DETECTING SUSPICIOUS TRANSACTIONS ON BITCOIN NETWORK USING UNSUPERVISED LEARNING

YOSSAPOL WITAYANONT

Graduate School Srinakharinwirot University

2023

การตรวจจับธุรกรรมต้องสงสัยบนเครือข่ายบิทคอยน์ด้วยการเรียนรู้แบบไม่มีผู้สอน

ยศพล วิทยานนท์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการข้อมูล
คณะวิทยาศาสตร์ มหาวิทยาลัยศรีนครินทรวิโรฒ
ปีการศึกษา 2566

DETECTING SUSPICIOUS TRANSACTIONS ON BITCOIN NETWORK USING UNSUPERVISED LEARNING

YOSSAPOL WITAYANONT

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of MASTER OF SCIENCE

(Data Science)

Faculty of Science, Srinakharinwirot University

2023

THE THESIS TITLED

DETECTING SUSPICIOUS TRANSACTIONS ON BITCOIN NETWORK
USING UNSUPERVISED LEARNING

BY

YOSSAPOL WITAYANONT

HAS BEEN APPROVED BY THE GRADUATE SCHOOL IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE MASTER OF SCIENCE
IN DATA SCIENCE AT SRINAKHARINWIROT UNIVERSITY

(Assoc. Prof. Dr. Chatchai Ekpanyaskul, MD.)
Dean of Graduate School

ORAL DEFENSE COMMITTEE

..... Major-advisor
(Asst. Prof.Waraporn Viyanon, Ph.D.)

..... Chair
(Asst. Prof.Akara Prayote, Ph.D.)

..... Committee
(Asst. Prof.Sirisup Laohakiat, Ph.D.)

Title	DETECTING SUSPICIOUS TRANSACTIONS ON BITCOIN NETWORK USING UNSUPERVISED LEARNING
Author	YOSSAPOL WITAYANONT
Degree	MASTER OF SCIENCE
Academic Year	2023
Thesis Advisor	Assistant Professor Waraporn Viyanon , Ph.D.

This research is the study and development of unsupervised learning algorithms to detect suspicious entities on the Bitcoin network. The objective is to develop a practical model for detecting anomalies in the Bitcoin network. This study was divided into two tasks, which are transaction and wallet address. The statistical techniques are applied for feature engineering and a Histogram-based Outlier Score (HBOS) and Isolation Forest (IForest) algorithms are trained and evaluated. The evaluations utilized were visualization, dual, and known-thieves evaluations. The result showed a similar detection for both algorithms. While HBOS has a higher wallet visualization score at 0.423, Isolation Forest yields better scores on transaction visualization, dual, and known-thieves evaluations with scores of 0.713, 0.681, and 0.035, respectively.

Keyword : Anomaly Detection, Unsupervised Learning, Bitcoin

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Waraporn Viyanon, for her invaluable guidance and support throughout my master's program. Her expertise and encouragement helped me to complete this research and write this thesis.

I would also like to thank the thesis committee for providing valuable feedback and suggestions. Their insights and guidance were instrumental in helping me to shape my research and write this thesis.

Finally, I would also like to thank my friends and family for their love and support during this process. Without them, this journey would not have been possible.

YOSSAPOL WITAYANONT

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Background

Recently, cryptocurrencies have been adopted as an asset as well as a payment method. Among all existing cryptocurrencies, Bitcoin is one of the most well-known, well-adopted, and widespread across the globe. It has enabled financial services for a relatively low price, especially for international transactions. The interesting feature is that it is available for everyone not only for the rich, but also for the unbanked.

As Bitcoin has been rapidly adopted, it attracts different types of people, not only investors and online sellers but also scammers and wrongdoers. Bitcoin is being abused because of its characteristic of pseudo-anonymity. The utopia idea of financial service for everyone (i.e., unbanked) opens a door for abusers to hide their identities. Consequently, Bitcoin has become a preferred payment method by criminals. According to the digital footprint on a public blockchain, the ransomware market was worth USD 12,768,536 in 2017 (Baek, Oh, Kim, & Lee, 2019).

Money laundering has been a huge challenge for years because it has devastating economic consequences and is closely related to terrorist financing (Mabunda, 2018). The corrupted cycle originates from the lack of income and tax declaration of underground businesses. Therefore, their operating costs are lower than their competitors, which is a great competitive advantage. As a result, regulators gain nothing in return while underground businesses collect massive number of profits none of which contributes to the development of the country. Consequently, governments receive fewer funds, and due to limited resources, many regulations are weakened, including the Anti-Money Laundering (AML) regulation which is responsible for the issue. As AML is weakening, underground businesses are operating without any obstructions and gaining even more profit. Then, the cycle is repeated.

Bitcoin should not be blamed for fueling money laundering. Yet admittedly, its anonymity obscures the investigation. It does not only hide the identity of participants

but their exact geolocation as well. Therefore, identifying fraud in Bitcoin is tougher than fiat currency.

Since cryptocurrencies are relatively new to the world, there is no common ground on how to tackle their misbehavior. This research aims to combine some techniques and create a benchmark for detecting anomaly transactions by using Bitcoin as a case study.

Objectives of the study

1. To create anomaly detection models that can detect suspicious transactions and wallets.
2. To determine which model performs the best for detecting suspicious transactions and wallets.
3. To study features that influence the decision-making of each model.
4. To study normal and abnormal patterns in Bitcoin transactions.

Significance of the study

Anonymity is an important concern when dealing with money laundering because it takes away non-repudiation (inability to deny). Bitcoin is pseudo-anonymous by nature because of the use of cryptography. While anonymity protects users' privacy, it is an obstruction to identify users when needed. Evildoers take advantage of this characteristic of Bitcoin to increase their chance of success in illegal transactions.

Immutability and decentralization are other challenges in preventing money laundering in terms of cryptocurrency. Bitcoin is based on blockchain technology in which information stored is copied across the network and they are unchangeable. These two characteristics of blockchain make it nearly impossible to prevent real-time transactions. Due to an absence of central administration, unlike SQL and NoSQL databases, no one can control the Bitcoin network unless he or she can control more than half of the nodes in the network, which is nearly impossible. In addition, reverting transactions is very unlikely as well, because the data stored is proof of cryptographic

work which means editing data will invalidate cryptographic hash, in other words, editing is rejected by the network.

