TRACKING MALICIOUS TRANSACTIONS IN CRYPTOCURRENCIES

Bhavish Dhanda
Supervisor: Dr Hassan Asghar
Co-Supervisor: Dr Benjamin Zhao

Department of Computing and Engineering
Macquarie University
Australia

ACKNOWLEDGEMENTS

I would like to acknowledge and give my warmest thanks to my supervisor Dr Hassan Asghar and co-supervisor Dr Benjamin Zhao who made this work possible. Their guidance and advice carried me through all the stages of writing my project.

STATEMENT OF CANDIDATE

I hereby declare that the work, which is being presented in the Thesis, entitled "Tracking malicious transaction in Cryptocurrencies", in partial fulfillment of the requirement for the award of the degree of Bachelor of Software Engineering (Honours) in the department of Engineering, Macquarie University. This thesis is an original piece of research work under the guidance of Dr Hassan Asghar and Dr Benjamin Zhao. The matter embodied in this thesis has not been submitted for the award of any other degree of any other academic institution.

Contents

List of Figures

Listings

Abstract

Cryptocurrencies form what is called the new decentralized finance world of digital finances. They have become increasingly popular in the recent few years and with this technology becoming more and more accessible, the number of users of this form of payments have grown exponentially. On of the other side of coin, this mass usage of this technology gives a small percentage of people of hide in plain sight and do illegal activities. These people who are involved in such malicious activites are also supported by the anonymity features of this form of payments. In this paper we take a deep dive into this illegal usage of cryptocurrencies and and how we can potentially track this down.

1 Introduction

Cryptocurrencies have become the modern form of decentralized finance giving the general public a level of visibility and control over financial systems that we could never imagine while using traditional financial systems. Their adoption has increased exponentially in the last few years with a few countries even deciding to make it a legal tender for their day to day transactions. Such countries include but are not limited to Central African Republic, El Salvador [1] where Bitcoin has been deemed as a legal form of tender by the local authorities.

The striking difference between so-called decentralized finance[2] formed by cryptocurrencies and traditional banking systems the type of ledger they maintain, and how the currency is controlled. Decentralized as the word suggests means that the control is not the in hands of a single person or entity rather the entire network, in this case, all the people using and servicing the cryptocurrency influence all the decisions that are made which include, the price largely controlled by supply and demand and the number of people mining and other factors, validation of transactions is normally done by validation nodes inside the networks and similar functions all are done by the public nodes inside the network. The second major factor that contributes to this decentralization is the public ledger these currencies maintain in contrast to a centralized secured ledger maintained by traditional banking systems where only certain authorized users can access the ledger, in the case of cryptocurrencies, anyone can access the ledger and see all sorts of information stored in a typical currency ledger which includes account balances, transactions made etc.

Now with this increased popularity there is a growing problem of illegal use of cryptocurrencies both through the dark web markets and also through direct P2P transactions. Sesha Kethineni and Ying Cao [3] have made substantial progress bringing the issues we are trying to tackle into the public domain. They explain how modern cryptocurrencies such as Monero[4], Dash[5] and a few others who are built upon existing currencies with the sole purpose of improving privacy are being more and more used for illegal activities. Along with that they explain how some countries as planning to introduce cryptocurrencies as their mainstream form of payment. As much as it seems like this move this move is disliked by the more powerful countries like US, emerging countries see it as a way of achieving independence from the US payment systems and having their own influence and scrutiny in their economies. But this increasing accessibility gives people who are involved in illegal activities the perfect opportunity to hide in plain sight.

> **Our goal in this thesis is to find ways to track down malicious transactions and to try and get as close as possible to the real world entity behind these activities.**

In the following sections we will go through everything mentioned above in detail starting with background about cryptocurrencies and the problem itself, followed by reviewing some work already done by researchers in this field to

understand what can we learn from their work, and then we will discuss some ethical considerations. In the last few sections we will go through our methodology and the reasoning behind including some decisions made and why we made them, after which we will do some critical analysis of this work, followed by some ideas of potential future work people can undertake taking into account lessons learnt from our research and finishing with concluding remarks about the thesis.

2 Background

2.1 What are cryptocurrencies?

Cryptocurrencies are a modern form of digital decentralized currencies which aim to put all the control of a currency and how it works in the hands of the consumers. The distinguishing factor between the conventional currencies we normally use and crypto is first its operation of a public ledger and second decentralized control. Now, every currency needs to maintain a ledger to keep track of the consumers, their activites i.e. the transactions, their account balances, in traditional banking systems all of this information is confidential and only a select group of people have access to it whereas in the cryptocurrencies everything is public, anyone can download the blockchain on their systems or access it through a number of APIs available these days to look at this information. However with all this information, it doesn't mean that you know how much money someone has because cryptocurrency ledgers hide their users' identity behind wallet addresses and there is no direct link between the real work identity of a person and his wallet address. So you can check how much money there is in a wallet address, the transactions the address has made but cannot check who operates/owns the wallet address.

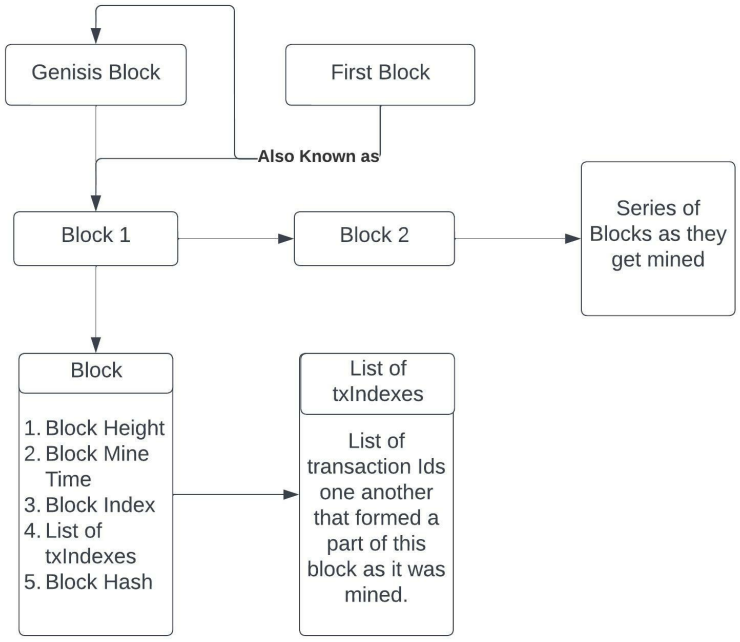

Figure 1: A simple blockchain data structure

2.2 How do cryptocurrencies work?

Before we actually start exploring our area of research we need to be aware of how cryptocurrencies technically operate. To simplify things we will be using Bitcoin as our reference to explain the technical architecture. As said earlier all currencies need a ledger to work so in case of cryptocurrencies it is the blockchain itself. Blockchain could is essentially a Linked List of Blocks. Each Block inside the blockchain is a collection of transactions with their associated metadata. Modern cryptocurrencies work on maintaining a public ledger which is the distinguishing factor of these currencies with the standard paper currencies that we have been using for ages.

In terms of actual decentralized operation of the currency, it works on the concept of a P2P network, that means in order to make a transaction on the currency you can connect directly to the other party without the oversight/involvement of a third party in the middle. Apart from that a large number of nodes/clients join the P2P network to host the currency and perform additional tasks such as mining, validation in return for some reward which depends on the currency itself. This is how the decentralization is enable the transactions are not managed or approved by a single entity or person rather a network of nodes on the P2P network known as validator nodes, same goes for miners, there are nodes on the network that perform mining and they are interacting directly with the blockchain so this eliminates the need to having to keep control of the currency in the hands of a single person or an entity.

2.3 How do cryptocurrencies preserve privacy?

Cryptocurrencies work on the concept of wallet addresses which is similar to having a bank account at any traditional financial institutions. Wallet account is the bank account number you get when opening up a bank account for the first time. That means any future transactions, enquiries, would be using that account number as a reference. In the same way cryptocurrencies use the wallet address for any transactions, also as your identity on the currency network and ledger to store information.

The point at which the difference arises between a traditional financial institution like a bank and a cryptocurrency is where the account number / wallet address is generated. Normally a bank is required to perform know you customer also known as KYC steps before they can issue an account number to validate the person's identity who is opening the bank account but in cryptocurrencies there is no such validation. A user can just create his own wallet address by providing no information at all and this way he can issue as many wallet addresses for himself or anyone, in fact users of cryptocurrencies are actually encouraged to create a new wallet address for every transaction they perform on the blockchain to maximise their privacy.

This way when a customer has wallet address all his interactions to and within the blockchain would be through the address which has no information at all linked to his real world personal identity. He can do whatever he wants on

the blockchain but all that would be left in the end is the wallet address which in no way shape or form would have enough information to even closely link to his real identity. But in contrast if this link is revealed by any way possible such as leaking of this information, the customer is completely exposed meaning, everyone can see how much money the user holds, how many transactions were made and with whom and everything linked to his wallet address. So, not only it is the target account that is exposed but also every other account that the target account has interacted with. This way cryptocurrencies are known to be pseudo-anonymous as they are normally and also ideally externally anonymous but in case a leakage of information happens, it will not be able to preserve any privacy at all.

2.4 What are Mixers?

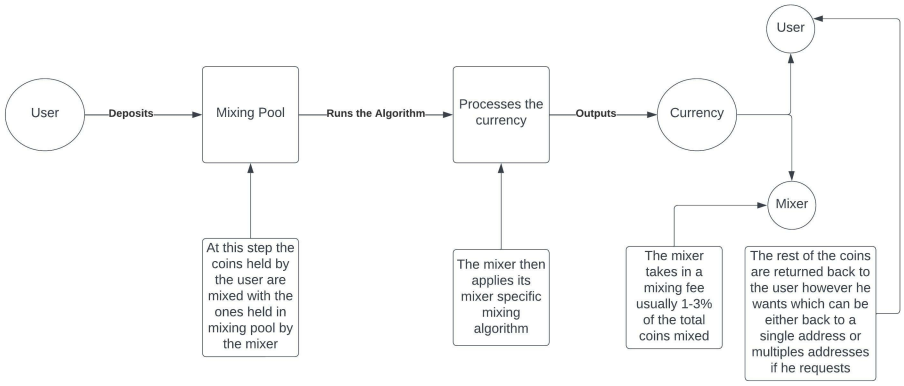

Figure 2: Working of a basic mixing service

Now that we are across how cryptocurrencies work, it is important we discuss about mixers as well as they introduce another dimension of complexity in the problem we are trying to solve.

Mixers were developed with the sole purpose to improve the privacy of users using the blockchain and reduce traceability of transactions. Mixers as the word suggest perform mixing, what they mix is the addresses with others so that there is no direct link between the source and target address thereby giving the appearance that the money in the target address came from the address generated by the mixing service. We also need to keep in mind that by using mixers the traceability of the source to the target decreases but is never eliminated so these can in fact when searched extensively can be broken through to some extent, we will cover an example of such work in the related works section about mixers below.

The way mixers work as highlited in fig 2 is

1. The user transfers his currency to be mixed from his wallet to the mixing service's wallet. At this point all the user has is a withdrawal note from the mixer which indicates that the user holds funds in the mixing pool. Now this note can differ from case to case on how the user requested the funds to the be dispatched. For instance some users do not want their funds to be released immediately so that hold on to that note until they want them back, some want their funds to be sent to a different address and some mixing services also give the option to send money to multiple output address so that is what the withdrawal note states.

