Detecting Ransomware Addresses on the Bitcoin Blockchain using Random Forest and Self Organizing Maps

HarvardX PH125.9x Final Capstone CYO Project

Kaylee Robert Tejeda

11/11/2021

Abstract

Ransomware is a persisent and growing threat in the world of cybersecurity. A specific area of focus involves detecting and tracking payments made to ransomware operators. While many attempts towards this goal have not made use of sophisticated machine learning methods, even those that have often result in models with poor precision or other performance issues. A two-step method is developed to address the issue of false positives.

Contents

Introduction

Ransomware attacks are of interest to security professionals, law enforcement, and financial regulatory officials.[1] The pseudo-anonymous Bitcoin network provides a convenient method for ransomware attackers to accept payments without revealing their identity or location. The victims (usually hospitals or other large organizations) come to learn that much if not all of their important organizational data have been encrypted with a secret key by an unknown attacker. They are instructed to make a payment to a specific Bitcoin address by a certain deadline to have the data decrypted or else it will be deleted automatically.

The deeper legal and financial implications of ransomware attacks are inconsequential to the work in this report, as we are merely interested in being able to classify bitcoin addresses by their connection to ransomware transactions. Many researchers are already tracking illicit activity (such as ransomware payments) around the Bitcoin blockchain as soon as possible to minimize financial losses. Daniel Goldsmith explains some of the reasons and methods of blockchain analysis at Chainalysis.com.[2] For example, consider a ransomware attack conducted towards an illegal darknet market site. The news of such an attack might not be published at all, let alone in popular media. By analyzing the transaction record with a blockchain explorer such as BTC.com, suspicious activity could be flagged in real time given a sufficiently robust model. It may, in fact, be the first public notice of such an event. Any suspicious addresses could then be blacklisted or banned from using other services, if that is so desired.

Lists of known ransomware payment addresses have been 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 our data set and the baseline to 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 28 known ransomware address groups. Addresses with no known ransomware associations are classified as *white*. The blockchain is then considered 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.

Any given address on the Bitcoin network may appear many times, with different inputs and outputs each time. The Bitcoin network data has been divided into 24-hour time intervals with the UTC-6 timezone as a reference. This way, variables can be defined in a specific and meaningful way. For example, *speed* can be defined as the number of blocks the coin appears in during a 24-hour period, and provides information on how quickly a coin moves through the network. *Speed* may be an indicator of money laundering or coin mixing, as normal payments only involve a limited number of addresses in a given 24 hour period, and thus have lower speeds when compared to "mixed" coins. The temporal data can also help distinguish transactions by geolocation, as criminal transactions tend to cluster in time.

With the graph specified as such, the following six numerical features[2] are 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) *Looped* - The number of starter addresses connected to this address by more than one path

These variables are defined rather conceptually, viewing the blockchain as a topological graph with nodes and edges. The rationale behind this approach is to quantify specific transaction patterns. Akcora[3] gives a thorough explanation in the original paper of how and why these features were chosen. We shall treat the features as general numerical variables and will not seek to justify their definitions beyond that. Machine learning methods will be applied to the original data set from the paper by Akcora[3], and the new results will be compared to the original ones.

Data

This data set was found while exploring the UCI Machine Learning Repository[4] as suggested in the project instructions. The author of this report, interested in Bitcoin and other cryptocurrencies since (unsuccessfully) mining them on an ASUS netbook in rural Peru in late 2010, used *cryptocurrency* as a preliminary search term. This brought up a single data set entitled "BitcoinHeist: Ransomware Address Data Set". The data set was downloaded and the exploration began.

A summary of the data set shows the range of values and size of the sample.

Table 1: Summary of data set

address	year	day	length	weight	count	looped	neighbors	income	label
Length:2916697	Min. :2011	Min. : 1.0	Min. : 0.00	Min. : 0.0000	Min. : 1.0	Min. : 0.0	Min. : 1.000	Min. :3.000e+07	Length:2916697
Class :character	1st Qu.:2013	1st Qu.: 92.0	1st Qu.: 2.00	1st Qu.: 0.0215	1st Qu.: 1.0	1st Qu.: 0.0	1st Qu.: 1.000	1st Qu.:7.429e+07	Class :character
Mode :character	Median :2014	Median :181.0	Median : 8.00	Median : 0.2500	Median : 1.0	Median : 0.0	Median : 2.000	Median :2.000e+08	Mode :character
NA	Mean :2014	Mean :181.5	Mean : 45.01	Mean : 0.5455	Mean : 721.6	Mean : 238.5	Mean : 2.207	Mean :4.465e+09	NA
NA	3rd Qu.:2016	3rd Qu.:271.0	3rd Qu.:108.00	3rd Qu.: 0.8819	3rd Qu.: 56.0	3rd Qu.: 0.0	3rd Qu.: 2.000	3rd Qu.:9.940e+08	NA
NA	Max. :2018	Max. :365.0	Max. :144.00	Max. :1943.7488	Max. :14497.0	Max. :14496.0	Max. :12920.000	Max. :4.996e+13	NA

A listing of the first ten rows provides a sample of the features associated with each observation.

Table 2: First ten entries of data set

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 previously (*income, neighbors, weight, length, count, loop*), two temporal features in the form of *year* and *day* (of the year as 1 to 365), and a categorical feature called *label* that categorizes each address as either *white* (meaning not connected to any ransomware activity), or one of 28 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 January 2009 to December 2018. Based on a 24 hour time interval, daily transactions on the network were extracted and the Bitcoin graph was formed. Network edges that transferred 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, whereas the entire network sees up to 800,000 addresses daily.[5]

Goal

The goal of this project is to apply different machine learning algorithms to the same data set used in the original paper, producing an acceptable predictive model for categorizing ransomware addresses correctly. Improving on the results of the original paper in some way, while not strictly necessary for the purposes of the project, would be a notable sign of success.

Outline of Steps Taken

1. Analyze data set numerically and visually, look for insights in any patterns.
2. Binary separation using Self Organizing Maps.
3. Fast binary separation using Random Forest.
4. Categorical classification using Self Organizing Maps.

5. Visualize clustering to analyze results further.
6. Generate confusion matrix to quantify results.

Data Analysis

Hardware Specification

All of the analysis in this report was conducted on a single laptop computer, a Lenovo Yoga S1 from late 2013 with the following specifications.

- CPU: Intel i7-4600U @ 3.300GHz (4th Gen quad-core i7 x86_64)
- RAM: 8217MB DDR3L @ 1600 MHz (8 GB)
- OS: Slackware64-current (15.0 RC1) `x86_64-slackware-linux-gnu` (64-bit GNU/Linux)
- R version 4.0.0 (2020-04-24) – "Arbor Day" (built from source using scripts from slackbuilds.org)
- RStudio Version 1.4.1106 "Tiger Daylily" (2389bc24, 2021-02-11) for CentOS 8 (converted using `rpm2tgz`)

Data Preparation

It is immediately apparent that this is a rather large data set. The usual practice of partitioning out 80% to 90% of the data for training results in a training set that is too large to process given the hardware limitations. For reasons that no longer apply, the original data set was first split in half with 50% reserved as *validation set* and the other 50% used as the *working set*. This working set was again split in half, to give a *training set* that was of a reasonable size to deal with. This produced partitions that were small enough to work with, so the partition size ratio was not further refined. This is a potential area for later optimization. Careful sampling was carried out to ensure that the ransomware groups were represented in each sample.

Exploration and Visualization

By graphing a values, we can get an idea of how the data is distributed across the various features.

