Detecting potential money laundering addresses in the Bitcoin blockchain using unsupervised machine learning

Hilmar Páll Stefánsson
Dept. of Computer Science
Reykjavík University
hilmarpall98@hotmail.com

Huginn Sær Grímsson
Dept. of Computer Science
Reykjavík University
huginngri@gmail.com

Jón Kristinn Þórðarson
Dept. of Computer Science
Reykjavík University
jonkrona@gmail.com

María Óskarsdóttir
Dept. of Computer Science
Reykjavík University
mariaoskars@ru.is

Abstract

Money laundering is a serious problem worldwide, especially in the crypto market. This is mostly because of the anonymity that many cryptocurrencies offer. That is one of the reasons why cryptocurrencies are a haven for money laundering, because it is easier for criminal entities to buy the currency and then trade it for real fiat money. Detecting money laundering in cryptocurrency can be tricky because the crypto network is large and convoluted and nearly impossible to analyze by hand. What we can do is look at addresses that took part in transactions as actors and then use machine learning to predict what addresses are possibly laundering money.
In this paper we intend to analyze methods that can be used to detect money laundering in Bitcoin using machine learning to empower investigators to more accurately and efficiently determine whether a suspicious activity is money laundering.

1. Introduction

Cryptocurrency has been increasing rapidly in popularity since 2008 when Bitcoin, the first peer-to-peer financial system by Satoshi Nakamoto was first described [1]. As of the 26$^{\text{th}}$ of April 2021, Bitcoin's market capitalization is around 700 billion dollars and the market value for all cryptocurrencies is around 1.6 trillion dollars [2]. There is a significant amount of money that goes through this system and because of the general anonymity in cryptocurrencies it is a haven for money laundering and other criminal activities. A great example of this is the website Silk road which was an online marketplace known for facilitating sales of illegal products where the only allowed currency was Bitcoin [3]. The congress of the United States passed the National Defense Authorisation Act in January 2021 which is intended to improve corporate transparency and enhance coordination among law enforcement and the federal and state agencies responsible for taking anti money laundering(AML) actions [4]. Because of this, increased development of AML techniques is foreseen in the coming years with substantial development in the crypto market because of the high presence of money laundering activity and the constant growth of the crypto market.

In this paper, we investigate and describe in detail how to start building an AML detection and investigation tool for cryptocurrencies and blockchain assets. We focus on the Bitcoin blockchain because of its popularity and the easy access to the entire Bitcoin blockchain through Bitcoin Core. In addition, much research on Bitcoin already exists as well as publicly available datasets with labeled addresses which we utilize. Similar methods could be applied to different cryptocurrencies. Our goal is to detect anomalous addresses in the Bitcoin data, since anomalies can be an indication of suspicious activity, such as money laundering. Anomaly detection has been used before to detect fraud and other threats in financial networks. Paula et al. showed that anomaly detection using deep learning models can predict fraud, including money laundering, in Brazilian exports [5]. Such methods are also common in the anti money laundering industry, for example Lucinity[1] uses anomaly detection, among other approaches, to detect potential money laundering in fiat environments with great success [6].

The methodology of this paper consists of two main steps. Firstly, there is the data ingestion. This entails extracting raw blockchain data and processing the data into features for Bitcoin addresses. We focus on manually defined and engineered features to facilitate interpretability of our results. Included in these features are results from supervised machine learning models that predict whether or not an address is owned by a cryptocurrency exchange, gambling entity or service provider which is described more in section 3.2. Secondly, there is a machine learning (ML) step where we use unsupervised learning to detect anomalies

[1]Lucinity is an AML software company that uses advanced AI systems to discover money laundering.

URI: https://hdl.handle.net/10125/79526
978-0-9981331-5-7

in the data, which is a commonly used approach to detect fraudulent behaviours [7]. We use local feature importance to explain why an address is an anomaly and then we dig deeper to verify whether or not particular anomalous addresses are possible instances of money laundering.
The rest of this paper is organized as follows. In the next section, we discuss related work on analysis of the blockchain and money laundering methods. Then we describe our methodology in terms of data ingestion, feature engineering and machine learning, followed by the results of the machine learning models. We conclude the paper with a discussion on the implication of our work and suggest directions for future work.

2. Related projects and topics

In this section we discuss related ML approaches to AML as well as research on Bitcoin.

2.1. Characterization in the crypto network

Many researchers have investigated which features are important in the cryptocurrency network. Ron et al. found that a large amount of addresses, transactions and Bitcoins were controlled by a few entities in the Bitcoin network [8]. An example of this kind of entity is a cryptocurrency exchange. Hu et al. discovered that money laundering transactions tend to have a higher in-degree/out-degree ratio, and a more uniform sum, mean and standard deviation of outputs and a slightly smaller number of weakly connected components compared to regular transactions [9]. Based on this work, we included some of these suggested features in our unsupervised models.