2. After the money is deposited in the mixing pool, it becomes a part of the mixing pool with no direct association to the person who deposited the money. Now at this stage the mixing pool applies it mixing algorithm which is usually unique to the pool itself with the end goal of reducing traceability.

3. Now that the mixer has done his job the output currency from the mixing algorithm is supposed to be sent back to the user. This is the point where the mixer deducts its fee which is usually 1 - 3 percent and sends back the rest of the amount to the user wherever he requested them.

2.5 What are Dark Web Markets?

To understand the role of cryptocurrencies in malicious activity it is important we understand what dark web markets are, and how they work to preserve the privacy of its users. Dark markets are essentially online shopping platforms like Amazon, Alibaba, etc but are used and developed for illegal activities like buying and selling illegal goods and services. Dark Web Markets are extremely popular amongst people trying to hide their online presence because they can only be accessed behind a hidden network and a private browser which ensures there is no leakage of information. These are usually access through The Onion Router also known as TOR[6] which is basically an open source software for enabling anonymous communication. The malicious activity on these networks has only increased after the launch of cryptocurrencies, as most of them have now started to accept cryptocurrencies, breaking further the single point where there could be traceability. Normal online transactions can be traced by the financial institutions where the account belongs, one can be caught performing cash transaction but introducing cryptocurrencies breaks this final link as well.

These are an extremely old type of illegal markets which have managed to remain in circulation even after various blow-backs from the authorities by having a successor every time one is taken down. Silk Road[7] was one such example which was one of first marketplace of such kind, which was founded in 2011 by a group of group of people, to only a select group of people and later expanded into the wider audience. As silk road was opened to the wider audience its number of users and the income increase as well along with sparking an interest of the authorities towards this. Eventually in 2013, this was shut down by the authorities but the concerning fact was that this did not give a

lesson to the people to stop these services instead this lead and inspired to people working to launch Silk Road 2.0 shortly after. Another example of such service which came in the highlights was AlphaBay which was founded by Christin who was the one of the people who originally launched Silk Road. As highlited by Hiramoto Et al[8] in their research that AlphaBay was the second biggest dark web market during the period it was active based on its total revenue. And out of this revenue it was estimated that there were around 450 millions worth of 4 million transactions just in Bitcoin and introduction of mixing services further fueled this popularity of the market.

Hiramoto Et al[8] describe in their research how serious is this problem of malicious transactions on these dark web markets indicated by the facts that these markets, Silk Road and AlphaBay which were active only for roughly 2 and 3 years respectively had a revenue of 200 and 300 million US dollars. To top that off Bitcoin accounted for 99.8 percent of transactions and 80 percent of those transactions were for malicious purposes such as ponzi schemes, money laundering, etc.

2.6 What are malicious transactions?

In this thesis we will at times refer to a transaction or address being malicious. What we mean when are saying that an address or transaction is malicious, is that the particular address or the addresses involved in a particular transaction are trying to do something illegal. These illegal activites can include money laundering, malware, scams, ponzi schemes amongst other ways to cause harm to the general public. The basic intention over here to signify that the entity controlling the address or performing a transaction is doing something which would be deemed illegal in the context of local laws and is thereby prohibited by law.

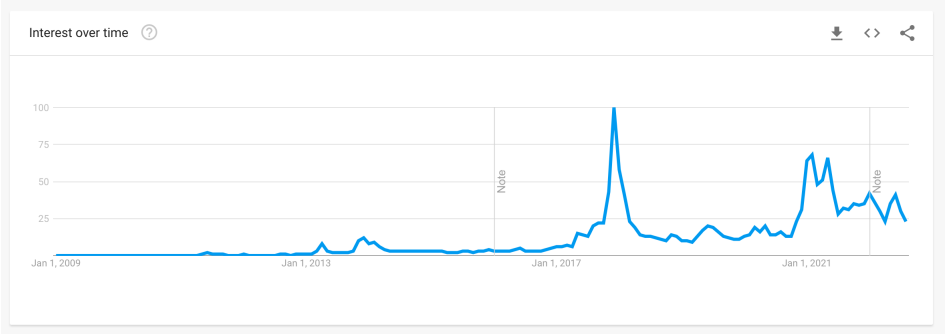

Figure 3: Popularity of BTC in Google Searches

Figure 4: Price trend of BTC

2.7 How serious is this problem

As mentioned earlier this issue of illegal use of cryptocurrencies is becoming worse and worse every with a sharp increase in the last few years and also a sharp increase expected in the next couple of years due to the increasing popularity of these services. Moreover this problem is only getting worse with the introduction of privacy coins whose sole purpose is to preserve the privacy of users with the downside of a bit higher fees but this makes our point clear that a lot of people are working towards even strengthen privacy in cryptocurrencies with the main motivation being to be able to perform illegal activities on these currencies with a low chance of getting caught.

As we can see in the figure how the relative interest of BTC rose from its early days. There were a very few early investors in BTC which looking at the current prices yielded extremely good results. But as the service became more and more popular, the supply and demand rule kicked in, increasing the price of the currency. This increase in price in turn attracted even more people to use cryptocurrencies. This was further fueled by the launching of trading platforms like Binance[9], Coinbase[10] which enabled basically anyone with basic internet access to use cryptocurrencies. This increase didn't really increase criminal activity directly but indirectly it resulted in a wider population available for malicious users to target and thereby increasing their probability of hitting a jackpot i.e. successfully scamming someone.

Kethineni Et al[3] in their paper describe the seriousness of this problem and how is this only growing. The head of Europol[11], Rob Wright estimates that 3 to 4 percent or 3 to 4 billion Euros of criminal activites are financed through crypto-currencies. Just by the time these currencies were becoming popular, in 2018 crypto-mining attacks saw a 1000 percent surge which alerted the authorities even further. Out of this there is a little bit of hope as there were a few instances where authorities were actually able to trace back to the

person responsible for conducting these activities, highlights of these being Amit Bhardwaj from India who ran a Ponzi Scheme for a value close to 300 Billions US Dollars under three companies, GB Minors, Gain BTC, and GB2. The other case was of Trendon Shavers, the founder of BTC Savings and Trust, who in short ran a Ponzi Scheme eventually getting fined with 40 Million US Dollars and eventually sent to prison for 18 months. To make things worse, as highlighted in their paper, there have been cases in Thailand, Turkey and New York amongst other places where people were extorted to transfer money to the criminals in BTC which ranges from a few hundred thousand to even millions in some cases. And finally this decentralized finance has also paved the way forward for dark web markets, which are only famous for providing illegal goods and services only.

3 Related Work

In this section we will be going through the work that has been previously done in this area of research. As we will go through this section we will realise that a lot of people are already working to break through this issue of governance in cryptocurrencies. As we saw in the last section that illegal and malicious transactions in cryptocurrencies is an ever increasing problem so its very important that some action is taken to add some sort of regulation.

3.1 Methods of De-anonymization

Before we dive deep into our own methodology of trying to de anonymize cryptocurrencies it is important we consider and study what methods other people in this field have adopted. In this section we will be going some of those and critically analyze them and use their learnings in our work.

Starting with the most basic ones Narayan Et al[12] attempted to de anonymize cryptocurrencies to some extent wherein they were aiming to identify what was the nature of the entity on the other end. What they did was simply performed transaction with a huge number of addresses which is definitely not feasible in the current landscape of any cryptocurrencies. This approach fails at a lot of places if we take into account the current situation of cryptocurrencies, which includes their prices, their popularity and finally the huge landscape of mixing and shadowing services including privacy coins out there with the sole purpose of preserve privacy of their customers.

Following this a lot of people worked on automating this process and this was widely accelerated with the advancements in machine learning. Xueshuo Et al[13] developed an address classifier to classify addresses based on their network activity. They used public APIs from blockchain.com[14] to get data directly from the blockchain. They collected data about transactions and mapped them as vectors in a graph network. After that they developed their model to look out for patterns exhibited by address and were eventually able to classify addresses into various categories like wallets, exchange services, scams, etc with a high success rate. This piece of work shows that it is in deed possible to analyze the blockchain data to group the address and target the malicious ones from there directly vastly bringing down our scope of search when we are only after malicious one.

Finally we move on to study some advanced methods of de-anonymization. Alex Et al[15] run a full fledged attack on Bitcoin, Z-Cash and Monero by exploiting the basic principles of a peer to peer network. To achieve this they developed a custom client called bc-client on top of the default BTCore[16] client with its primary purpose being data collection such log all the messages, IP addresses of the parties involved, the transactions and their related information and also the timestamps of various events which the default one doesn't do. Once they had this client running they basically observed all the activity on the network. Their basic principle behind the attack was the principle which forms P2P networks, that is all nodes need to remain in sync with each other.

So in the case of cryptocurrencies what happens is whenever a new node or transaction is registered on the currency, it has to be propagated to the entire network. So, what happens is whenever this propagation happens, in the case of a transaction for example, the first node to propagate the transaction has to be either the sender or the receiver. So, once they had identified that a transaction is malicious they could trace this propagation back to the node with initially started the transmission. Now their concept is a really good one, but if we take into consideration the situation of cryptocurrencies right now , it is next to impossible to keep track of all the active nodes as there are so many of them, and this is only getting harder and harder for all cryptocurrencies as time passes due to the increasing popularity and accessibility.

3.2 De-anonymization of cryptocurrencies

Muller et al[17] tried to achieve a similar goal as ours but in a different fashion. The team identified a couple of dark web markets and went on to hunt for the vendors of illegal services. Dark web markets are extremely popular for the sale of illegal drugs, weapons and services and people involved in such activities usually tend to use these services because of the privacy they offer and with the introduction of cryptocurrencies, they have become even more anonymous. They started with public information available on the markets, which is the customer reviews, and from there tried to deduce important information such as time of transaction, value of transaction and the vendor on the market to start with and also took any other information they found. Once they were able to estimate a rough time when the transaction might have occurred they added a buffer of 1 day and looked for a transaction in the blockchain with the same order value. Now the chances that there will be another transaction with the exact value in the specified time frame are negligibly low, so it was relatively easy to mark down the transaction. After they were able to extract the transaction, they took the destination of the money and mapped it to the vendors account.

In this research they have a done a very good job in identifying the wallet addresses of the vendors, but there are a couple of things we need to critically evaluate as well. Starting with the good, a very good evaluation they did, was they took into account the payment systems of the market and changed their algorithm to identify the vendor accordingly as some markets perform direct transactions whereas some use an escrow service for this. A big downside of their research is the the limited expansion. For instance, they are only working with a few dark web markets, and their research purely relies on the customer leaving reviews for the vendor so their approach would be way more effective in markets where customers are encouraged to put reviews and so are vendors to improve their visibility but in markets where this is not the case or reviews are not enforced or even the case where reviewing is not available at all, this approach would either have a very poor performance, or in the last case would completely fail.

3.3 Anti-Money Laundering in cryptocurrencies

With this recent surge in popularity of cryptocurrencies and the related malicious activity governments and financial regulatory authorities all over the world are working on finding ways to track down on this. Governments in some countries have even set up individual authorities and teams with the sole purpose of regulating cryptocurrencies. The most popular way to prevent this in traditional financial institutions is by imposing KYC upon as a regulatory condition. KYC also known as know your customers is a technique which basically all the centralized financial institutions perform to get as much information as possible about their customer before actually letting them inside their institution to safeguard themselves. This way they have all the information about the customer to determine their credit history, their lending power, how good of a customer they would be and finally in-case something goes down to track them down. But cryptocurrencies which form the larger part of decentralized finance were made in the first place to avoid this step and because of this any user can open as many accounts / wallets in a currency without essentially providing any information at all.