The proportion of ransomware addresses in the original data set is 0.0141986. The total number of NA or missing values in the original data set is 0.

The ransomware addresses make up less than 2% of the overall data set. This presents a challenge as the target observations are sparse within the data set, especially when we consider that this is then divided into 28 subsets. In fact, some of the ransomware groups have only a single member, making categorization a dubious task. At least there are no missing values to worry about.

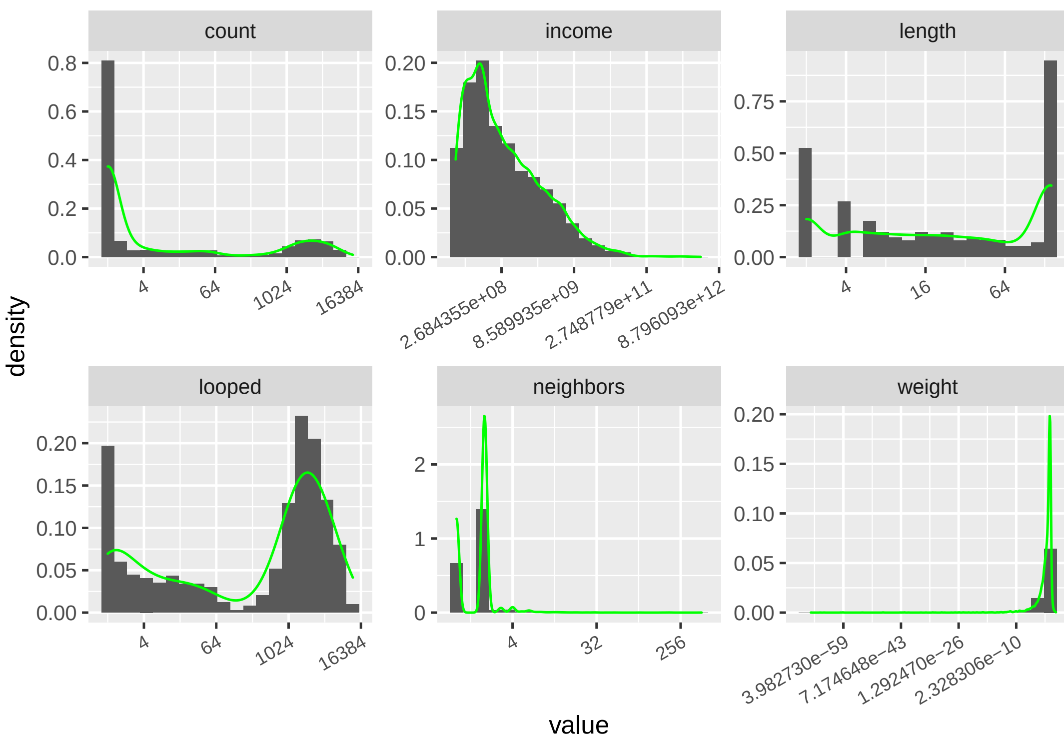
Let's take a look at the distribution of the different features. Note how skewed the non-temporal features are, some of them being bimodal. Looks better on a log scale x-axis.

Table 3: Ransomware group labels and frequency counts for full data set

	x		x		x
montrealAPT	11	montrealGlobe	32	montrealXLocker	1
montrealComradeCircle	1	montrealGlobeImposter	55	montrealXLockerv5.0	7
montrealCryptConsole	7	montrealGlobev3	34	montrealXTPLocker	8
montrealCryptXXX	2419	montrealJigSaw	4	paduaCryptoWall	12390
montrealCryptoLocker	9315	montrealNoobCrypt	483	paduaJigsaw	2
montrealCryptoTorLocker2015	55	montrealRazy	13	paduaKeRanger	10
montrealDMALocker	251	montrealSam	1	princetonCerber	9223
montrealDMALockerv3	354	montrealSamSam	62	princetonLocky	6625
montrealEDA2	6	montrealVenusLocker	7	white	2875284
montrealFlyper	9	montrealWannaCry	28		

Table 4: Coefficients of Variation for each feature

	x		x		x
income	36	weight	6	count	2
neighbors	8	length	1	looped	4

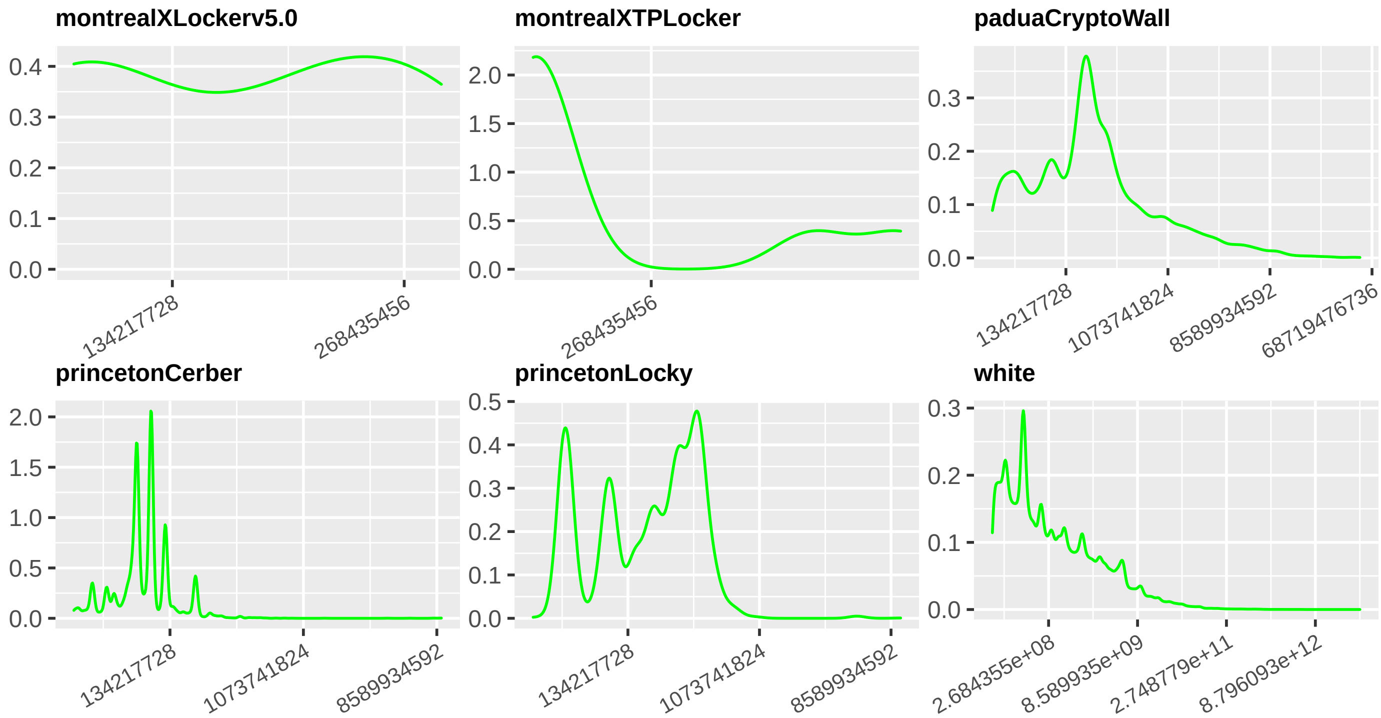


Now let us compare the relative spread of each feature by calculating the coefficient of variation for each column. Larger coefficients of variation indicate larger relative spread compared to other columns.

