ISSN No:-2456-2165

Investigating Machine Learning Approaches for Bitcoin Ransomware Payment Detection Systems

Kirat Jadhav
Department of Electronics,
Ramrao Adik Institute of Technology,
DY Patil University, Navi Mumbai, India

Abstract:- Cryptocurrencies have revolutionized the process of trading in the digital world. Roughly one decade since the induction of the first bitcoin block, thousands of cryptocurrencies have been introduced. The anonymity offered by the cryptocurrencies also attracted the perpetuators of cybercrime. This paper attempts to examine the different machine learning approaches for efficiently identifying ransomware payments made to the operators using bitcoin transactions. Machine learning models may be developed based on patterns differentiating such cybercrime operations from normal bitcoin transactions in order to identify and report attacks. The machine learning approaches are evaluated on bitcoin ransomware dataset. Experimental results show that Gradient Boosting and XGBoost algorithms achieved better detection rate with respect to precision, recall and F-measure rates when compared with k-Nearest Neighbor, Random Forest, Naïve Bayes and Multilayer Perceptron approaches.

Keywords:- *Blockchain, Bitcoin, Cybercrime, Machine Learning, Ransomware.*

I. INTRODUCTION

Bitcoin is a cryptocurrency introduced in 2008 by Satoshi Nakamoto. Bitcoin is the best-known cryptocurrency. It was implemented in 2009 using an open source code. It is a digital banking system without a physical central banking system without any specific country of origin. Bitcoin is a decentralized type of payment system where the public ledger is properly supported in a distributed manner. Some unknown anonymous candidates called miners, executing a protocol that maintains and extends a distributed public ledger that records bitcoin transactions is called a Block Chain. Block chain is implemented as a chain of blocks. Bitcoin is the best-known cryptocurrency.

The transactions of bitcoin are completely digital and anonymous to a great extent. This situation has led many cyber crime perpetuators to use bitcoin as a safe haven for illegal transactions such as ransomware payments. Ransomware is malicious software that affects the payment gateway in return of ransom that has to be paid. Machine Learning approaches may be employed to pour over the previous transactions as training data inorder to correctly predict the individuals or groups to whom ransomware payments are being made. This paper tries to explore the efficacy of different machine learning approaches in detecting such payments.

The remainder of the paper is structured as follows: related work in the area is discussed in section 2. Section 3 discusses the features and characteristics of the Bitcoin Ransomware Dataset evaluated in this paper. Section 4 presents the experiment results, followed by conclusions in section 5

II. RELATED WORK

There is a surge in the number of online users investing and trading in cryptocurrencies (for example, bitcoin) recently. However, the anonymity lent by the cryptocurrencies were misused by operators of ransomware. This section aims to identify the works which have focused on identifying ransom payments in cryptocurrencies, especially in terms of bitcoins.

Agcora et al [1] have utilized topological data analysis techniques to automatically identify new malicious addresses in the ransomware family. The authors have designed a bitcoin graph model as a directed weighted graph. New addresses belonging to the ransomware family are identified based on the payments made to the known addresses of ransomware family. Initially, the ransomware addresses are clustered into 20,000 groups. The resulting clusters are then analyzed for any relation between ransomware families. Both Topological Data Analysis (TDA) as well as DBSCAN clustering algorithm are employed to detect and predict ransomware transactions.

Liao et al [2] have performed analysis on CryptoLocker, a family of ransomware. A framework which automatically detects the ransom payments made to bitcoin addresses that belong to the CryptoLocker.Blockchain analysis and data sourced from online forums such as reddit and BitcoinTalk were utilized to perform measurement analysis on the data. The timestamps based on the ransom payments by the victims are then extracted. Using this data, the trends in the times ransom amounts were paid were analyzed.

Conti el al [3] explored the security and privacy issues in bitcoin. The work discussed how the veil of anonymity provided by the bitcoin ecosystem is encouraging the cyber criminals to resort to illegal activities such as ransomware, tax evasion and money laundering.

