Detecting Ransomware in Bitcoin Transaction Using Machine Learning

Conference Paper · May 2022

CITATIONS
0

READS
49

1 author:

Baraa Abu Sallout
Islamic University of Gaza
1 PUBLICATION **0** CITATIONS

SEE PROFILE

Detecting Ransomware in Bitcoin Transaction Using Machine Learning

Baraa Wael Abu Sallout
Department of Computer Engineering
I*slamic University Of Gaza*
Gaza, Palestine
baraasallout@gmail.com

Hussein Ali Ashour
Department of Computer Engineering
I*slamic University Of Gaza*
Gaza, Palestine
hussein.a02@gmail.com

Abstract— **Bitcoin cryptocurrency system enables users to transact securely and pseudo-anonymously by using an arbitrary number of aliases (Bitcoin addresses). Cybercriminals exploit these characteristics to commit immutable and presumably untraceable monetary fraud, especially via ransomware; a type of malware that encrypts files of the infected system and demands ransom for decryption. Traditional machine learning algorithms tend to be biased towards the more frequently occurring class categories and fail to capture the general pattern or structure of the ransomware bitcoin addresses. Hence, we believe that traditional machine learning algorithms do not perform well enough to be used in such a data security problem. Deep Neural Networks have proven to work well with time-series data as well as multi-class classification and clustering problems. In this paper, we will use deep neural networks to analyze how these transactions take place and try to build models that predict if a given Bitcoin address is being used for malicious intent or not.**

Keywords—Ransomware, Deep Neural Networks, KNN, Random Forest, Naive Bayes, Decision Tree, Machine Learning, AdaBoost

I. INTRODUCTION

Ransomware is a sophisticated malware that has grown quickly in recent years, that prevents users from accessing their data using encryption techniques until a ransom is paid to the attacker. This results in huge losses for businesses and individuals. Ransomware can be classified into numerous types, the most violent and virulent being crypto-ransomware. Crypto ransomware not only encrypts user data but also tries to encrypt information on both mapped and unmapped network devices, putting a whole department or company to a standstill if just one machine is infected. In this type of attack, the attacker does not benefit by selling user data on illegal websites. They reap the benefits from the value associated with the victim's data and by the money paid by the victim to release their data. This attack causes temporary or permanent loss of valuable information and blocks regular operations. Crypto ransomware prefers the Bitcoin network as, during the ransomware attack, the victim's system remains fully functional thus allowing the victim to pay the ransom in Bitcoins on the system.

Satoshi Nakamoto 2008 proposed a decentralized cryptography-based electronic currency called Bitcoin. Bitcoin is a digital asset that leverages a peer-to-peer network to facilitate the transfer of value without intermediation from banks or central authority. Bitcoin is a digital currency, with no physical bitcoins in circulation.

According to the annual threat report-2017 published by Symantec Inc. , ransomware continued to be the most dangerous cyber-crime threat to individual users and enterprises in 2016. Compared to the previous year, the number of detected ransomware infection increased by 36% during 2016. Moreover, average ransomware detection rate reached over 1,500 incidents per day at the yearend. In particular, the average ransom amount rose 266% from USD 294 in 2015 to 2015 to USD 1,077.

Various Machine Learning algorithms have been proposed to detect the Ransomware Bitcoin addresses, but they do not generalize enough to classify Bitcoin addresses belonging to different Malware families and the test results need to be validated on more recent Bitcoin addresses.

In this paper, we tried out different machine learning approaches to detect the Bitcoin ransomware addresses and we found out that the neural network model performs slightly better compared to other models in metrics like Receiver Operating Characteristic (ROC), F1 score, and accuracy. A neural network captures the information granularity at various layers which helps to detect ransomware addresses in a more generalized way.

II. RELATED WORK

There has been a surge in the use of machine learning approaches to detect and prevent ransomware attacks. A detailed study was made by Martina Jose Mary.M, Usharani.S , Thirugnanam.P in their paper studied the detailed working of a ransomware attack and the different types of ransomware. They analyzed the features like CPU user usage, system usage, RAM usage, receive packet and byte, send packets, send bytes, receive packets, receive bytes, and netflows of the dataset. They used various machine learning algorithms such as KNN, Naïve Bayes, Random Forest, SGD, SVM, Logistic Regression, Bayesian Network to get desired output. They showed the comparison of all the accuracies and results as well. But every algorithm has its own advantages and disadvantages, so the best suited model which can be handled by the system is to be chosen. We improve on this study further by using different machine learning algorithms. We used the behaviour of the ransomware trace files to train and test the model.