From this, it appears that income has the widest range of variability, followed by neighbors. These are also the features that are most strongly skewed to the right, meaning that a few addresses have really high values for each of these features while the bulk of the data set has very low values for these numbers.

Taking the feature with the highest variation income, let us take a look at the distribution for individual ransomware families. Perhaps there is a similarity across families.


montrealAPT
0.20
0.15
0.10
0.05
0.00
67108864
134217728
268435456
536870912
1073741824
montrealCryptXXX
3
2
1
0
67108864
268435456
1073741824
montrealCryptoLocker
0.20
0.15
0.10
0.05
0.00
268435456
4294967296
68719476736
montrealCryptoTorLocker2015
0.15
0.10
0.05
0.00
134217728
1073741824
8589934592
montrealDMALocker
0.2
0.1
0.0
134217728
1073741824
8589934592
montrealDMALockerv3
0.4
0.3
0.2
0.1
0.0
67108864
268435456
1073741824
montrealFlyper
2.5
2.0
1.5
1.0
0.5
0.0
33554432
67108864
montrealGlobe
0.4
0.3
0.2
0.1
0.0
33554432
67108864
134217728
montrealGlobeImposter
0.15
0.10
0.05
0.00
134217728
1073741824
8589934592
montrealGlobev3
60
40
20
0
67108864
134217728
268435456
montrealJigSaw
0.6
0.4
0.2
0.0
33554432
67108864
134217728
montrealNoobCrypt
0.2
0.1
0.0
67108864
268435456
1073741824
montrealRazy
0.06
0.04
0.02
0.00
2.684355e+08
4.294967e+09
6.871948e+10
1.099512e+12
montrealSamSam
0.20
0.15
0.10
0.05
0.00
268435456
1073741824
4294967296
montrealWannaCry
0.4
0.3
0.2
0.1
0.0
33554432
67108864
134217728



It appears that, although the income distribution (as an example feature to consider) for ransomware groups does differ from the distribution pattern for *white* addresses, it also varies from group to group. For this reason, this makes a good feature to use in the training of the models.

The percentage of wallets with less than one hundred bitcoins as their balance is 0.0147151. I have no idea why this is meaningful, but I can calculate it at least.

Insights gained from exploration

After visually and statistically exploring of the data, it becomes clear what the challenge is. Ransomware related addresses are very sparse in the data set, making up less than 2% of all addresses. This small percentage is also further classified into 28 groups. Perhaps the original paper was a overly ambitious in trying to categorize all the addresses into 29 categories, including the vastly prevalent *white* addresses. To simplify our approach, we will categorize the addresses in a binary way as either *white* or *black*, where *black* signifies an association with ransomware transactions. Asking this as a "ransomware or not-ransomware" question allows for application of methods that have been shown to be impractical otherwise.

Modelling approach

Akcora, et al. applied a Random Forest approach to the data, however "Despite improving data scarcity, [...] tree based methods (i.e., Random Forest and XGBoost) fail to predict any ransomware family".[3, 11] Considering all ransomware addresses as belonging to a single group may improve the predictive power of such methods, making Random Forest worth another try.

The topological description of the data set inspired a search for topological machine learning methods, although one does not necessitate the other. Searching for *topo* in the documentation for the `caret` package [6] resulted in the entry for Self Organizing Maps (SOMs), supplied by the `kohonen` package. The description at CRAN [7] was intriguing enough to merit further investigation.

Initially, the categorization of ransomware into the 28 different families was attempted using SOMs. This proved to be very resource intensive, requiring more time and RAM than was available. Although it did help to illuminate how SOMs are configured, the resource requirements of the algorithm became a deterrent. It was at this point that the SOMs were applied in a binary way, classifying all ransomware addresses as merely *black*, initially in an attempt to simply get the algorithm to run to completion without error. This seemed to reduce RAM usage to the point of being feasible on the available hardware.

Self Organizing Maps were not covered in the coursework at any point, therefore a familiar method was sought out to compare the results to. Random Forest was chosen and applied to the data set in a binary way, classifying every address as either *white* or *black*, ignoring the ransomware families. Surprisingly, not only did the Random Forest approach result in an acceptable model, it did so much quicker than expected, taking only a few minutes to produce results.

At this point, it was very tempting to leave it there and write up a comparison of the two approaches to the binary problem, by classifying all ransomware related addresses as *black*. However, a nagging feeling that more could be done eventually inspired a second look at the categorical problem of grouping the ransomware addresses into the 28 known families. Given the high accuracy and precision of the binary Random Forest approach, the sparseness of the ransomware in the larger set has been eliminated completely, along with any chances of false positives. There are a few cases of false negatives, depending on how the randomization is done during the sampling process. However, the Random Forest method does not seem to produce many false positive (if any), meaning it never seems to predict a truly white address as being black. Hence, by applying the Random Forest method first, we have effectively filtered out any possibility of false positives by correctly identifying a very large set of purely *white* addresses, which are then removed from the set. The best model used in the original paper by Akcora, et al. resulted in more false positives than true positives. This low precision rate is what made it impractical for real-world usage.[3]

This all inspired a two-part method to first separate the addresses into *black* and *white* groups, and then further classify the *black* addresses into ransomware families. We shall explore each of these steps separately.

Method Part 0: Binary SOMs

The first working model that ran to completion without exhausting computer resources did not make use of the ransomware family labels and instead the two categories of *black* and *white*. The `kohonen` package provides algorithms for both supervised and unsupervised model building. A supervised approach was used since the data set includes information about the membership of ransomware families that can be used to train the model.

After training the model, weobtain the confusion matricies for the test set and the validation set, separately.

Table 5: test set

	black	white
black	10353	0
white	0	718800

Table 6: validation set

	black	white
black	20706	0
white	1	1437605

This is a very intensive and somewhat inaccurate method compared to what follows. It was left out of the final version of the script and has been included here only for model comparison and to track developmental evolution.

Method Part 1: Binary Random Forest

A Random Forest model is trained using ten-fold cross validation and a tuning grid with the number of variables randomly sampled as candidates at each split (`mtry`) set to the values $= 2, 4, 6, 8, 10, 12$, each one being checked for optimization.

The confusion matrix for the test set shows excellent results, specifically in the areas of accuracy and precision.

Here are the confusion matrices for the test set and the full set resulting from the Random Forest model, respectively.

The confusion matrix for the full ransomware set is very similar to that of the test set.

Overall results for test and full sets show good results.

Results by class for the test and full sets. What can you say about these, specifically?

This is a much quicker way of removing most of the *white* addresses, and will be used in the final composite model to save time.

Table 7: confusion matrix for test set

	black	white
black	99	0
white	5	7188

Table 8: confusion matrix for full set

	black	white
black	40027	0
white	1386	2875284

Table 9: test set overall results

	x
Accuracy	0.9993143
Kappa	0.9750220
AccuracyLower	0.9984006
AccuracyUpper	0.9997773
AccuracyNull	0.9857378
AccuracyPValue	0.0000000
McnemarPValue	0.0736383

Table 10: full set overall results

	x
Accuracy	0.9995248
Kappa	0.9827404
AccuracyLower	0.9994991
AccuracyUpper	0.9995495
AccuracyNull	0.9858014
AccuracyPValue	0.0000000
McnemarPValue	0.0000000

Table 11: test set results by class

	x
Sensitivity	0.9519231
Specificity	1.0000000
Pos Pred Value	1.0000000
Neg Pred Value	0.9993049
Precision	1.0000000
Recall	0.9519231
F1	0.9753695
Prevalence	0.0142622
Detection Rate	0.0135765
Detection Prevalence	0.0135765
Balanced Accuracy	0.9759615

Table 12: full set results by class

	x
Sensitivity	0.9665322
Specificity	1.0000000
Pos Pred Value	1.0000000
Neg Pred Value	0.9995182
Precision	1.0000000
Recall	0.9665322
F1	0.9829813
Prevalence	0.0141986
Detection Rate	0.0137234
Detection Prevalence	0.0137234
Balanced Accuracy	0.9832661

Method Part 2: Categorical SOMs

Now we train a new model after throwing away all *white* addresses. The predictions from the Random Forest model are used to isolate all *black* addresses for further classification into ransomware addresses using SOMs. The reduced set is then categorized using a supervised SOM method with the 28 ransomware families as the target classification groups.

