Using Machine Learning to assist users in making decisions while trusting cryptocurrency addresses for online purchases

Suraj Shamsundar Jain
Department of Computer Science
Texas A&M University
College Station, United States
surajsjain@tamu.edu

Reid Oboyle
Department of Computer Science
Texas A&M University
College Station, United States
oboylere@tamu.edu

Sankrandan Loke
Department of Computer Science
Texas A&M University
College Station, United States
lokesankrandan@tamu.edu

***Abstract*—Cryptocurrency fraud is an increasingly important problem in today's society. Because cryptocurrencies are decentralized and anonymized, it makes it very difficult to know the legitimacy of cryptocurrency addresses. There are an increasing number of scams using cryptocurrencies occurring and there is no easy way to know if you can trust a cryptocurrency address that we see on websites. The model proposed in this paper aims to help the users decide whether to trust a particular Bitcoin or Ethereum address that they see on the internet, especially on e-commerce sites.**

***Index Terms*—Cryptocurrency, Bitcoin, Ethereum, Transaction, Security**

I. Introduction

Most of the e-commerce websites that allow users to make purchases with cryptocurrency payments on the internet today follow a checkout procedure where a particular cryptocurrency address is displayed to the user and the user is asked to transfer the purchase amount to that address for them to process the order. There are also a growing number of phishing websites that are leveraging cryptocurrencies as the main payment method in order to maintain anonymity and get away with crimes.

Also, it is not possible to shut down a particular cryptocurrency address as it is decentralized. So, the most popular mitigation to this problem today is maintaining websites such as bitcoinabuse.com and cryptoscamdb.org, where users can report a particular cryptocurrency address after they have been scammed by a website using it. However, users of the internet today either do not know about the websites where the scams are listed, or they just do not bother to check if a particular cryptocurrency address is listed in a scam record database. Also, there might be times when the cryptocurrency address that the user is looking at on a website is not listed on a scam record database but might be linked to an address that is listed on the scam record database or another website that looks like a phishing website. People can also get into serious trouble if the account that they are sending the amount to has been blacklisted by the government agencies as they might be belonging to terrorist organizations or nations (such as North Korea) that are deemed to be dangerous to the national security of the home country of the user.

Therefore, in this project, we will be helping the users in making decisions whether to trust a Bitcoin or an Ethereum address that they are seeing on a webpage or not to trust them by having a browser extension that shows them the reasons to trust an address, as well as the reasons not to trust an address that they are viewing on the web page that they are on.

II. Overview of the methodology

At first, the browser extension will detect if there is a Bitcoin or an Ethereum address on the web page that the user is viewing. If there are Bitcoin or Ethereum addresses, they will be sent to the back-end.

The back-end will serve the following three functions:

- Predicting if a particular address is fraudulent or not, based on its transaction activities.
- Checking if the address displayed to the user is present on another website, and if it is, a model will be run to predict if the website has a negative sentiment or not. (The website might have a negative sentiment news)
- Performing a custom graph search algorithm to check the other accounts linked to the account number that the user is viewing and perform the first two operations above on them as well.

Therefore, there will be two different machine learning models used - one model for predicting if the account looks suspicious based on the transaction activities, and another model for predicting if a website has negative sentiment news if the requested address or an address linked to the requested address is found to be present on the website.

III. Transaction activity-based fraud detection in cryptocurrencies

We found a labeled dataset on Kaggle that contained account numbers, various statistics related to the transactions made by that account and labeled the accounts to be fraudulent or not [1]. We wanted to build a model that would work on both Ethereum as well as Bitcoin just by converting the currency values using the exchange rates. Therefore, we trained a random forest classifier with 100 estimators and a maximum depth of 10 to predict if an account is fraudulent or not, based on the details of the account provided.

A. Description of the dataset and training the model to predict Ethereum fraud

The dataset contained many features that are present on Ethereum accounts and not Bitcoin. This is because Ethereum supports smart contracts and Ethereum users can also hold and make transactions in other ERC20 tokens launched by organizations or communities for interacting with their smart contracts, whereas this is not possible with Bitcoin. So, we trained our first model by only considering the features in Ethereum that are common to Bitcoin. These features include the basic transaction summary statistics such as average time between the outgoing transactions, the average time between the incoming transactions, the time difference between the first and the last transactions (to tell how long the account has been active for), the number of unique addresses that the account has made incoming and outgoing transactions with, the current balance in the account, and the minimum, maximum and the average amounts for the incoming and outgoing transactions.