Hu Et all[18] developed a machine learning model to identify upcoming clusters of addresses which were extremely likely to be involved in malicious transactions. They started by running the basic bitcoin core client to get transaction data and parsed that into JSON format, and along with that evaluated some ground truth data which was further classified into different categories like malicious users, consumers, etc. After they collected the data they extracted basic features about every transaction such as timestamp, inputs, outputs, with 14 extended features which included but not limited to linked inputs and output UXTOs (Unspent Transaction Outputs) and used all of this to form a transaction graph. Once they had all this data ready they trained a machine learning model to predict upcoming instances of clusters of malicious addresses. In order to validate their approach they pre determined 3 clusters of addresses which were involved in money laundering and ran cycles of their algorithm multiple times each time leaving out one of the three instances and checking if they were able to identify the third one correctly. In the end they were able to do this successfully proving that it is indeed possible to automate this kind of discovery where machine learning algorithms are put in place to detect these illegal activities as there is an extremely high chance this will be missed by a human eye if one is analyzing a transaction graph manually. What we need to note is that these sort of addresses exhibit a pattern of transactions, they have a relatively higher in-degree and out-degree and a large number of weakly connected components, meaning they transaction with a lot of either single use addresses or the addresses they interact with, they avoid multiple interactions with the same one.

One such example of a government agency taking matters related to crimes in cryptocurrencies such as money laundering, scams etc is in the US where the Biden administration has set up a task force[19] with more than 150 federal prosecutors. The reason for a specialized team is as we discussed earlier that

tackling cryptocurrency related crimes require way more technical expertise and stopping and tracking them is extra effort on top. Good thing with the task force is that the official already understand the complexity of the problem as to how these sort of crimes can be spread across multiple borders, currencies. They do plan on on-boarding more people from different disciplines such as tax, criminal, civil, national security and environmental after upskilling them on this technology to tackle this problem from all possible landscapes. Along with this they have also imposed new sanctions in the recent month to bring down the number of platforms that allow anonymous transactions in cryptocurrencies. They are also working with exchange service providers, finding technical ways to track down on such offenders, one such case was when the authorities seized 30 million dollars[20] which was stolen through an online game by hackers from North Korea. This is just one example but such actions are being taken in other countries as well, if not already there at-least a lot of them are planning already how to tackle this problem.

We can thus conclude from here a lot of work is already being done to tackle this massive problem and our work in some way shape or form will give people working on this some more insights on how to get closer to the offenders.

3.4 Tracking money across cryptocurrency borders

In contrast to centralized currencies like dollar, euros, cryptocurrencies, do not have a concept of borders. What that means is that unlike dollar which e.g. is the currency officially used as a legal tender in the United States, and in other countries one has to convert it into the local currency like one would have to convert their dollars into Euros in Europe if they wish to use, cryptocurrencies have no such thing. Its a decentralized form of payment used in the same way worldwide. The only border in decentralized currency is the currency itself, i.e. when the user wants to use a different currency so he has to go through an exchange to get this done and then use it.

Yousaf Et al.[21] worked to trace cryptocurrency transaction across multiple currencies. Now this is where they explored beyond the borders of cryptocurrencies. Their research was focused on using a platform called Shape-Shift[] which allows users to perform transactions on different types of block chains and exchange between currencies as needed. They used the public API available for the platform to perform transactions and get data about past transactions as well for their validation. Their concept to track money flowing through multiple currencies was simple that whatever goes in has to come out essentially meaning that whenever a user initiates a transaction to move money from one currency to another there has to be an output transaction in the source currency and an input transaction the target currency of roughly the same value and very close to each other in time. To evaluate all this they collected data from these platforms and also ran full nodes of various currencies to validate their heuristic, having in total around 654GB of raw blockchain data and 434 GB of parsed blockchain data loaded into Apache Spark[] to analysis. They identified three patterns of transactions to look for which are pass through transactions where

the money just moves from one currency to another, second U-turns where the user sends money to a different currency and then immediately returns back to the original one, and finally round trip in which the user sends money to a different currency and then shifts back but with a diversion in the middle. Now there could be multiple reasons why one would want to do this, it could either be to benefit from the exchange due to prices of the 2 currencies or one might just think that once they move money like this the output they get is clean money as opposed to input which could be obtained from illegal activities. To identify the link between the address they identified some heuristic one being that if 2 or more addresses send or receive money in the same address either can they sent money to the same address in the output currency or receive money in the same address in the input currency then they have a common social relationship. On top of this they identified until January 30 2019, out of 6374 scams reported on EtherScamDb[] for Eutherium related scams 194 addresses roughly 9 percent were involved in 853 transactions on shape-shift out of which 688 were complete indicating the scammers were successfully able to move their money into some other currency making it even harder to track them. A total amount of 1797 Eutherium was shifted to other currencies made up by 74 percent to BTC, 19 percent to Monero, 3 percent to BTC cash and finally 1 percent went into ZCash. In their analysis they also saw how introduction of KYC saw a decline in the number of transactions on the platform strongly indicating that people were hesitant to provide their identification while performing such transactions, so they could be potentially hiding illegal activity. This work thus indicates that if expanded systematically, can be used to track money across ledgers in the world of decentralized finance. This work however is just restricted to one platform but does prove the point in target.

3.5 Breaking Mixers

Now that we are across the concept of mixers and how they work, it is important we go through some of prior work done in order to break through them and be able to trace the transactions that are going through mixers.

Wang Et all [22] worked on solving this very problem to trace the money that was going inside a mixer by first actually determining if a mixer was used in the transaction or not, and if it was they identified some common patterns of people using mixers and mapped the transaction behaviour to the patterns to get to the target address eventually. What they essentially did was pick up a source address and traced the transactions for it until they found a link back to the main address or until they found an indirect link through the value of the transaction. This did involve a very extensive search and traversal of the transactions but they were in the end able to determine if a user used a mixing service or not, and if they did what kind of mixing they perform, i.e. did they just send money to another address and receive it back or did they send money and received it to an entirely different address or even if they sent it and received it back in multiple addresses.

This piece of work does show that it is indeed possible to break through

mixers, identify them or find the link between the source and target address in case a mixing service was user. As exhaustive as this would be, it does in the end to some extent cracks through the mixers.

4 Ethical Considerations

Now that we have a good understanding of the problem at our hand and what we are trying to achieve there are a few ethical considerations we need to take into account before we actually start diving into what has been already done and what we will be doing.

The main point we need to keep in mind here is although we are working with public blockchain data most of the time, we are however trying to get to the real world entity for the associated address, and that is where these ethical concerns kick in. The reason for that is once and if we do reach the real world entity we are dealing with personal data about people which they would not want exposed to the general public. This can however can be explained in the case of malicious users as the authorities are chasing them up but the work we are doing can potentially be used for non-malicious addresses as well although results may vary.

The main takeaway from this piece of work is that we are not reaching or disclosing any personal information about the transactions or the people involved in the transactions, we are only reporting the country/city level information which we are extracting from the IP addresses and the domains from public sources which have reports about the target addresses.

5 Detailed Methodology

Since there is not much information about cryptocurrencies except the information in the public ledger we will need to look outside that scope and see what information we can find about addresses and transactions to make some justified conclusion so as to eventually draw a suspicion explaining how is the address malicious and the conclusions we are making to justify its link with the real world entity.

The aim over here is now to find a target address either directly or through a transaction and find all the information I can related to it. We have shortlisted a few sources which keep records of information reported by the public, so we will need to scrape through all that information and its related reports to see if we can find something which can identify the person itself. If not we will need to move to the neighbour address and perform a similar search to observe the behaviour of the neighbour address and draw some conclusions. Once we have found some data about the address or the neighbours we would need to analyze it and draw up a justifiable algorithm to extract personally identifiable information from it.

In this section we will be going through step by step our process. We will explain how we derived the data and then how we used it in our application and eventually how we came up with the results presented in the following section.

5.1 Selection of Data Set

For the data to be used in the experiment to check if our approach yields any results or not we rounded our radius down to the below 4 data sets and in the end used the last one due to the various reasons highlighted below, one of them being the quality of data i.e. we had a lot of meaningful data for our work in this dataset and secondly the dataset was a very recent report in our case the last 30 days before from 22 September 2022.

- Elliptic Data Set[23] [24]
- BitcoinHeistRansomwareAddressDataset[25]
- Twitter Data[]
- BTCAbuse Report Data[26]

All data sets have enough information for us to be able to perform our simulation to get data about the address and its neighbours. The elliptic data set has an advantage because of the number of features it maintains for each and every address but then again we are not performing any machine learning analysis at this point that those features would be of use for us. Also, after some manual experimentation we found that that elliptic data set does not have real data rather the addresses were masked so that made the data set essentially impossible for us to use since we are looking for real world on information

about the addresses on the internet. Moving on we used the BitcoinHeistRansomwareAddressDataset for our work but found several limitations that made us use BTCAbuse Report Data in the end.

After performing multiple simulations with the BitcoinHeistRansomwareAddressDataset data-set we discovered that the data was extremely old and was not providing good enough results for our work. For instance only roughly 2 percent of malicious addresses in the data-set had any information about them on the sources we are scraping. This got even worse when it came to their neighbours, all the addresses had roughly 4 to 5 neighbour but rarely they had any information available for us to scrape. So, due to all of these issues we decided to use a CSV report available from BTCAbuse which is known to be the largest known database of Bitcoin addresses reported as malicious.

BTC Abuse is known as the largest database of BTC addresses reported to be malicious. They have various APIs that can be used for our research, the main ones for us being first, an API to tell whether an address is reported as malicious or not, and second that one that we based our work off on is the API to get a csv of all the addresses reported as malicious in the specified time frame. For our work we used a report with 30 days worth of addresses. A snippet of this data is attached in the appendix B.1.

5.2 Data Sets

Below in this section we discuss the 2 dataset we considered in our research and why we choose one over the other.

5.2.1 The Bitcoin Heist Ransomware Address Dataset

To start with the data set is a collection of 2916697 addresses, out of which 41413 have been labelled to being involved in one or other identified scam, with the following headers and the ones of our interest specifically have been explained.

1. address : The address about which the row contains information.
2. neighbours : The number of recorded neighbours of the address.
3. income : This is the income of the address in Satoshi amount (1 bitcoin = 100 million satoshis).
4. label : This is label associated with each address to identify what is the nature of the address. 'White' was used to indicate a non-malicious addresses.

From this entire data set, we are primarily interested in the malicious ones which are 41413 out of the total roughly 3 million addresses. This however does show that this is a very unevenly distributed data set. But for the sake of our experiment we will only deal with the malicious ones as our primary goal is to extract data about addresses involved in these kinds of transactions. The dataset also has the identified neighbours of every address, identified at

time of data collection which will help us prioritize the addresses over others as addresses with more neighbours are more likely to give us better results than others. We ran a few simulations with this dataset but we weren't getting much results and upon further evaluation we suspect that it was because the dataset is extremely old.

5.2.2 BTC Abuse Data Set

In this section we go through the data returned by BTC Abuse when we queries a csv report containing all the reports filed for various BTC addresses in the last 30 days. From the API we got a CSV containing 16401 row but that does not mean we had a set of 16401 addresses as we get reports not addresses from the source. We then transformed that into a set to get a list of unique addresses which resulted in 1250 addresses reported as malicious.

In the csv they provided us with multiple columns and the ones of interest are explained below.