When selecting the grid size for a Self Organizing Map, there are at least two different schools of thought. The two that were tried here are explained (with supporting documentation) on a Researchgate forum.[8] The first method is based on the size of the training set, and in this case results in a larger, more accurate map. The second method is based on the number of known categories to classify the data into, and in this case results in a smaller, less accurate map. For this script, a grid size of 27 has been selected.

A summary of the results for the categorization of black addresses into ransomware families follows. For the full table of predictions and statistics, see the Appendix.

Here are the overall results of the final categorization.

Table 13: overall categorization results

	x
Accuracy	0.9998501

	x
Kappa	0.9998046
AccuracyLower	0.9995619
AccuracyUpper	0.9999691
AccuracyNull	0.3062122
AccuracyPValue	0.0000000
McnemarPValue	NaN

Here are the final results by class.

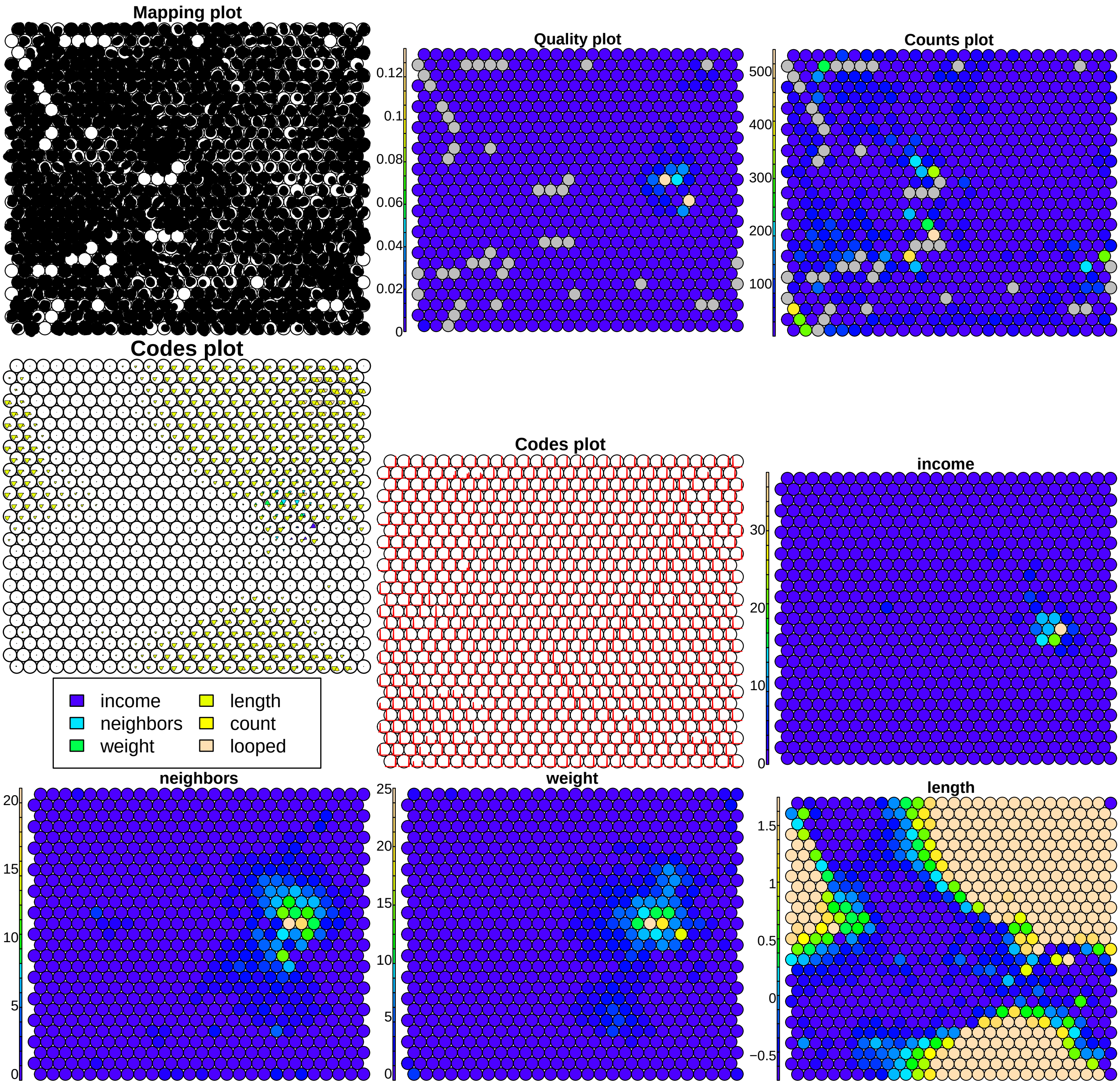
Table 14: categorization results by class

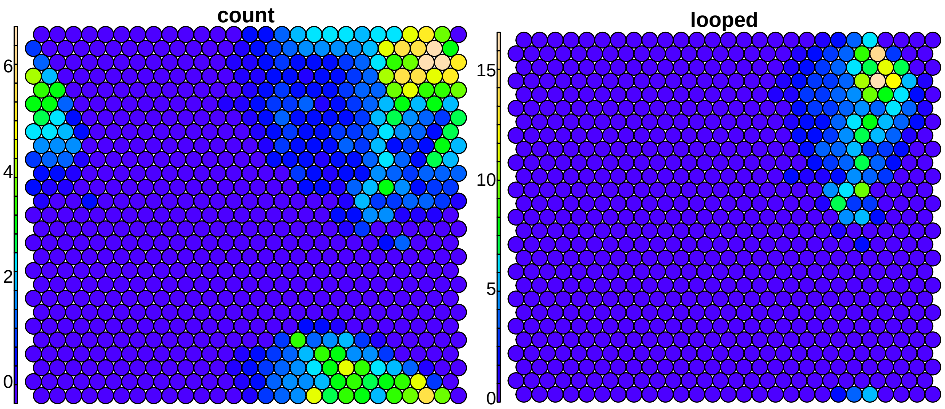
	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value	Precision	Recall	F1	Prevalence	Detection Rate	Detection Prevalence	Balanced Accuracy
Class: montrealAPT	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: montrealComradeCircle	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: montrealCryptConsole	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: montrealCryptXXX	1.0000000	0.9999467	0.9992013	1.0000000	0.9992013	1.0000000	0.9996005	0.0625219	0.0625219	0.0625718	0.9999733
Class: montrealCryptoLocker	0.9997851	1.0000000	1.0000000	0.9999349	1.0000000	0.9997851	0.9998926	0.2325953	0.2325454	0.2325454	0.9998926
Class: montrealCryptoTorLocker2015	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: montrealDMALocker	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: montrealDMALockerv3	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: montrealEDA2	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: montrealFlyper	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: montrealGlobe	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: montrealGlobeImposter	0.9722222	1.0000000	1.0000000	0.9999499	1.0000000	0.9722222	0.9859155	0.0017992	0.0017492	0.0017492	0.9861111
Class: montrealGlobev3	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: montrealJigSaw	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: montrealNoobCrypt	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: montrealRazy	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: montrealSam	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: montrealSamSam	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: montrealVenusLocker	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: montrealWannaCry	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: montrealXLocker	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: montrealXLockerv5.0	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: montrealXTPLocker	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: paduaCryptoWall	1.0000000	0.9998559	0.9996737	1.0000000	0.9996737	1.0000000	0.9998368	0.3062122	0.3062122	0.3063122	0.9999280
Class: paduaJigsaw	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: paduaKeRanger	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA
Class: princetonCerber	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	0.2299965	0.2299965	0.2299965	1.0000000
Class: princetonLocky	0.9997005	1.0000000	1.0000000	0.9999400	1.0000000	0.9997005	0.9998502	0.1668749	0.1668249	0.1668249	0.9998503
Class: white	NA	1.0000000	NA	NA	NA	NA	NA	0.0000000	0.0000000	0.0000000	NA