2.2. FATF virtual assets red flag indicators

Late in 2020, the FATF (Financial Action Task Force) published a document with an overview of possible red flag behaviours in virtual assets that can indicate illicit behaviours. They talk about a few general classes of behaviours that will be of guidance when detecting outliers. Transaction size and frequency are one of the mentioned behaviours. They also point out that if an actor sends multiple transactions with values under the FATF limit (which is a 1000$) or is sending high value transactions it is an example of a behavior that should be flagged. Unusual transaction patterns are also considered to be suspicious. This includes, but is not limited to, frequent transactions to the same VASP (Virtual Asset Service Provider) and making a large initial deposit with a VASP. These are the behaviours that we will be taking a look at when detecting outliers [10].

2.3. Anomaly detection in blockchains and other environments

A stream of research is focused on anomalies and suspicious behaviour in the Bitcoin blockchain using data science and machine learning. In an attempt to find anomalous transactions, Pham and Lee extracted features from the transaction network, from the origin until 2014, and applied k-means clustering to find outliers [11]. Similar approaches have been proposed by other researchers [12, 13]. Some studies investigate certain types of suspicious behaviours. Firstly, to identify Ponzi schemes, transactions and wallets related to known schemes were extracted and compared to regular transactions and wallets in a supervised learning setting [14]. Secondly, researchers have looked into money laundering specifically, using network methods, in particular network representation learning and supervised machine learning models [9]. Recently, Elliptic[2] introduced a public dataset that contains several sub-networks for the blockchain transaction network, with rich node features and labels for licit and illicit transactions. Researchers have trained several supervised learning methods to detect illicit transactions and compared their performance [15]. Others have also worked with the Elliptic dataset [16, 17, 18], for example using active learning to address the high class imbalance in the dataset [18]. Although useful, the main drawback of the Elliptic dataset is that it is very poor in terms of feature labeling. In a similar way, Goldberg et al. used anomaly detection to identify insider threats in user data. They also did temporal aggregation experimentation to find the most unusual user in a time period [19].

3. Methodology

In this section we describe the methodology and processes that we used to detect anomalies in the blockchain data. First, we explain the data sources that were used for the raw data and the labeling of the addresses in the Bitcoin network. In section 3.3 we detail the process of finding anomalies or possible money laundering and describe the unsupervised models we deployed.

[2]Elliptic is a cryptocurrency intelligence company focused on safeguarding cryptocurrency ecosystems from criminal activity.

3.1. Data sources

Our goal was to create a dataset with easily interpretable features so we could later use those features to explain our anomaly detection models. To do that we first had to get our hands on the raw Bitcoin data. The raw dataset we found is included in the Bitcoin Core open source software. Bitcoin Core runs a local server that hosts the entire Bitcoin blockchain and allows API requests with bitcoin-cli commands. We created a python program that sent bitoin-cli API requests with the appropriate commands and received JSON strings back from Bitcoin Core with all the information stored on a single block. The next step was to get labeled data scraped from Walletexplorer[3]. This data is composed of around 30 million Bitcoin addresses, labeled as belonging to a cryptocurrency exchange, gambling entity, service, miner or historic [20, 21]. We did not use the labels mining and historic since the historic label was badly defined and the mining label was too sparse. Exchanges offer a way to buy and sell Bitcoin or other cryptocurrencies [22]. The gambling label marks known addresses that are owned by a gambling entity. Lastly there was the service label, the definition of a service is broad and unclear since it includes exchanges, mixers and anything that offers some kind of service on the Bitcoin network. Some exchanges are also labeled as service in this dataset [23]. Because most of the labeled addresses were focused on a period in January 2017, we decided to study a part of this period specifically, and thus limited the blockchain data to the same period.

3.2. Feature engineering

Feature engineering was an important step in this process, the more informative and rich features that were created, the easier it would be to interpret the models' results. The first step in this process was to get the raw data for all blocks in the desired time period. Then important information was extracted and stored in tables in a database. The data stored in the previously mentioned database was extracted and from it two feature tables were created, called tables A and B. Feature table A is made up of rows, each row represents all information for a single address for a single date, if an address does not make any transaction on that day that address - date pair does not appear in the table. Each feature in the table is a list of values. Feature table B is composed of rows, where each row represents aggregated features from table A for an address in a predetermined date range. This includes, but is not limited to, mean, standard deviation, median, min and max of input and output values along with frequency of transactions in the given time period, time difference between transactions and etc. Lastly one more feature table was added, table C. Feature table C only consists of features representing predictions of whether or not an address was predicted to be a Bitcoin exchange, gambling entity or service by supervised models. The supervised model used was the XGBoost regressor which we trained on feature table B. The supervised model scored each address from a range of zero to one, where one corresponds to an exchange, gambling entity or service. However, the training of the supervised models is out of scope for this research and will not be described in detail.