1. id : An id of the report submitted by the user generated by BTC Abuse
2. address : The address being reported
3. abuse-type-id : A number code given to the kind of abuse that was reported
4. abuse-type-other : The type of abuse being reported such as malware, scam, etc
5. abuser : This is the suspected entity involved in the scam reported by the person
6. description : A description of the report submitted by the user

Out of all this data the main thing we are interested is basically the address as these address are reported by multiple people as being malicious so it helps us validate our scraping as we would expect some information about the address on BtcWhoIsWho. Finally after running a few simulations we decided to go with this dataset as it was returning good results.

In the next section we will be going through how we are processing the data-set and how we are getting the information from the scraped data.

5.3 Data Processing

For the processing of the dataset we imported the csv file using a Python script to extract all the addresses from the csv into our program. Once we had those our first goal is to download all the data onto our device and then process it locally because waiting for API calls just makes the script extremely slow and unfeasible to be used at a large scale as we have highlited in the section below. Once we have all the data available locally we extract the specific scam alerts part from the scraped data we need and then perform natural language processing to get the personally identifiable data. The steps for this are explained below.

5.3.1 Downloading and Scraping HTML Files

Since we have built a web scraping script it will need the HTML pages to scrap and external websites or APIs can be extremely slow so what we did was create a script just to download the HTMLs. To put things into context we had around 1250 target address to aim for and they had a total of roughly 115000 neighbours combined. Doing this one by one was extremely slow, we tried this to start with but we were getting less than 40 downloads per around 12 hours which was not feasible at all. So we converted out code to be completely asynchronous using python's asyncio[27] library. This way we were able to improve the performance of the data collection script to almost 10000 HTMLs per hours.

Going step by step we basically used the pythons inbuilt requests[28] module to make a get call to the webpage that is rendered when we open up the bitcoinwhoiswho website for a particular page and simply save it for future. The URL for the request was simply the host followed by the address and the response was the HTML. URL https://www.bitcoinwhoswho.com/address/address Now at this point we were dealing with hundreds of thousands of addresses whose HTMLs we needed so it was not feasible as highlighted above to make this run synchronously meaning we wait for every HTML to be downloaded one by one before proceeding onto the next one. So to overcome this we implemented python's asyncio[27] library and ran the out method to get the HTMLs in the run-until-complete method of the library. This improved the results for us but was still not enough as this library had a certain timeout which we could not figure how to override and after downloading almost 10000 HTMLs it would throw a Timeout Exception which was also not ideal as then we had to wait for every run to finish and start another one. So what we did was create another script to simply run our original script again which would internally run the downloading of HTMLs asynchronously as future tasks and after we got a timeout error it would simply restart with the next batch of addresses. To have an upper limit on the execution we added a few guards such as not to download the same HTML again and also the script would only restart if I had not downloaded yet the number of files I was expected which I had computed ahead basically the sum of number of target addresses plus the number of neighbours of each address. The code for this section is in appendix A.1.

5.3.2 Extracting the Neighbours

Now that we are working on the target address we also need to know information about their neighbours, so to do that we need to know all the neighbours of a particular target address. In this section we will go through how we got the neighbours and how we used that moving forward.

For that we will be utilizing an API from Blockchain.info[14] to get basic information about the address.

Url : GET

`https://blockchain.info/rawaddr/{{address_id}}`

This API takes in the wallet address which is the publicly available on the blockchain as a path variable and gives all the information about it directly from the blockchain. The sample response is shown above from which we will observe the interactions of the address inside the blockchain.

Sample Response:

Listing 1: A single address from the blockchain


```
{
    "hash160": "5e9b23809261178723055968d134a947f47e799f
        ",
    "address": "19dENFt4wVwos6xtgwStA6n8bbA57WCS58",
    "n_tx": 10638,
    "n_unredeemed": 160,
    "total_received": 6672872758903,
    "total_sent": 6609785308209,
    "final_balance": 63087450694,
    "txs": [
        {
            "hash": "68a0b7e0ee87f5...",
            "ver": 2,
            "vin_sz": 1,
            "vout_sz": 2,
            "size": 217,
            "weight": 760,
            "fee": 0,
            "relayed_by": "0.0.0.0",
            "lock_time": 0,
            "tx_index": 5048564692439748,
            "double_spend": false,
            "time": 1660349777,
            "block_index": 749197,
            "block_height": 749197,
            "inputs": [
                {
                    "sequence": 4294967295,
                    "witness": "01200000000000000...",
```

```
                    "script": "038d6e0b0451edf6..." ,
                    "index": 0,
                    "prev_out": {
                        "tx_index": 0,
                        "value": 0,
                        "n": 4294967295,
                        "type": 0,
                        "spent": true,
                        "script": "",
                        "spending_outpoints": [
                            {
                                "tx_index":
                                    5048564692439748,
                                "n": 0
                            }
                        ]
                    }
                }
            ],
            "out": [
                {
                    "type": 0,
                    "spent": false,
                    "value": 638621401,
                    "spending_outpoints": [],
                    "n": 0,
                    "tx_index": 5048564692439748,
                    "script": "76a9145e9b23...",
                    "addr": "19
                        dENFt4wVwos6xtgwStA6n8bbA57WCS58"
                },
                {
                    "type": 0,
                    "spent": false,
                    "value": 0,
                    "spending_outpoints": [],
                    "n": 1,
                    "tx_index": 5048564692439748,
                    "script": "6a24aa21a9ed8855481..."
                }
            ],
            "result": 638621401,
            "balance": 63087450694
        },
        {
            "hash": "dd2dde28f1b7a...",
```

```
            "ver": 2,
            "vin_sz": 1,
            "vout_sz": 2,
            "size": 217,
            "weight": 760,
            "fee": 0,
            "relayed_by": "0.0.0.0",
            "lock_time": 0,
            "tx_index": 7278413324283513,
            "double_spend": false,
            "time": 1660349287,
            "block_index": 749196,
            "block_height": 749196,
            "inputs": [
                {
                    "sequence": 4294967295,
                    "witness": "0120000000000....",
                    "script": "038c6e0b0468ebf6...",
                    "index": 0,
                    "prev_out": {
                        "tx_index": 0,
                        "value": 0,
                        "n": 4294967295,
                        "type": 0,
                        "spent": true,
                        "script": "",
                        "spending_outpoints": [
                            {
                                "tx_index":
                                    7278413324283513,
                                "n": 0
                            }
                        ]
                    }
                }
            ],
            "out": [
                {
                    "type": 0,
                    "spent": false,
                    "value": 645134567,
                    "spending_outpoints": [],
                    "n": 0,
                    "tx_index": 7278413324283513,
                    "script": "76a9145e9b23809....",
                    "addr": "19
```

```
                        dENFt4wVwos6xtgwStA6n8bbA57WCS58"
                },
                {
                    "type": 0,
                    "spent": false,
                    "value": 0,
                    "spending_outpoints": [],
                    "n": 1,
                    "tx_index": 7278413324283513,
                    "script": "6a24aa21a9ed668b12b36b8b1
                        ..."
                }
            ],
            "result": 645134567,
            "balance": 62448829293
        },
        {
            "hash": "More transactions depending on the
                address"
        }
    ]
}
```


The main information we need to extract from here is the interactions of the address throughout the blockchain. As we can see from the sample response we get a list of transactions in the object though which we can loop through to get to the immediate neighbours of the address. We refer immediate neighbours to addresses which directly interact with the address though a transaction where our target address can either be the sender or the receiver. The code responsible for this function is attached in appendix A.2. What we did was in order to avoid any external API calls while processing the data, we extracted all the neighbours of the known target addresses and then stored it in a JSON file where each object had the address and a list of all its neighbours. An extract from the file is attached in appendix B.2.

After we had the data we ran the step 1 for neighbours to download their HTMLs and stored it locally for processing later on.

5.3.3 Scraping HTML files for Scam Reports

Now the next task at our hand was to extract the scams from the downloaded HTMLs to actually get the data we need for processing later on. In this section we will go through in detail how we parsed and processed the HTML file to get the section of the page we actually need.

What we did for this was utilise from Python's bs4[29] library the Beautiful Soup module which is very popular for these use cases where we need to scrape HTML files. The code for this part is attached in appendix A.8. Essentially

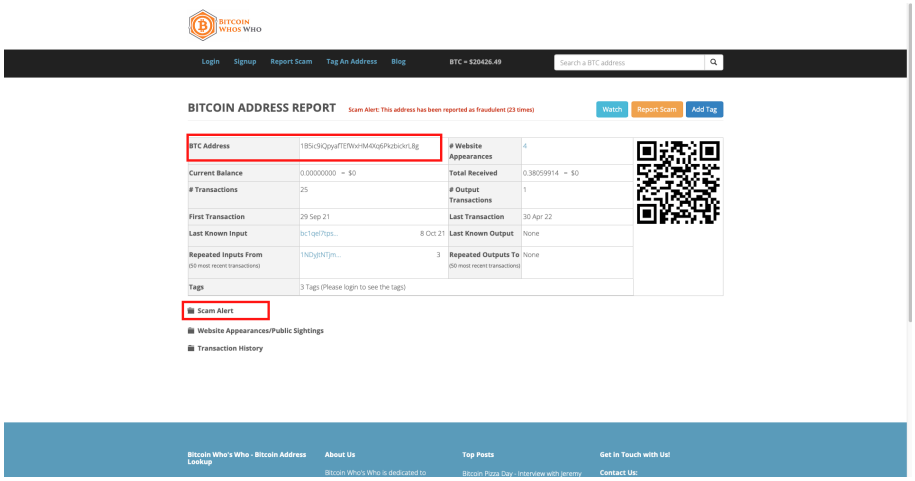

Figure 5: A screenshot from "Bitcoin Who's Who"

what we did was create a separate function into which we would pass the HTML file and it would load the file into an Beautiful Soup object from which we can go down section by section the point where we would get the scam alerts section of the page. This method alongside looping into all the tags of the pages to get to the scam alerts would also count the number of scam alerts that were reported by the user and return an object which contains the number of scam alerts and the HTML of the section which has the scam alerts which would be used later on for processing. What this method essentially does is loop through all the tags in the page until the one we have defined is reached. Whenever we call the find-all method on the beautiful soup object it returns a list of the sections that match the information passed in our case the class name or the id of tag. This way when we reach the table of the page which contains the scam alerts we evaluate that there are 2 divs for each scam alert thereby dividing by half the number of sections to get the number of scam alerts and we then add the scam alerts in a list and return it back.

5.3.4 Extracting data from parsed HTML sections

Now that we have the data available locally we need to extract the meaningful information from it. We applied multiple techniques to get all the information about an address we could from the HTMLs and evaluated it. Primarily we were interested in the data that would take us closer to the real world entity associated to the address. So for that our primary target was IP addresses and domain names followed by email address and labelled data from a natural language processing library.

1. Extracting IPs : To extract the IPs from the HTML we use regular expression matching patterns through the python's re module[30]. The code for this section is attached in appendix A.3. This returned us a list of IP addresses that were present in the report that was passed which we would use later on for reverse look-ups.

2. Extracting Domains : To extract the domains we used a similar approach as IPs where we used regular expression matching to see if we have anything that starts with an http followed by some url specific patterns to ensure we are picking the right urls. This way we would get a list of urls found in the report. We will at a later stage pass this through a Url validator to ensure we get rid of any URLs that might not be valid and also in the process we would remove the URLs that we know are trusted to reduce the False positives. The code for this action is attached in appendix A.6.

3. Extracting Emails : For email extraction also we used regular expression pattern matching. The main goal to extract emails is to see if we can get to know more about the details of the people involved in the scam which can either be the receive of the email which is the person being targeted by the criminal or if we can get the senders details that would get us closer to the criminal itself. The code for this section is in appendix A.4.

