APPLIED MACHINE LEARNING REPORT
Predicting fraudulent transactions in cryptocurrency trading

The university of Adelaide
Manik Marwaha (a1797063)

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, I 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 decentralized 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. Analyzing 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 (GCN) 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 [5] and its variant Evolve GCN (which takes into account the temporality of thedataset) [1] in predicting the illicit transactions. Apart from this, the authors have provided a timeline-wise analysis of the dataset where they mention an ab- normality in the dataset which occurs after time-step43, they have attributed this phenomenon to the darkmarket crash in bitcoin network. The authors also propose and implement a novel UMAP based visualization software which visualizes the graph taking into account the temporality.

2.2. Anomaly Detection in Bitcoin Network Using Unsupervised Learning Methods

The authors [2] 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, there- fore 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 theedges, 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 Ire able to detect some cases of theft and losses.

3. Dataset Description

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

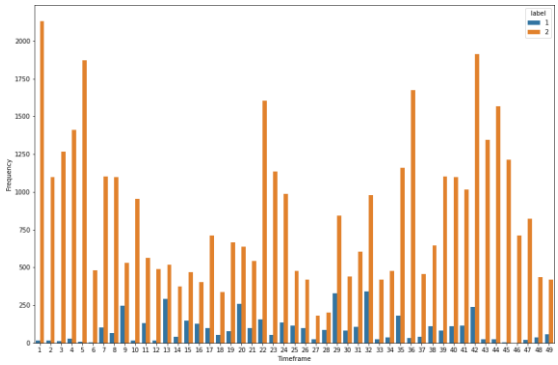

Figure 1: Time-slot wise distribution of licit and illicit transactions

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 rep- resent 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 center node - giving the maximum, mini- mum, standard deviation and correlation coefficients of the neighbor transactions for the same information data.

3.1. Graph Analysis

- Number of nodes: 203769
- Number of edges: 234355
- Average degree: 2.3002
- Density: 1.1288341056918834e-05
- Average Clustering: 0.013762190724244798

From the above stats, I 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 Ire removed, and Labels Ire changed to numerical values: 1 for illicit, 2 for licit and 3 for unknown. Transactions with "unknown" labels Ire removed from the dataset.

Columns which Ire meant for identification of a node, such as a node id, time step Ire removed. I tried different preprocessing strategies to handle outliers and for normalization such as scaling each feature such that their mean is 0 and the standard deviation is 1. Detecting Collinearity in features and removing feature vectors showing > 95% collinearity with other feature vectors.

These two-preprocessing resulted in worse performances. I hypothesize this may be because of the nature of the dataset where I do not have information about the nature of the features. Additionally, I removed the rows which had columns with values having a $z\ score > 5$. Z-score for a column is defined as

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

A $z\text{-}score < 5$ 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 $z\text{-}score > 3$ for the respective column.

4. Methodology

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

I trained various machine learning models, and the performances of some of them can be seen in the

results section. I have tried SVM, Logistic Regression, XGBoost, Random Forest and Multi-layer perceptron. Further, grid search was applied on these models to achieve the best score. I 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 I did notobserve such phenomenon.
The dataset has a lot of imbalances and to overcome the issue, which is causing models to perform poorly, I have implemented a few techniques as described below.

1. Under sampling: I perform stratified under-sampling on the two classes by removing random observations from to create test and train sets.

2. Gaussian Mixture Model [7]: 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 In- formation Criterion parameter is used to estimate the number of components required. I found the optimal number of components to be6 *fig*.2.

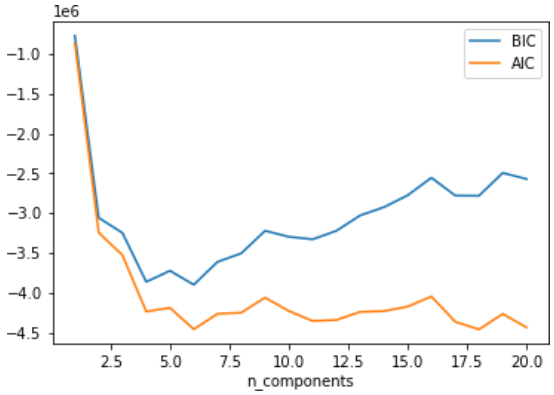

Figure 2: Bayesian information criterion used to prevent overfitting in GMM

3. SMOTE [8]: SMOTE is a widely used algorithm for oversampling which oversamples the minority class by using an approach similar to K-nearest neighbor. I, however, found SMOTE to be inferior to GMM when it comes to estimatingprobability distribution and sampling from it.