A paper written by Ban Mohammed Khammas is a static method of detection. They used a random forest classifier to detect and found out that a high accuracy can be achieved by changing the seed value to 1 and tree numbers 100. They also used Frequent Pattern Mining technique to directly extract the features from raw byte. This showed an increase in efficiency.

Ahmad O. Almashhadani et. al. in their paper designed a network based intrusion detection system with two independent classifiers, packet classifier and flow-based classifier, working in parallel on packet and flow levels to detect the packet-level and flow-level feature vectors coupled with a decision unit which detects any suspicious activity.

On a packet-based data set using Random Forest algorithm they were able to achieve a F1 score of 0.979 and an accuracy of 98.72 % . On a flow-based data set, Bayes Net algorithm achieved a F1 score of 0.971 and 99.83% accuracy.

III. Methodology and Technical Detils

Given a bitcoin address along with some meta-data pertaining to that address, we are challenged to predict if that address has been used to receive ransoms in the past. Upon finding that an address has indeed been used for malicious intent, quick action can be taken against that bitcoin address, such as banning it from any future transactions or blacklisting it to prevent further online scams. Although Bitcoin is open source, it takes in donations. It is in Bitcoin's best interests that the platform has a good reputation, as more people starting using it, and because of that, it generates higher donations. Thus, to keep a good PR, Bitcoin would have to keep its platform from being advertised as a convenient means to scamming. Seeing that the outcome of misclassification by our model can be extremely high, we will assume that Bitcoin or a third-party has employed domain-expert(s) who will verify the outputs of our model.

A. The data

We use a dataset from the UCI Machine Learning Repository that contains parsed Bitcoin transaction graphs from 2009 January to 2018 December. This data-set contains labeled data of transactions and if whether they are white or if they belong to a class of Ransomware.

B. Data-cleaning and Feature Engineering

Our dataset has ~30,00,000 rows and 10 columns. Out of these 10 columns, we have 9 predictors and one target column. Each row relates to a particular transaction.

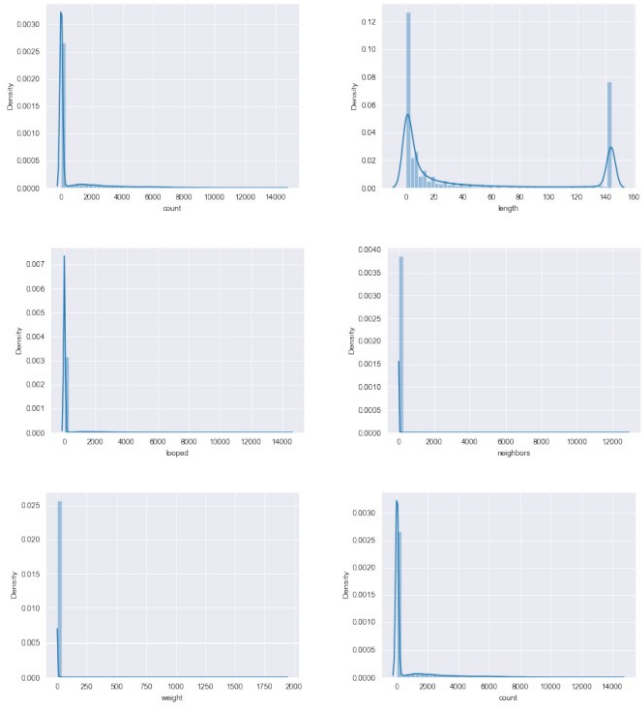
Bitcoin Heist Ransomware Address dataset was used to perform bitcoin ransomware detection. The dataset was obtained by parsing the bitcoin transaction graph. The graph consists of daily Bitcoin transactions from 2009 to 2018. Since ransom amounts are rarely below 0.3 Bitcoins, the network edges that transferred less than the threshold were removed. The dataset consists of 10 attributes in which the last attribute is the target label of the bitcoin addresses. The dataset attributes are described in TABLE I.