Turner et. al [4] have tried to analyze the transaction patterns of ransomware attacks. The patterns are analyzed to collect intelligence to counteract the ransomware attacks. Ransomware seed addresses were used to model a target network for pattern analysis. Different graph algorithms were employed to analyze the cash-in and cash-out patterns. The results show distinguishable paths related to the input and output side of the ransomware graphs.

Huang et. al [5] have performed measurement analysis of two-year data of ransomware payments including the details regarding the victims as well as operators. A comprehensive dataset from multiple data sources such as ransomware binaries, victim telemetry as well as vast list of bitcoin addresses was formed. This information was used to bitcoin-trail right from when the victim acquires bitcoins to the point where the operators cash out the bitcoins. The results claim improved coverage and detection of the ransomware when compared with existing algorithms.

Alhawi et. al [6] have proposed NetConverse which uses J48 based decision-tree classifier to detect ransomware samples from features that were derived from network traffic communications. Results show the the proposed approach returned better detection aret when compared to other conventional machine learning approaches such as Bayes Network, k-Nearest Neighbor, Multi-layer perceptron, Random Forest and Logistic Model tree.

Poudyal et. al [7] have proposed a framework for detecting ransomware using machine learning techniques. Evaluation of the eight different supervised machine learning algorithms has been conducted at two levels viz., assembly and dll. The results indicate that the ransomware detection rate of more than 90%

III. DATASET DESCRIPTION

The dataset for training the machine learning algorithms on the ransomware payments over bitcoin network is sourced from [1]. The dataset was downloaded from the bitcoin transaction graph from 2009January to 2018 December. Daily transactions from the network were extracted and the network links having less than 0.3billion were filtered out as ransomware amounts were usually above this threshold. The dataset contains 24,486 addresses selected from 28 ransomware families. The “Bit Coin Heist Ransomware Address Dataset” contains 9 descriptive attributes and a decision attribute. A summary of the dataset is presented in Table 1.

Attribute Id	Attribute Name	Attribute Type	Category/Description
1	Address	String	Address of the transaction. The transaction could be ransomware or white.
2	Year	Integer	Year of transaction as integer
3	Day	Integer	1 is and 365 is last day of the year
4	Length	Integer	Number of non-starter transactions on its longest chain.
5	Weight	Float	Sum of fraction of coins that originate from starter transaction and end up reaching the address.
6	Count	Integer	Number of starter transactions connected to the address through a chain
7	Looped	Integer	Number of starter transactions connected to the address with more than one directed arc.
8	Neighbors	Integer	Number of transactions which have the address as output.
9	Income	Float	Total number of coins output to the address
10	Label	String	The class to which the transaction belongs to. either white (non-ransomware) or ransomware (one of the 27 ransomware families)

Table 1:- Description of the BitCoinHeistRansomware Address dataset

The bitcoin trasactions have been implemented as a Bitcoin Graph model with the help og a directed acyclic graph. Along with the bitcoin address and its year and day time stamp, six other features have been associated with the address. The income attribute is used to represent payment made in number of bitcoins. The length attribute is used to identify the number of non-starter transactions on the longest chain. A starter-transaction is any one of the earlier transaction in a 24-hour window which did not receive any payments. The attribute weight corresponds to the fraction of coins originating from the starter transaction and ultimately ending up at the corresponding bitcoin address. The attribute length defines the number of non-starter transactions in its longest chain. The chain is implemented as a directed acyclic graph, originating from any starter-transaction and ending at given address. Count attributes defines the number of starter transactions connected to the given address. Loop of an address is the number of starter transactions connected to the address via more than one directed path.

Each of the transactions in the dataset are associated with a label indicating whether the transaction is benign (white) or belongs to one of the 27 ransomware families. In toto, the dataset is a multi-class dataset which is extremely imbalanced in nature. A dataset is said to be imbalanced if the representations of different classes are roughly not equal. The class distributions of the label attribute is summarized in Table 2. The percentage of imbalance of the most frequent ransomware class viz., paduaCryptoWall with respect to the majority class viz., white is 0.43%. The representation of other less frequent ransomware families is almost n egligible. Most of the conventional classifiers, driven by accuracy-based evaluation metrics may fail in effectively predicting the ransomware attacks. This work aims to study the effect of different classifiers in such extremely imbalanced scenario.