B. Testing the model's flexibility to also detect Bitcoin fraud

The free Bitcoin APIs from blockchain.com and python blockchain packages were used to get the input features required for the model to make the predictions. Scripts were written to generate the values for the features required by the model when a bitcoin address is provided to it. Then, the features that involved Bitcoin currency values were converted into Ethereum values using the exchange rate.

When it comes to testing the model, we downloaded the "Bitcoin Heist ransomware" dataset from UCI and clustered the data using k-means classification. We had planned to use the Bitcoin wallet addresses from the Bitcoin Heist dataset, use blockchain.com's API and the custom scripts to get the features required by the first model, classify the wallet addresses to be fraudulent or not using the first model, and compare the labels generated by the first model to the cluster that the address is associated with using k-means. However, when it comes to blockchain.com's API to get the required account properties, the free version of the API had a limit to the number of requests that could be made to it by a particular IP address within a given timeframe, after which it would block the IP address from making the API calls for a while. Therefore, we had to use a python package to collect the transaction data on cryptocurrencies.

IV. Checking the presence of the wallet address on websites other than the one that the user is looking at

To do this for Bitcoin, we came across a website called bitcoinwhoswho.com. This website does not have an API but, if we enter the wallet address in the search bar of the website, it walks us to a page whose URL will be like www.bitcoinwhoswho.com/address/<searchAddress>/. Therefore, we used the urllib in Python and read the contents of the webpage and extracted URLs of the website appearances from the table in the resulting webpage of bitcoinwhoswho.com using beautifulsoup. After extracting the URLs, we got the HTML webpage content for each URL using urllib. After getting the HTML content of the web pages, we used beautifulsoup again to remove all the HTML tags to get the content of the website.

A. Sentiment analysis on the web-page contents

We then perform sentiment analysis on it using the NLTK library. NLTK already has a built-in, pretrained sentiment analyzer called VADER (Valence Aware Dictionary for sEntiment Reasoning). We used VADER to determine the sentiment of the contents of the website. We extracted the sentiment analysis for each webpage's 'title' and 'body' sections separately. The sentiment analysis of the body is done by aggregating all the sentiment analysis scores (non-zero negative scores and non-zero positive scores) of each sentence in the body section. NLTK tokenizer and Regex filters were used to separate individual sentences. If the negative score in either of the 'title' or 'body' sections crosses a threshold, we flag the site to the user. Since VADER is pretrained, you can get results more quickly than with many other analyzers.

We tested the contents of the webpage against this sentiment classification model. If the webpage is a news article and it's negative, it means that it might be dangerous for the user to send money to the address. The positive and negative scores obtained from the model lie between 0 and 1. With the thresholds as 0.1 for the title section and 0.3 for the body section, we were able to identify almost all such websites.

V. Exploring related wallet addresses using graph search algorithms and providing the confidence score to the user

For Bitcoin, we can get the transactions associated with a wallet address that can be found using python blockchain packages. For Ethereum, we can use the APIs provided by etherscan.io to get the transactions associated with an account.

We perform a controlled depth-limit search by selecting neighboring addresses in order of the total amount transacted between the accounts and do a depth-first search on the selected neighbor. For production purposes, the code will be made open-source and if any businesses want to put it to production, they can set the depth to be explored as well as the total number of transactions to consider per address, based on the computing power of the servers.

Whenever we come across a new address, we run the first model to determine if it is fraudulent, based on transactions that it has made, and after that, we check for its presence on other web pages across the internet. If any of the linked addresses in the search are present on any other websites and the website is found to be legitimate or contains an article with a positive sentiment (ex: a travel blog where the author has asked for donations in Bitcoin), we assign a positive score to search. But if we come across a website that is determined to be a phishing website, we assign a negative score to the search and alert the user about it.

In the end, we display all the links of the wallet address that the user is looking at, the other websites that the address that they are looking at may be related to, as well as if the other websites are good or bad. In the future, this will be displayed on the browser extension so that the user can also look at the other related websites and then decide whether to trust the wallet address or not.