4. Natural Language Processing : This was the last part of scraped data processing we did to get as much meaningful information as possible to get to the physical entity associated with the address. For this we utilized Python's spacy[31] library which is one of the most popular natural language processing libraries available with already available pre-trained models. Out of the available models we used 'en-core-web-trf' and what this does internally is tokenize the entire string of data into individual tokens and then would try to recognise what is type of word as in is a name, a pronoun, verb, etc. The reason to do this was to see if we can get any personally identifiable information directly from the scams but turns out people usually are very protective of their privacy and because the criminal is basically, it hinders this approach drastically. The code for this task is attached in the appendix A.5.

A sample output containing all of these extract features is available in listing 5.4.

5.3.5 Processing the IPs and the Domains

Now that we have all the data related to the address such as the neighbours, associated IPs, the domains that are in the report, we need to extract meaningful data from them before actually collecting the statistics and the results. IPs and URLs can be a really good source of the geolocation potentially bringing us closer to the criminal or at-least from the neighbours proving more insight into the address itself.

1. Extracting Location from the IPs : To extract the location from the IPs we used an external IP that takes in a list of IPs in a specified format and returns all the information about the IP in a JSON array in response. The code to perform this is attached in appendix A.7

   URL : `http://ip-api.com/batch`

   Sample Request Body :

   Listing 2: Sample Request to get IP Information

```
[
    "ipA",
    "ipB"
]
```

   Sample Response :

   Listing 3: Sample Response for IP Information

```
[
    {
        "status": "success",
        "country": "Switzerland",
        "countryCode": "CH",
        "region": "ZH",
        "regionName": "Zurich",
        "city": "Zurich",
        "zip": "8000",
        "lat": 47.3769,
        "lon": 8.54169,
        "timezone": "Europe/Zurich",
        "isp": "DataCamp Limited",
        "org": "CDNEXT Zurich",
        "as": "AS212238 Datacamp Limited",
        "query": "169.150.197.153"
    },
    {
        "status": "success",
        "country": "Netherlands",
        "countryCode": "NL",
```

```
        "region": "ZH",
        "regionName": "South Holland",
        "city": "Naaldwijk",
        "zip": "2671",
        "lat": 51.9934,
        "lon": 4.2158,
        "timezone": "Europe/Amsterdam",
        "isp": "CUSTOMERPANEL",
        "org": "",
        "as": "AS49981 WorldStream B.V.",
        "query": "93.190.142.127"
    },
    {
        "status": "fail",
        "message": "private range",
        "query": "10.217.130.145"
    }
]
```


As we can see from the sample response that the API does in fact provide quite a lot of information about the IP address passed in which is returned in the query string. We need to look at the difference between the first and the last response, so there are some IPs that are visible on the internet but fall under a private range so for them we received a failure status code and a clear message indicating that it was a private IP so for these we will just need to ignore the results. However, in the case of successes we do get the main information we need which the country and the associated region for now. At this stage we are only focusing on the country and will evaluate regions if time allows. So once we get all the data we are storing this locally on our machine to be accessed later and ensure that when we are generating our statistics we have no external API calls throttling our script.

2. Extracting Location from the URLs : In some of the addresses we also had mentions of URLs and we highly they would be related to the address could be either used by the scammer to trap people, thereby clear indication that the scammer owns it either directly or indirectly or it could be used to spread the scam further thereby having some indirect connection. To get the location information from the URL we used an external IP that was basically perform a reverse DNS on the URL and we would extract the A records from it which is where the URL is hosted. The reason for this is there is a high chance the URL is usually hosted near where the person is so giving a high chance of getting us closer to the criminal. Also to avoid false positives we evaluated all the URLs we received and filtered and removed the ones we know are legit to ensure only the ones that might be associated to the scammer are targeted and also

to reduce the number of external API calls as they only slow down our script. The code to perform this is attached in appendix **number** POST URL : https://api.geekflare.com/dnsrecord

Sample Request Body :

Listing 4: Sample Request to get Domain Information

```
{
    "url" : "www.mq.edu.au"
}
```

Sample Response :

Listing 5: Sample Response From GeekFlare

```
{
    "timestamp": 1667525360330,
    "apiStatus": "success",
    "apiCode": 200,
    "meta": {
        "url": "www.mq.edu.au",
        "test": {
            "id": "7nqxjmf0077frjb70d1yj7fw7ohndcg9"
        }
    },
    "data": {
        "A": [
            {
                "address": "203.82.26.7",
                "ttl": 299
            }
        ],
        "AAAA": [],
        "CAA": [],
        "CNAME": [
            "mqu.squizedge.net"
        ],
        "MX": [],
        "SRV": [],
        "TXT": []
    }
}
```

Once we had got all the data from the API we needed we stored it locally in a JSON file with each object containing the URL and its associated A records we can use to read later. Below is a part of the JSON file we stored locally to retrieve later.

input listing

5.4 Interpreting the Results

Once we have got all the data from the sources above we dump everything together into a single JSON file to try and visualise the relationships between the target addresses and its neighbours with the vital information to make some justified reasonable conclusions. **We need to keep in mind our goal in the end is to get close to the real world entity associated with the address, not identify whether its legal or illegal as plenty of studies have already proposed methods with a very high success rate to ascertain whether an address is legal or illegal.** Below is an extract of the report we generated with all the possible attributes for an address.

Listing 6: Extract from the report generated for all the target addresses


```
[
    [
        {
            "address": "
                bc1qmghwkrrxlh62k4r530lgfxucum65087ya00wvz
                "
        },
        {
            "numberOfAlerts": 11.0
        },
        {
            "numberOfNeighbours": 2
        },
        {
            "numberOfNeighboursWithScams": 0
        },
        {
            "numberOfScamsInNeighbours": 0
        },
        {
            "numberOfScamsInAddress": 11.0
        },
        {
            "parsedAddressData": {
                "urls": [
                    [
                        "https://.com",
                        "https://.com"
                    ],
                    [
                        "http://legadoo.com"
                    ],
                    [
```

```
                "https://cmohr-konzeption.de",
                "https://cmohr-konzeption.de"
            ],
            [
                "https://www.",
                "https://www."
            ]
        ],
        "emails": [],
        "ipAddresses": [
            "169.150.197.153"
        ],
        "spacy_data": [
            {
                "text": "First",
                "label": "ORDINAL"
            },
            {
                "text": "2500",
                "label": "MONEY"
            },
            {
                "text": "First",
                "label": "ORDINAL"
            },
            {
                "text": "2500",
                "label": "MONEY"
            },
            {
                "text": "7 days",
                "label": "DATE"
            },
            {
                "text": "google",
                "label": "ORG"
            },
            {
                "text": "First",
                "label": "ORDINAL"
            },
            {
                "text": "2500",
                "label": "MONEY"
            },
            {
```

```
                        "text": "First",
                        "label": "ORDINAL"
                    },
                    {
                        "text": "2500",
                        "label": "MONEY"
                    },
                    {
                        "text": "2500",
                        "label": "MONEY"
                    },
                    {
                        "text": "First",
                        "label": "ORDINAL"
                    },
                    {
                        "text": "2500",
                        "label": "MONEY"
                    },
                    {
                        "text": "First",
                        "label": "ORDINAL"
                    },
                    {
                        "text": "2500",
                        "label": "MONEY"
                    }
                ]
            }
        },
        {
            "parsedNeighbourData": []
        }
    ],
    [
        {
            "address": "
                bc1qgjxx4upjvqpa6xutmxa3kw6k0gs6hnkkeqlxup
                "
        },
        {
            "numberOfAlerts": 1.0
        },
        {
            "numberOfNeighbours": 4
        },
```

```
        {
            "numberOfNeighboursWithScams": 1
        },
        {
            "numberOfScamsInNeighbours": 19.0
        },
        {
            "numberOfScamsInAddress": 1.0
        },
        {
            "parsedAddressData": {
                "urls": [],
                "emails": [],
                "ipAddresses": [],
                "spacy_data": []
            }
        },
        {
            "parsedNeighbourData": [
                {
                    "urls": [],
                    "emails": [],
                    "ipAddresses": [
                        "93.190.142.127"
                    ],
                    "spacy_data": [
                        {
                            "text": "R10.000",
                            "label": "MONEY"
                        }
                    ]
                }
            ]
        }
    ]
]
```


Above listing shows all the results we extracted for a particular target address. The code used to achieve this is attached in appendix **number**.

Within each object there are multiple attributes which we need to familiar as to what purpose they solve so below is a short description of each one of them

1. address : This is the target address about which we are finding the information.

2. numberOfAlerts : This is a deprecated attribute replaced by numberOfS-

camsInAddress.

3. numberOfNeighbours : This refers to the number of addresses our target address has interacted it can be either it received money from it or sent money to it.

4. numberOfNeighboursWithScams : This is the number of neighbours which had reported scam alerts on BitcoinWhoIsWho.com

5. numberOfScamsInNeighbours : An address can have multiple scams reported by multiple people so this is the total number of reported scams on BitcoinWhoIsWho.com amongst all the neighbours.

6. numberOfScamsInAddress : This attributes tells the number of scam reports that were submitted against the target address in question.

7. parsedAddressData : This contains all the data that was extracted from the scams sections of the address HTML and includes emails, urls, ip addresses and keywords from spacy natural language processing.

8. parsedNeighbourData : This contains all the data that was parsed scams section of all the neighbours of the target address and includes the ipAddresses, urls, emails and keywords from spacy natural language processing library.

9. urls : This is the list of URLs read from the parsed HTML using regular expression matching.

10. emails : This is the list of emails detected in the parsed HTML again using regular expression matching.

11. ipAddresses : This is the list of ipAddresses detected in the parsed HTML data using regular expression matching.

12. spacy-data : This is a list of key value pair where the key is called text and is the actual keyword from the data that was fed into the natural language processing library and label is the type of keyword detected by spacy which can be anything like MONEY, PERSON, NUMBER, etc amongst a few that are shown in the example.