Clustering Visualizations

Heatmaps and K-means clustering

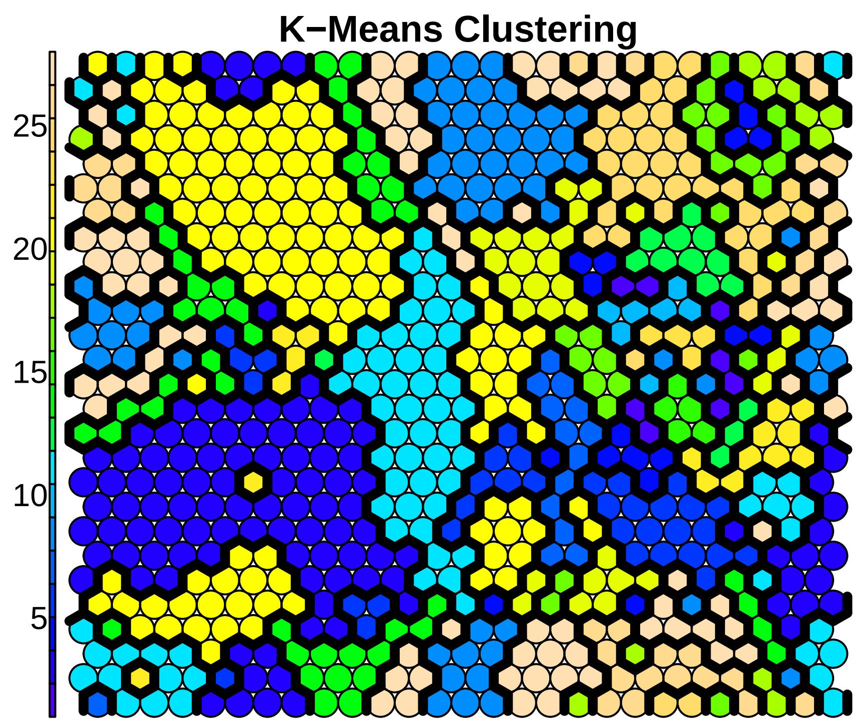
Toroidal nerual node maps are used to generate the models, and can be visualized n a number of ways.





K-means clustering offers a nice way of visualizing the final SOM grid and the categorical boundaries that were formed by the model. Say a bit more about it here. . . .

K-means clustering categorizes the SOM grid by adding boundaries to the classification groups. This is the author's favorite graph in the entire report.



Results & Performance

Results

The first attempt to isolate ransomware using SOMs resulted in a model with an accuracy of 0.999999314275683 and precision 1.

The the second attempt to isolate ransomware using Random forest resulted in a model with an accuracy of 0.999524804942029 and precision 1.

Classifying the ransomware predicted by the second attempt into 28 ransomware families resulted in a model with an overall accuracy of 0.999850067469639 and minimum nonzero precision of 0.999201277955272.

Performance

The script runs on the aforementioned hardware in less than five minutes and uses less than 4GB of RAM. Given that the Bitcoin network produces one new block every ten minutes on average, then real-time analysis could theoretically be conducted on each block as it is announced using even moderate computing resources. Just for kicks, the final script was also run on a more humble computer with the following specifications:

ASUS Eee PC 1025C

- CPU: Intel Atom N2600 @ 1.600GHz (64-bit Intel Atom quad-core x86)
- RAM: 3911MB DDR3 @ 800 MT/s (4 GB)

This is a computer known for being slow and clunky. Even on this device, which runs the same operating system and software as the hardware listed previously, the total run time for the script is around 1665 seconds. At nearly 28 minutes, this is not fast enough to analyze the Bitcoin blockchain in real time, but it does show that the script can be run on very modest hardware to completion.

Pine64 Quartz64 Model A

- CPU: Rockchip RK3566 SoC aarch64 (64-bit quad-core ARM)
- RAM: DDR4 8080MB (8 GB)

Single board computer / Development board. This was run to benchmark a modern 64-bit ARM processor. The script runs in about 860 minutes on this platform, nearly half of that for the Atom processor above.

Summary

Comparison to results from original paper

In the original paper by Akcora et al., they tested several different sets of parameters on their TDA model. According to them, "In the best TDA models for each ransomware family, we predict **16.59 false positives for each true positive**. In turn, this number is 27.44 for the best non-TDA models."[3] In fact, the highest Precision (a.k.a. Positive Predictive Value, defined as TP/(TP+FP)) they achieved was only 0.1610. By comparison, although several of our predicted classes had zero or NA precision values, the lowest non-zero precision value is 0.999201277955272, with many well above that, approaching one in a few cases.

One might say that we are comparing apples to oranges in a sense, because their method was one single model, while these results are from a two-method stack. Still, given the run time of the final script, I think the two-model approach is superior in this case, especially when measured in terms of precision and avoiding false positives.

Limitations

SOMs seem like they are easy to misconfigure. Perhaps a dual Random Forest approach would be better. this has not been attempted yet, as the two method approach presented here was satisfactory enough to present in a report.

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. Also, a dual Random Forest approach to first isolate the ransomware addresses and also

The script itself has a few areas that could be further optimization. The sampling method does what it needs to do, but the ratios taken for each set could possibly be optimized.

Conclusion

This paper/report presents a reliable method for classifying Bitcoin addresses into known 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 much harder it would be to perform the same task for ransomware that uses privacy coins. Certain cryptocurrency networks utilize privacy coins, such as Monero, that obfuscate transactions from being analyzed in the same way that the Bitcoin network has been analyzed here. Some progress has been made towards analyzing such networks[9], but the developers of such networks continually evolve the code to complicate transaction tracking. This could be another good area for future research.