These challenges make detecting illicit transactions important because it is going to be the first step to taking down wrongdoers. As mentioned earlier, Bitcoin has no central administration, therefore, accounts and transactions cannot be frozen from a single point of control. AML inspectors must further their investigation by locating the abuser and cooperating with authorities to arrest them. It is going to be costly if inspectors do not choose the first step wisely because all efforts can go to waste and put them back to square one.

Currently, the Anti-Money Laundering (AML) authority of Thailand enforces a law that requires financial institutions to report any electronic transactions above 100,000 THB as suspicious transactions. Law in general is rarely updated and since it is public information, criminals are aware which means they can easily prevent their activities from being monitored and inspected.

While this study refers to money laundering frequently, this work still lacks information to make the detection of money laundering happen. Hence, an assumption of anomaly is applied. Since anomaly refers to a minority of events or occurrences, money laundering falls into this definition perfectly. Thus, money laundering is referred to as anomaly detection in this study.

Therefore, this research studies anomaly patterns in the Bitcoin network and creates models that help inspectors make a better decision on their first step of investigation. This study aims to improve the effectiveness of current practice and reduce the effort spent on irrelevant cases.

Scope of the study

Dataset

There are 2 options used to gather blockchain data. Google BigQuery API is utilized for data exploration because it is user-friendly. However, it does not provide up-to-date data since it has incomplete data for 2018 and later (BigQuery, 2019). On the other

hand, a blockchain explorer website (Blockchain.com) has complete data up to the latest block. Therefore, Blockchain.com's API is used for later processes in the study.

The Bitcoin data between 1st July and 14th July 2021 is used in the study. As Bitcoin is categorized as a high-risk asset, the price of BTC is diverse. Therefore, the selected timeframe is chosen because of a low fluctuation of its price. This helps to prevent rate conversion with the least effect of bias.

Features

This research is about detecting anomalies in the Bitcoin network by studying related works, public Bitcoin data, and the architecture of the Bitcoin network. There are three groups of features studied as follows:

1. Raw features

Raw features are features that can be found in the Bitcoin network. These features are public transactions stored in the blockchain. Qualitative variables of raw features are not being used in this study. Raw features are:

- Number of receivers
- Input amount in BTC
- Output amount in BTC (to each receiver)

2. Basic statistics

Basic statistical features are calculated from raw features. Basic statistics features are:

- Ratio of receiving transactions to all transactions
- Average spending amount per transaction
- Average receiving amount per transaction
- Ratio of payback amount
- The frequency of spend at a certain BTC amount
- The frequency of receive at a certain BTC amount

3. Extra statistics

Extra statistical features are derived from public information on wallet addresses. Extra statistics features are:

- Lifespan of wallet
- Total BTC spent
- Total BTC received
- Total number of transactions
- Total number of spent transactions
- Total number of received transactions
- Total number of coinbase transactions
- Total number of payback transactions
- Average balance after each transaction
- Standard deviation of balance after each transaction

Algorithms

- Isolation Forest (IForest)
- Histogram-based Outlier Score (HBOS)

Evaluations

- Visualization
- Dual
- Known thieves

Definition of terms

Blockchain is a relatively new architecture introduced to the financial system. The underlying structure is composed of complex elements that have specific names for the technology. Therefore, it is crucial to explain important terms for better understanding.

Table 1 Definition of terms

Terms	Definition
Block height	The number identifying the block starting from 0 (alternative identifier for block hash)

Terms	Definition
BTC	A representation of the Bitcoin unit.
Consensus mechanism	A mechanism used in blockchain systems to achieve an agreement on a single data value
Coinbase	A transaction received as a reward for mining.
Fiat	Currency issued by governments, such as U.S. Dollars and Thai Baht.
Genesis block	The first block of a blockchain.
Mining	A process of verifying transactions by solving mathematical problems.
Minting	A process of creating a new entity in blockchain, for example, a new unspent transaction output.
Node validator / Node	A computer that stores a copy of blockchain information and performs computation to ensure data stored is secure.
Unbanked	People who are not served by a bank, because they do not meet banks' requirements.
Payback / Change	BTC is sent back to the sender as the remaining balance of the transaction.
Unspent transaction output (UXTO)	Analogues to a coin, hold a certain amount of value.
Swamping	A normal instance is mistakenly identified as anomalous
Masking	An Anomaly instance is wrongly identified as a normal

Conceptual of the study

This research analyzes anomaly transactions on the Bitcoin network within a two-week range. Anomalies detected are separated into two schemes: transaction-based and address-based.

The transaction-based attempts to detect suspicious transactions and is studied in two settings. The first setting is for detecting global outliers by using only publicly available features or engineered features. The first setting is studied to explore the general characteristics of outliers of the network, which is useful for overview analysis. The second setting adds wallet address information into the context. The second setting is explored to overcome rule-based detection.

On the other hand, address-based strives to detect suspicious wallet addresses. Unlike transaction-based, address-based only concentrates on local outliers. A wallet address without transaction context is meaningless because it only contains one feature, which is the amount of BTC that a certain wallet address has. And the amount of BTC alone cannot make a suspicious wallet. Hence global outlier detection is ignored.

This research utilizes three anomaly detection algorithms, namely Isolation Forest (IForest) and Histogram-based Outlier (HBOS) along with feature engineering and feature selection.

Hypothesis of the study

1. Unsupervised learning techniques can effectively identify patterns in Bitcoin transaction data that deviate from normal behavior, thereby flagging potentially suspicious transactions.
2. Statistical features can improve models' performance.

Limitations of the study

Bitcoin wallet address is generated by an algorithm with no identity involved or attached to it. Therefore, the wallet address alone cannot identify its owner. Since this research is studying anomaly detection on the Bitcoin network based on its public information, this research implies an assumption that each wallet address is

independent of the other. In other words, one wallet address is assumed to be owned by a single user and each user owns only one wallet. In practice, a single user can own multiple wallet addresses. The assumption is made because detecting criminal networks is out of the scope of this study.

Another limitation is that the labeled dataset is costly. The process of labeling is expensive because it is time-consuming, and requires experts. Therefore, a publicly labeled dataset is rare and incomplete in some ways. For example, a largely labeled public dataset called the Elliptic dataset takes away all the names of its features, which are practically valid for training a new model but invalid to use on the real Bitcoin network.

Expected benefits

1. Models can reliably detect suspicious Bitcoin transactions and wallets.
2. A reliable model can be used for reporting suspicious transactions to the Anti-Money Laundering Organization (AML).
3. Discover fundamental anomalous patterns that can apply to other cryptocurrencies on different types of blockchain architecture.

