Detecting Ransomware Addresses on the Bitcoin Blockchain

HarvardX Final Capstone CYO Project

Kaylee Robert Tejeda

10/31/2021

Abstract

Ransomware is a persisent and growing threat in the world of cybersecurity. There is much interest in detecting and tracking transactions made to ransomware operators. While many of the attempts towards achieving this have not relied on sophisticated machine learning methods, even those that do have resulted in models with poor specificity. A two-step method is developed to address the issue of false positives and improve on previous results.

Contents

Introduction

Ransomware attacks have gained the attention of security professionals and are of specific interest to international law enforcement and financial regulatory officials.[1] The pseudo-anonymous nature of the Bitcoin blockchain makes it a convenient payment method for attackers who deploy ransomware to accept payments without revealing their identity or location. The victims (usually hospitals or other large organizations) first come to find that all of their important organizational data has been encrypted with a secret key, and are then instructed to make a payment to a specific Bitcoin address in order to have their data decrypted by a certain deadline, otherwise the data will be deleted forever.

For the purposes of this paper, we will ignore the legal and financial implications of ransomware attacks. It will suffice to say that certain parties are interested in tracking and tracing illicit activity on and around the Bitcoin blockchain, and that ransomware transactions are a good example of such activity. For a more detailed overview of how and why such analysis is carried out, the reader is referred to Daniel Goldsmith's work at Chainalysis.com.[2] It can be said that there is significant interest in detecting illicit activity on the Bitcoin blockchain as soon as possible to minimize financial losses. For example, it could be the case that a ransomware attack is being perpetrated on an illegal darknet market site. The news of such an attack might not be published at all, let alone in popular media. However, by analyzing the transaction record with a blockchain explorer, such as BTC.com, we might be able to flag suspicious activity in real time if we have a model that is sufficiently robust. It may, in fact, be the first public notice of such an event. At that point, the suspicious address can be blacklisted or banned from using other services.

Ransomware attackers provide their victims with a payment address, allowing for a list of known ransomware payment addresses to be compiled and analyzed using various methods. One well known paper entitled "BitcoinHeist: Topological Data Analysis for Ransomware Detection on the Bitcoin Blockchain"[3] will be the source of the data set and the baseline by which we will compare our results. In that paper, Akcora, et al. use Topological Data Analysis (TDA) to classify addresses on the Bitcoin blockchain into one of 29 known ransomware address groups. Otherwise they are classified as "white", meaning there is no ransomware activity associated with that address. They then consider the blockchain as a heterogeneous Directed Acyclic Graph (DAG) with two types of nodes describing *addresses* and *transactions*. Edges are formed between the nodes when a transaction can be associated with a particular address.

Addresses on the Bitcoin network may appear many times, with different inputs and outputs each time. The Bitcoin network data is divided into 24-hour long windows with the UTC-6 timezone as a reference. Doing so provides information on how quickly a coin moves through the network, with speed measured as the number of blocks the coin appears in during a 24-hour period, with the maximum being 144 blocks per 24 hours (at an average rate of one block every ten minutes). This speed can be an indicator of money laundering. The temporal data can also help distinguish transactions by geolocation, as criminal transactions tend to cluster in time.

With the graph formed this way, the following six numerical features[2] can be associated with a given address:

1) Income - the total amount of coins sent to an address
2) Neighbors - the number of transactions that have this address as one of its output addresses
3) Weight - the sum of fraction of coins that reach this address from address that do not have any other inputs within the 24-hour window, which are referred to as "starter transactions"
4) Length - the number of non-starter transactions on its longest chain, where a chain is defined as an acyclic directed path originating from any starter transaction and ending at the address in question
5) Count - The number of starter addresses connected to this address through a chain
6) Loop - The number of starter addresses connected to this address by more than one path

These variables are defined in a somewhat abstract way, viewing the blockchain as a topological graph with nodes and edges. The rationale is to be able to quantify specific transaction patterns. For a deeper discussion of how and why these variables were chosen, Akcora[3] gives a thorough explanation in the original paper. For the purposes of this report, we will just be treating the variables as abstract numerical features rather than trying to justify their definitions. Instead, we will run the same data set as used in the original paper by Akcora[3] through a few different machine learning methods to see how closely we can come to their results.

Data

This data set was discovered while exploring the UCI Machine Learning Repository[4] as suggested in the instructions for this project. The author of this report, having been interested in Bitcoin and other cryptocurrencies since (unsuccessfully) mining for them on a ASUS netbook in rural Peru in late 2010, found "cryptocurrencies" to be a natural search term. This brings up a single data set entitled BitcoinHeist: Ransomeware Address Data Set. The data set was downloaded and the exploration began.