VI. Results

A. Testing the model on Ethereum Fraud

20% of the dataset was reserved for testing the model, and the model has an accuracy of 94.66%, a precision of 95.04% and a recall of 98.31% on the test dataset containing Ethereum addresses. The confusion matrix for the model's performance on the test set is shown in the figure 1 and the detailed classification report is shown in figure 2.

B. Testing the Model on Bitcoin Fraud

Because the bitcoin dataset we used was not labeled there was no way to clearly test our model accuracy on the dataset. However, in 2020 one of the authors was a victim of trusting the wrong wallet to hold the bitcoins and all the bitcoins were stolen from him. So, the properties of all the accounts linked to the robbery were passed to the model. The accounts included the account owned by the exchange where the author got his Bitcoins from by trading Ardor (exchange account), the author's account (legitimate account), the account number of the account where the stolen bitcoins went to and the accounts in the chain the amount in the hacker's account was transferred to (fraudulent accounts). It was found that the model classified the account that stole from the author, as well as all the other accounts where the bitcoins were passed to from that account (also owned by the same hacker group) were flagged to be fraudulent by the model. The accounts that were owned by the exchange, the author's account, and the last account where the author's stolen bitcoins were transferred to were labeled not to be fraudulent. The last account where all the stolen bitcoins were transferred has just one incoming transaction. Although this is one scenario we could confirm, we believe that our model would be accurate on bitcoin because it shares the same features as our Ethereum dataset and those features are good indicators of fraud in both bitcoin and Ethereum.

VII. Related Work

When it comes to other related works, it includes some work of researchers who have done individual parts of what we are doing in this work. For instance, in the paper by "Unsupervised learning for robust Bitcoin fraud detection" by Monamo, P., Marivate, V. and Twala, B. [8], they use k-means clustering to detect whether a particular bitcoin address is fraudulent or not. One problem with their approach is that they can only run their model on a dataset with a given number of transactions. In our model, our approach can find new transactions in real time because it uses blockchain API's to traverse through the live blockchain map. We hope that our model will be able to find hackers quicker than previous approaches that rely on a static dataset. There are similar works on Ethereum using random forest such as [9], and also works such as [10] for detecting Ponzi schemes deployed as Ethereum smart contracts on the blockchain, which when users interact with will be subjected to fraud. Although [9] and [10] are good solutions for ethereum fraud, our approach will be able to run detection on both bitcoin and ethereum. Just using ethereum limits the number of hacks they can detect. By running analysis on the two most popular cryptocurrencies we hope our model can detect a majority of hacks in the blockchain.

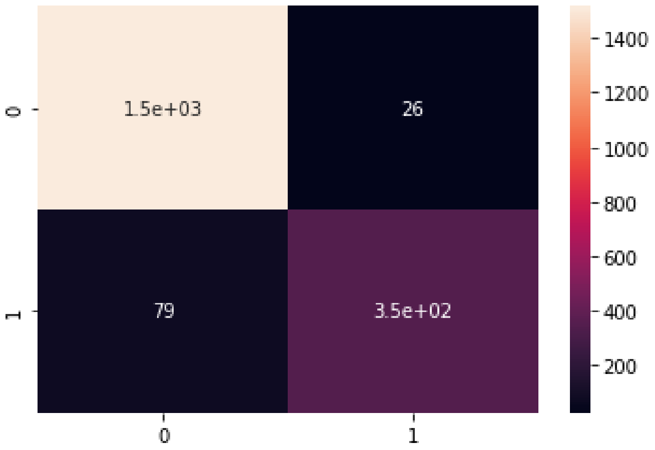
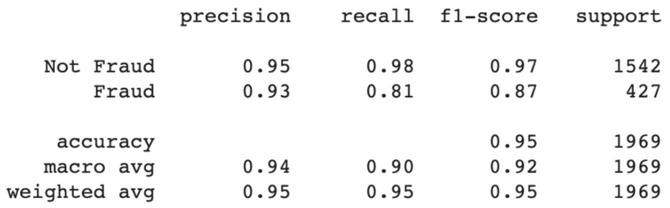


Fig. 1. Confusion matrix for the test set results of the random forest model used to classify the fraudulent Ethereum accounts.



```
              precision    recall  f1-score   support

   Not Fraud       0.95      0.98      0.97      1542
       Fraud       0.93      0.81      0.87       427

    accuracy                           0.95      1969
   macro avg       0.94      0.90      0.92      1969
weighted avg       0.95      0.95      0.95      1969
```

Fig. 2. Detailed classification report for the test set results of the random forest model used to classify the fraudulent Ethereum accounts.