CHAPTER 2
THEORIES AND RELATED WORKS

Money laundering in cryptocurrencies is a growing concern, prompting research into the application of machine learning and anomaly detection algorithms for Bitcoin transaction analysis. The topics of research are as follows:

1. Bitcoin
2. Money laundering
3. Machine learning
4. Anomaly detection algorithms
5. Literature review related to anomaly detection in the Bitcoin network.
6. Literature discussion

Bitcoin

Bitcoin was invented to eliminate trusted third parties from financial transactions, by a person or a group of persons called Satoshi Nakamoto. Nakamoto claims the shortcomings of a trust-based model include reversible transactions, high fees, and low transaction capacity. Reversible transactions are invented when electronic payment is introduced because transactions can be made without consent between two parties. Furthermore, the mediation comes with higher transaction fees and limiting minimum and maximum transaction size. Finally, Bitcoin was proposed with the ability to transact directly between two willing parties based on cryptographic proof instead of relying on trust. Transactions are non-reversible and stored in multiple machines to maximize security with the assumption that the network has more CPU power than a collective group of attackers.

Bitcoin transactions are based on digital signatures. Each transaction consists of digital signatures of previous transactions (using a private key) and the public key of the next owner. Therefore, the history of a coin transaction is a chain of ownership as shown in Figure 1. Though the receiver could verify the previous owner, it cannot verify if the

previous owner already spent the coin (sign the same coin on a different transaction). To overcome the issue of double spending, Bitcoin broadcasts its information publicly for public participants to determine the first transaction of the coin and agree on a single history of a coin's transaction. Hence, Bitcoin data is available to all participants in the network.

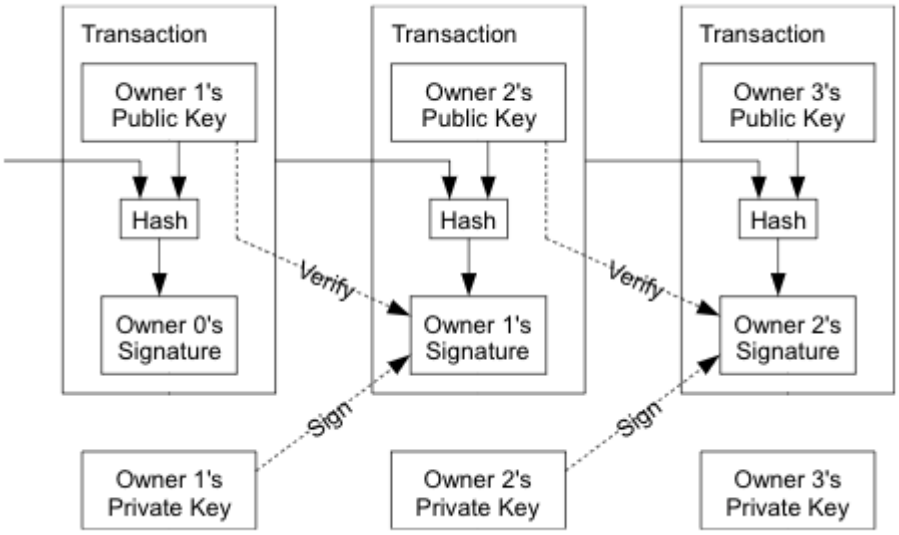


Figure 1 Digital Signature on Bitcoin Transaction

Source: (Nakamoto, 2008)

Bitcoin utilizes a timestamp server to chain all the blocks in the network together. A timestamp server marks the time on a hash of a block and broadcasts the hash. The timestamp ensures the existence of data at specific periods. Each timestamp contains the previous timestamp in its hash to form a chain as evident in Figure 2.

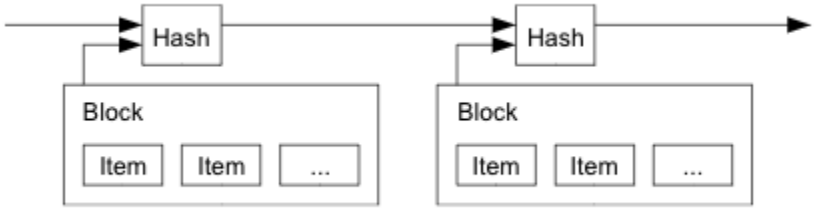


Figure 2 Bitcoin's Block Hash

Source: (Nakamoto, 2008)

Bitcoin implements the proof-of-work algorithm to determine a single source of truth. The proof-of-work is basically one-CPU-one-vote. Therefore, if the majority of CPU power is truthful nodes, then the truthful chain is going to beat the other chain and be accepted as the truth of the network. On the other hand, if attackers want to modify information in the chain, they must redo the proof-of-work of the block and every block afterward until it grows and exceeds the truthful chain, which is nearly impossible.

The proof-of-work difficulty is adjusted dynamically using a moving average, targeting an average number of blocks per hour. The difficulty is increased when the block is generated too fast, and vice versa. The difficulty of the proof-of-work is directly tied to the amount of BTC miners will receive as compensation for the CPU power spent in confirming new blocks.

Money Laundering

Money laundering is a process of disguising illicit money as a licit one, by mixing different types of transactions to hide the source of that money. Money laundering is a serious issue because it is related to many types of crimes, and it deceives the financial situation (Mabunda, 2018).

There are three stages in money laundering: placement, layering, and integration. The placement stage is an initial stage where illegal money is placed into a legitimate entity such as a small business. The placement is practiced by disguising the original crime. The next stage is layering where multiple transactions are made to distance the money from its source. This stage can be done by buying products for the small business. The second step can be repeated to increase the distance of the original source. In the last stage, integration, money is placed back into a legitimate financial institution. The final step is done by depositing directly into the bank or transferring via the bank. The integration phase ensures that the money is placed back into the system and can be used in any lawful transactions.

Cryptocurrency is being abused for placement and layering practices. Unlawful money is exchanged for cryptocurrency. Cryptocurrency offers anonymity which is a preferred characteristic for abusers. Moreover, converting fiat money to cryptocurrency is easier than establishing a small business or buying real estate, as both require bigger amounts of money compared to cryptocurrency which can be exchanged in a smaller amount. In addition, the layering process is even easier with pseudo-anonymity by transferring cryptocurrency across multiple wallets. Not to mention, creating a new wallet is free and requires no documents at all. As a consequence, some cryptocurrencies are utilized as a new option for money laundering.