Class	Label	Frequency	Class	Label	Frequency
0	white	2875284	14	montrealWannaCry	28
1	paduaCryptoWall	12390	15	montrealRazy	13
2	montrealCryptoLocker	9315	16	montrealAPT	11
3	princetonCerber	9223	17	paduaKeRanger	10
4	princetonLocky	6625	18	montrealFlyper	9
5	montrealCryptXXX	2419	19	montrealXTPLocker	8
6	montrealNoobCrypt	483	20	montrealCryptConsole	7
7	montrealDMALockerv3	354	21	montrealVenusLocker	7
8	montrealDMALocker	251	22	montrealXLockerv5.0	7
9	montrealSamSam	62	23	montrealEDA2	6
10	montrealGlobeImposter	55	24	montrealJigSaw	4
11	montrealCryptoTorLocker2015	55	25	paduaJigsaw	2
12	montrealGlobev3	34	26	montrealSam	1
13	montrealGlobe	32	27	montrealComradeCircle	1
			28	montrealXLocker	1

Table 2:- Frequency of occurrences of the class labels

IV. ECPERIMENTAL RESULTS

All the experiments were conducted on Intel Core i7-6500U CPU 2.5GHz PC with 16GB of RAM running on 64-bit operating system. The implementation is done using Python programming language on Jupyter Notebook. The experiments on “Bitcoin Heist Ransomware Address Dataset” are performed with randomly selected 90% of the dataset as training data and remaining as validation data.

A. Machine Learning Approaches

The machine learning approaches considered in this paper for building classification models for predicting the ransomware attacks are Naïve Bayes, Random Forest, Multi Layer Perceptron, k-Nearest Neighbor, Gradient Boosting and XGBoost.

Naive Bayes algorithm is based on the probability-based Bayes Theorem. The algorithm works on the principle of Class Conditional Independence. The class conditional independence states that the effect a feature has on the class label is independent of the effect of other features. The posterior probability of the unknown instance with respect to each class label is estimated and the class label which maximizes this conditional will be the predicted class label.

Random Forest (RF) is an ensemble classification framework which relies on the predictions from multiple weak learners, in order to make a single unified prediction. The ensemble approaches have been proven to perform better than conventional classification approaches, and ease the issues faced by the individual constituent classifiers. Random Forest approach creates a collection of multiple decision trees. The constituent decision trees are fed the data by applying random subset sampling on the instances as well as features. The predictions from these decision trees are aggregated to obtain the unified prediction.

Multilayer Perceptron (MLP) is a neural network based classification approach that attempts to learn the concept from the provided training data based on back propagation algorithm. The back-propagation algorithm searches for the weight values that minimize the error over the training instances. The algorithm repeatedly executes in two phases viz., forward and backward. The forward pass evaluates the output using the weights of the neural network. The deviation (error) with respect to actual labels is evaluated and the weights associated with the neural networks will be adjusted based on this error. The process is repeated in epochs until the training error becomes negligible.

k-nearest neighbor (k-NN) is a lazy learning approach in that the model for generalizing the provided training dataset is not prebuilt before examining the unknown instances. k-NN represents the provided training instances on the feature space in terms of similarity measures (Distance functions). “k” is a user specified parameter which selects the “k” number of training instance “closest” to a given unknown instance. The nearest neighbors are estimated using classical distance measures (Euclidean, Manhattan, or Minkowski) for continuous variables and hamming distance for categorical variables. Consensus among these measures provides the predicted class label for a given unknown instance.

Gradient Boosting is an ensemble learner which uses Decision Trees as base classifiers. The decision trees are added one at a time. Gradient Descent approach for minimizing loss function is employed while adding the trees, whenever new base classifier is to be added, then its correlation with the negative loss function is evaluated. Only those weak learners which are maximally correlated are added.