The raw data we got from the UCI Repository needs to be processed before we do any modeling. We also need to modify the distributions for our feature columns because some of the predictive models we will use might assume that the features are normally distributed.

TABLE I: Dataset Attribute Details

Attribute	**Description**
address	unique Bitcoin address
year	year of the transaction.
day	day of the year. 1 denotes 1st day, similarly 365 is the last day of the year.
weight	sum of the coins that come from a beginning transaction.
count	number of beginning transactions which the acyclic directed path connects the address node of interest to a number of beginning transactions.
looped	number of starter transactions connected to the address node of interest by more than one directed path.
neighbors	number of transactions with the output address of the address node of interest.
income	total quantity of coins produced to the address node of interest .
label	name of the ransomware variant (e.g., Cryptxxx, CryptoLocker etc) or white (i.e., not a ransomware).

Fig. 1: Distribution of each feature showing they are highly skewed



C. Feature Engineering

- Fixing skewness: Some of our newer features are boxcox or other transformations of our vanilla features.
- Engineering new features:
  - Number of addresses: This feature keeps track of how many times a given Bitcoin address has been encountered in the training dataset. The hypothesis is that, if a Bitcoin address has been used for malicious intent, it might be involved in more transactions than usual, to receive ransoms. We will see later whether or not this hypothesis holds.

- Day of week: Another feature we engineer is the day of the week in which the transaction took place. We generate this information, by using the year and day data in the dataset and use python's datetime library to retrieve the day. We will see later whether or not this feature correlates with the target.
- Is close to a holiday: Using the datetime object, we generated earlier, we can see if this transaction took place around an important holiday/festival. The hypothesis is that, if a transaction took place before or after a week of a major holiday, it might be white. We may understand this by thinking that people may want to transfer money to their family or friends during holidays. Once again, during EDA we will verify or disprove this assumption.

IV. RESULTS

Logistic Regression was applied on our cleaned dataset. However, the model did not perform well on the training set. The model achieved 69% accuracy on the test set and had low precision, recall and F1 scores. Logistic Regression assumes that the data is linear and tries to fit linear decision boundaries on the training data to perform classification.

Knearest Neighbour (KNN) was trained on the dataset. Since KNN is a lazy learning algorithm that performs classification of new example based on the classes of k nearest neighbours. KNN does not perform well when the data is large. KNN is also sensitive to nosiy data and is not robust to outliers. Because of which, KNN does not perfom well on our test dataset resulting in 72% accuracy.

Support Vector Machines (SVM) was the next model that we tried. SVMs work well with both linear and non-linear data. SVMs use different kernels that transform the input features into a much higher dimensions and make predictions on them. SVM achieved an accuracy of 72% on our test data. However, it did not improve in precision, recall and F1 scores.

Decision Trees were used for ransomware classification. In our experiments, we noticed that the decision trees performs the worst compared to all the other algorithms we used. This maybe due to the fact that a small change in the data causes the whole structure of the decision tree to change. This leads to instability as the tree structure keeps changing during training. Decision Trees obtained only 67% accuracy on our test set.

RandomForest models were also tested on our test splits. The model achieves about 72% accuracy on our test set and there is a slight increase in precison, recall and F1 scores.

AdaBoost and XGBoost were two boosting algorithms that we also tried for detecting ransomware addresses. Both of the models showed a decent improvements in all the metrics that we used to evaluate the model.

Finally, we tried applying a deep neural network on the problem and checked if it performs well compared to other algorithms. In our experiments, we found out that neural networks works well with our dataset and scores the highest in all the metrics.

TABLE II: Model Comparison Study

Model	Acuracy	Recall	precision	F1 Score	ROC
Logistic Regression	0.69	0.62	0.73	0.67	0.69
KNN	0.72	0.69	0.73	0.71	0.71
SVM	0.72	0.61	0.79	0.69	0.72
Decision Tree	0.67	0.68	0.67	0.68	0.67
Random Forest	0.72	0.70	0.73	0.71	0.72
AdaBoost	0.73	0.64	0.78	0.70	0.72
XGBoost	0.74	0.65	0.79	0.72	0.73
Neural Network	0.74	0.69	0.79	0.72	0.74

V. CONSTRAINTS AND CHALLENGES

Note that in this project, we only focus on banning/blacklisting bitcoin addresses used for malicious intents in the past and NOT in real-time. This project can very easily be extended to work in real-time as well, by having an API that checks if a receiver's bitcoin address has been flagged by our model before any new transaction is made, and if so, immediately notifies the authorities who can take further action. However, for this project, we will not be extending our use-case to real-time.