Machine Learning

Machine learning (ML) is a process of training a machine to predict the future based on historical data. The main characteristic of ML is generalization, the ability to permit the system to work well on unknown data (Awad & Khanna, 2015).

Historical dataset is split into two or three datasets in the learning process usually called training, testing, and validating datasets. The name of each dataset suggestively describes how it is being used. For example, a training dataset is used for training an algorithm.

Machine learning is divided into 4 categories:

1. Supervised learning

Supervised learning is a technique for discovering the association between independent variables and a desired dependent variable. Supervised learning requires historical data with known labels. Label is a desired dependent variable. There are 2 types of problems in supervised learning which are classification and regression.

2. Unsupervised Learning

Unsupervised learning is a technique of grouping similar entities without predefined variables. Unsupervised learning does not require labels. Clustering and dimensionality reduction are examples of unsupervised learning.

3. Semi-supervised Learning

Semi-supervised learning is a combination of supervised and unsupervised learning. Semi-supervised is usually conducted with a small amount of labeled data and a huge amount of unlabeled because labeling requires human resources and it is costly, while unlabeled data is readily available.

4. Reinforcement Learning

Reinforcement learning is explorative and adaptive learning. An intelligent agent is usually introduced in reinforcement settings. The intelligent agent is learning from the configuration of reward and penalty.

Anomaly detection algorithms

There are several algorithms that can be used for anomaly detection in Bitcoin transactions, all falling under the umbrella of unsupervised learning since Bitcoin transaction data is typically unlabeled. Here are selected approaches:

1. Isolation Forest (IForest)

Isolation Forest (IForest) is an anomaly detection algorithm that utilizes decision trees to create partitions on instances, called isolation trees. Unlike other anomaly detection algorithms which are usually based on profiling normal instances before measuring outliers, IForest does not build the inlier profiles, rather it explicitly isolates the anomalies. IForest does not utilize distance or density measures. Hence, it is faster in computation. Moreover, it has a linear time complexity with low constant and requires low memory. A final remark on IForest is that it can scale up to handle large and high-dimensional datasets (Liu, Ting, & Zhou, 2008).

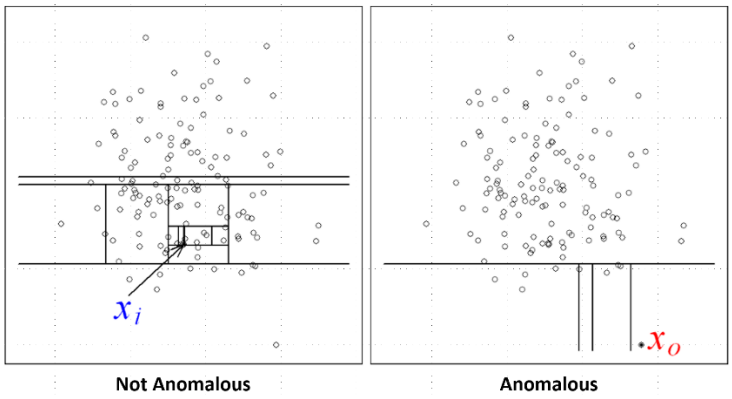


Figure 3 Isolation Forest

Source: (Liu et al., 2008)

IForest creates trees to separate instances by randomly selecting a feature and randomly selecting a value between the minimum and maximum of the selected feature. Then, outliers can be observed from the average path length. While normal instances will have an average path length close to the overall average path length, anomalies have a shorter path length.

There are 2 parameters to the IForest algorithm, which are sub-sampling size and the number of trees. The sub-sampling size controls the training data size (Liu et al., 2008). The number of sub-sampling sizes should be small because IForest works well on detecting swamping and masking on smaller sample sizes. The number of trees is the number of isolated trees created for each sub-sampling. The path lengths usually converge before 100 trees. Any number beyond the convergence is unnecessary as it does not increase in anything but time.

2. Histogram-based Outlier Score (HBOS)

Histogram-based Outlier Score (HBOS) utilizes histograms to distinguish between normal instances and anomalous instances. HBOS is categorized as proximity-based or density-based because it assigns feature values into histogram bins, which literally calculate the distribution of values, and bins with low values indicate greater distance on feature space.

HBOS calculates each feature by constructing an univariate histogram (Goldstein & Dengel, 2012). Then calculate the frequency of the feature. For categorical

data, frequency is obtained by counting the number of instances that fall into each category. There are two methods for calculating the frequency of the feature for numerical data: (1) static bin-width and (2) dynamic bin-width.

Static bin-width is built using the standard histogram technique by splitting all values according to the number of bins. Hence, each bin has a fixed range of values. As a result, data points in high-density bins are less likely to be outliers. On the other hand, dynamic bin width is calculated by the total number of samples divided by the number of bins, which is the expected number of instances for each bin. Then values are arranged from the lowest to the highest and assigned to the bin. Therefore, a bin with a bigger range of values is a low-density bin and it is likely to be the location where outliers reside.

$$HBOS(p) = \sum_{i=0}^{d} log\left(\frac{1}{hist_i(p)}\right) \quad (1)$$

Each dimension, represented by *d*, is calculated separately and then added up for the final outlier score. The term *hist(p)* is normalized between 0 and 1. For any *hist(p)* closer to 1, the result from the *log* operation will be closer to 0, which increases the likelihood of an inlier. The author of HBOS (Goldstein & Dengel, 2012) claims that summarization is less sensitive to error compared to multiplication and also less computational power is required while preserving the order of the scores.

Literature review

Anonymity Analysis of Bitcoin Transactions Using Unsupervised Machine Learning.

The paper (Bivin S. Nair, 2018) utilizes graph theory and the Isolation Forest algorithm to detect anomalous transactions in the Bitcoin network. The graph theory is applied to users' wallet addresses by calculating the in-degree, out-degree, unique in-degree, unique out-degree, in-transaction rate, and out-transaction rate. Later, a Random Forest is experimented with to find the optimal threshold value. In the end, the average isolation

depth is assigned to each instance as an anomalous score. The anomalous score indicates the level of anomaly, a higher anomalous score indicates the higher level of anomaly, and vice versa. The anomalous score is not limited to a positive number. The Random Forest model marks 10,553 transactions from 1,048,576 transactions to be anomalous, or in other words, 1% of transactions are anomalous. The model found an anomalous score of -0.21385 to a threshold to determine whether a given transaction is anomalous or not. Any score above that number is labeled as anomalous.

