Bitcoin Blockchain Clustering Analysis for Ransomware Detection

Thiago R. C. de Lima

February 2, 2021

Abstract

Bitcoin is the most popular digital currency. It is not controlled by any sort of central bank or government institution and is the preferred payment method requested by cyber criminals through ransomware. This type of malware encrypts a victim's files and forces them to pay a ransom in order to regain access. In this short paper, Bitcoin transaction data of tenyears is analyzed by generating a K-Means clustering model, using it to predict each sample's cluster, and then creating a confusion matrix and evaluating the results (Rand Score).

Introduction

Different forms of digital currency have been proposed and implemented over the decades [1] [2] and now the Blockchain technology allows decentralized currencies to exist and reliable transactions to be made, while preventing double-spending using consensus algorithm relying on proof of work [3]. This is the basis of *cryptocurrencies*, specially Bitcoin, which is often considered the first one and has now gathered a lot of attention [4], from researchers, to investors, and, due to its lack of regulation, criminals as well [5].

Recently the internet has been targeted by a wave of new types of malware, categorized as *ransomware.* Ransomware is a type of malware that once it successfully infects a target's computer system, it then encrypts files and private data so that the victim cannot access them anymore [6]. The victim is then presented with a message explaining the situation and also containing instructions on how to regain full access to their own system and this typically involves paying a sum of money via Bitcoin. There are several groups of hackers and variants, also known as *families*, of such malware, yet their defining characteristic is requiring payment to release access to the captured data.

On this research I will analyze several transactions gathered from the Bitcoin Blockchain to determine whether it is possible to easily categorize each transaction as being from a specific ransomware family or not pertaining to any of them at all. Relevant research has been conducted on this field in [7], [8] and [9].

Development

Data

The data set used for this research was provided by Cuneyt Gurcan Akcora, Yulia Gel and Murat Kantarcioglu and is currently available at the UCI Machine Learning Repository [10], hosted by the University of California Irvine [11]. This data set contains a total of 2916697 rows with 10 columns each, collected from January 2009 to December 2018.

Each row represents a transaction from the Bitcoin Blockchain and contains the following set of attributes: *address*, *year*, *day*, *length*, *weight*, *count*, *looped*, *neighbors*, *income* and *label.* A full description of each of these features can be found in [12]. For the purpose of this research only the meaning of *address* and *label* are of importance.

The *label* attribute is used to determine the known family of ransomware each transaction relates to, if any. There are 28 possible families of ransomware within the dataset. The entries that are not related to any of those families have been labeled as *white* by *Akcora et al.* It is imperative to state that this does not necessarily mean such transactions are known to be completely unrelated to any sort of ransomware, as there might exist families unbeknownst to the analysts that put the data set together.

The *address* feature represents the Bitcoin address and is not used for the purposed of this research. The reasoning behind this decision revolves around two issues. The first issue is this attribute being a string value, therefore not quantifiable. The second, although still related to the previous, pertains how such values are generated and what kind of meaning they would bring to a clustering process. The K-Means algorithm is based on the positioning of each sample in an N-dimension *Euclidean Space*, where closer samples tend to have a higher similarity. That is, the position in each axis has a meaning. Bitcoin addresses do not hold any sort of sequential meaning since they are randomly generated hashes.

Tools

I have developed a script using the *Python* language, and *SciKi-Learn* and *Pandas* packages. Python is a general purpose programming language [13] [14] that has gained notoriety in the data science field [15] [16]; SciKit-Learn is tool set for machine learning [17]; and Pandas is a library for data analysis [18]. For this particular research, I used SciKit-Learn for generating the K-Means model and feeding it with the data set again to retrieve the relationship between each transaction and the expected cluster. Then, Pandas is used to manage the data set, cross-reference the known labels and the predicted clusters, and evaluating the level of similarity. The source code has been published under MIT License and can be found in my repository on GitHub [19]: `https://github.com/thiagorcdl/bitcoin_heist`.