6 Results

Result Attribute	Value
Total Number of Target Addresses	1251
Total Number of Target Addresses With Data	128
Average Number of Reports Per Target Address	8.53125
Average Number of Neighbours Per Target Address	920.8515625
Average Number of Scam Reports in Neighbours	0.014753
Average Number of Neighbours With Reports Per Target Address	2.296875
Total Number of Reports in Neighbours	1739
Total Number of Addresses With Location Data	36
Average Number of Target Addresses With Location Data	0.28125

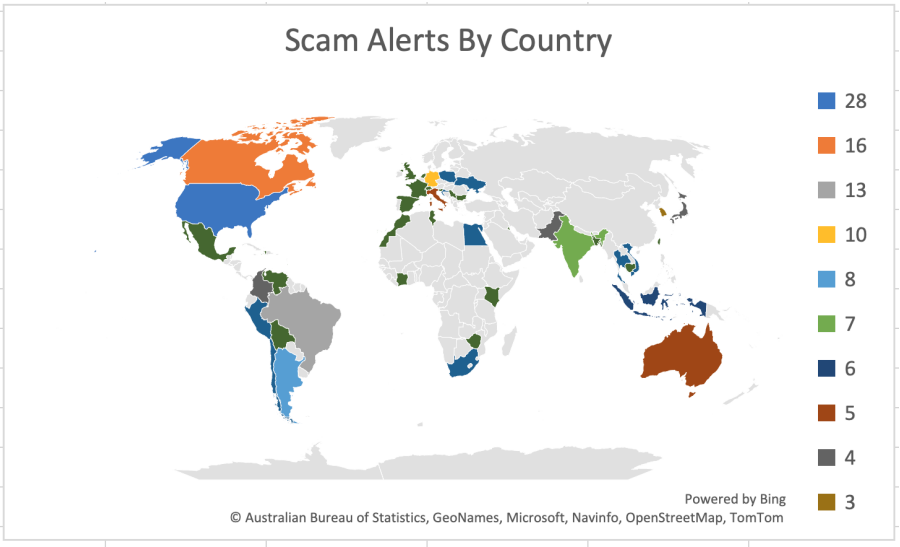

Figure 6: Scam Alerts By Country in the Target Address Reports

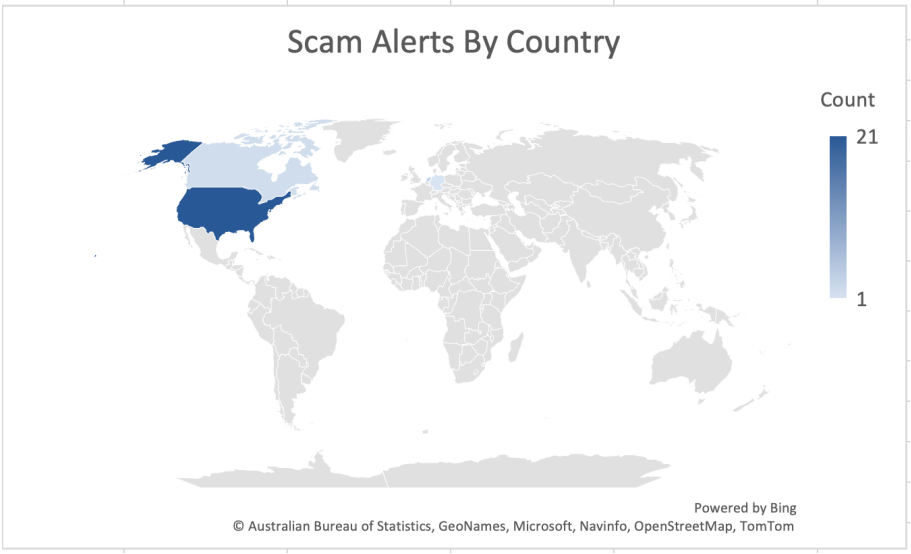

Figure 7: Scam Alerts By Country in the Neighbours Reports

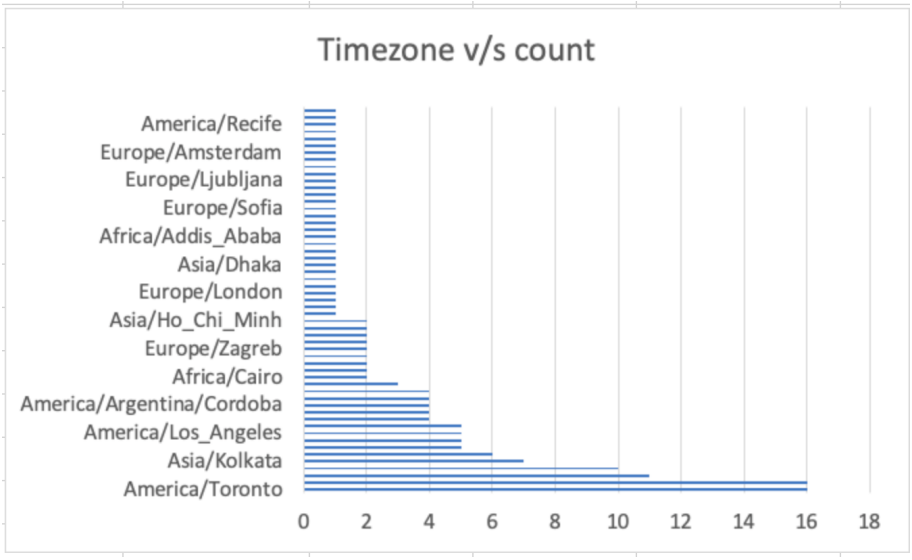

Figure 8: Count of various Time-zones in the Address Reports

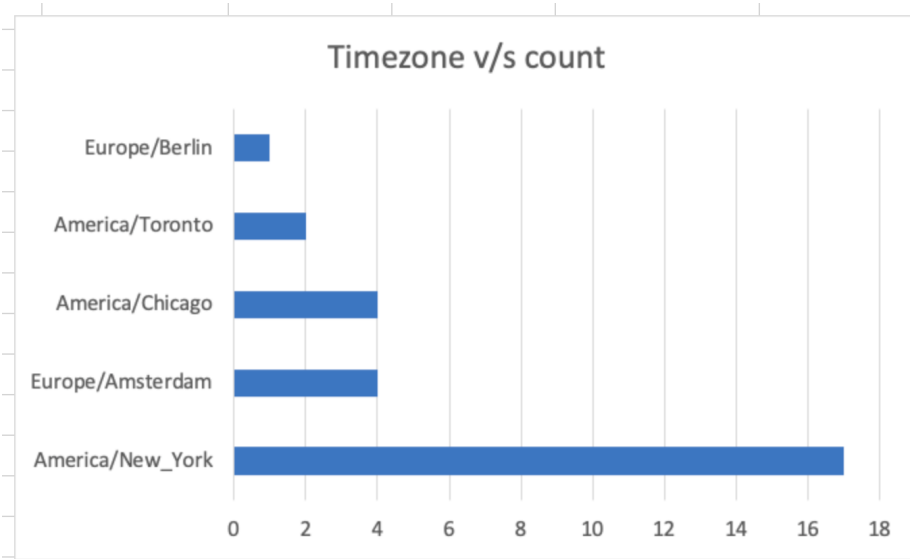

Figure 9: Count of various Time-zones in the Neighbour Reports

Figures above show the results we obtained after running our analysis. It does indeed show that there is quite a lot of data available the source we are scrapping. We are specifically interested in the counties and regions that are more popular as based on the screenshot in 6 there is a high chance that the region we detect in the report belongs to the address itself. As we can see from our statistics in figure 6, we had a total of 128 addresses out of 1251 where where we had some address reports submitted by the user which is roughly 10 percent. What we need to realise here is all of this data is data reported by people who have been affected by these addresses so it does show that people are becoming aware about these forums to report and as they get more popular, this only serves as fuel for our analysis. Another feature of these reported malicious addresses we can observe is that they tend to have a large number interactions which we believe would be to move money around to different addresses as much as possible to minimize traceability, which is a very common behaviour of malicious addresses as we have seen in the past work that has been done in this area.

Along with this if we look into what we received about the neighbours it seems like roughly only 1 percent neighbours had any reports indicating that the neighbour addresses were most likely single use addresses not primarily used to perform the crime rather store the money, this could be a reason why around 98 percent of neighbours did not come into the limelight of the people who got scammed. It does seem however that in some cases the neighbours might have been used to actually perform the scam which we think would be in the cases where we had the report which we can indicate from the 1739 reports from all

the neighbours. There is also a possibility that the scammer used a mixing service to move his money around from the target address to hid own. Mixing services as we discussed earlier were designed to reduce traceability so which is definitely what the scammer desires and this could also be the reason why so few neighbours have any reports as they could be single use addresses owned and managed by mixers.

Also looking at the number of addresses where we had location information available almost 28 percent of addresses had a mention of location which is being through IP addresses or URLs makes it even a better claim than just being said by someone. We will need to be however cautious that these IP addresses could be relayed IP address i.e. there might be instances that the IP address we have in the report might be a remote server the hacker is accessing rather than his actual IP. But this does give us an indication for some of the addresses that it could be the correct location at least the country or maybe in some cases the city of the scammer.

7 Limitations and Future Work

Every research is a problem unsolved and so we also faced a few issues in our work. In this section we will be going through the problems we faced while performing this research and what solutions we came up to overcome them and also what can one learn from our work if they decide to take forward our methodology.

7.1 Scarcity of Data

This is the essence of the problem we are trying to solve. Directly the blockchain provides us with all the information stored in the ledger which is basic financial information such as money held, transactions made but nothing about the personal entity the address is associated with. The only information which we can exploit is the sources available online which allow people to submit reports about information in relation to a particular address, other than that there is essentially nothing available.

To overcome this issue we explored multiple datasets such as BTCHeistData and BTCAbuse database. We had no luck with the former due to how old the data is but we did get some results with the later set. BTCAbuse's API[26] gave us multiple options for the length of the report we want which were either 1d, 30d or forever. We tried retrieving the forever report but it gave us a server error so we had to go for the next best option which is the 30 days report we are using for our work.

7.2 Issues with APIs

With the APIs that we were utilizing for retrieving the data related directly to the address we had a couple of issues with them as they are not really built for research purposes. So for that we came up with alternative measures such a putting a sleep after every call, using web scraping instead of APIs. Below we have listed some of the issues we came accross along with the solutions we used to overcome them.

7.2.1 Lack of reliable APIs

All the sources we used above to collect our data have APIs available but were not accessible to us and also had rate limits which would hinder our work. Most of the services had free APIs available and we used them wherever feasible considering the rate limit. For instance we used blockchain.info's free API to collect neighbours of an address as we didn't need to make thousands of calls over there considering the rate limit was maximum 10 calls per minute so we were able to get all the required data within a span of a couple of hours. For the rest of the data which we scraped online from BitcoinWhoIsWho, we could not use the APIs. The reasons for that being, first, we applied for an API key in the early days however failed to receive any response from the party responsible for

issuing an API key and also we needed to access that page more than tens of thousands of times so no API would have such a generous limit for us to be able to do that. So, to overcome this we built the web scraping application which just imitates a user on the website and the website would just consider it as a lot of people visiting it rather than through an API where they could consider a Denial of Service Attack and block us.

7.2.2 IP getting blocked from One of the APIs

We were constantly accessing Blockchain.info API to get neighbour addresses of the target. We were trying to hit the API nearly 100 times plus a second and tried over a number of days. After digging a bit deeper we found that the API has a rate limit of 10 calls maximum per minute so we adapted our program to do that but turns out doing that also too many times blocks our IP. So in the last few weeks we had one of our IPs blocked due to making too many requests and every time we tried accessing it through the IP, we got a HTTP-Status 429[32] which is an error status for making too many calls. So, for this we had to switch networks and adapted our approach to collect all the data in one go and then process it locally as required.

7.2.3 Speed of Data Collection

The Python script we developed initially was a completely synchronous program which meant one step ran after another one has finished. This was working fine for test data-sets where we were just checking if our program works or not with a couple of address but when we actually started performing our experiment with thousands of addresses our program throttled exponentially. There were multiple reasons for this and overall this resulted into such a slow program that it was estimated that our experiment would take over a month to just execute and then we had to analyze the results after that which was definitely not feasible considering the time frame we had. So, to overcome this we refactored the entire script to be asynchronous and concurrently perform multiple tasks. For this we used the Python's AsyncIo[27] library which basically collects all the methods/tasks to be run together and runs them in parallel using co-routines. This way were we were able to analyze hundreds of addresses at once and also their neighbours which meant that at one point we were processing thousands of addresses and that too didn't take more than a few minutes for every execution.

7.3 Future Work

As we can see from above that there we certain challenges that were faced while getting this results, but it does show that there is potential in this work and if someone does decided to continue work on our methodology, it would be in the best interest to overcome and avoid these issues at first.