1. Multi-class Bitcoin-enabled Service Identification Based on Transaction History Summarization.

The paper (Toyoda, Ohtsuki, & Mathiopoulos, 2018) focuses on identifying different types of services available on the Bitcoin network based on transaction history summarization. The paper proposes two schemes of transaction history retrieval, which are address-based and owner-based. Address-based is straightforward, all transactions related to the address are retrieved. On the other hand, owner-based applied address clustering to group some addresses together before retrieving transactions from all addresses in the group as if it is controlled by a single person. There are three steps taken after transaction data is retrieved. First, the amount of change is removed from the retrieved data as it does not contain meaningful information. Second, BTC is converted to USD to reduce the effect of volatility of its price. Third, USD is converted to two significant digits to capture how big a transaction is. After preprocessing, a basic calculation is applied to transaction data. Calculated features are transaction frequency, the ratio of received transactions, the ratio of received coinbase, frequency of received transaction, the ratio of payback, the mean value of inputs in the spent transaction, and the mean value of outputs in the spent transaction. Random Forest is then trained to evaluate the significant distribution of the engineered features. The model achieved 0.72 accuracy on the owner-based and 0.70 on the address-based scheme. The paper concluded that the top three contributed features are transaction frequency, frequency of received transaction, and the ratio of received transactions which contributed 0.52, 0.44, and 0.35 bits respectively.

2. An Evaluation of Bitcoin Address Classification based on Transaction History Summarization.

The paper (Lin, Wu, Hsu, Tu, & Liao, 2019) studies multiclass classification on Bitcoin wallet addresses based on transaction history summarization. The study is conducted as an extension to "Multi-class Bitcoin-enabled Service Identification Based on Transaction History Summarization" with few adjustments. First, an owner-based scheme is renamed as an entity-based scheme. Second, engineered features on that paper are used with the name of basic statistics. The paper proposes two groups of features extracted from transaction data called extra statistics and transaction moments. The extra statistics comprise of lifetime (defined by first and last transactions of the wallet), spent amount in BTC, received amount in BTC, spent amount in USD, received amount in USD, number of transactions, number of spent transactions, number of received transactions, number of coinbase transactions, number of payback transactions, mean value of balance after each transaction in BTC, standard deviation of balance after each transaction in BTC, mean value of balance after each transaction in USD and standard deviation of balance after each transaction in USD. The transaction moment is calculated on block height to capture transaction distribution. The transaction moments consist of overall, spent, received, coinbase, payback, and interval moments. Eight classification algorithms and basic statistics were utilized for evaluating the features. Models are best performed when all groups of features are utilized. Neural Network and LightGBM are the best performers. LightGBM achieves a Micro-F1 score of 0.87 and a Macro-F1 score of 0.86 on an address-based scheme. On an entity-based scheme (owner-based), Neural Network achieves 0.91 and 0.78 for Micro-F1 and Macro-F1 scores. The top five important features in the study are transaction frequency, the mean value of inputs in the spent transaction, the mean value of outputs in the spent transaction, the number of received transactions, and the interval moment.

3. Anomaly Detection in Bitcoin Network Using Unsupervised Learning Methods.

The paper (Pham & Lee, 2016) utilizes graph structure and three unsupervised algorithms to detect anomalous entities in the Bitcoin network. The research is divided

into detecting transactions and users. There are many network features calculated during feature extraction, for example, in-degree, unique in-degree, average in-transaction, average time interval between in-transactions, and similar calculations for out-transactions. There are three algorithms trained for the study, which are k-Means Clustering, Mahalanobis Distance Based Method, and Unsupervised SVM. The k-Means is used as a baseline for evaluation. Though the dataset contains approximately 38 million transactions, only 100,00 data points are utilized for each model during the training process. There are a few evaluation methods used to evaluate the performance of each algorithm, which are Visualization Evaluation, Dual Evaluation, and Known-Thieves Evaluation. The visualization evaluation is calculated by the relative distance between the detected outliers and the centroids from the k-Means algorithm. As anomalies are expected to be far from normal data, a higher value indicates better performance. The dual evaluation assumes that suspicious users create suspicious transactions. Therefore, the calculation is calculated on overlapping predicted values on users and transactions, where a higher value indicates better performance. The third evaluation is observed from real-world known thieves in the dataset. The result shows that Mahalanobis achieved 0.76 on the user graph and 0.83 on the transaction graph for visualization evaluation. Unsupervised SVM achieved similar results, 0.72 on the user graph and 0.86 on the transaction graph. On dual evaluation, Mahalanobis achieved 0.03, while Unsupervised SVM achieved 0.11. On the last evaluation, Mahalanobis detected one known thief, while Unsupervised SVM detected one known loss transaction.

4. A Case Study of Cluster-based and Histogram-based Multivariate Anomaly Detection Approach in General Ledgers.

This paper (Becirovic, Zunic, & Donko, 2020) attempts to detect outliers from general ledgers using cluster-based and histogram-based algorithms on a real company's data from Bosnia and Herzegovina. A general ledger is a record of transactions during the life of an operating company, organized by accounts containing both credit and debit transactions. The raw data contains 4.5 million rows of

transactions, but around 1 million records are filtered out during preprocessing. k-Means and Histogram-based Outlier Score (HBOS) are trained on approximately 3.5 million records. An evaluation is done through simulation with false data. Histogram-based detects 2% of the entire dataset to be outliers, hence 2% margin is used for cluster-based. The result shows that cluster-based is better in precision, yet histogram-based is fast in execution which can be beneficial on real-time anomaly detection.

Literature discussion

An interesting study that highly influences this study is “Anomaly Detection in Bitcoin Network Using Unsupervised Learning Methods” (Pham & Lee, 2016). The study has trained multiple unsupervised algorithms, many of which are density-based, to detect anomalous transactions and wallets utilizing graph structure as features, which yields decent results for detecting anomalous on unlabelled datasets. The study has introduced an interesting framework for anomaly detection and evaluations of unlabelled data. It splits tasks into transaction and wallet, along with visualization, dual, and known-thieves evaluations.

Two years later, Bivin S. Nair (2018) has studied in a similar setting. Both studies utilized unlabelled data and graph structure. While the previous study focused on utilizing density-based algorithms, this study opted for tree-based. However, this study did not inherit the evaluations. Hence, it is difficult to compare these two works.

In the same year, Toyoda et al. (2018) has introduced another trend of study on Bitcoin. The study introduced a different setting, by utilizing a labeled dataset, statistical features, and a tree-based algorithm. Also, tasks are split into address and owner, which emphasize identifying identity in the Bitcoin network.