4. CTGAN [4]: I also explore CTGAN (Conditional GAN for Tabular Data), a GAN based synthesizer developed explicitly for generating tabular data. It is built upon TGAN and uses several tricks to improve upon it, for preprocessing CTGAN uses Variational Gaussian Mixture Model to detect models of continuous columns. CTGAN uses fully connected networks and to prevent model collapse and stabilize learningit uses WGAN-GP (WGAN Gradient Penalty, which uses Wasserstein distance).

I also perform unsupervised learning on the unlabelled transactions. I have used K-nearest neighbor (k = 2) and Gaussian Mixture Models ($n\_components = 2$) for this purpose. I label the smaller cluster as illicit, because the illicit transactions are lesser in the bitcoin network. On the clustered data points, I perform classification using Random Forest.
Moreover, I generated node embeddings using DeepWalk and trained Machine learning models on them. Furthermore, I 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 I tried, Random Forest unequivocally gives the best results when the features provided by the dataset Ire used. This is also confirmed by the findings of Weber et. al. [1].

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.
I 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*.3, I ended up withworse results than the original. Thus, I set Stratified

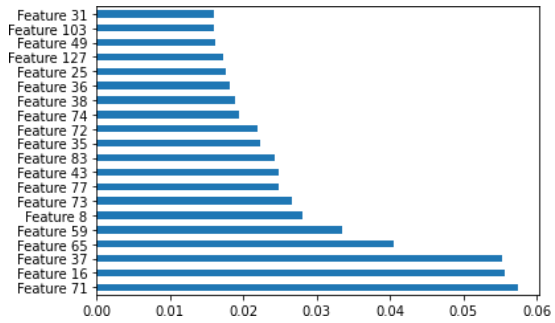

Figure 3: Feature Importance of Random Forest

sampling based Random Forest (Hyperparameter selection done using Grid Search)

5.2. Under sampling and Oversampling

Stratified sampling led to superior results compared to without under sampling which is expected because the class imbalance issue gets partially resolved. Oversampling using GMM vs SMOTE: I used t-SNE plots to compare the probability capturing ability of both the models *fig*.4.

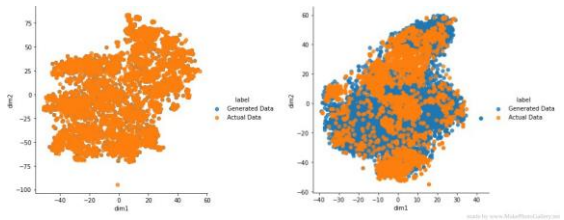

Figure 4: t-SNE plots for SMOTE and GMM

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.

I also report that GMM and ctGAN Ire used for synthesizing only the illicit class. Therefore, I are aware that if the synthesizers failed to capture the probability distribution and synthesized samples which are significantly different from the original population, then there is an inherent bias that has been introduced in the dataset, where samples from 1 class are significantly different from the other class. Thus, since the population distribution from CTGAN is significantly different from the original population, the results for the same might be biased.

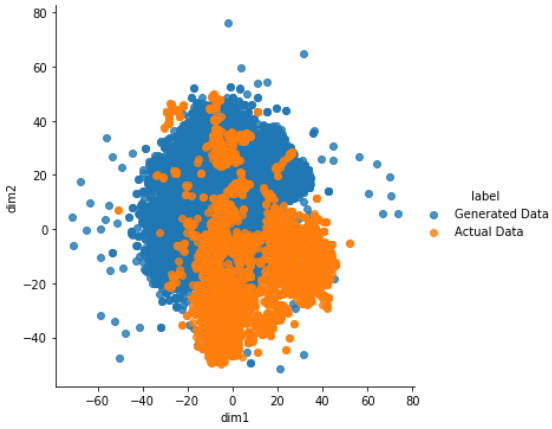

Figure 5: t-SNE plot for ctGAN

5.3. Clustering

The t-SNE plot of the unlabeled dataset is given in the left plot of *fig*.6. As can be seen from the t-SNE plot, the dataset does not show very clear demarcations among the two classes. I performed GMM clustering and KNN clustering both of which were used in the further pipeline to build a classifier. I assigned the smaller cluster to the illicit class because of the nature of the dataset. 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.