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] BitcoinHeist Ransomware Address Dataset
https://archive.ics.uci.edu/ml/datasets/BitcoinHeistRansomwareAddressDataset

[6] Available Models - The `caret` package http://topepo.github.io/caret/available-models.html

[7] Ron Wehrens and Johannes Kruisselbrink, Package '`kohonen`' @ CRAN (2019) https://cran.r-project.org/web/packages/kohonen/kohonen.pdf

[8] How many nodes for self-organizing maps? (Oct 22, 2021) https://www.researchgate.net/post/How-many-nodes-for-self-organizing-maps

[9] Malte Möser, Kyle Soska, Ethan Heilman, Kevin Lee, Henry Heffan, Shashvat Srivastava, Kyle Hogan, Jason Hennessey, Andrew Miller, Arvind Narayanan, and Nicolas Christin (April 23, 2018) An Empirical Analysis of Traceability in the Monero Blockchain

Appendix:

Categorical SOM prediction table and confusion matrix

Here are the full prediction results for the categorization of *black* addresses into ransomware families. It is assumed that all *white* address have already been removed.

```
## Confusion Matrix and Statistics
##
##                                Reference
## Prediction                      montrealAPT montrealComradeCircle
##   montrealAPT                             0                     0
##   montrealComradeCircle                   0                     0
##   montrealCryptConsole                    0                     0
##   montrealCryptXXX                        0                     0
##   montrealCryptoLocker                    0                     0
##   montrealCryptoTorLocker2015             0                     0
##   montrealDMALocker                       0                     0
##   montrealDMALockerv3                     0                     0
##   montrealEDA2                            0                     0
##   montrealFlyper                          0                     0
##   montrealGlobe                           0                     0
##   montrealGlobeImposter                   0                     0
##   montrealGlobev3                         0                     0
##   montrealJigSaw                          0                     0
##   montrealNoobCrypt                       0                     0
##   montrealRazy                            0                     0
##   montrealSam                             0                     0
##   montrealSamSam                          0                     0
##   montrealVenusLocker                     0                     0
##   montrealWannaCry                        0                     0
##   montrealXLocker                         0                     0
##   montrealXLockerv5.0                     0                     0
##   montrealXTPLocker                       0                     0
##   paduaCryptoWall                         0                     0
##   paduaJigsaw                             0                     0
##   paduaKeRanger                           0                     0
##   princetonCerber                         0                     0
##   princetonLocky                          0                     0
##   white                                   0                     0
##                                Reference
## Prediction                      montrealCryptConsole montrealCryptXXX
##   montrealAPT                                      0                0
##   montrealComradeCircle                            0                0
##   montrealCryptConsole                             0                0
##   montrealCryptXXX                                 0             1251
##   montrealCryptoLocker                             0                0
##   montrealCryptoTorLocker2015                      0                0
##   montrealDMALocker                                0                0
##   montrealDMALockerv3                              0                0
##   montrealEDA2                                     0                0
##   montrealFlyper                                   0                0
##   montrealGlobe                                    0                0
##   montrealGlobeImposter                            0                0
##   montrealGlobev3                                  0                0
##   montrealJigSaw                                   0                0
```

```
##    montrealNoobCrypt                               0                0
##    montrealRazy                                    0                0
##    montrealSam                                     0                0
##    montrealSamSam                                  0                0
##    montrealVenusLocker                             0                0
##    montrealWannaCry                                0                0
##    montrealXLocker                                 0                0
##    montrealXLockerv5.0                             0                0
##    montrealXTPLocker                               0                0
##    paduaCryptoWall                                 0                0
##    paduaJigsaw                                     0                0
##    paduaKeRanger                                   0                0
##    princetonCerber                                 0                0
##    princetonLocky                                  0                0
##    white                                           0                0
##                              Reference
## Prediction                    montrealCryptoLocker montrealCryptoTorLocker2015
##    montrealAPT                                     0                           0
##    montrealComradeCircle                           0                           0
##    montrealCryptConsole                            0                           0
##    montrealCryptXXX                                0                           0
##    montrealCryptoLocker                         4653                           0
##    montrealCryptoTorLocker2015                     0                           0
##    montrealDMALocker                               0                           0
##    montrealDMALockerv3                             0                           0
##    montrealEDA2                                    0                           0
##    montrealFlyper                                  0                           0
##    montrealGlobe                                   0                           0
##    montrealGlobeImposter                           0                           0
##    montrealGlobev3                                 0                           0
##    montrealJigSaw                                  0                           0
##    montrealNoobCrypt                               0                           0
##    montrealRazy                                    0                           0
##    montrealSam                                     0                           0
##    montrealSamSam                                  0                           0
##    montrealVenusLocker                             0                           0
##    montrealWannaCry                                0                           0
##    montrealXLocker                                 0                           0
##    montrealXLockerv5.0                             0                           0
##    montrealXTPLocker                               0                           0
##    paduaCryptoWall                                 1                           0
##    paduaJigsaw                                     0                           0
##    paduaKeRanger                                   0                           0
##    princetonCerber                                 0                           0
##    princetonLocky                                  0                           0
##    white                                           0                           0
##                              Reference
## Prediction                    montrealDMALocker montrealDMALockerv3
##    montrealAPT                                  0                   0
##    montrealComradeCircle                        0                   0
##    montrealCryptConsole                         0                   0
##    montrealCryptXXX                             0                   0
##    montrealCryptoLocker                         0                   0
##    montrealCryptoTorLocker2015                  0                   0
##    montrealDMALocker                            0                   0
##    montrealDMALockerv3                          0                   0
```

```
##   montrealEDA2                                0                   0
##   montrealFlyper                              0                   0
##   montrealGlobe                               0                   0
##   montrealGlobeImposter                       0                   0
##   montrealGlobev3                             0                   0
##   montrealJigSaw                              0                   0
##   montrealNoobCrypt                           0                   0
##   montrealRazy                                0                   0
##   montrealSam                                 0                   0
##   montrealSamSam                              0                   0
##   montrealVenusLocker                         0                   0
##   montrealWannaCry                            0                   0
##   montrealXLocker                             0                   0
##   montrealXLockerv5.0                         0                   0
##   montrealXTPLocker                           0                   0
##   paduaCryptoWall                             0                   0
##   paduaJigsaw                                 0                   0
##   paduaKeRanger                               0                   0
##   princetonCerber                             0                   0
##   princetonLocky                              0                   0
##   white                                       0                   0
##                             Reference
## Prediction                   montrealEDA2 montrealFlyper montrealGlobe
##   montrealAPT                           0              0             0
##   montrealComradeCircle                 0              0             0
##   montrealCryptConsole                  0              0             0
##   montrealCryptXXX                      0              0             0
##   montrealCryptoLocker                  0              0             0
##   montrealCryptoTorLocker2015           0              0             0
##   montrealDMALocker                     0              0             0
##   montrealDMALockerv3                   0              0             0
##   montrealEDA2                          0              0             0
##   montrealFlyper                        0              0             0
##   montrealGlobe                         0              0             0
##   montrealGlobeImposter                 0              0             0
##   montrealGlobev3                       0              0             0
##   montrealJigSaw                        0              0             0
##   montrealNoobCrypt                     0              0             0
##   montrealRazy                          0              0             0
##   montrealSam                           0              0             0
##   montrealSamSam                        0              0             0
##   montrealVenusLocker                   0              0             0
##   montrealWannaCry                      0              0             0
##   montrealXLocker                       0              0             0
##   montrealXLockerv5.0                   0              0             0
##   montrealXTPLocker                     0              0             0
##   paduaCryptoWall                       0              0             0
##   paduaJigsaw                           0              0             0
##   paduaKeRanger                         0              0             0
##   princetonCerber                       0              0             0
##   princetonLocky                        0              0             0
##   white                                 0              0             0
##                             Reference
## Prediction                   montrealGlobeImposter montrealGlobev3
##   montrealAPT                                    0               0
##   montrealComradeCircle                          0               0
```