Later year, Lin et al. (2019) has extended the previous study by training multiple algorithms, including artificial neural networks. The study also introduced two more sets of statistical features, which are called advanced and moments. Features from both groups have highly contributed to the algorithms, especially advanced features.

A year later, Becirovic et al. (2020) has utilized HBOS on big data of 4.5 million records. Though, it is not the best performer, it is being recognized for its speed.

Though the trend of unlabelled data is slowing down, it is not well explored. Pham and Lee (2016) has established a framework and its results still have room for improvement. Since later studies have shown that statistical features have improved the performance of algorithms. This study will combine the framework presented by Pham and Lee (2016) with features from Toyoda et al. (2018) and Lin et al. (2019), also, utilizing a popular algorithm and a fast algorithm which are utilized in Bivin S. Nair (2018) and Becirovic et al. (2020).

CHAPTER 3

RESEARCH METHODOLOGY

The procedure of this research includes the following:

1. Data collection and preparation
2. Feature engineering
3. Modeling
4. Model evaluation
5. Preliminary

Overview of the research process

This research utilizes unsupervised learning algorithms to detect suspicious transactions and wallet addresses in the Bitcoin network. Data collection is the first step after the scope of the study is defined. Raw data from blockchain requires transformation before use. After the required data is acquired, basic statistics and extra statistics are applied to features. Then, unsupervised algorithms are trained on the data along with parameter tuning. Afterward, all models are evaluated with three different types of evaluation. Finally, discussion and comparison are held against each algorithm.

Table 2 Overview of research process

Activities	Months											
	1	2	3	4	5	6	7	8	9	10	11	12
Study theories and related works	✓	✓										
Define the research topic and research conceptual framework			✓									
Research planning and preparation				✓								
Data collection and preparation					✓	✓						

Activities	Months											
	1	2	3	4	5	6	7	8	9	10	11	12
- Collect data from API - Reshape data into the desired format												
Feature engineering - Basic statistics - Extra statistics							✓					
Modeling - Train models - Fine-tuning								✓	✓			
Model evaluation - Evaluate each model - Compare between models									✓	✓		
Research summary											✓	✓

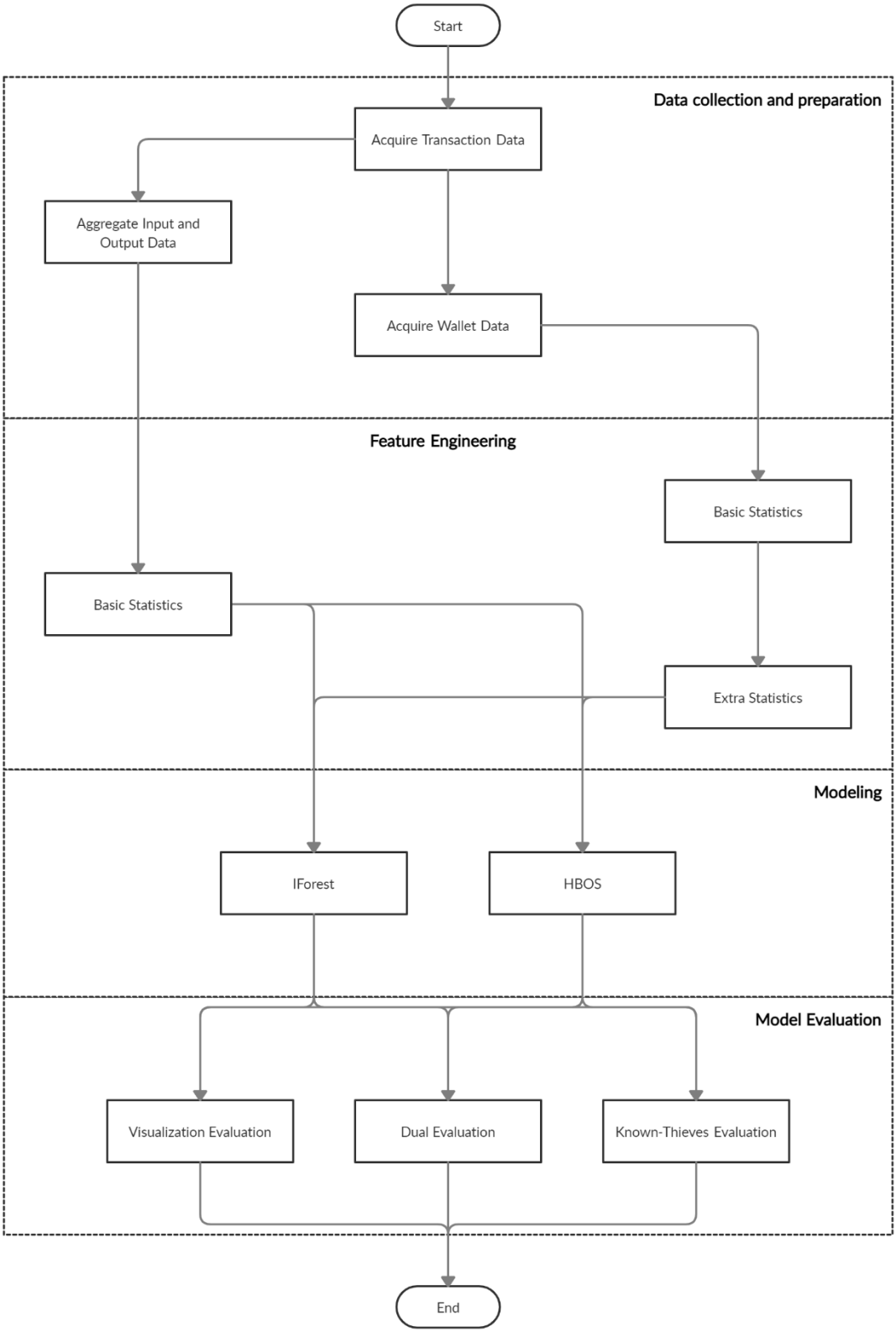


Figure 4 Overview of research process.

Data collection and preparation

Source of data

The Bitcoin network is decentralized which means that its data is distributed across its network. Therefore, Bitcoin data is publicly available to every node participating in the network. However, to participate in the network is computationally expensive and requires a great storage capacity because Bitcoin data is immersive.

There are services that provide Bitcoin information to the public to reduce the stress of Bitcoin users running a node by themselves. These service providers are called blockchain explorers. Every block, transaction, and wallet address can be searched and obtained. A blockchain explorer website is one source of data for this research.