XGBoost is an optimized version of Gradient Boosting algorithm. XGBoost has algorithmic as well as system enhancements over GradientBoosting algorithm. The sequential tree addition in Gradient Boosting is parallelized in XGBoost. Also, XGBoost constrains the growth of the constituent decision trees using maximum depth as a parameter. Hardware based optimizations are also included by allocating cache buffers to store the gradient details. The usage of regularization methods like LASSO and Ridge further makes XGBoost superior to Gradient Descent algorithm.

B. Evaluation Metrics

The classification models suggested by the learning algorithms cannot be deployed directly as models derived from active learners suffer from the overfitting problem. Therefore,

		Actual	
		Positive	***Negative***
Predicted	***Positive***	TP	FP
	Negative	FN	TN

Table 3:- Confusion Matrix

The classification model is validated against a separate test dataset. Once the evaluation metrics returned satisfactory values, the classification model is presumed to be ready for deployment.

The evaluation parameters for classification model are based on the confusion matrix. Table 3 shows the confusion matrix for a basic two-class problem. The confusion matrix is comprised of TP (true positives), TN (true negatives), FP (false positives), FN (false negatives).

The most common evaluation metrics considered are Accuracy, Precision, Recall and F-Measure. Accuracy is defined as proportion of total number of predictions made that are correct. True Positive Rate is defined as the ratio of correctly classified positive examples to the total number of positive examples. Precision is another widely used metric in information retrieval which estimates the percentage of relevant objects out of the retrieved ones. Recall corresponds to the number of relevant instances retrieved out of all relevant ones. F-Measure is the harmonic mean of Precision and Recall. Accuracy has been shown in several studies is biased towards majority class. In case of the bitcoin dataset which is extremely skewed in nature, accuracy may not be considered as a good evaluation measure. Hence the results were drawn on the validation dataset for Precision, Recall and F-measure values.

C. Results

The validation dataset corresponds to 10% of the randomly subsampled instances from the BitcoinRansomware dataset. The addresses as well as the class label attributes of the entire dataset have to be transformed using Label Encoding process for some classification algorithms to begin modeling the data. The resulting class label and the frequency counts of individual class labels are provided in the result tables for clear understanding. The validation dataset also can be noticed as extremely imbalanced in nature. The results in terms of Accuracy, Precision, Recall and F-measure are depicted in Tables 4-7.

Class	Accuracy	Average Precision	Average Recall	Average F-Measure
Naïve Bayes	0.38	0.98	0.38	0.55
Random Forest	0.99	0.99	0.99	0.99
Multi Layer Perceptron	0.99	0.97	0.99	0.98
k-Nearest Neighbor	0.98	0.98	0.98	0.98
Gradient Boosting	0.99	0.99	0.99	0.98
XGBoost	0.99	0.98	0.99	0.98

Table 4:- Comparison of Evaluation metrics for overall validation data

A cursory glance at the results of evaluation metrics in Table 4 show that the Naïve Bayes Classifier did not perform well on the dataset whereas other classifiers return good values. However, detailed analysis of these results with respect to individual attacks as displayed in Tables 5-7 indicate that good values for averages of Precision, Recall and F-measure cannot be equated to good prediction of attacks. Tables 5-7 examine the values of the evaluation metrics in a more granular level. Class 28 corresponds to the white label and is not a kind of attack. Classes 0 to 27 represent 28 different kinds of ransom payments made to cybercrime perpetuators. Hence, the efficacy of a classification model in the current dataset is not characterized by how well the classifiers predict class 28 but rather how well the attacks represented by classes 0 to 27 are identified.