We can observe that our dataset was quite large to begin with around 16k reports but when we rounded it down to the number of addresses in question,

we were only left with 1251 addresses so a good starting point for continuing work would be to extract a larger dataset. As we can see from the results there is a good chance that with an even larger data set we will be able to get closer to the scammer provided we run a detailed enough analysis of the reports we obtain. Another way to improve results would be add more sources to scrape to the investigation, we had to bound ourselves to one due to the time constraints. One such source that can be added is hashXp.org[33] which is a similar platform to Bitcoin Who's who where users can come and anonymously report about various cryptocurrency addresses.

Apart from this another extension to this work would taking into account mixers. We have explored the neighbours of the target address and their related scam information. Mixers as we discussed earlier were designed to reduce traceability so there is an extremely high chance that a scammer wont just cash out the currency from the address he used to perform the scam, rather he would pass the money though multiple hops to reduce his traceability from the scam address. Now this can be done through a mixer as well where the mixer does the hopping and outputs the money into the addresses specified. So, based on this there is a high chance that the neighbours we explored of the target addresses, some of those neighbours actually belong to some mixing service thereby are single use addresses most of times. So, one can further take this up possibly take inspiration on how to track mixers from previous research and apply our methodology.

8 Conclusion

We can thereby conclude from here that it is in fact possible to get closer to the criminals even though we might not be able to get the pin point location. This piece of work does show some improvement as to what is currently done in this area which is limited to identification of malicious vs non-malicious but an extension of this could some day lead to better regulation and less crimes in cryptocurrencies. One thing is for sure that the world of decentralized finance is going to keep changing as technology evolves and people whose focus is to preserve privacy will keep on working to find ways to mask their identity so this sort of work has to be a continuing research in order to stay on top of the new and upcoming technologies and tools.

Appendices

A Code

The entire code is available on my GitHub[34] as a public repository.

A.1 Code to Download HTMLs

Listing 7: Code to Download HTMLs from bitcoinwhoiswho.om

```
import os
from progressbar import Percentage, ProgressBar,Bar,ETA
import pandas as pd
import aiohttp
import asyncio
import json


pbar = ProgressBar(widgets=[Bar('>', '[', ']'), ' ',
    Percentage(), ' ',ETA()])

async def extractHTML(session, address) :
    btcWhoIsWhoUrl = "https://www.bitcoinwhoswho.com/
        address/" + (address)
    async with session.get(btcWhoIsWhoUrl) as response:
        data = await response.text()
        print(data)
        return data

async def findIfBtcWhoIsWhoHasReport(address,session) :
    if not (os.path.exists("./btcabuseNeighbours/" + str(
        address) + ".html")) :
        print("Extracting HTML")
        print(address)
        html = await extractHTML(session, address)
        if html is not None:
            print(html)
            writeToFile(html, address)

def writeToFile(addressData, x) :
    print("Writing to file for " + x)
    with open("./btcabuseNeighbours/" + str(x) + ".html",
        "w") as outfile :
        outfile.write(str(addressData))

```

```
async def validateResults(data) :
    async with aiohttp.ClientSession() as session:
        addresses = []
        addresswithNeighbours = []
        tasks = []
        for x in pbar(data) :
            result = asyncio.ensure_future(
                findIfBtcWhoIsWhoHasReport(x, session))
            tasks.append(result)
        await asyncio.gather(*tasks)
    return {"addresses" : addresses, "
       addresswithNeighbours" : addresswithNeighbours}


maliciousAddressesToInvestigate = []

df = pd.read_csv("latestReport.csv")

list = df.address.to_list()

for x in list :
    maliciousAddressesToInvestigate.append(str(x))

FILE_NAME = "neighborData.json"

with open(FILE_NAME, "r") as f :
    data = f.read()
    data = json.loads(data)

addressesToInvestigate = []
for x in data :
    if len(x["neighbours"]) > 0 :
        for y in x["neighbours"] :
            addressesToInvestigate.append(y)


loop = asyncio.get_event_loop()
loop.run_until_complete(validateResults(set(
   addressesToInvestigate)))
```

A.2 Code to Extract Neighbours

Listing 8: Code to Extract Neighbours

```
import requests

def extractNeighbours(address) :
```

```
    addressUrl = "https://blockchain.info/rawaddr/" +
        address
    data = requests.request("GET", addressUrl)
    result = []
    if (data.status_code == 200) :
        data = data.json()
        if (data.__contains__("txs")):
            for x in data["txs"]:
                for y in x["out"]:
                    if(y.__contains__("addr")):
                        if(y["addr"] is not None):
                            result.append(y["addr"])
                for y in x["inputs"]:
                    if(y.__contains__("addr")):
                        if(y["addr"] is not None):
                            result.append(y["addr"])
    return result
```

A.3 Code to Extract IP Addresses

Listing 9: Code to Extract IP Addresses

```
import re

def parseHTML(html) :
    pattern = re.compile(r'(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d
        {1,3})')
    ipAddresses = []
    for x in html :
        x = str(x)
        if pattern.search(x) is not None:
            ipAddresses.append(pattern.findall(x)[0])
    return ipAddresses
```

A.4 Code to Extract Emails

Listing 10: Code to Extract Email Addresses

```
import re

def parseHTML(html) :
    emails = []
    for x in html :
        x = str(x)
        email = re.findall(r"[a-z0-9\.\-+_]+@[a-z0-9\.\-+
            _]+\.[a-z]+", x)
        if len(email) > 0 :
            emails.append(email)
```

```
    return emails
```

A.5 Code to Perform Natural Language Processing

Listing 11: Code to Perform Natural Language Processing

```
import spacy

nlp = spacy.load('en_core_web_trf')

def parseHTML(html) :
    spacy_data = []

    for x in html :
        x = str(x)
        spacy_parser = nlp(x)
        for entity in spacy_parser.ents:
            spacy_data.append({'text': entity.text, '
                label': entity.label_})

    return spacy_data
```

A.6 Code to Extract Domains

Listing 12: Code to Extract Domains

```
import re

def parseHTML(html) :
    urls = []
    for x in html :
        x = str(x)
        url = re.findall('https?://(?:[-\w.]|(?:%[\da-fA-
            F]{2}))+', x)
        if len(url) > 0 :
            urls.append(url)

    return urls
```

A.7 Code to Extract Location From IP Addreses

Listing 13: Code to Extract Location From IP Addresses

```
# App to store all IP Addresses and their data in KV
    pairs in a JSON file.
import json
from requests.adapters import HTTPAdapter
from urllib3.util.retry import Retry
```

```
import requests

with open("allAddressesData.json") as f:
    data = json.load(f)

# Get IPs from Domains
with open("allUrlsData.json") as f:
    d = json.load(f)
    ips = []
    for x in d :
        for y in x['aRec']:
            ips.append(y['address'])

def extractItems(data) :
    result = []
    for x in data :
        result.append(x)
    return result

allIps = []

for x in data :
    ipAddressesFromNeighbours = []
    urlsFromNeighbours = []
    ipAddressesFromAddress = extractItems(x[6]['
        parsedAddressData']['ipAddresses'])
    if (len(x[7]['parsedNeighbourData']) > 0) :
        ipAddressesFromNeighbours = extractItems(x[7]['
            parsedNeighbourData'][0]['ipAddresses'])
    urlsFromAddress = extractItems(x[6]['
        parsedAddressData']['urls'])
    if (len(x[7]['parsedNeighbourData']) > 0) :
        urlsFromNeighbours = extractItems(x[7]['
            parsedNeighbourData'][0]['urls'])
    allIps = allIps + ipAddressesFromAddress +
        ipAddressesFromNeighbours


allIpData = []

if len(allIps) > 0:
    data = "["
    for x in allIps:
        data = data + '"' + x + '"' + ", "

    data = data + '"' + "1.1.1.1" + '"' + "]"
```

```
46
47      session = requests.Session()
48      retry = Retry(connect=3, backoff_factor=0.5)
49      adapter = HTTPAdapter(max_retries=retry)
50      session.mount('http://', adapter)
51      session.mount('https://', adapter)
52
53      allIpData = session.post('http://ip-api.com/batch',
            headers= {'Content-Type': 'application/x-www-form-
            urlencoded'}
                        , data=data).json()
54
55
56
57  with open('allIpData.json', 'w') as f:
58      json.dump(allIpData, f)
```

A.8 Code to Extract Scams from the HTML

codeToScrapeHTML.py > ...

```
1   from bs4 import BeautifulSoup
2
3   x = "the HTML File"
4   parsedHtml = BeautifulSoup(x, "html.parser")
5   res = parsedHtml.body.find_all("div", {"id": "wrapper"})
6   finalSection = []
7   numScamAlerts = 0
8   for result in res :
9       for d in result.find_all("section",
10                               {"id": "content"}) :
11          for x in d.find_all("div",
12                              {"id" : "search_address_index", "class" : "container"}) :
13              for y in x.find_all("div",
14                                  {"class" : "row"}) :
15                  for z in y.find_all("div",
16                                      {"class" : "col-lg-12"}) :
17                      for w in z.find_all("div",
18                                          {"class" : "row"}) :
19                          for z in w.find_all("div",
20                                              {"class" : "col-lg-12 float_left_box"}) :
21                              for x in z.find_all("div",
22                                                  {"class" : "row text-center"}) :
23                                  for z in x.find_all("div",
24                                                      {"class" : "col-lg-12"}) :
25                                      for y in z.find_all("div",
26                                                          {"class" :
27                                                           "float_left_box flb_scam_records_table"}) :
28                                          for x in y.find_all("div",
29                                                              {"class" : "collapse",
30                                                               "id" : "scam_records_table"}) :
31                                              numScamAlerts = len(x.find_all("div",
32                                                                             {"class" :
33                                                                              lambda x: x
34                                                                              and
35                                                                              x.startswith('row row_')}))/2
36                                              for z in x.find_all("div",
37                                                                  {"class" :
38                                                                   lambda x: x
39                                                                   and
40                                                                   x.startswith('row row_')}) :
41                                                  finalSection.append(z.find_all("div", {"class" : "col-md-11"}))
42  {"HTML" : finalSection,"numberOfAlerts" : numScamAlerts}
```

Figure 10: Code to Extract Scams from the raw HTML

B Data

This section of appendix contains some snippets of the dataset used and extracted.