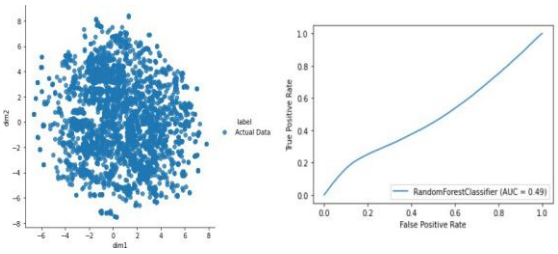

Figure 6: Results from KNN clustering

5.4. Experiments on the nature of the graphs

I analyzed the graph at certain time steps and found that a lot of illicit transactions (marked in blue in the *fig*.7) consisted of clusters tightly bound together. Now, the nodes represent transactions

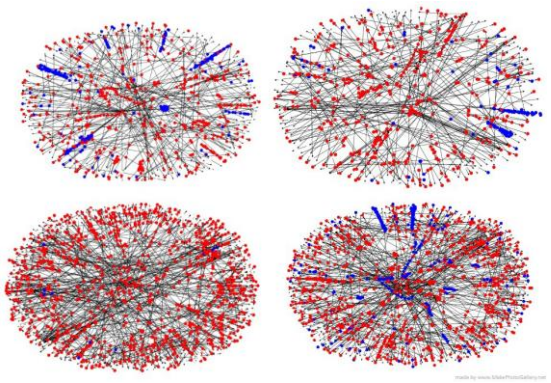

Figure 7: Graph visualization of different timesteps, illicit vs licit

while edges refer to the flow of coins. I 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*.8. I also hypothesize that these clusters are unique to illicit transactions only. I also hypothesize that these clusters are unique to illicit transactions only. These are manifested in the embeddings I obtained using Deep Walk, leading to better separability between the classes.

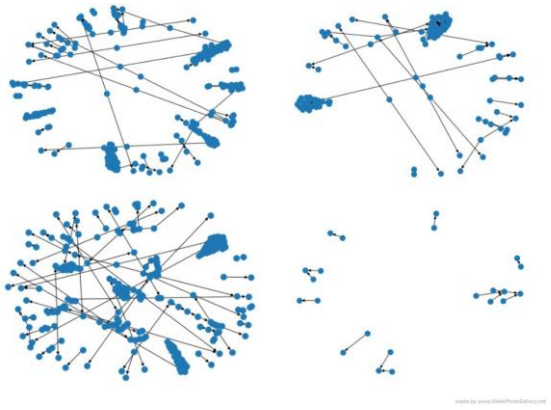

Figure 8: Graph visualization of different timesteps, illicit

The dataset is temporal in nature. It is known that there was a dark shutdown that occurred in the later timesteps. Thus, to explore this property I trained a Random Forest model for n-1 timesteps and tested on the nth timestep, this is done n in the range 2, 47. I have plotted illicit F1 score vs timestep in the *fig*.9 The graph shows that there is a major dip in the illicit F-1 score showing that there is a significant amount of unpredictability introduced in the dataset after the 33rd timestep.

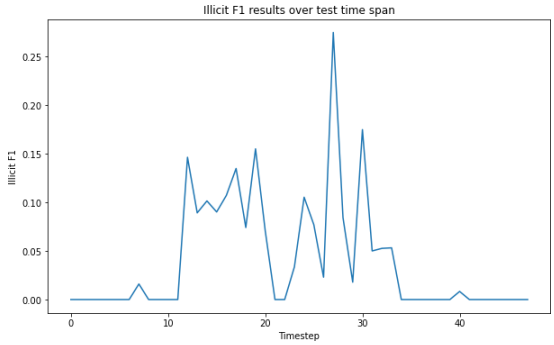

Figure 9: Illicit F1 results over test time span

5.5. DeepWalk

Deepwalk is a novel approach to learn representation of graphs by walking on vertices. [3] As hypothesized earlier, illicit transactions generally are part of a larger flow of illicit transactions. This would mean that using Graph Convolution or by generating representations of graphs, I could further prove our hypothesis. Since, GCN's Ire out the scope, I used DeepWalk.

DeepWalk uses local information from its random walks to learn representations of the graph. The random walks can be thought of as small sentences of phrases. These random walks are essential in capturing local community information in the graph which ultimately give a better understanding of the features and the neighborhood of a node. The DeepWalk algorithm then uses the information retrieved to generate the embeddings of the graph.