Class	#instances	Naïve Bayes	Random Forest	Multi Layer Perceptron	k-Nearest Neighbor	Gradient Boosting	XGBoost
0	2	0	0	0	0	1	0
1	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0
3	248	0	0.93	0	0.5	0.93	0.92
4	922	0	0.89	0	0.3	0.94	1
5	4	0	0	0	0.5	0	0
6	29	0	1	0	0.5	0.92	1
7	41	0	1	0	0.83	0.74	0
8	1	0	0	0	0	0	0
9	1	0	0	0	0	0	0
10	6	0	0	0	0	0.12	0
11	2	0	0	0	0	1	1
12	3	0	0	0	1	1	1
13	0	0	0	0	0	0	0
14	45	0	1	0	0	0.74	1
15	0	0	0	0	0	0	0
16	1	00	0	0	0	0	0
17	4	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	3	0	0	0	0	0	0
20	1	0	0	0	0	0	0
21	1	0	0	0	0	0	0
22	1	0	0	0	0	0	0
23	1245	0	0.78	0	0.32	0.65	0
24	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0
26	926	0.01	0.85	0	0.21	0.78	0
27	680	0	0.83	0	0.18	0.82	0.87
28	287503	0.99	0.99	0.99	0.99	0.99	0.99

Table 5:- Comparison of class-based Precision values

After observing the results in Tables 5-7, it is evident that Naïve Bayes is not a good learner for the bitcoin ransomware dataset. However, the other classifiers which returned high average values for evaluation metrics, have completely failed to correctly identify many of the attack classes. The Gradient Boosting and XGBoost classifiers were able to identify more attack instances when compared with other classifiers considered. The multi-layer perceptron algorithm was not able to identify any attack-based classes, and its bias towards the majority class is clearly evident.

Class	#instances	Naïve Bayes	Random Forest	Multi Layer Perceptron	k-Nearest Neighbor	Gradient Boosting	XGBoost
0	2	0	0	0	0	1	0
1	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0
3	248	0	0.79	0	0.4	0.87	0.76
4	922	0	0.25	0	0.07	0.16	0.03
5	4	0	0	0	0.25	0	0
6	29	0	0.1	0	0.14	0.83	0.52
7	41	0	0.1	0	0.46	0.41	0
8	1	0	0	0	0	0	0
9	1	0	0	0	0	0	0
10	6	0	0	0	0	0.17	0
11	2	0	0	0	0	1	1
12	3	0	0	0	0.33	1	0.33
13	0	0	0	0	0	0	0

14	45	0	0.09	0	0	0.96	0.02
15	0	0	0	0	0	0	0
16	1	0	0	0	0	0	0
17	4	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	3	0	0	0	0	0	0
20	1	0	0	0	0	0	0
21	1	0	0	0	0	0	0
22	1	0	0	0	0	0	0
23	1245	0	0.2	0	0.09	0.02	0
24	0	0	0	0	0	0	00
25	0	0	0	0	0	0	00
26	926	0.99	0.46	0	0.07	0.27	0
27	680	0	0.61	0	0.04	0.43	0.12
28	287503	0.38	1	1	1	1	1

Table 6:- Comparison of class-based Recall values

Class	#instances	Naïve Bayes	Random Forest	Multi Layer Perceptron	k-Nearest Neighbor	Gradient Boosting	XGBoost
0	2	0	0	0	0	1	0
1	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0
3	248	0	0.85	0	0.44	0.9	0.83
4	922	0	0.39	0	0.12	0.27	0.05
5	4	0	0	0	0.33	0	0
6	29	0	0.19	0	0.22	0.87	0.68
7	41	0	0.18	0	0.59	0.53	0
8	1	0	0	0	0	0	0
9	1	0	0	0	0	0	0
10	6	0	0	0	0	0.14	0
11	2	0	0	0	0	1	1
12	3	0	0	0	0.5	1	0.5
13	0	0	0	0	0	0	0
14	45	0	0.16	0	0	0.83	0.04
15	0	0	0	0	0	0	0
16	1	0	0	0	0	0	0
17	4	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	3	0	0	0	0	0	0
20	1	0	0	0	0	0	0
21	1	0	0	0	0	0	0
22	1	0	0	0	0	0	0
23	1245	0	0.31	0	0.14	0.04	0
24	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0
26	926	0.01	0.6	0	0.1	0.4	0
27	680	0	0.71	0	0.07	0.56	0.22
28	287503	0.55	0.99	0.99	0.99	0.99	0.99