```
##   montrealCryptConsole                       0               0
##   montrealCryptXXX                           0               0
##   montrealCryptoLocker                       0               0
##   montrealCryptoTorLocker2015                0               0
##   montrealDMALocker                          0               0
##   montrealDMALockerv3                        0               0
##   montrealEDA2                               0               0
##   montrealFlyper                             0               0
##   montrealGlobe                              0               0
##   montrealGlobeImposter                     35               0
##   montrealGlobev3                            0               0
##   montrealJigSaw                             0               0
##   montrealNoobCrypt                          0               0
##   montrealRazy                               0               0
##   montrealSam                                0               0
##   montrealSamSam                             0               0
##   montrealVenusLocker                        0               0
##   montrealWannaCry                           0               0
##   montrealXLocker                            0               0
##   montrealXLockerv5.0                        0               0
##   montrealXTPLocker                          0               0
##   paduaCryptoWall                            1               0
##   paduaJigsaw                                0               0
##   paduaKeRanger                              0               0
##   princetonCerber                            0               0
##   princetonLocky                             0               0
##   white                                      0               0
##                              Reference
## Prediction                    montrealJigSaw montrealNoobCrypt montrealRazy
##   montrealAPT                              0                 0            0
##   montrealComradeCircle                    0                 0            0
##   montrealCryptConsole                     0                 0            0
##   montrealCryptXXX                         0                 0            0
##   montrealCryptoLocker                     0                 0            0
##   montrealCryptoTorLocker2015              0                 0            0
##   montrealDMALocker                        0                 0            0
##   montrealDMALockerv3                      0                 0            0
##   montrealEDA2                             0                 0            0
##   montrealFlyper                           0                 0            0
##   montrealGlobe                            0                 0            0
##   montrealGlobeImposter                    0                 0            0
##   montrealGlobev3                          0                 0            0
##   montrealJigSaw                           0                 0            0
##   montrealNoobCrypt                        0                 0            0
##   montrealRazy                             0                 0            0
##   montrealSam                              0                 0            0
##   montrealSamSam                           0                 0            0
##   montrealVenusLocker                      0                 0            0
##   montrealWannaCry                         0                 0            0
##   montrealXLocker                          0                 0            0
##   montrealXLockerv5.0                      0                 0            0
##   montrealXTPLocker                        0                 0            0
##   paduaCryptoWall                          0                 0            0
##   paduaJigsaw                              0                 0            0
##   paduaKeRanger                            0                 0            0
##   princetonCerber                          0                 0            0
```

```
##   princetonLocky                                0                 0            0
##   white                                         0                 0            0
##                              Reference
## Prediction                    montrealSam montrealSamSam montrealVenusLocker
##   montrealAPT                           0              0                   0
##   montrealComradeCircle                 0              0                   0
##   montrealCryptConsole                  0              0                   0
##   montrealCryptXXX                      0              0                   0
##   montrealCryptoLocker                  0              0                   0
##   montrealCryptoTorLocker2015           0              0                   0
##   montrealDMALocker                     0              0                   0
##   montrealDMALockerv3                   0              0                   0
##   montrealEDA2                          0              0                   0
##   montrealFlyper                        0              0                   0
##   montrealGlobe                         0              0                   0
##   montrealGlobeImposter                 0              0                   0
##   montrealGlobev3                       0              0                   0
##   montrealJigSaw                        0              0                   0
##   montrealNoobCrypt                     0              0                   0
##   montrealRazy                          0              0                   0
##   montrealSam                           0              0                   0
##   montrealSamSam                        0              0                   0
##   montrealVenusLocker                   0              0                   0
##   montrealWannaCry                      0              0                   0
##   montrealXLocker                       0              0                   0
##   montrealXLockerv5.0                   0              0                   0
##   montrealXTPLocker                     0              0                   0
##   paduaCryptoWall                       0              0                   0
##   paduaJigsaw                           0              0                   0
##   paduaKeRanger                         0              0                   0
##   princetonCerber                       0              0                   0
##   princetonLocky                        0              0                   0
##   white                                 0              0                   0
##                              Reference
## Prediction                    montrealWannaCry montrealXLocker
##   montrealAPT                                0               0
##   montrealComradeCircle                      0               0
##   montrealCryptConsole                       0               0
##   montrealCryptXXX                           0               0
##   montrealCryptoLocker                       0               0
##   montrealCryptoTorLocker2015                0               0
##   montrealDMALocker                          0               0
##   montrealDMALockerv3                        0               0
##   montrealEDA2                               0               0
##   montrealFlyper                             0               0
##   montrealGlobe                              0               0
##   montrealGlobeImposter                      0               0
##   montrealGlobev3                            0               0
##   montrealJigSaw                             0               0
##   montrealNoobCrypt                          0               0
##   montrealRazy                               0               0
##   montrealSam                                0               0
##   montrealSamSam                             0               0
##   montrealVenusLocker                        0               0
##   montrealWannaCry                           0               0
##   montrealXLocker                            0               0
```

```
##   montrealXLockerv5.0                         0               0
##   montrealXTPLocker                           0               0
##   paduaCryptoWall                             0               0
##   paduaJigsaw                                 0               0
##   paduaKeRanger                               0               0
##   princetonCerber                             0               0
##   princetonLocky                              0               0
##   white                                       0               0
##                               Reference
## Prediction                    montrealXLockerv5.0 montrealXTPLocker
##   montrealAPT                                   0                 0
##   montrealComradeCircle                         0                 0
##   montrealCryptConsole                          0                 0
##   montrealCryptXXX                              0                 0
##   montrealCryptoLocker                          0                 0
##   montrealCryptoTorLocker2015                   0                 0
##   montrealDMALocker                             0                 0
##   montrealDMALockerv3                           0                 0
##   montrealEDA2                                  0                 0
##   montrealFlyper                                0                 0
##   montrealGlobe                                 0                 0
##   montrealGlobeImposter                         0                 0
##   montrealGlobev3                               0                 0
##   montrealJigSaw                                0                 0
##   montrealNoobCrypt                             0                 0
##   montrealRazy                                  0                 0
##   montrealSam                                   0                 0
##   montrealSamSam                                0                 0
##   montrealVenusLocker                           0                 0
##   montrealWannaCry                              0                 0
##   montrealXLocker                               0                 0
##   montrealXLockerv5.0                           0                 0
##   montrealXTPLocker                             0                 0
##   paduaCryptoWall                               0                 0
##   paduaJigsaw                                   0                 0
##   paduaKeRanger                                 0                 0
##   princetonCerber                               0                 0
##   princetonLocky                                0                 0
##   white                                         0                 0
##                               Reference
## Prediction                    paduaCryptoWall paduaJigsaw paduaKeRanger
##   montrealAPT                               0           0             0
##   montrealComradeCircle                     0           0             0
##   montrealCryptConsole                      0           0             0
##   montrealCryptXXX                          0           0             0
##   montrealCryptoLocker                      0           0             0
##   montrealCryptoTorLocker2015               0           0             0
##   montrealDMALocker                         0           0             0
##   montrealDMALockerv3                       0           0             0
##   montrealEDA2                              0           0             0
##   montrealFlyper                            0           0             0
##   montrealGlobe                             0           0             0
##   montrealGlobeImposter                     0           0             0
##   montrealGlobev3                           0           0             0
##   montrealJigSaw                            0           0             0
##   montrealNoobCrypt                         0           0             0
```

```
