Anomaly Detection Using Unsupervised Learning In Blockchain Network

Mrinal Anand
IIT Gandhinagar
mrinal.anand@iitgn.ac.in

Abstract

Bitcoin is a cryptocurrency that features a distributed, decentralized and trustworthy mechanism, which has made Bitcoin a popular global transaction platform. The transaction efficiency among nations and the privacy benefiting from address anonymity of the Bitcoin network has attracted many activities such as payments, investments, gambling and even money laundering in the past decade. Unfortunately, some criminal behaviors which took advantage of this platform are very hard to detect due to its anonymity. These actions has discouraged many governments to support cryptocurrency. Thus, the capability to identify anomaly in the Bitcoin network has become an important issue to address. In financial network, thieves and illegal activities are often regarded as anomalous in the nature. Many Machine Learning techniques have been proposed to deal with this problem. In this challenge I will explore unsupervised techniques on the Bitcoin transaction network. Our goal is to detect which users and transactions are the most suspicious; in this case, anomalous behavior is a proxy for suspicious behavior. To summarize I generated two Bitcoin transaction graphs one graph has users as nodes, and the other has transactions as nodes and used unsupervised techniques like k-means, Isolation Forest, Local Outlier Factor.

1 Introduction

Network structures have appeared for a long time, and along with them are those who behave abnormally within the system. We can refer to these people or their illegal activities as anomalies. With respect to financial transactional networks, anomalies can include those who execute fraudulent transactions. In these networks, a common goal is to detect those anomalies to prevent future illegal actions.
In this project, I am seeking to detect anomalies or suspicious activities in this anonymous network. I used four unsupervised learning techniques including k-means clustering, DBSCAN, Isolation Forest, Local Outlier Factor on two graphs generated by the Bitcoin transaction network; one graph has users as nodes, and the other has transaction as nodes.

2 Data Set and Prepossessing

2.1 Data Collections and Cleaning

I used the Bitcoin transaction data set provided by "ELTE Bitcoin Project".All Bitcoin transactions are documented in a public ledger and are in the currency unit called the Bitcoin (BTC). The data set contains all Bitcoin transactions beginning from the networks creation until Dec 28th, 2013. For each transaction, there can be multiple sender and receiver addresses. Furthermore, multiple addresses can belong to single user. Finally, users are also anonymous

in that there are no names or personal information associated with a given user.

The dataset is large with 277443 blocks, 24698959 address in the database and total of 30048983 transactions happened in the network. I created two graphs out of the raw data, the first is called **User Graph** - users (where each user owns a list of addresses) are nodes and transaction between users are edges.The second way, which we will call the **Transaction Graph** - models transactions as nodes and Bitcoin flow between transactions as edges.
Both the graphs are useful in detection of suspicious behaviour. The user graph will give suspicious users involved in transaction and transaction graph will give suspicious transaction involved.

2.2 Feature Extraction

Feature engineering is very important part of machine learning pipeline. For each node in the graph we extract a set of features. In the graph user of the data, we extract the following In-degree, Out-degree, Balances of each node in BTC. For Transaction graph, the set of selected features are In-degree, Out-degree, net-incoming BTC, net-outgoing BTC.
Note - The timestamp of each transaction might not prove to be useful feature but Average time interval of the transactions can be useful feature but unfortunately we don't have that feature in dataset.

3 Unsupervised Learning

3.1 k-Means Clustering

The purpose of the k-means methods is to partition m points(i.e m nodes in the graph) into k groups of similar characteristics. Technically speaking, k-means clustering itself is not a method for anomaly detection; however, it can be useful. Because we expect outliers to stay far away from the centroids found by k-means.

For this method to work, we first represent each node as a multi-dimensional vector in the Euclidean space;
This method produces a set of m points $(x_1, x_2, ..., x_m)$ in which $x_i \in \mathbb{R}^n$ (where $n = 4$or3 depending on graphs) for each $i = 1, ..., m$. We seek to partition these m points into k clusters $S = (S_1, ..., S_k)$ to solve

$$\min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2,$$

where μ_i is the mean of the points in S_i for each $i = 1, ..., k$. We use the k-means clustering algorithm as a heuristic method to solve this problem. Our final note in this part is that we used different types of normalization techniques e.g. Min-Max Scaler, Log Normalization, but the normalized log seems to work better than any other normalization. In case of Min-Max Normalization, since the feature have high variance scaling it to [0,1] yields features that are very close to 0 or very close to 1, which in-turns harms the model.
After plotting the cross-cluster entropy we set the k=7 user graph and k=8 for the transaction graph.

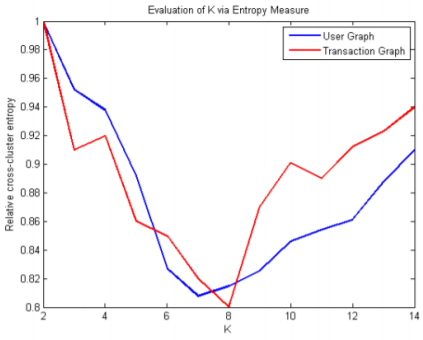

Figure 1. Cluster Entropy vs. k

Used **K-means++** which selects initial cluster centers for k-mean clustering in a smart way to speed up convergence.

3.2 Isolation Forest

Isolation Forest, like any tree ensemble method, is built on the basis of decision trees. In these trees, partitions are created by first randomly selecting a feature and then selecting a random split value between the minimum and maximum value of the selected feature.
As with other outlier detection methods, an anomaly score is required for decision making. In the case of Isolation Forest, it is defined as:

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}}$$

where h(x) is the path length of observation x, c(n) is the average path length of unsuccessful search in a Binary Search Tree and n is the number of external nodes.