The script starts by preparing the data set: loads the CSV file containing the data, converts the type from strings to floating point representation when appropriate and the address column, as previously explained. Then a copy of the data is created without the label to avoid taking it into consideration, and the K-Means model for k clusters is generated using the remaining 8 attributes: *year*, *day*, *length*, *weight*, *count*, *looped*, *neighbors*, and *income*. The result of this model is a set of k *centroids*, which represent the center of each cluster in an 8-dimension Euclidean Space, and are written to a CSV file. Then, the model is used to predict the cluster for each of the entries in the data set and the result is then used, along with the previously known labels, to generate a Confusion Matrix, which is written to a separate CSV file.

Methodology

The effectiveness of the classification using K-Means is calculated by retrieving the *Rand Index*, which measures the similarity between how the known labels and how the predicted clusters are plotted [20] in the 8-dimension space. Finally the Adjusted Rand Index is also calculated, taking into consideration the similarity to random model clustering [21].

Considering the purpose of this research is to find out whether it is possible to segment the transactions in such way to allow the easy categorization of whether they represent a ransomware-related transaction or not, I decided to feed the script with two different values for k, that is to create two different sets of clusters. The first one is to create 29 cluster, ideally one for each family of ransomware plus the *white* ones. Since clustering is an unsupervised classification algorithm, the k clusters do not have a pre-established meaning. In other words, unless there was a perfect segregation of the 29 possible labels, it is not necessarily possibly to know which cluster relates to which label. Thus, the second attempt was to simply create two cluster, which a perfect fit could separate the transactions that are known to be malicious from those that have no evidence.

Results

Upon running the script for 29 clusters and 2 clusters, respectively, I gathered the information on two confusion matrices and four Rand Index values.

For the scenario with 29 clusters, the confusion matrix can be found in *Table* 1. The resulting Rand Index value was *0.8550275740313222*, which relatively close to 1. This value by itself could mean a good level of similarity. However, looking at the confusion matrix, it is possible to note the vast majority of samples were categorized on the very same cluster 0 and even though the white label dominates every other cluster, it is also the majority in cluster 0. That means that even though one could say all clusters but 0 have a high probability of not being related to ransomware, there isn't enough confidence to actually pinpoint the ones that are. That could mean a high rate of False Negatives . This becomes clearer when taking into account the value of Rand Index adjusted for chance, which yielded -*2.918553945107504*. A value below 0 is not reliable.

In *Table* 2 the confusion matrix for the binary scenario is presented. The *negative* label was assigned to all rows with original label *white*, and *positive* to the rest, that is to the ones that are known to belong to some family of ransomware. The Rand Index values were a bit better in this scenario: *0.9719860133842922* and -*0.49493780907250057* for the regular and the adjusted values, respectively, which still means an unreliable level of similarity. The data from the confusion matrix follows the same path as the findings of the previous scenario. Even though there is a really high True Negative Rate (*0.9857911192*), the True Positive Rate is zero, which means the model was unable to correctly guess a single suspicious transaction.

Observations

To summarize, in this research I utilized an extensive data set of Bitcoin transactions to discover whether there are natural clusters with strong correlation to ransomware occurrences. I developed a script and used it for trying to segment the data set both for all 29 original labels and also for a binary partitioning. I used the K-means clustering algorithm in the attempts of finding a clear separation between suspicious and not suspicious transactions.

Although for both scenarios the plain Rand Index is close to 1, when evaluating the similarity adjusting for chance the scores drop below zero, which indicates low reliability. This leads me to conclude the combination of features, algorithm and consequently the generated model used do not represent a good method for classifying suspicious transactions in the Bitcoin Blockchain.

For future work I would consider comparing the results when using the address attribute, and also trying out different classification algorithms.