3.3. Detecting money laundering with unsupervised models

In this paper we use unsupervised learning to detect anomalies among Bitcoin addresses. This works since money launderers often behave abnormally which appears as anomalies. Here, we explain the methods used for the money laundering detection part of the project.
To tailor the anomaly detection towards finding money laundering addresses instead of other anomalous behaviours we created another feature table, feature table D. Feature table D consists of handpicked features from table B, table C and aggregated features from table B, that try to encapsulate behaviours outlined by the FATF, see Section 2.2. These features were used to train the unsupervised models, with the goal of reducing the amount of addresses human investigators would need to analyze and cut down the time needed to analyze each address by focusing on the anomalies and the features that define them.

3.3.1. Isolation Forest The first model used was Isolation Forest. This model returns an anomaly score based on how often features have to be randomly split until each feature of an address is isolated from the same feature of other addresses. This is repeated multiple times for each feature or feature vector and the final score is an aggregate score from all the splits [24]. Isolation Forest is useful since there are a couple of techniques that can be used to get the local feature importance for the observations in the model. While regular feature importance is used to explain what features are important in a model, local feature importance is used to explain what features are important for a single prediction. The techniques that are used for local and regular feature importance are

[3] https://www.walletexplorer.com

called MI-local-DIFFI and SHAP respectively [25, 26]. We used the default parameters for this model since it gave the best results.

3.3.2. Local-DIFFI and SHAP Carletti et al. proposed a local feature importance measurement called Local-DIFFI [25]. Local-DIFFI allows for interpreting individual predictions made by an isolation forest model. We used another implementation called MI-Local-DIFFI which was shown to have better runtime and performance [26]. The other feature importance tool used was SHAP. The SHAP method is used to explain individual model predictions by computing individual feature contributions, it is based on the game theory of optimal Shapley values. The SHAP value is an estimation of the Shapley values using either kernelSHAP or treeSHAP [27]. SHAP is mostly model agnostic but works well with Isolation Forest since the tree version of SHAP can be used with it resulting in better speed and performance [28].

3.3.3. Deep Autoencoder Autoencoders are a type of neural networks. They are often used for data compression. They start with an input layer that has the same number of neurons as features. After that there is the encoder phase, where each layer has fewer neurons than the previous one. This continues until the layer called the bottleneck is hit, this is the layer with the fewest neurons in the network. After this comes the decoding phase where the neural network tries to reconstruct the original data from the bottleneck. Each layer has more neurons than the previous one until the model reaches the output layer which has the same number of neurons as the feature set and input layer.

Autoencoders can be used in anomaly detection by viewing the mean squared error or the difference between the input and output layer. If the difference is low then the data is easily reconstructable from the bottleneck layer and therefore not an outlier. However if the difference is high then the Autoencoder has a hard time reconstructing the data and that datapoint is considered an outlier [29, Chapter 14]. When training the Autoencoders, we experimented with the various hyperparameters and the architecture, and found that the following combination gave the best results: learning rate$= 0.0001$, $\beta_1 = \beta_2 = 0.9$, batch size=128, 6 layers, with a varying number of nodes per layer depending on the number of features we trained on. The data was scaled so the higher value features would not dominate when training the models. To calculate the score of each feature the difference between the original value of the record and the predicted value from the model is calculated. The distribution of this difference turned out to be too large and the values were not all positive. To solve this we took the absolute and log value of the differences. This feature score was then calculated for every record for each feature and the final score of each prediction was the sum of these feature scores.

3.3.4. K-Means K-means is a clustering algorithm. It starts by receiving the dataset and inputs x random points called centroids. Then for each point in the dataset, it calculates what the closest centroid is, using Euclidean distance. After assigning each datapoint to a centroid it calculates the center of those clusters and sets the centroids to the center of the corresponding cluster. This is repeated for n iterations. It is possible to tune how many clusters the model has and how many iterations it goes through. Since the centroids are chosen at random at the beginning, this process is repeated k times and chooses the best run based on the sum of squared error from each point to the closest centroid [30]. Running K-means on the outliers helps to see if the anomalies are clustering together or not. Another approach is combining the outliers with either non-outliers or the rest of the data, in other words running K-means on the combined data where we know the outliers. The clusters would then show how many data points it contains and how large the ratio between outliers and non-outliers in each cluster is.

3.3.5. Unsupervised learning procedure The models, Isolation Forest and Deep Autoencoder, were trained on all the features in the Unsupervised features table and receive an anomaly score for all rows in the data. Then the fraction of the rows that received the highest anomaly score were inspected, marked as outliers and sent through a few local feature importance tests. In a production setting all anomalies would be flagged and sent to investigators along with an explanation of why it is an anomaly, i.e. the features it scored highly on in the feature importance test. This process was repeated for different combinations of features to see what sets of outliers get flagged. This was done to see if the original outlier detection, with all features, missed important outliers and to use the feature combinations to define a certain money laundering behavior. These outliers were saved in different files to analyze the intersections of addresses between different feature sets. The different feature combinations and the money laundering behaviour they define can be seen in Table 1.

Table 1. Feature combinations and money laundering behaviours

Behaviour	Description
Transacting in similar amounts (SA)	This feature combination defines a money laundering behaviour that looks at addresses that have a similar amount of income and outcome.
High Value Transactions (HV)	This feature combination defines a money laundering behaviour that looks at addresses that transact in high volumes.
Transaction patterns (P)	This feature combination defines a money laundering behaviour that looks at addresses that follow certain patterns when transacting.