3.3 Local Outlier Factor

The local outlier factor is based on a concept of a local density, where locality is given by k nearest neighbors, whose distance is used to estimate the density. By comparing the local density of an object to the local densities of its neighbors, one can identify regions of similar density, and points that have a substantially lower density than their neighbors. These are considered to be outliers.

The local density is estimated by the typical distance at which a point can be "reached" from its neighbors. The definition of "reachability distance" used in LOF is an additional measure to produce more stable results within clusters. Local Outlier Factor (LOF) is a score that tells how likely a certain data point is an outlier/anomaly.
LOF $\approx 1 \Rightarrow$ no outlier
LOF $\gg 1 \Rightarrow$ outlier

4 Evaluations and Results

With unlabeled data, evaluating our methods is a difficult challenge. Due to the nature of the network-type data set we have, we propose three evaluation methods.

- Using k-means as a baseline, we can calculate the relative distances between the detected outliers and the centroids. If these values are small, then we conclude that our methods are not good enough. If these values are sufficiently large as compared to maximum distance between data points and their corresponding clusters then our methods works good.
- Since we represent our data in two ways with nodes and edges somehow exchanging, we can test for our methods' consistency by checking if detected suspicious users own detected suspicious transactions. This is called "Dual Evaluation."

Specifically, with the user graph we can get the top N user outliers and with the transaction graph we can get the top M transaction outliers. I choose N = M = 100. We then determine X_N - the set of transactions corresponding to the top N node outliers and Y_M - the set of users corresponding to the top M transaction outliers defined above.

$$A_1 = \frac{|X_N \cap \text{top } X_N \text{ transaction outliers}|}{|X_N|}$$

and

$$A_2 = \frac{|Y_M \cap \text{top } Y_M \text{ transaction outliers}|}{|Y_M|}$$

Finally, the define the Dual Evaluation Metric m_{DE} by

$$m_{DE} = \frac{A_1 + A_2}{2}$$

Note that $m_{DE} \in [0, 1]$ and the bigger it is the more accurate our model is.

4.1 Visualization

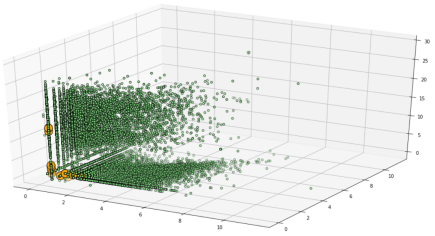

Figure 1 Anomaly Detection using k-means Factor In User Graph

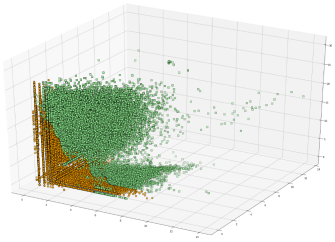

Figure 2 Anomaly Detection using Isolation Forest In user Graph

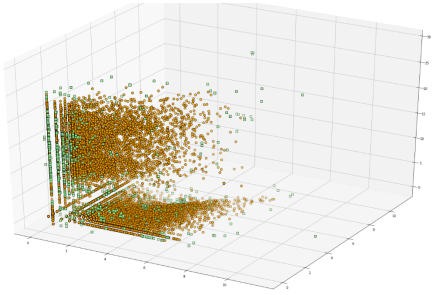

Figure 3 Anomaly Detection using Local outlier Factor In transaction Graph

4.2 Results

- Using k-means clustering method, we get k clusters with corresponding k centroids. For each graph type (user and transaction), calculate the average of the ratios of detected anomaly distances to corresponding centroids over max distances from those centroids to their assigned points for the top 100 outliers. It came out to 0.652 for user graph and $.728$ for transaction graph. These values are large as we expect them to far from the cluster centers.
- For k-means, $A1$ comes out to be 0.12 and $A2$ to be 0.1 and $m_{DE} = 0.11$. This value is quite small, it comes out not as a surprise given the simplicity of our model and lack of computation power.

5 Conclusion and Further Improvements

In this challenge, we have investigate the Bitcoin network. I first represent the data with two focuses: users and transactions. We then use three main social network techniques to detect anomalies, which are potential anomalous users and transactions.
Few improvements that can be done are; explore more different kinds of normalization (data is very messy) good normalization will definitely increase the accuracy. Secondly, Increase the data - I limited myself only to about 1/5 of data due to lack of computation power. Use Mahalanobis distance metric rather than euclidean distance, Mahalanobis distance is an effective multivariate distance metric that measures the distance between a point (vector) and a distribution.

6 References

1. Jason Hirshman,Yifei Huang,Stephen Macke, "*Unsupervised Approaches to Detecting Anomalous Behavior in the Bitcoin Transaction Network* "

2. Thai T. Pham, Steven Lee "*Anomaly Detection in Bitcoin Network Using Unsupervised Learning Methods*",

3. Yu-Jing Lin,Po-Wei Wu, Cheng-Han Hsu,I-Ping Tu, d Shih-wei Liao "*An Evaluation of Bitcoin Address Classification based on Transaction History Summarization*".

4. Kondor D, Pósfai M, Csabai I, Vattay G *Do the Rich Get Richer? An Empirical Analysis of the Bitcoin Transaction Network*

5. https://towardsdatascience.com/outlier-detection-with-isolation-forest-3d190448d45e

6. https://towardsdatascience.com/local-outlier-factor-for-anomaly-detection-cc0c770d2ebe

7. All about dataset - http://www.vo.elte.hu/bitcoin/default.htm