Another source of data is from Google's data warehouse via BigQuery. BigQuery has a public endpoint for querying Bitcoin historical data using an intuitive structured query language (SQL). Data retrieved from BigQuery is in a ready-to-use state, nevertheless, it does not provide recent data. The latest data available to query is 2 years back. Hence it is only used for exploration.

Data collection methods

There are three API endpoints used for collecting data and arranging it in chronological order:

1. Date endpoint

The date endpoint provides information about blocks that are verified on that specific date. The date endpoint accepts one parameter which is a date in Unix timestamp format. The date endpoint is used as a time frame for the dataset. Information retrieved from this endpoint can only be used on the next endpoint for gathering full information about a block.

2. Block endpoint

Block endpoint provides information on a single block. Each block contains tons of transactions in full detail. Block endpoint accepts one parameter which is a block hash. Most of the block information is discarded and only transaction information is kept.

3. Wallet address endpoint

The wallet address endpoint provides information about a specific wallet address. The wallet address endpoint accepts one parameter which is a wallet address. This endpoint provides information about the wallet, for example, spent, received, and balance.

Data preparation

Data retrieved from API is transformed from raw data stored in blockchain into JavaScript Object Notation (JSON). Unlike Comma-separated Values (CSV), JSON does not have a tabular format. Therefore, data requires a little transformation to fit into a data frame. Moreover, block information contains system-specific information, for example, block merkle root and cryptographic nonce, which are not relevant to the study. Hence irrelevant data is excluded during this step.

Feature engineering

Features retrieved from API are limited because the Bitcoin network stores minimum information. Unlike online banking where customers' locations can be obtained, blockchain does not store any clues about its users. Therefore, this research applies statistics to existing features to increase the number of features as well as provide more information about existing features for the algorithms.

All features related to transactions are summarized in Table 3 and every feature related to wallet address is listed on Table 4.

Table 3 Transaction's Features

	Features	Description
Raw features	n_input	Number of inputs in a transaction
	n_output	Number of outputs in a transaction

	Features	Description
	time	The hour in UTC timezone
Basic statistics	total_btc	The total amount of BTC in a transaction

Table 4 Wallet Address's Features

	Features	Description
Raw features	n_tx	Number of transactions
	total_spent_btc	Amount of BTC spent
	total_received_btc	Amount of BTC received
	current_balance_btc	The current balance of BTC
Basic statistics	r_spent	The ratio of spent transaction
	r_received	The ratio of received transaction
	avg_spent	Average spent amount in BTC
	avg_received	Average received amount in BTC
	r_coinbase	The ratio of received transactions from mining
	r_payback	The ratio of spent transaction that has a returned change
	freq_spent_more_1	Number of spent transactions more than 1 BTC
	freq_spent_between_1_01	Number of spent transactions between 1 and 0.1 BTC
	freq_spent_between_01_001	Number of spent transactions between 0.1 and 0.01 BTC
	freq_spent_less_001	Number of spent transactions less than 0.01 BTC
	freq_received_more_1	Number of received transactions more than 1 BTC
	freq_received_between_1_01	Number of received transactions

	Features	Description
		between 1 and 0.1 BTC
	freq_received_between_01_001	Number of received transactions between 0.1 and 0.01 BTC
	freq_received_less_001	Number of received transactions less than 0.01 BTC
Extra statistics	n_coinbase	Amount of BTC received from mining
	n_payback	Number of transactions that contain returned changes
	lifespan	Number of days between the first and the latest transactions
	avg_balance_after_tx	The average balance in BTC after each transaction
	std_balance_after_tx	The standard deviation of balance in BTC after each transaction

Exploratory data analysis

The first exploration is on transaction data. As mentioned earlier, this study is using two-week Bitcoin data from 1st July 2021 to 14th July 2021. However, the specified time frame is based on the block confirmed, which means transactions are made earlier. Figure 5 shows that the majority of transactions are between 30th June 2021 and 13th July 2021. However, some transactions were made earlier such as the one in March. When inspected further, that particular transaction is being verified on 08th July 2021. This can happen because the user has specified a low transaction fee.

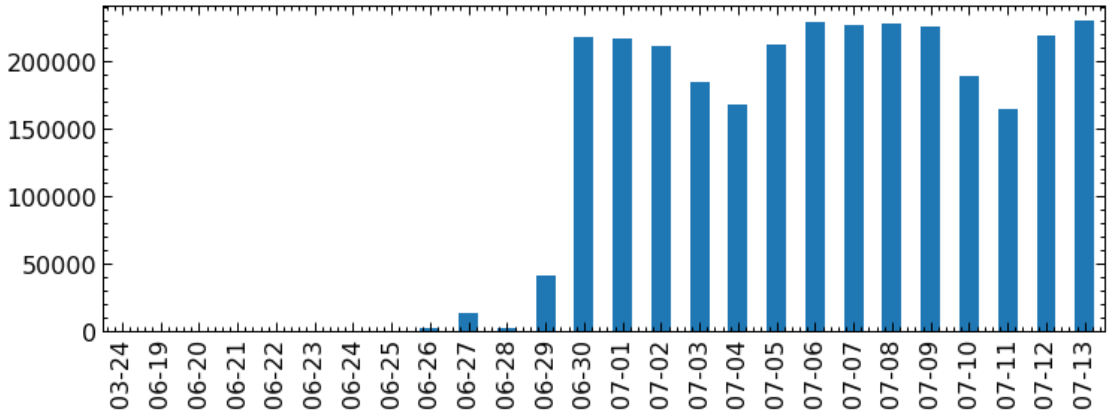


Figure 5 Transactions per day.

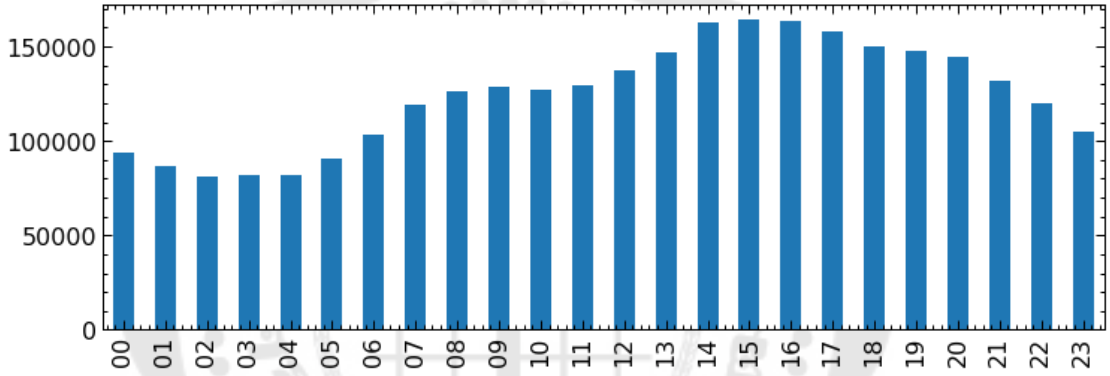


Figure 6 Transaction by hour.