Now let us see the main business constraints.

- The cost of misclassifying a positive data point can be high. This is because if an address that been used for malicious intent is flagged by our model to be white, that address can continue to be used to receive ransoms and scamming people.
- There are no latency concerns since we are not considering real-time use cases. However, if we do, latency becomes a very important constraint.
- Interpretability is important because our domain expert would require logical reasoning behind the model's predictions so he can either verify or conduct further investigation

VI. CONCLUSION

We started out with a dataset with extremely skewed features and with the goal of predicting if a bitcoin address was used for malicious intent or not. We performed a lot of data analysis and found out that the existing features needed some tweaking and new features had to be engineered to give our models some more juice.

We performed various transformations on our existing features and used the same to come up with brand new features which would later increase our predictive power substantially.

We looked at various kinds of models; distance-based, tree-based, and even stacked models. On

investigating each model's performance, we concluded that GBDT was the best performer.

Going back to our initial business constraints, we satisfy all three. Firstly, we maximized recall so as to minimize misclassification of a positive data point. The second constraint was that we did not have any strict latency constraint, which is why we were able to build such complicated models. The final constraint was that interpretability is very important. We can very easily have our model give out an explanation behind its outputs because its base-estimators (Decision Trees) are extremely interpretable. For feature importances and other graphs, please go through the notebook.

Because neural networks have the capacity to learn on their own and produce output which is not limited by inputs, they can learn from past events and apply what they have learned whenever a similar circumstance arises, enabling them to cope with real-time problems, therefore it performs better than conventional machine learning algorithms. It would have performed much more significantly if the proportion of white to black labels was more, if the data had not been skewed and even with log transformations the data is still not normal enough to show good metric results.

VII. REFERENCES

[1] study of locky ransomware," Ieee Access, vol. 7, pp. 47 053–47 067, 2019.

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

[3] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Decentralized Business Review, p. 21260, 2008.

[4] D. Y. Huang, M. M. Aliapoulios, V. G. Li, L. Invernizzi, E. Bursztein, K. McRoberts, J. Levin, K. Levchenko, A. C. Snoeren, and D. McCoy, "Tracking ransomware end-to-end," in 2018 IEEE Symposium on Security and Privacy (SP). IEEE, 2018, pp. 618–631.

[5] Mohurle, S.; Patil, M. A brief study of Wannacry Threat: Ransomware Attack 2017. Int. J. Adv. Res. Comput. Sci. 2017, 8

[6] Oosthoek, K.; Doerr, C. From Hodl to Heist: Analysis of Cyber Security Threats to Bitcoin Exchanges. In Proceedings of the 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Toronto, ON, Canada, 2–6 May 2020.

[7] Paquet-Clouston, M.; Haslhofer, B.; Dupont, B. Ransomware payments in the Bitcoin ecosystem. J. Cybersecur. 2019, 5, tyz003.

[8] Erfani, S.; Ahmadi, M. Bitcoin Security Reference Model: An Implementation Platform. In Proceedings of the 2019 International Symposium on Signals, Circuits and Systems (ISSCS), Iasi, Romania, 11–12 July 2019.

[10] Biryukov, A.; Pustogarov, I. Bitcoin over Tor isn't A Good Idea. In Proceedings of the 2015 IEEE Symposium on Security and Privacy, San Jose, CA, USA, 17–21 May 2015.

[11] S. Kok, A. Abdullah, N. Jhanjhi, and M. Supramaniam, "Prevention of crypto-ransomware using a pre-encryption detection algorithm," Computers, vol. 8, no. 4, p. 79, 2019.

[12] K. Cabaj, M. Gregorczyk, and W. Mazurczyk, "Software-defined networking-based crypto ransomware detection using http traffic characteristics," Computers & Electrical Engineering, vol. 66, pp. 353–368, 2018.