviation and correlation coefficients of the neighbour transactions for the same information data.

3.1. Graph Analysis

- Number of nodes: 203769
- Number of edges: 234355
- Average degree: 2.3002
- Density: 1.12883410569188834e-05
- Average Clustering: 0.013762190724244798

From the above stats, we observe that the graph is very sparse as the density is very less. Moreover, the average clustering is also very less, which shows that the clustering present among the nodes in the graph is also very less.

3.2. Preprocessing

Rows with all None values were removed, and Labels were changed to numerical values: 1 for illicit, 2 for licit and 3 for unknown. Transactions with “unknown” labels were removed from the dataset.

Columns which were meant for identification of a node, such as a node id, time step were removed. To handle outliers and for normalisation, we scaled each feature such that their mean is 0 and the standard deviation is 1. Collinearity in features was detected and feature vectors showing $> 95\%$ collinearity with other feature vectors were removed.
Additionally, we removed the rows which had columns with values having a $z - score > 5$. Z-score for a column is defined as

$$Z = \frac{x - \mu}{\sigma}$$

A $z - score < 5$ would mean that all values of that column are within 5 standard deviations of the mean. Interestingly, All the transactions have at least one value which has a $z - score > 3$ for the respective column.

4. Methodology

Dataset obtained after preprocessing was used for PCA and TSNE analysis, but both were unable to differentiate the two classes.

We trained various machine learning models, and the performances of some of them can be seen in the results section. We have tried SVM, Logistic Regression, XGBoost, Random Forest and Multi-layer perceptron. We also tried the models with different combinations of features. Weber et. al. [1] have mentioned that the local features sometimes give better results when compared to the total features, however in our experiments we did not observe such phenomenon.
The dataset has a lot of imbalance and to overcome the issue, which is causing models to perform poorly, we have implemented a few techniques as described below.

1. Undersampling: We undersampled the licit class by removing random observations from licit class.
2. Gaussian Mixture Model: GMM is an unsupervised learning algorithm which is similar to K-nearest neighbors but instead uses Expectation maximization to estimate normal distributions which fit the data. It can be used as a generative model once normal distributions have been estimated. To prevent overfitting Bayesian Information Criterion parameter is used to estimate the number of components required. We found the optimal number of components to be 6 $fig.2$.
3. SMOTE: SMOTE is a widely used algorithm for oversampling which oversamples the minority class by using an approach similar to K-nearest neighbor. We, however, found SMOTE to be inferior to GMM when it comes to estimating probability distribution and sampling from it.

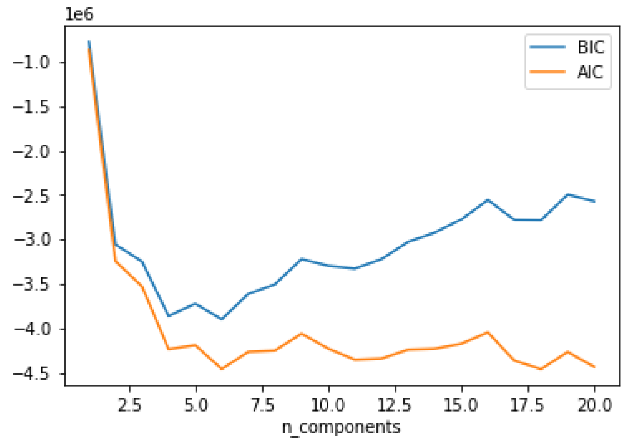

Figure 2.