There have been works that utilize a graph structure to parse through cryptocurrency transactions such as in [13]. Researchers use groups of known hacking addresses to classify new addresses. They look at addresses from subnetworks of the hacking groups and compare different attributes among them. They then hope to be able to classify and group new addresses as hacking or not. This research mainly focuses on further expanding hacking groups based on an already known hacking group. Our work focuses more on finding new addresses and classifying it based on our model's prediction rather than classifying it to a group of hackers. We believe we will be able to find more undetected suspicious addresses using our approach.

There are also a couple of other works on sentiment analysis and malicious website classification using CNNs and URL features such as [11] and [12], but there has been no other work that combines cryptocurrency fraud detection with exploring the links to other accounts, their presence on websites, and classification of those websites as far as we have explored. Therefore, we believe that this is a "first of the kind" work on cryptocurrency fraud detection with link exploration.

VIII. Conclusions

Throughout this work we have identified multiple three different effective methods for identifying a fraudulent cryptocurrency. We implemented these methods and our results confirmed that these are effective in detecting cryptocurrency fraud. In future work, we would like to implement a browser extension that can scan a cryptocurrency address and use these three methods together to decide the legitimacy of a given address. We believe that putting these into one browser extension could help a user confirm legitimacy before sending money to a cryptocurrency address. In doing so, we could help decrease the amount of cryptocurrency fraud occurring on the internet and prevent users from scams. Cryptocurrency is becoming more popular every day and the need for a tool like this increases more and more. Because blockchain transactions are freely available on the internet, our model can learn from new data and become more accurate as more information is discovered. There is a lot of opportunity for future work on this project and we believe this project could eventually become a very valuable tool on the internet.

References

[1] Aliyev, Vagif. "Ethereum Fraud Detection Dataset." Kaggle, 3 Jan. 2021, https://www.kaggle.com/vagifa/ethereum-frauddetection-dataset.

[2] UCI Machine Learning Repository: Bitcoinheistransomwareaddressdataset Data Set, University of California-Irvine, 2019, https://archive.ics.uci.edu/ml/datasets/BitcoinHeistRansomwareAddressDataset.

[3] "Bitcoin Abuse Database." Bitcoin Abuse Database, https://www.bitcoinabuse.com/.

[4] "Home." Crypto Scam DB, https://cryptoscamdb.org/.

[5] Anon, Bitcoin Address Lookup. BitcoinWhosWho. Available at: https://www.bitcoinwhoswho.com/ [Accessed November 17, 2021].

[6] The most trusted Crypto Company. Blockchain.com - The Most Trusted Crypto Company. (n.d.). Retrieved November 17, 2021, from https://www.blockchain.com/.

[7] Ethereum API endpoints. Etherscan. (n.d.). Retrieved November 17, 2021, from https://docs.etherscan.io

[8] Monamo, P., Marivate, V. and Twala, B., 2016, August. Unsupervised learning for robust Bitcoin fraud detection. In 2016 Information Security for South Africa (ISSA) (pp. 129-134). IEEE.

[9] Jung, E., Le Tilly, M., Gehani, A. and Ge, Y., 2019, July. Data mining-based ethereum fraud detection. In 2019 IEEE International Conference on Blockchain (Blockchain) (pp. 266-273). IEEE.

[10] W. Chen, Z. Zheng, E. C. . -H. Ngai, P. Zheng and Y. Zhou, "Exploiting Blockchain Data to Detect Smart Ponzi Schemes on Ethereum," in IEEE Access, vol. 7, pp. 37575-37586, 2019, doi: 10.1109/ACCESS.2019.2905769.

[11] Liu, D. and Lee, J.H., 2020. CNN based malicious website detection by invalidating multiple web spams. IEEE Access, 8, pp.97258-97266.

[12] Aldwairi, M. and Alsalman, R., 2012. Malurls: A lightweight malicious website classification based on url features. Journal of Emerging Technologies in Web Intelligence, 4(2), pp.128-133.

[13] Goldsmith, Daniel, et al. "Analyzing Hack Subnetworks in the Bitcoin Transaction Graph." Applied Network Science, vol. 5, no. 1, 2020, https://doi.org/10.1007/s41109-020-00261-7.