4. Results

In this section the results of the unsupervised learning models will be presented. First the distribution of anomaly scores is shown and compared to the anomaly scores of the outliers found by the two models mentioned in Section 3.3. Then the feature importance is shown, first for the models trained on all features and then for the ones trained on feature combinations mentioned in table 1. In Section 4.3 the results from the clustering analysis are presented. Finally Section 4.4 shows a few addresses that were found to be outliers.

4.1. Outliers detected

The figures in this section show the distribution of addresses based on the anomaly scores. The first figure shows the anomaly score distribution for all of the rows in the training set but the next figure shows the outliers, or the 1% of the addresses with the highest anomaly score, which sum up to 14661 outliers. The x-axis on the graphs shows the amount of addresses in each column while the y-axis shows the anomaly score. A low anomaly score for the Isolation Forest model is around or below 0.5. Higher values represent more anomalous addresses.

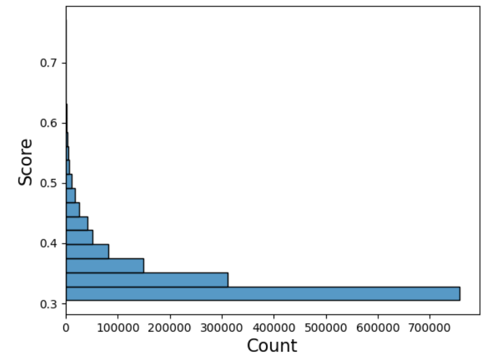

Figure 1. Anomaly scores w. All features for all rows

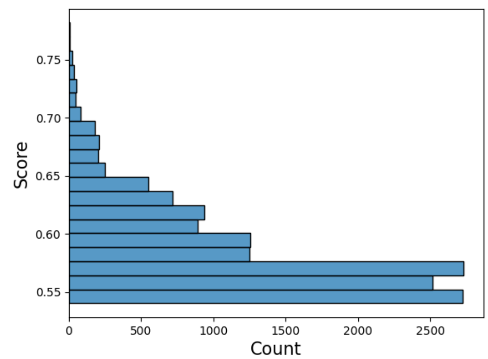

Figure 2. Anomaly scores w. All features for outliers

4.1.1. Isolation Forest The figures in this section show the graphs from the Isolation Forests. Figure 1 is a histogram of all anomaly scores when it was run with all features and Figure 2 shows the same histogram but with a cutoff so it only shows the top 1% of outliers with the highest anomaly score.

Similar histograms were created for the money laundering behaviors that were defined in Table 1 but we decided not to include them in this paper[4]. The results from those models were comparable to the ones shown in Figure 2, with the main difference being that the outliers from the feature combination models had a more uniform distribution. This is most likely because each outlier in the all features model might have a lot of anomalous features but they might also have features that are not anomalous that "pull" them back towards the non-outliers. This does not happen as much for the other three feature combination models since they have fewer features and the features they do have are more correlated, i.e an address with an anomalous value in one feature is likely to also have them in the other features.

[4] Authors are happy to share those figures on request.

4.1.2. Deep Autoencoder The figures in this section, Figures 3 and 4 show the distribution of the anomaly score for the addresses trained on the deep Autoencoder model. It is important to note that the anomaly scores for Isolation Forest and Autoencoder are not related. i.e. an address with a 0.8 anomaly score for Autoencoder is not explicitly more anomalous than an address with a 0.7 in the Isolation Forest model and vice versa. However, we noticed that the shape of the histograms is similar for both models where most of the addresses have a low score and tend to group together while only a few addresses receive higher anomaly scores. The number of addresses in figure 4 decreases as the model score increases. This means that the model can predict the behavior of most addresses quite easily but there are a few outliers that are harder to predict and they receive higher scores than the others. This was also done for the 3 feature combinations described in table 1. This was similar to the results in section 4.1.1, where we can see the same patterns in the figures.