	F1(1)	F1(2)	Rec(1)	Rec(2)	Acc
$LogReg$	0.00	0.99	0.00	1.00	0.98
SVM	0.06	0.89	0.25	0.82	0.81
RF	0.09	0.81	0.71	0.68	0.68
$GS\ SVM$	0.05	0.90	0.22	0.84	0.82
$GS\ RF$	0.09	0.78	0.76	0.64	0.64
MLP	0.07	0.80	0.69	0.65	0.66

We also perform unsupervised learning on the unlabelled transactions. We have used K-nearest neighbor (k = 2) and Gaussian Mixture Models ($n\_components = 2$) for this purpose. We label the smaller cluster as illicit, because the illicit transactions are lesser in the bitcoin network. On the clustered data points we perform classification using Random Forest.
Furthermore, we also analyzed the provided graph using networkx library to test a hypothesis which is 'Do bitcoins generally flow through longer chain of illicit transactions'.

5. Results and analysis

5.1. Random Forest

Random Forest performs the best for classification task. Among all the models we tried, Random Forest unequivocally gives the best results. This is also confirmed by the findings of Weber et. al. [1]. Following are the results:

SVM performed the best in terms of accuracy but was not able to predict illicit transactions which can be owed to the heavy class imbalance in the dataset. Random Forest performed better on predicting illicit transactions but also predicted many false positives. Incidentally since the dataset is concerned with Anti-money laundering, the main purpose of this exercise should be to minimize the false negatives and a tight threshold on the false positives.

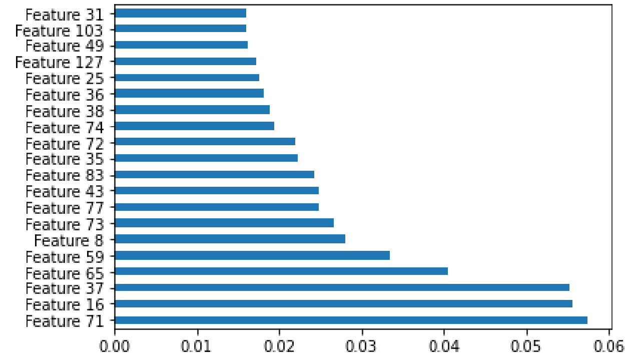

Figure 3.

We extracted the features which are the most important for the random forest model.

Upon trying to build a classifier with the top 20 features extracted as in fig. 2, we ended up with worse results than the original.

5.2. Undersampling and Oversampling

Naive undersampling led to inferior results compared to without undersampling, hence we chose to discard it. Oversampling using GMM vs SMOTE: We used t-SNE plots to compare the probability capturing ability of both the models $fig.4$.

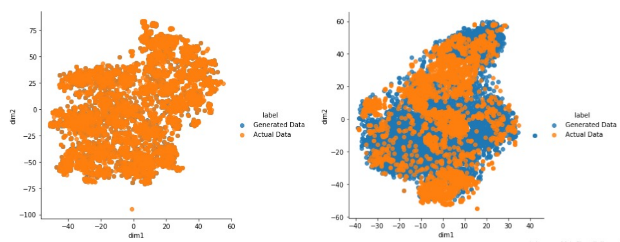

Figure 4.

SMOTE tends to overfit to the data which would mean that it is trying to replicate the samples. However, it appears that there is a better capture of the probability distribution in case of GMM. Augmenting the original matrix with the generated samples from illicit class resulted in amazing results.

	precision	recall	f1-score
1.0	0.99	0.82	0.90
2.0	0.91	0.99	0.95

Accuracy: 0.93

However, more testing is required with data augmentation methods, specifically Train Synthetic Test Real methodology needs to be adopted.

5.3. Clustering

The t-SNE plot of the unlabelled dataset is given in the left plot of $fig.5$. As can be seen from the t-SNE plot, the dataset does not show very clear demarcations among the two classes. We performed GMM clustering and KNN clustering both of which were used in the further pipeline to build a classifier. We assigned the smaller cluster to the illicit class because of the nature of the datest. The classification performance resulting from using RF classifier on the clustered dataset resulted in a considerably worse performance where the AUC was less than that of random classification.