Embeddings generated by DeepWalk Ire trained on SVM and Random Forest. SVM gave an accuracy of 95% which is the highest accuracy I have received.

Figure 10: DeepWalk algorithm

Results of Different Classifiers

Classifiers	F1 Score(1)	F1 Score(2)	Recall(1)	Recall(2)	Accuracy
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
DW RF	0.21	0.93	0.76	0.87	0.87
DW SVM	0.4	0.97	0.73	0.95	0.95
OS CTGAN RF	0.94	0.95	0.90	0.98	0.94
OS GMM RF	0.90	0.95	0.82	0.99	0.93
OS SMOTE	0.21	0.93	0.16	0.95	0.88
FTRS RF	0.07	0.80	0.69	0.65	0.66
PCA RF	0.10	0.80	0.72	0.69	0.68

6. Conclusion

I have applied a variety of machine learning techniques to the dataset so far. Synthetic data generated using GMM has a good distribution capture compared to the original data. There is temporality in the data which is explored using n-1 train and n test method. Also, the graph structure provides more insights from an abstraction point which I have explored in the context of connected illicit transactions but there still are possibilities which could yield a deeper understanding to the nature of the transactions. DeepWalk helped us to achieve our best results and further our hypothesis on the nature of illicit transactions. Although, Random Forest performed better in terms of F1 score, graph features proved to be better identifiers of illicit transactions and SVM performed much better on the embeddings because the data is sparse making it harder for the Random Forest model to find distinctions in the features.

7. Knowledge Learnt from this project

In thus section, I will conclude what I learnt by working on this project.

Firstly, over the duration of this course, I was required to present different stages of my project to lecturers and the audience. This helped me a lot to improve my presentation skills as I am more confident now talking in front of an audience.

Secondly and most importantly, I learnt a lot about machine learning. Mostly before this project, I had worked on machine learning projects that required the use of classical supervised and unsupervised machine learning models. But since the data for this project was in the form of graph, I was able to combine my existing knowledge with graph-based neural network models. During the pre-processing stage, I worked with NetworkX library that helped me to conduct graph analysis and understand the requirements of this graph. Since there was a high-class imbalance, I was also presented with an opportunity to work with different under sampling and over sampling techniques like stratified under sampling, GMM, SMOTE and CTGAN that I had never worked with before. This project also helped me to have a deeper understanding about correctly choosing and using different evaluation metrics. Earlier I wasn't very clear with this concept. Finally, I got a chance to work graph-based models like DeepWalk that helped me to find node embeddings or node features for nodes withing the graph. I also got a brief introduction about Graph Neural Network (GCN) models during the literature review as it was used by Weber et al [1]. Several other papers [9,10,11] that I read to understand the graph-based models showed how important they are in many different domains like social networks, recommendation systems, node classification and more.

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].

[3] Perozzi, Bryan, et al. "DeepWalk: On-line Learning of Social Representations." Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '14, 2014, pp.701–10.arXiv.org, doi:10.1145/2623330.2623732.

[4] Xu, Lei, et al. "Modeling Tabular Data Using Conditional GAN." ArXiv:1907.00503 [Cs, Stat], Oct. 2019. arXiv.org, http://arxiv.org/abs/1907.00503.

[5] Chiang, W.L., Liu, X., Si, S., Li, Y., Bengio, S. and Hsieh, C.J., 2019, July. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 257-266).

[6] Dataset - https://www.kaggle.com/ellipticco/elliptic-data-set

[7] Rasmussen, C.E., 1999, November. The infinite Gaussian mixture model. In *NIPS* (Vol. 12, pp. 554-560).

[8] 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*, pp.321-357

[9] Santos, L, Piwowarski, B, Denoyer, L & Gallinari, P 2018, 'Representation Learning for Classification in Heterogeneous Graphs with Application to Social Networks', *ACM Transactions on Knowledge Discovery from Data*, vol. 12, no. 5, pp. 1–33.

[10] Zhang, C., Swami, A. and Chawla, N.V., 2019, January. Shne: Representation learning for semantic-associated heterogeneous networks. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining* (pp. 690-698).

[11] Yang, C., Xiao, Y., Zhang, Y., Sun, Y. and Han, J., 2020. Heterogeneous Network Representation Learning: A Unified Framework with Survey and Benchmark. *IEEE Transactions on Knowledge and Data Engineering*.

[12] https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/