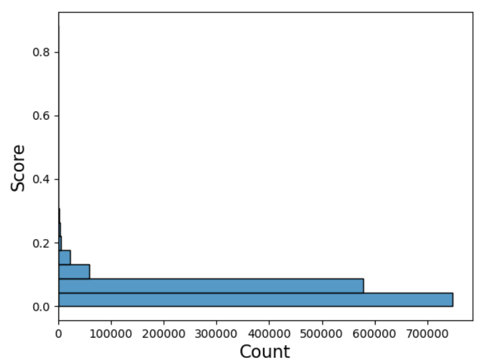

Figure 3. Anomaly scores w. All features for All rows

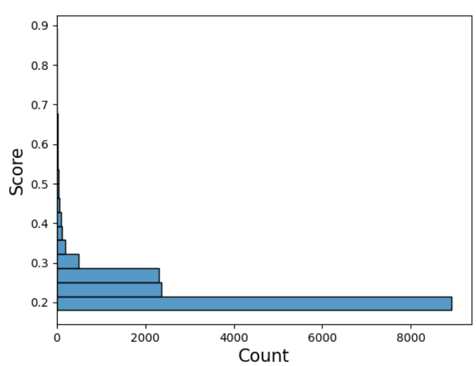

Figure 4. Anomaly scores w. All features for outliers

4.2. Feature Importance

Figures 5 and 6 show the distribution and the most important features, according to SHAP, for a sub-sample of addresses and outlier addresses, respectively. The sample in the SHAP plot in figure 5 contains all 14661 outliers and 73305 non-outliers chosen at random while the SHAP plot in figure 6 only includes the 14661 outliers. Each dot on the plot represents a single feature for a single address. The y-axis represents the most important features with the most important feature at the top and the least important at the bottom while the x-axis represents the SHAP values. A low SHAP value implies that the corresponding feature was anomalous for that address. Blue dots represent low feature values while red dots represent high feature values for the corresponding features.

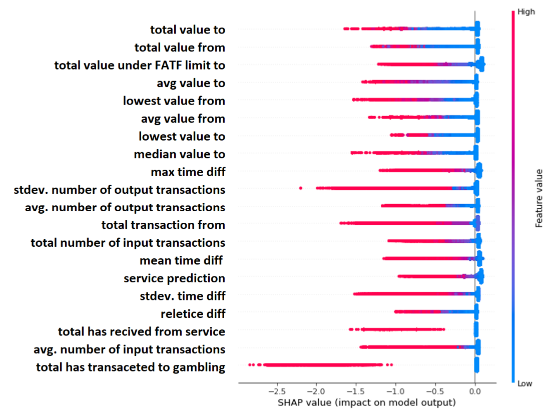

Figure 5. SHAP summary plots, for all features in the whole sample

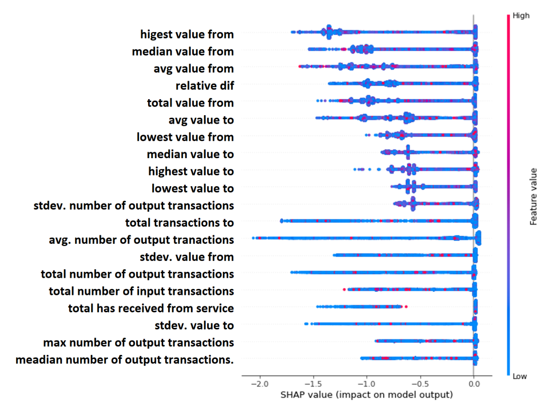

Figure 6. SHAP summary plots, for all features for the outliers

The SHAP values for the sample set (see Figure 5) clearly favor high feature values as more anomalous, meanwhile the outlier set (see Figure 6) contains mostly blue dots, representing low feature values compared to other outliers. The red dots on the outlier sets presumably have really high feature values compared to other outliers. This means high feature values have a strong correlation to outliers. The SHAP plots for the all features model(AF) show that the features that seem to be the most correlated to anomalous behaviour are value related, i.e a high value transaction is anomalous. This was further confirmed when taking a look at the intersection of outliers for the different feature combinations.

As can be seen in Table 2, which shows the intersection of outliers between different feature combinations, the high value feature combination has the largest intersection with AF which means that the AF model leans towards high value related anomalies.

Table 2. Intersection between behaviors for Isolated Forest

Intersection	Count
$Count(AF \cap SA)$	2929
$Count(AF \cap HV)$	9386
$Count(AF \cap P)$	1273
$Count(SA \cap HV)$	3504
$Count(SA \cap P)$	94
$Count(HV \cap P)$	2929

4.3. K-Means

The goal of the K-means model was to see if our outliers would cluster together. The dataset consisted of 186974 datapoints that were not marked as outliers by the Autoencoder models, and 56698 outlier datapoints, consisting of the union of outliers found by all Autoencoder models. The outlier datapoints are around 23% of all the datapoints. The K-means clustering algorithm was set up with 10 centroids, the number of different iterations was 20 and the max iterations for each iteration was set at 600. The results can be seen in Table 3 in a way that shows the total number of datapoints in each cluster and how many outliers are in the cluster. The table shows that most of the clusters contain either high or low percentages of outliers. The results are particularly interesting for clusters 6 and 4, where over 80% of the datapoints are outliers. It implies that these datapoints have some distinct behaviour that defines them. The non-outlier datapoints in the outlier heavy clusters could be investigated further since they share commonalities with the outliers. An avenue of research would be to see what distinguishes these datapoints from the outliers.

4.4. Suspicious individual addresses

This section takes a closer look at four interesting addresses, their feature importance scores and their feature values. These addresses were all marked as outliers in one or more models described above. In Table 4 we can see the address id[5] of the 4 addresses we analyzed.