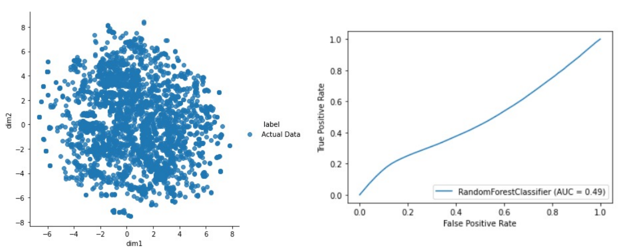

Figure 5.

5.4. Experiments on the nature of the graphs

We analyzed the graph at certain time steps and found that a lot of illicit transactions (marked in blue in the $fig.6$) consisted of clusters tightly bound together. Now, the nodes represent transactions while edges refer to the flow of coins. We hypothesize that illicit transactions generally are part of a larger flow of illicit transactions. This can somewhat be confirmed by looking at the visualizations of graphs with only the illicit transactions $fig.7$. A firm statistical test ought to provide more insights in our hypothesis which we leave as work to be done.

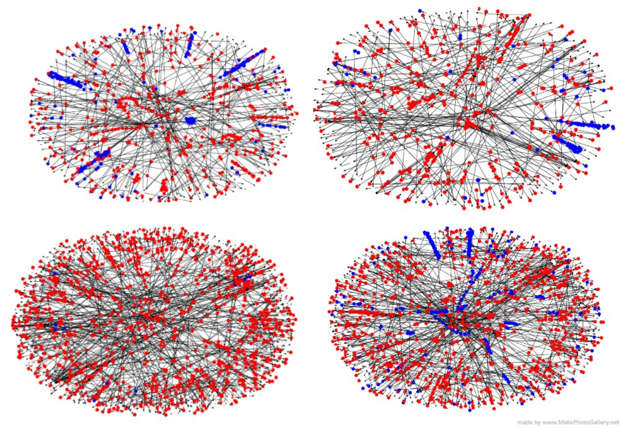

Figure 6.

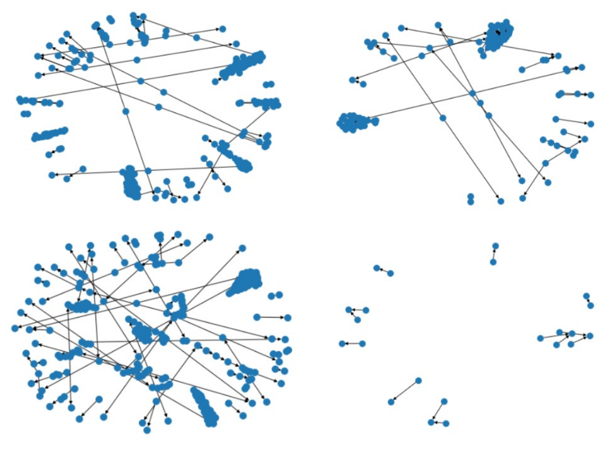

Figure 7.

6. Conclusion

We have applied a variety of machine learning techniques to the dataset so far. Oversampling helped us to achieve our best accuracy and the synthetic data generated using GMM has a good distribution capture compared to the original data. There is temporality in the data which ought to be explored in detail. Also, the graph structure provides more insights from an abstraction point which should be explored in the context of connected components. Since Graph Convolution Networks are beyond the scope of this course, our future work would focus on extracting embeddings from the graphs using different methods such as DeepWalk, EvolveGCN and train the embeddings on machine learning models. Also, we would work on the explainability of the results we have obtained so far.

Individual Contributions:

- Bhavay: EDA, Training Classifiers and t-SNE visualizations
- Prasham: Literature Review, Graph Analysis and Documentation using LaTeX
- Saad: Methodology, Oversampling and Undersampling and Clustering

6.1. References

[1] Weber, M. et al. (2019) 'Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics', arXiv:1908.02591 [cs,

q-fin].

[2] Pham, T. and Lee, S. (2017) ‘Anomaly Detection in Bitcoin Network Using Unsupervised Learning Methods’, arXiv:1611.03941 [cs].