##   montrealRazy                            0           0             0
##   montrealSam                             0           0             0
##   montrealSamSam                          0           0             0
##   montrealVenusLocker                     0           0             0
##   montrealWannaCry                        0           0             0
##   montrealXLocker                         0           0             0
##   montrealXLockerv5.0                     0           0             0
##   montrealXTPLocker                       0           0             0
##   paduaCryptoWall                      6127           0             0
##   paduaJigsaw                             0           0             0
##   paduaKeRanger                           0           0             0
##   princetonCerber                         0           0             0
##   princetonLocky                          0           0             0
##   white                                   0           0             0
##                              Reference
## Prediction                    princetonCerber princetonLocky white
##   montrealAPT                             0              0     0
##   montrealComradeCircle                   0              0     0
##   montrealCryptConsole                    0              0     0
##   montrealCryptXXX                        0              1     0
##   montrealCryptoLocker                    0              0     0
##   montrealCryptoTorLocker2015             0              0     0
##   montrealDMALocker                       0              0     0
##   montrealDMALockerv3                     0              0     0
##   montrealEDA2                            0              0     0
##   montrealFlyper                          0              0     0
##   montrealGlobe                           0              0     0
##   montrealGlobeImposter                   0              0     0
##   montrealGlobev3                         0              0     0
##   montrealJigSaw                          0              0     0
##   montrealNoobCrypt                       0              0     0
##   montrealRazy                            0              0     0
##   montrealSam                             0              0     0
##   montrealSamSam                          0              0     0
##   montrealVenusLocker                     0              0     0
##   montrealWannaCry                        0              0     0
##   montrealXLocker                         0              0     0
##   montrealXLockerv5.0                     0              0     0
##   montrealXTPLocker                       0              0     0
##   paduaCryptoWall                         0              0     0
##   paduaJigsaw                             0              0     0
##   paduaKeRanger                           0              0     0
##   princetonCerber                      4602              0     0
##   princetonLocky                          0           3338     0
##   white                                   0              0     0
##
## Overall Statistics
##
##                Accuracy : 0.9999
##                  95% CI : (0.9996, 1)
##     No Information Rate : 0.3062
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9998
##
##  Mcnemar's Test P-Value : NA
```

```
##
## Statistics by Class:
##
##                      Class: montrealAPT Class: montrealComradeCircle
## Sensitivity                          NA                           NA
## Specificity                           1                            1
## Pos Pred Value                       NA                           NA
## Neg Pred Value                       NA                           NA
## Prevalence                            0                            0
## Detection Rate                        0                            0
## Detection Prevalence                  0                            0
## Balanced Accuracy                    NA                           NA
##                      Class: montrealCryptConsole Class: montrealCryptXXX
## Sensitivity                                   NA                 1.00000
## Specificity                                    1                 0.99995
## Pos Pred Value                                NA                 0.99920
## Neg Pred Value                                NA                 1.00000
## Prevalence                                     0                 0.06252
## Detection Rate                                 0                 0.06252
## Detection Prevalence                           0                 0.06257
## Balanced Accuracy                             NA                 0.99997
##                      Class: montrealCryptoLocker
## Sensitivity                               0.9998
## Specificity                               1.0000
## Pos Pred Value                            1.0000
## Neg Pred Value                            0.9999
## Prevalence                                0.2326
## Detection Rate                            0.2325
## Detection Prevalence                      0.2325
## Balanced Accuracy                         0.9999
##                      Class: montrealCryptoTorLocker2015
## Sensitivity                                         NA
## Specificity                                          1
## Pos Pred Value                                      NA
## Neg Pred Value                                      NA
## Prevalence                                           0
## Detection Rate                                       0
## Detection Prevalence                                 0
## Balanced Accuracy                                   NA
##                      Class: montrealDMALocker Class: montrealDMALockerv3
## Sensitivity                                NA                         NA
## Specificity                                 1                          1
## Pos Pred Value                             NA                         NA
## Neg Pred Value                             NA                         NA
## Prevalence                                  0                          0
## Detection Rate                              0                          0
## Detection Prevalence                        0                          0
## Balanced Accuracy                          NA                         NA
##                      Class: montrealEDA2 Class: montrealFlyper
## Sensitivity                           NA                    NA
## Specificity                            1                     1
## Pos Pred Value                        NA                    NA
## Neg Pred Value                        NA                    NA
## Prevalence                             0                     0
## Detection Rate                         0                     0
## Detection Prevalence                   0                     0
```

```
## Balanced Accuracy                    NA                     NA
##                      Class: montrealGlobe Class: montrealGlobeImposter
## Sensitivity                           NA                     0.972222
## Specificity                            1                     1.000000
## Pos Pred Value                        NA                     1.000000
## Neg Pred Value                        NA                     0.999950
## Prevalence                             0                     0.001799
## Detection Rate                         0                     0.001749
## Detection Prevalence                   0                     0.001749
## Balanced Accuracy                     NA                     0.986111
##                      Class: montrealGlobev3 Class: montrealJigSaw
## Sensitivity                             NA                    NA
## Specificity                              1                     1
## Pos Pred Value                          NA                    NA
## Neg Pred Value                          NA                    NA
## Prevalence                               0                     0
## Detection Rate                           0                     0
## Detection Prevalence                     0                     0
## Balanced Accuracy                       NA                    NA
##                      Class: montrealNoobCrypt Class: montrealRazy
## Sensitivity                               NA                  NA
## Specificity                                1                   1
## Pos Pred Value                            NA                  NA
## Neg Pred Value                            NA                  NA
## Prevalence                                 0                   0
## Detection Rate                             0                   0
## Detection Prevalence                       0                   0
## Balanced Accuracy                         NA                  NA
##                      Class: montrealSam Class: montrealSamSam
## Sensitivity                         NA                    NA
## Specificity                          1                     1
## Pos Pred Value                      NA                    NA
## Neg Pred Value                      NA                    NA
## Prevalence                           0                     0
## Detection Rate                       0                     0
## Detection Prevalence                 0                     0
## Balanced Accuracy                   NA                    NA
##                      Class: montrealVenusLocker Class: montrealWannaCry
## Sensitivity                                 NA                      NA
## Specificity                                  1                       1
## Pos Pred Value                              NA                      NA
## Neg Pred Value                              NA                      NA
## Prevalence                                   0                       0
## Detection Rate                               0                       0
## Detection Prevalence                         0                       0
## Balanced Accuracy                           NA                      NA
##                      Class: montrealXLocker Class: montrealXLockerv5.0
## Sensitivity                             NA                         NA
## Specificity                              1                          1
## Pos Pred Value                          NA                         NA
## Neg Pred Value                          NA                         NA
## Prevalence                               0                          0
## Detection Rate                           0                          0
## Detection Prevalence                     0                          0
## Balanced Accuracy                       NA                         NA
##                      Class: montrealXTPLocker Class: paduaCryptoWall
```

```
## Sensitivity                             NA                 1.0000
## Specificity                              1                 0.9999
## Pos Pred Value                          NA                 0.9997
## Neg Pred Value                          NA                 1.0000
## Prevalence                               0                 0.3062
## Detection Rate                           0                 0.3062
## Detection Prevalence                     0                 0.3063
## Balanced Accuracy                       NA                 0.9999
##                      Class: paduaJigsaw Class: paduaKeRanger
## Sensitivity                          NA                   NA
## Specificity                           1                    1
## Pos Pred Value                       NA                   NA
## Neg Pred Value                       NA                   NA
## Prevalence                            0                    0
## Detection Rate                        0                    0
## Detection Prevalence                  0                    0
## Balanced Accuracy                    NA                   NA
##                      Class: princetonCerber Class: princetonLocky Class: white
## Sensitivity                            1.00                0.9997           NA
## Specificity                            1.00                1.0000            1
## Pos Pred Value                         1.00                1.0000           NA
## Neg Pred Value                         1.00                0.9999           NA
## Prevalence                             0.23                0.1669            0
## Detection Rate                         0.23                0.1668            0
## Detection Prevalence                   0.23                0.1668            0
## Balanced Accuracy                      1.00                0.9999           NA
```