[5]Unique identifier used to identify a particular address

Table 3. Results from K means

Label	Count	Outliers (%)
1	43460	28010(64.5%)
2	55723	3568(6.4%)
3	5248	1962(37.4%)
4	12438	10286(82.7%)
5	43013	505(1.2%)
6	9275	8214(88.6%)
7	51683	849(1.6%)
8	8963	467(5.2%)
9	7669	28(0.4%)
10	6200	2809(45.3%)

Table 4. Addresses

Address nr.	Address id
Address 1	1KwA4fS4uVuCNjCtMivE7m5ATbv93UZg8V
Address 2	17W4PZ2PS3KksDy5T6yE7FaJsgjmnYMXuP
Address 3	356fU64uSTydGcu87cBEnLtSDseVA785KV
Address 4	17gR9ybNYTWA1W1br8QYH3RoQuYU95Bn22

4.4.1. Address 1 The first address we discuss is the most anomalous address found by the Isolation Forest model with all features with an anomaly score of 0.786.

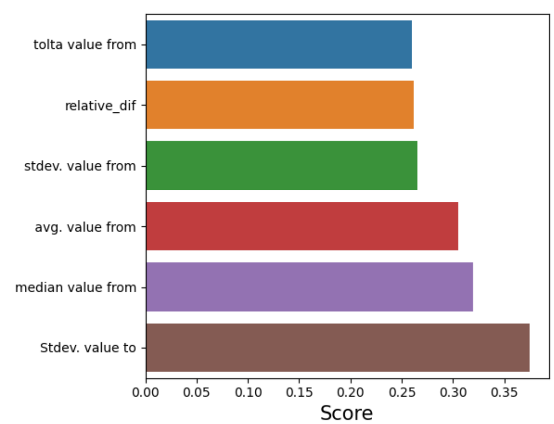

Figure 7. Feature importance scores for address 1

We can see from table 5 that this address transacts high values during the three day examination period. It is still active today[6]. This address has transacted around 73.000 times. This behavior is considered suspicious, and the address would be flagged and sent to an investigator.

[6]See blockchain.com

Table 5. Values for address 1

Feature	Value
std. dev. value to	688.24btc
median value from	1602.3btc
avg. value from	4380.0btc
std. dev. value from	5713.99btc
relative dif.	17351.7btc
total value from	17520.23btc

4.4.2. Address 2 The next address we take a closer look at is the most anomalous address found by the Isolation Forest model with transaction pattern features with an anomaly score of 0.865. Figure 8 shows the local feature importance of this address according to the MI Local DIFFI method.

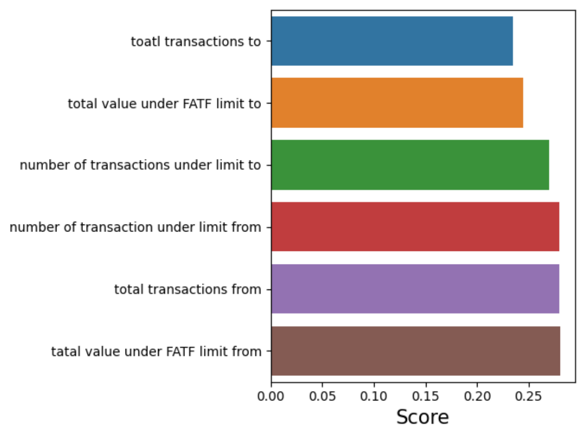

Figure 8. Feature importance scores for address 2

Table 6. Values for address 2

Feature	Value
total value under FATF limit from	10841$
total transactions from	33
number of transactions under limit from	29
number of transactions under limit to	136
total value under FATF limit to	4281$
total transactions to	146

Table 6, shows the values of the 6 most important features for this address. As can be seen this address is transacting quite often and almost always under the FATF threshold of a 1000$ as mentioned in Section 2.2. This could mean they are exploiting the limit by transacting often in amounts below the limit and avoiding binary detection methods that only look at one transaction at a time. As with the other addresses this would be flagged as a potential money launderer for further investigation.

4.4.3. Address 3 The third address is one of the most anomalous addresses found by the Autoencoder model with all features.

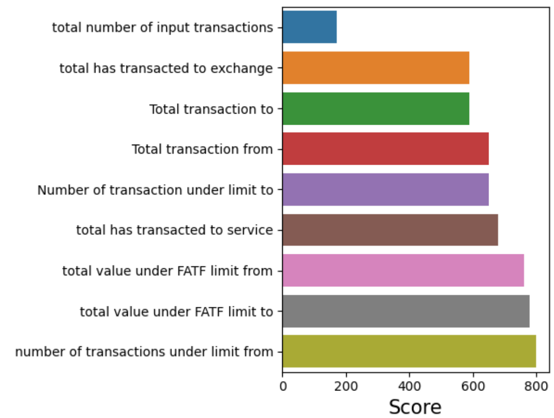

Figure 9. Feature importance for address 3

Table 7. Values for address 3

Feature	Value
number of transactions under limit from	5534
total value under FATF limit to	2452236$
total value under FATF limit from	2439177$
total has transacted to service	1256
Number of transaction under limit to	5731$
Total transaction from	11681
Total transaction to	11891
total has transacted to exchange	923
total number of input transactions	27738
exchange prediction	0.0216