We can inspect the first ten observations to get an idea of what features are present.

address	year	day	length	weight	count	looped	neighbors	income	label
111K8kZAEnJg245r2cM6y9zgJGHZtJPy6	2017	11	18	0.0083333	1	0	2	100050000	princetonCerber
1123pJv8jzeFQaCV4w644pzQJzVWay2zcA	2016	132	44	0.0002441	1	0	1	100000000	princetonLocky
112536im7hy6wtKbpH1qYDWtTyMRAcA2p7	2016	246	0	1.0000000	1	0	2	200000000	princetonCerber
1126eDRw2wqSkWosjTCre8cjjQW8sSeWH7	2016	322	72	0.0039063	1	0	2	71200000	princetonCerber
1129TSjKtx65E35GiUo4AYVeyo48twbrGX	2016	238	144	0.0728484	456	0	1	200000000	princetonLocky
112AmFATxzhuSpvtz1hfpa3Zrw3BG276pc	2016	96	144	0.0846140	2821	0	1	50000000	princetonLocky

This data set has 2,916,697 observations of ten features associated with a sample of transactions from the Bitcoin blockchain. The ten features include *address* as a unique identifier, the six features defined above (*income, neighbors, weight, length, count, loop*), two temporal features in the form of *year* and *day* (of year as 1-365), and a categorical factor called *label* that categorizes each address as either "white" (meaning not connected to any ransomware activity), or else one of the 29 known ransomware groups, as identified by three independent ransomware analysis teams (Montreal, Princeton, and Padua)[3] .

The original research team downloaded and parsed the entire Bitcoin transaction graph from 2009 January to 2018 December. Based on a 24 hour time interval, daily transactions on the network were extracted and the Bitcoin graph was formed. Network edges that transfer less than ฿0.3 were filtered out since ransom amounts are rarely below this threshold. Ransomware addresses are taken from three widely adopted studies: Montreal, Princeton and Padua. "White" Bitcoin addresses were capped at one thousand per day while the entire network has up to 800,000 addresses daily.[5]

Goal

The goal of this paper is to apply different machine learning algorithms to the same data set as in the original paper to produce a predictive model without some of the drawbacks that were present there.

Outline of Steps Taken

1) Analyze data set numerically and visually. Notice any pattern, look for insights.

2) Binary classification using Random Forests.

3) Binary classification using Self Organizing Maps.

4) Categorical classification using Self Organizing Maps.

5) Two step method using Random Forests and Self Organizing Maps.

6) Visualize clustering to analyze results.

7) Generate Confusion Matrix to quantify results.

Exploration & Visualization (chunk #2)

Notes on Graphs (remove later)

I need better graphs. I have plenty, but I need them to look better and/or have more labels, etc.

Ideas:

1) Show skewness of the non-temporal variables.
2) Show the rarity of the target addresses.
3) Note how sparse some of the groups are.
4) List group counts in a table

Other fancy graph ideas? Look through sample work for possibilities

Notes End Here

Data Analysis (chunk #2.5)

List computer specs here. Laptop, OS, and R versions.

Preparation

What did I do to prepare the data?

Exploration and Visualization

Insights Gained from Exploration

Maybe its better to approach this as a binary problem? At least at first, lets see how far that gets us....

Modeling approach (Chunk #3, mostly done, just need to clean up a bit)

An overview of why I picked the methods that I did. Based on from original paper, that Random Forests were hard to apply here, and that it was all topological data to begin with, hence that lead me to SOMs. Also, describe the reasoning behind the binary approach. Describe what you learned about SOMs.

Random Forests

Self Organizing Maps

Method 1: Binary Random Forests

If we ask a simpler question, is this a useful approach? Mentioned to not work well in original paper. Try it using a binary black/white approach. change all instances of "grey" in the code to "bw". show how this simplification leads to (near)-perfect accuracy. Confusion Matrix?

Method 2: Binary SOMs

If we ask the same question to a more sophisticated and topological approach, how good is the model? Mention how the original paper was toplogical in nature, an how this lead to the investigation of SOMs. Repeat the binary "b/w" approach using SOMs. This accuracy is still pretty good, but not *as* good as the random forest method. Point out how SOMs are really used for classification into *many* groups. This leads to an Insight! (see above) What if we first *isolate* the "black" addresses using Random Forest, and then categorize the black only subset ($< 2\%$) using categorical SOMs. This leads to a 2-part system...

Method 3: Categorical SOMs

Describe categorical SOM work here, show results. This is where the pretty colored hex-graphs show up.

Final Method: Combined Methods 1 and 3

Using the results from Random Forest, isolate the black addresses first, and then run that subset through an SOM algorithm. Compare final results to original paper. These go in a "results" section. (below)

Results & Performance (Chunk #4)

Results

Performance

In terms of what? Time? RAM?

Summary

Comparison to original paper and impact of findings

Limitations

Future Work

I only scratched he surface of the SOM algorithm which seems to have many implementations and parameters that could be investigated further and possibly optimized via cross-validation, somehow.

Conclusions

Get Monero!
This paper/report presents a reliable method for classifying bitcoin addresses into know ransomware families, while at the same time avoiding false positives by filtering them out using a binary method before classifying them further. It leaves the author of the paper wondering how long before we see ransomware using privacy coins such as Monero. Find and cite a recent paper on the untracability of the Monero blockchain.

References

[1] Adam Brian Turner, Stephen McCombie and Allon J. Uhlmann (November 30, 2020) Analysis Techniques for Illicit Bitcoin Transactions

[2] Daniel Goldsmith, Kim Grauer and Yonah Shmalo (April 16, 2020) Analyzing hack subnetworks in the bitcoin transaction graph

[3] Cuneyt Gurcan Akcora, Yitao Li, Yulia R. Gel, Murat Kantarcioglu (June 19, 2019) BitcoinHeist: Topological Data Analysis for Ransomware Detection on the Bitcoin Blockchain

[4] UCI Machine Learning Repository https://archive.ics.uci.edu/ml/index.php

[5] BitcoinHeistRansomwareAddressDataset Data Set https://archive.ics.uci.edu/ml/datasets/BitcoinHeistRansomwareAddre