B.1 Bitcoin Abuse DB Data

Below is a snippet of the top few rows of the data we obtained and used from Bitcoin Abuse DB.

id	address	abuse_type_id	abuse_type_other	abuser	description	from_country	from_country_code	created_at
633114	bc1qm34lsc65zpw79lxes69zkqmk6ee3ewf0j77s3h	99	FRAUD - Cryptocurrency Trading Scheme	https://tradingtech.io (Mr. Konstantinos Makris)	A user has just reported a scam involving a supposed investment schema out of the UK and Russia called "Trading Tech" that was recommended by friends. He sent over .47 BTC and never received it back - this aligns with similar reports from others over the past few months. Clients have reported to Interpol, FCA, FBI and NFC. PLEASE, BE CAREFUL! - Jon from Digital Block (jon.g@digitalblock.info) a tracking and recovery	United States	US	2022-09-09T22:10:30.000000Z
633124	3JTVr8uhUBW7yiRgDMxh5g5UfUXDVaUyqK	99	Scam	Impersonation	Impersonation of Georgian Legion attempting to solicit donations	United States	US	2022-09-09T22:22:50.000000Z
633134	1CRmsktJESGQionPDbvu7L5XnqAqkfj8Ed	99	Scammer	Kyle Duncan	Scams people out of money	United States	US	2022-09-09T22:32:13.000000Z
633144	bc1qdxf9gedsncq9mavpzm3vnuq5k8ed3e8zmyy3mz	3		https://shapeofmars.com/?mars	Elon Musk crypto giveaway scam	United States	US	2022-09-09T22:38:18.000000Z
633154	1KK7guZhwuw5vv3EXd5DnpUZUqT8NgiN52	4		desarrollo@rfv-truck-parts-accessories.com.mx	Claims to have infected your computer with a trojan and observed you watching	United States	US	2022-09-09T22:51:09.000000Z
633164	18YFLJHGufQQukMFRkJJJiz51mp21qN96p	5	phishing	No name stated	content beginning: I am sorry to inform you but your device was hacked. That's what happened. I have used a Zero Click vulnerability with a special code to hack your device through a website. A complicated software that requires precise skills that I posess. This exploit works in a chain with a specially crafted unique code and such type of an attack goes undetected. You only had to visit a website to be infected, and unfortunately for you it's that simple for me. [,Ä¶] From email header: Received: from [179.49.245.210] (port=43107 helo=cgnat-210-245-49-179.clickprovedorfibra.com.br) by SUBDOMAIN_REDACTED.bluehost.com with esmtp (Exim 4.95) (envelope-from <MY_EMAIL_REDACTED>) id 1oWmXA-002gg4-HG for MY_EMAIL_REDACTED; Fri, 09 Sep 2022 16:31:52 -0600 that's a brazillian domain if it's not spoofed too	United States	US	2022-09-09T23:28:08.000000Z
633174	LVH942cyEoS49ay5nvjFGyHwmNyqRJuahv	5		Rachel Lee	Sextorion claiming to have a video of me. Asking for money or they would emai	United States	US	2022-09-09T23:37:48.000000Z
					Hello, The site http://iifz7vplkeovyrruirnridqb6ewbvirjxkb2w333lrjeytxlt3gqbsyd.onion/#NK is about to show young children (in erotic/"sexual driven" poses) to adults for erotic/sexual entertainment and (may) contain illegal content like CHILD-PORN / CHILD-EROTIC or any like this. They ask for "Donation" to make the "Service" stay online:			

Figure 11: An Extract from Bitcoin Abuse CSV Report

B.2 Address and its Neighbours Data

Listing 14: Sample report of Address with its neighbours


```
[
    {
        "address": "18Jro9LNFqBQarcc63WYGf3w7PdDAiwXpk",
        "neighbours": [
            "1HLDhceB1HF8Uzf8AyZ4qLs3J4tGxD1sm8",
            "35szf5NPv3CWaXJrgEcAn34ttev1GbWUFN",
            "355L7iQJozifhiP1qskT75YRP5AVAfBubr",
            "15FWyyWHXdtgbxPd8a1s9UVG3VGCWGpsgV",
            "3CK2BrAsYckRkiEdnCbhEi1KGLJ99SAEbw",
            "19jednwZQ1UNBZ6fcrLxtp5gaQL6L3pMuS",
            "3527cxJcs3jjV7nG7mDGBTcULMWmcUoYoF",
            "31nKJmksRAyiQ2Qj2SPXDyDm5KXhu5EEAy",
            "1HqQUZMz3S2guCmshKuGEy7iP2TSRQb13Z",
            "1Ei73Wtp9G8A1CCfSbRjqRePA1PArUhCUW",
```

```
                "bc1q60rg6hfr79v7fee0m70kxug5utalhq7uwn5pel",
                "18Jro9LNFqBQarcc63WYGf3w7PdDAiwXpk",
                "3NbeS1bAV9Kh3arFNBtVfmKgE5zZSaHHXX",
                "bc1qu73n7ezl8q9qefwd8xy8lfjte4quz4ythksqht",
                "18Jro9LNFqBQarcc63WYGf3w7PdDAiwXpk",
                "1AzbY7CUP3xQoCyoXTofFSAdWynRF418uT",
                "18boKrPj1gyyHNUXuqNr5ad7wTeJ7L5qb8",
                "14ZE1UpGeabpAebTpewkxfWcGFB9HTwpu7",
                "342q2FGRex3j3iE3UhxtN1TaxYAa4oiVHA",
                "34caE1x7tWsV81vEojjqra9MbTgNW5SsHz",
                "3QibiFH3Hx3xcDaQi5JRbGCofTJm9pUsrc"
            ]
        },
        {
            "address": "3PFgmTVFbRD6ZCB7G6QEb5G9SAuhyd2Z66",
            "neighbours": []
        }
]
```

B.3 Sample Report Used for generating Statistics

Listing 15: Extract from the report generated for all the target addresses

```
[
    [
        {
            "address": "
                bc1qmghwkrrxlh62k4r530lgfxucum65087ya00wvz
                "
        },
        {
            "numberOfAlerts": 11.0
        },
        {
            "numberOfNeighbours": 2
        },
        {
            "numberOfNeighboursWithScams": 0
        },
        {
            "numberOfScamsInNeighbours": 0
        },
        {
            "numberOfScamsInAddress": 11.0
        },
        {
            "parsedAddressData": {
```

```
                "urls": [
                    [
                        "https://.com",
                        "https://.com"
                    ],
                    [
                        "http://legadoo.com"
                    ],
                    [
                        "https://cmohr-konzeption.de",
                        "https://cmohr-konzeption.de"
                    ],
                    [
                        "https://www.",
                        "https://www."
                    ]
                ],
                "emails": [],
                "ipAddresses": [
                    "169.150.197.153"
                ],
                "spacy_data": [
                    {
                        "text": "First",
                        "label": "ORDINAL"
                    },
                    {
                        "text": "2500",
                        "label": "MONEY"
                    },
                    {
                        "text": "First",
                        "label": "ORDINAL"
                    },
                    {
                        "text": "2500",
                        "label": "MONEY"
                    },
                    {
                        "text": "7 days",
                        "label": "DATE"
                    },
                    {
                        "text": "google",
                        "label": "ORG"
                    },
```

```
                        {
                            "text": "First",
                            "label": "ORDINAL"
                        },
                        {
                            "text": "2500",
                            "label": "MONEY"
                        },
                        {
                            "text": "First",
                            "label": "ORDINAL"
                        },
                        {
                            "text": "2500",
                            "label": "MONEY"
                        },
                        {
                            "text": "2500",
                            "label": "MONEY"
                        },
                        {
                            "text": "First",
                            "label": "ORDINAL"
                        },
                        {
                            "text": "2500",
                            "label": "MONEY"
                        },
                        {
                            "text": "First",
                            "label": "ORDINAL"
                        },
                        {
                            "text": "2500",
                            "label": "MONEY"
                        }
                    ]
                }
            },
            {
                "parsedNeighbourData": []
            }
        ],
        [
            {
                "address": "
```

```
                bc1qgjxx4upjvqpa6xutmxa3kw6k0gs6hnkkeqlxup
                "
        },
        {
            "numberOfAlerts": 1.0
        },
        {
            "numberOfNeighbours": 4
        },
        {
            "numberOfNeighboursWithScams": 1
        },
        {
            "numberOfScamsInNeighbours": 19.0
        },
        {
            "numberOfScamsInAddress": 1.0
        },
        {
            "parsedAddressData": {
                "urls": [],
                "emails": [],
                "ipAddresses": [],
                "spacy_data": []
            }
        },
        {
            "parsedNeighbourData": [
                {
                    "urls": [],
                    "emails": [],
                    "ipAddresses": [
                        "93.190.142.127"
                    ],
                    "spacy_data": [
                        {
                            "text": "R10.000",
                            "label": "MONEY"
                        }
                    ]
                }
            ]
        }
    ]
]
```

References

[1] R. Browne, "Central african republic becomes second country to adopt bitcoin as legal tender," Apr 2022.

[2] D. A. Zetzsche, D. W. Arner, and R. P. Buckley, "Decentralized finance," Sep 2020.

[3] S. Kethineni and Y. Cao, "The rise in popularity of cryptocurrency and associated criminal activity," *International Criminal Justice Review*, vol. 30, no. 3, p. 325–344, 2019.

[4] R. Spagni, "The monero project."

[5] E. Duffield, "Dash is digital cash you can spend anywhere," Jul 2022.

[6] D. G. Paul Syverson, Michael Reed, "The tor project: Privacy freedom online."

[7] J. Frankenfield, "Silk road."

[8] N. Hiramoto and Y. Tsuchiya, "Measuring dark web marketplaces via bitcoin transactions: From birth to independence," *Forensic Science International: Digital Investigation*, vol. 35, p. 301086, 2020.

[9] C. Zhao, "Binance — cryptocurrency exchange for bitcoin, ethereum & altcoins."

[10] F. E. Brian Armstrong, "Coinbase — buy & sell bitcoin, ethereum, and more with trust."

[11] R. Wright, "Europol."

[12] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press, 2016.

[13] X. Xueshuo, W. Jiming, Y. Junyi, F. Yaozheng, L. Ye, L. Tao, and W. Guiling, "Awap: Adaptive weighted attribute propagation enhanced community detection model for bitcoin de-anonymization," *Applied Soft Computing*, vol. 109, p. 107507, 2021.

[14] BlockchainExplorer, "Blockchain.com explorer: Btc: Eth: Bch."

[15] A. Biryukov and S. Tikhomirov, "Deanonymization and linkability of cryptocurrency transactions based on network analysis," in *2019 IEEE European symposium on security and privacy (EuroS&P)*, pp. 172–184, IEEE, 2019.

[16] S. Nakamoto, "btc-core."

[17] J. Schäfer, C. Müller, and F. Armknecht, "If you like me, please don't "like" me: Inferring vendor bitcoin addresses from positive reviews," *Proceedings on Privacy Enhancing Technologies*, vol. 2022, no. 1, pp. 440–459, 2022.

[18] Y. Hu, S. Seneviratne, K. Thilakarathna, K. Fukuda, and A. Seneviratne, "Characterizing and detecting money laundering activities on the bitcoin network," *arXiv preprint arXiv:1912.12060*, 2019.

[19] D. o. Justice, "Wsj news exclusive — justice department forms national network of prosecutors focused on crypto crime," Sep 2022.

[20] V. M. News, "Wsj news exclusive — u.s. recovers over 30 million us dollars in cryptocurrency stolen by north korean hackers," Sep 2022.

[21] G. K. Haaroon Yousaf and S. Meiklejohn, "Tracing transactions across cryptocurrency ledgers."

[22] Z. Wang, S. Chaliasos, K. Qin, L. Zhou, L. Gao, P. Berrang, B. Livshits, and A. Gervais, "On how zero-knowledge proof blockchain mixers improve, and worsen user privacy," *arXiv preprint arXiv:2201.09035*, 2022.

[23] Elliptic, "Blockchain analytics & crypto compliance solutions."

[24] M. Weber, G. Domeniconi, J. Chen, D. K. I. Weidele, C. Bellei, T. Robinson, and C. E. Leiserson, "Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics," *arXiv preprint arXiv:1908.02591*, 2019.

[25] C. G. Akcora, Y. Li, Y. R. Gel, and M. Kantarcioglu, "Bitcoinheist: Topological data analysis for ransomware detection on the bitcoin blockchain," *arXiv preprint [Web Link]*, 2019.

[26] B. A. D. Team, "Bitcoin abuse — api documentation."

[27] Python, "Asyncio - asynchronous i/o."

[28] Python, "Python requests."

[29] B. Soup, "Beautiful soup documentation."

[30] Python, "Re - regular expression operations."

[31] Spacy, "Spacy · industrial-strength natural language processing in python."

[32] M. D. Network, "429 too many requests - http: Mdn."

[33] H. XP, "Hash xp org."

[34] B. Dhanda, "Github - bhavish dhanda - tracking malicious transactions in cryptocurrencies."