Figure 9 shows the most important features of this address, i.e the features the Autoencoder had the most difficulty predicting. The most important features center around how many times it is transacting and the total value of transactions. That is not surprising as the address has transacted over 66,022 times and received a total of 154.514 BTC and has sent a total of 154.514 BTC[7]. By looking at table 7 we suspect that this address could be an exchange or service since a lot of the transactions are small and the address is transacting often, it is also transacting frequently with other exchanges which is a typical behaviour of an exchange. This address is however not labeled as an exchange or service in the walletexplorer dataset and neither was it predicted to be one by our supervise models. This does not exclude the possibility that it is

[7] See blockchain.com

doing something illegal since transacting to VASP's in large amounts can be seen as a suspicious behaviour as can be seen in section 2.2.

4.4.4. Address 4 The last address we examine in this paper was one of the most anomalous addresses found by the Autoencoder model with all features.

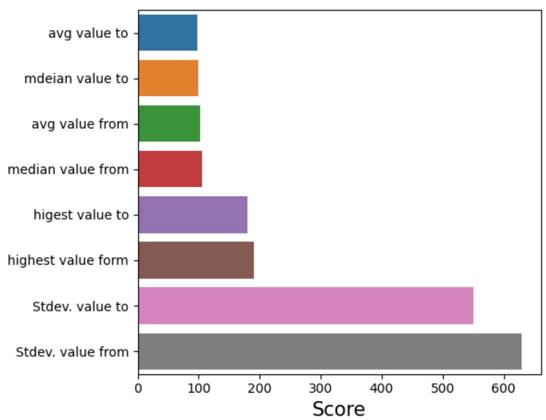

Figure 10. Feature importance for address 4

Table 8. Values for address 4

feature	value
Std. dev. value from	4163 btc
Std. dev. value to	4163 btc
Higest value from	5887 btc
Higest value to	5887 btc

Post-hoc analysis of this address revealed suspicious behavior. It has only transacted 4 times[8]. On the first transaction it receives a small amount of Bitcoin. A few minutes later a second transaction comes in where the address receives a large sum of Bitcoin. It then immediately sends the low value to another address and then the high value transaction to a different address. After these transactions this address is never used again. Because it is transacting in both low and high values we can see that the feature, "stdev. value from", has the highest feature importance in Figure 10. The total value of these transactions is 5887 Bitcoin, as can be seen in Table 8, that go in and out in just 40 minutes. This address is clearly transacting in ways that merits further investigation.

[8]see blockchain.com

5. Conclusion

Cryptocurrencies are an ever growing part of the financial market. With new currencies emerging and existing ones growing larger the need for investigating the legitimacy of the source of the funds is imperative. Through our research, we argue that using unsupervised models to find anomalous behaviours in the bitcoin network can lead to a better way of discovering illicit actors. We used two different models to find anomalous bitcoin addresses, Isolation Forest and Autoencoder, with 3 days' worth of data. We took the 1% of addresses with the highest anomaly scores and labeled them as outliers, giving a total of 14661 addresses. Out of the 1.5 million addresses in this dataset the two models found 7000 of the same addresses as anomalies when training on the same feature set. With this process we have decreased the total amount of addresses investigators would need to analyze to look for money laundering and other suspicious activities on the Bitcoin blockchain.
Investigators can use the local feature importance scores to get a better idea of why an address is anomalous. Explainable AI gives investigators a powerful tool to understand the specific elements of an actor's behaviour that may be suspicious and requires additional examination. Section 4.4 shows small examples of the data investigators would receive to conclude whether or not the address is laundering money.
The methods detailed in this paper have been shown to work to detect money laundering in fiat environments and similar methods are currently in use at Lucinity [5, 6]. We have shown that using these methods on the bitcoin network shows promising results on detecting addresses that should be investigated and can therefore conclude that using unsupervised models on the Bitcoin blockchain helps detect potential money laundering and empowers investigators with intelligent insights that improve the accuracy and efficiency of an investigation.

Our research shows that anomaly detection is a feasible approach for detecting possible money laundering in Bitcoin. However, our methods could be extended in several ways to improve such anti money laundering detection methods. One approach would be to group addresses by wallets since according to Reid et al.it is common practice for users of Bitcoin to own multiple addresses stored in one wallet in a one to many relationships between private and public keys. With this approach the feature set could more accurately represent the behavior of a singular actor [31].
Graph based approaches have also shown promise in anomaly detection with blockchain data. It would be interesting to see the results of a graph based model on this dataset [15].

6. Acknowledgement

The authors wish to thank Lucinity for giving us the chance to work on this project, and specially our supervisors at Lucinity, Justin Bercich, and Kristín Bergþórsdóttir, for their consistent support and guidance during the time of this project.

References

[1] S. Nakamoto, "Bitcoin whitepaper," *URL: https://bitcoin. org/bitcoin. pdf-(URL date: 17.07. 2019)*, 2008.