Labels	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
montrealAPT	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
montrealComradeCircle	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
montrealCryptConsole	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
montrealCryptXXX	2419	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
montrealCryptoLocker	8924	0	0	0	0	0	3	0	0	0	15	21	0	0	0	0	0	0	0	0	0	298	0	0	0	0	50	0	4
montrealCryptoTorLocker2015	54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
montrealDMALocker	246	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0
montrealDMALockerv3	354	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
montrealEDA2	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
montrealFlyper	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
montrealGlobe	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
montrealGlobeImposter	54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
montrealGlobev3	34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
montrealJigSaw	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
montrealNoobCrypt	483	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
montrealRazy	8	0	0	0	0	1	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
montrealSam	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
montrealSamSam	62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
montrealVenusLocker	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
montrealWannaCry	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
montrealXLocker	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
montrealXLockerv5.0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
montrealXTPLocker	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
paduaCryptoWall	12165	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	205	0	0	0	0	20	0	0
paduaJigsaw	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
paduaKeRanger	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
princetonCerber	9219	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0

Labels	1	0
positive	0	41413
negative	30	2875254

Table 2: Confusion matrix for 2 labels

References

[1]D. Chaum, “Blind Signatures for Untraceable Payments”, in *Advances in Cryptology*, Springer US, 1983, pp. 199–203.

[2]C. P. Mullan, *The digital currency challenge shaping online payment systems through US financial regulations.* Basingstoke: Palgrave Macmillan, 2014.

[3]S. A. Back *et al.*, “Enabling Blockchain Innovations with Pegged”, 2014.

[4]Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, “Blockchain challenges and opportunities: A survey”, *International Journal of Web and Grid Services*, vol. 14, p. 352, Oct. 2018, doi: 10.1504/IJWGS.2018.095647.

[5]A. Jeffries, “Suspected multi-million dollar Bitcoin pyramid scheme shuts down, investors revolt”. The Verge, 2012.

[6]R. Richardson and M. M. North, “Ransomware: Evolution, mitigation and prevention”, *International Management Review*, vol. 13, no. 1, p. 10, 2017.

[7]C. G. Akcora, Y. Li, Y. R. Gel, and M. Kantarcioglu, “BitcoinHeist: Topological Data Analysis for Ransomware Detection on the Bitcoin Blockchain”, *arXiv preprint [Web Link]*, 2019.

[8]D. Goldsmith, K. Grauer, and Y. Shmalo, “Analyzing hack subnetworks in the bitcoin transaction graph”, *Applied Network Science*, vol. 5, Apr. 2020, doi: 10.1007/s41109-020-00261-7.

[9]R. Rivera-Castro, P. Pilyugina, and E. Burnaev, “Topological Data Analysis for Portfolio Management of Cryptocurrencies”, in *2019 International Conference on Data Mining Workshops (ICDMW)*, 2019, pp. 238–243, doi: 10.1109/ICDMW.2019.00044.

[10]“UCI Machine Learning Repository: Data Sets”. University of California Irvine.

[11]“UCI Machine Learning Repository — re3data.org”. University of California Irvine.

[12]“BitcoinHeistRansomwareAddress Data Set”. University of California Irvine.

[13]G. van Rossum, “Python tutorial”, no. R 9526, Jan-1995.

[14]“What is Python? Powerful, intuitive programming”. InfoWorld, 13-Nov-2019.

[15]“Python eats away at R: Top Software for Analytics, Data Science, Machine Learning in 2018: Trends and Analysis - KDnuggets”. KDNuggets.

[16]“Python Developers Survey 2019 Results”. JetBrains.

[17]A. Géron, “Hands-on machine learning with Scikit-Learn and TensorFlow concepts, tools, and techniques to build intelligentsystems”, Apr. 2017.

[18]W. Mckinney, “pandas: a Foundational Python Library for Data Analysis and Statistics”, *Python High Performance Science Computer*, Jan. 2011.

[19]T. R. C. de Lima, “Bitcoin Heist Repository”. GitHub.

[20]W. M. Rand, “Objective Criteria for the Evaluation of Clustering Methods”, *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, Dec. 1971, doi: 10.1080/01621459.1971.10482356.

[21]N. Vinh, J. Epps, and J. Bailey, “Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance”, *Journal of Machine Learning Research*, vol. 11, pp. 2837–2854, Oct. 2010.