Table 7:- Comparison of class-based F-measure values

The k-nearest neighbor algorithm was able to correctly identify instances belonging to the minority classes. k-NN is actually a lazy classifier and postpones the classification task until the unknown instance is provided. The prediction is based on the consensus from ‘k’ similar training instances. It may be observed that k-NN correctly identifies attack classes better than MLP and Naïve Bayes Classifiers. Random Forest is an ensemble formed from base classifiers of decision trees. The weak learners are trained on data obtained by applying random subset sampling on the set of instances and features as well. This process ensures least correlation among the constituent decision trees. The class imbalance did not have as much effect on Random Forest as it did on Naïve Bayes, MLP and k-NN algorithms. It may be observed that Gradient Boosting classifiers return best results among the classifiers considered. Both Gradient Boosting and XGBoost algorithms use sequential process such that every time an instance is incorrectly classified, more focus is provided to such instances. From the results presented, it may be discerned that in datasets possessing extreme order of class imbalances, the class of Gradient Boosting algorithms may provide better classification results to the minority class.

V. CONCLUTIONS

This paper investigates the effect of different supervised machine learning approaches for effective identification of Bitcoin payments for Ransomware perpetuators. dataset considered is a multi-class extremely imbalanced in nature. Results on different evaluation metrics indicate that the Gradient Boosting and XGBoost algorithms correctly identified more of the attack classes than other classifiers considered namely Naïve Bayes, Multi-layer Perceptron, k-Nearest Neighbor and Random Forest Classifiers. The findings of the algorithms need further exploration on other datasets having extreme class imbalances as well. More emphasis may also be provided to classifiers which consider the representatives of the minority classes from the training data for making better reductions. Future work may also be done to validate the results on more recent spurious bitcoin transactions involving cybercrime such as ransomware payments and money launder

REFERENCES

[1]. Cuneyt G. Akcora, Yitao Li, Yulia R. Gel, Murat Kantarcioglu (2020). BitcoinHeist: Topological Data Analysis for Ransomware Prediction on the Bitcoin Blockchain. IJCAI 2020: 4439-4445J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[2]. K. Liao, Z. Zhao, A. Doupe and G. Ahn, "Behind closed doors: measurement and analysis of CryptoLocker ransoms in Bitcoin," 2016 APWG Symposium on Electronic Crime Research (eCrime), Toronto, ON, 2016, pp. 1-13, doi: 10.1109/ECRIME.2016.7487938. K. Elissa, “Title of paper if known,” unpublished.

[3]. M. Conti, E. Sandeep Kumar, C. Lal and S. Ruj, (2018).(2016)."A Survey on Security and Privacy Issues of Bitcoin," in IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 3416-3452, Fourthquarter 2018, doi: 10.1109/COMST.2018.2842460.

[4]. Turner, A.B., McCombie, S. and Uhlmann, A.J. (2020), "Discerning payment patterns in Bitcoin from ransomware attacks", Journal of Money Laundering Control, Vol. ahead-of-print No. ahead-of-print. https://doi.org/10.1108/JMLC-02-2020-0012.M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[5]. D. Y. Huang et al., "Tracking Ransomware End-to-end," (2018).2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, 2018, pp. 618-631, doi: 10.1109/SP.2018.00047.

[6]. Alhawi O.M.K., Baldwin J., Dehghantanha A. (2018) Leveraging Machine Learning Techniques for Windows Ransomware Network Traffic Detection. In: Dehghantanha A., Conti M., Dargahi T. (eds) Cyber Threat Intelligence. Advances in Information Security, vol 70. Springer, Cham. doi: 10.1007/978-3-319-73951-9_5

[7]. S. Poudyal, K. P. Subedi and D. Dasgupta, "A Framework for Analyzing Ransomware using Machine Learning," 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 2018, pp. 1692-1699, doi: 10.1109/SSCI.2018.8628743.