[2] "Cryptocurrency prices, charts and market capitalizations." Accessed: 26.04.2021.

[3] S. Kethineni, Y. Cao, and C. Dodge, "Use of bitcoin in darknet markets: Examining facilitative factors on bitcoin-related crimes," *American Journal of Criminal Justice*, vol. 43, no. 2, pp. 141–157, 2018.

[4] K. A. Hausfeld, G. Stratton, D. Hamid, J. L. Hare, and B. Krystek, "The new anti-money laundering act of 2020: A potential game-changer for enforcement and compliance." `https://www.dlapiper.com/en/europe/insights/publications/2021/01/new-aml-act-is-a-game-changer/`. Accessed: 26.04.2021.

[5] E. L. Paula, M. Ladeira, R. N. Carvalho, and T. Marzagao, "Deep learning anomaly detection as support fraud investigation in brazilian exports and anti-money laundering," in *2016 15th ieee international conference on machine learning and applications (icmla)*, pp. 954–960, IEEE, 2016.

[6] T. Bercich, "Xai fuels human ai to improve aml programs," *Lucinity*, Dec 2020.

[7] B. Baesens, V. Van Vlasselaer, and W. Verbeke, *Fraud analytics using descriptive, predictive, and social network techniques: a guide to data science for fraud detection*. John Wiley & Sons, 2015.

[8] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *International Conference on Financial Cryptography and Data Security*, pp. 6–24, Springer, 2013.

[9] 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.

[10] "virtual assets red flag indicators of money laundering and terrorist financing _2020," Sep 2020.

[11] T. Pham and S. Lee, "Anomaly detection in the bitcoin system-a network perspective," *arXiv preprint arXiv:1611.03942*, 2016.

[12] P. Monamo, V. Marivate, and B. Twala, "Unsupervised learning for robust bitcoin fraud detection," in *2016 Information Security for South Africa (ISSA)*, pp. 129–134, IEEE, 2016.

[13] P. M. Monamo, V. Marivate, and B. Twala, "A multifaceted approach to bitcoin fraud detection: Global and local outliers," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 188–194, IEEE, 2016.

[14] M. Bartoletti, B. Pes, and S. Serusi, "Data mining for detecting bitcoin ponzi schemes," in *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pp. 75–84, IEEE, 2018.

[15] 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.

[16] A. B. Turner, S. McCombie, and A. J. Uhlmann, "Discerning payment patterns in bitcoin from ransomware attacks," *Journal of Money Laundering Control*, 2020.

[17] I. Alarab, S. Prakoonwit, and M. I. Nacer, "Comparative analysis using supervised learning methods for anti-money laundering in bitcoin," in *Proceedings of the 2020 5th International Conference on Machine Learning Technologies*, pp. 11–17, 2020.

[18] J. Lorenz, M. I. Silva, D. Aparício, J. T. Ascensão, and P. Bizarro, "Machine learning methods to detect money laundering in the bitcoin blockchain in the presence of label scarcity," *arXiv preprint arXiv:2005.14635*, 2020.

[19] H. G. Goldberg, W. T. Young, A. Memory, and T. E. Senator, "Explaining and aggregating anomalies to detect insider threats," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pp. 2739–2748, IEEE, 2016.

[20] M. Jourdan, S. Blandin, L. Wynter, and P. Deshpande, "Characterizing entities in the bitcoin blockchain," in *Data Mining Workshop (ICDMW), 2018 IEEE International Conference on*, pp. –, IEEE, 2018.

[21] M. Jourdan, S. Blandin, L. Wynter, and P. Deshpande, "A probabilistic model of the bitcoin blockchain," in *Computer Vision and Pattern Recognition Workshop (CVPRW), 2019*, pp. –, IEEE, 2019.

[22] G. Hileman and M. Rauchs, "Global cryptocurrency benchmarking study," *Cambridge Centre for Alternative Finance*, vol. 33, pp. 33–113, 2017.

[23] "Walletexplorer." `https://www.walletexplorer.com/`. Accessed: 28.04.2021.

[24] F. T. Liu, K. Ting, and Z.-H. Zhou, "Isolation forest," pp. 413 – 422, 01 2009.

[25] M. Carletti, M. Terzi, and G. A. Susto, "Interpretable anomaly detection with diffi: Depth-based feature importance for the isolation forest," *arXiv preprint arXiv:2007.11117*, 2020.

[26] K. Bergþórsdóttir, "Local explanation methods for isolation forest: Explainable outlier detection in anti-money laundering," Aug 2020.

[27] C. Molnar, *Interpretable machine learning: A guide for making black box models explainable*. Leanpub, 2020.

[28] "shap." `https://shap.readthedocs.io/en/latest/`. Accessed: 05.05.2021.

[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org/contents/autoencoders.html`.

[30] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar, *Introduction to data mining*. 1 ed., 2014.

[31] F. Reid and M. Harrigan, "An analysis of anonymity in the bitcoin system," in *Security and privacy in social networks*, pp. 197–223, Springer, 2013.