The transaction is then grouped by the hour to determine peak time as shown on Figure 6. Though, the Bitcoin network is global which means transactions should be distributed equally. However, when plotted using UTC timezone, the transaction amount declines during late night. This may infer that dominant Bitcoin users are people who live at or close to the UTC zone.

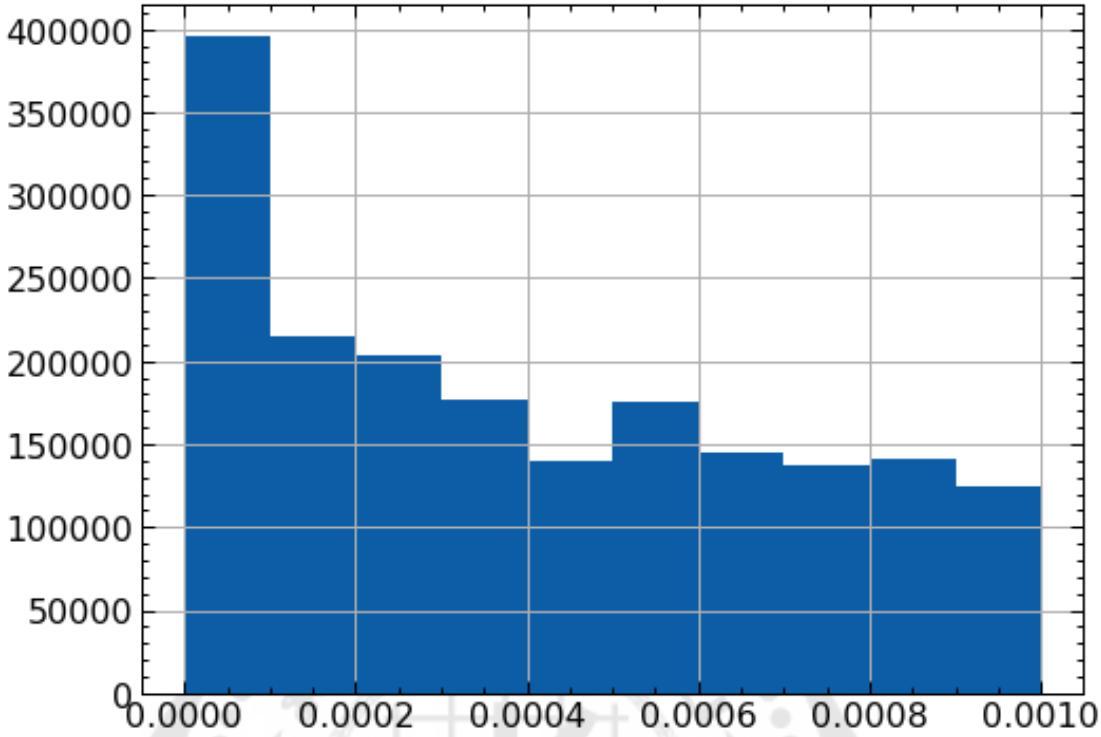


Figure 7 Distribution of input.

The distribution of the input amount is then explored. Since the amount of input and the amount of output should be identical, the distribution of input should represent the distribution of output as well. The distribution is skewed. Figure 7 shows that the majority of transactions are below 1 BTC.

Distribution of all wallet address's features is similar to the distribution of input amount where distribution either skews to the right or shares few values across instances. Distribution plots of 23 features are available on appendices represented by 20,000 data points.

Modeling

There are two anomaly detection algorithms trained to detect suspicious wallets and transactions on the Bitcoin network, which are Isolation Forest (IForest) and Histogram-based Outlier Score (HBOS). Both algorithms are trained with statistical features obtained from feature engineering which are already standardized on the same

scale. These algorithms are chosen based on performance and time complexity to handle a large dataset with high dimensions.

Both IForest and HBOS are similar to each other in terms of calculation. Each feature is calculated independently at the beginning. Then all output is recalculated altogether for the final outlier score. The fact that each dimension is calculated independently has reduced computation complexity tremendously. As a result, these algorithms are fast in training.

However, IForest and HBOS utilize different techniques to determine outliers. While IForest makes use of tree structure to separate anomalies, HBOS exercises histogram to represent outliers. IForest calculates the average path length after separation for a final outlier score. On the other hand, HBOS sums up derived values from histograms as a final score.

This study utilizes A Python Toolbox for Scalable Outlier Detection (PYOD) library. As the name suggests, the library focuses on outlier detection. Moreover, The author also (Zhao, Nasrullah, & Li, 2019) emphasizes the scalability of the library which makes it suitable for the study.

Model evaluation

This study is primarily focused on detecting outliers on the Bitcoin network where no label is available. Model evaluation is difficult in this study because it does not have the ground truth to validate against. Therefore, evaluation is held on the training dataset itself to evaluate its performance. There are 3 evaluations used in this study which are previously used in academic research: (1) Visualization Evaluation, (2) Dual Evaluation, and (3) Known-thieves Evaluation.

Visualization evaluation utilizes distance-based algorithms to calculate the distance of outliers from clusters' centroids. This evaluation assumes that outliers are further away from the centroids, therefore the greater amount of distance indicates better performance.

Visualization evaluation is first calculated by one hundred data points labeled as outliers with the highest distance ratio to its centroid, as shown in Equation (2), where *n* is the number of samples used for the calculation, *c* is the coordinate of the centroid, *t* is the coordinate of a detected outlier, *i* is the coordinate of an individual datapoint, and *C* represents clusters, while *d* is a distance function. The distance between each detected outlier is divided by the longest distance of its respective cluster to find its distance ratio. The output of the first equation is then re-scaled by Equation (3), which calculates the same amount of *n* records with the highest distance ratio to its centroid, in order to have the final score range between 0 and 1. Equation (4) shows how Equation (2) and Equation (3) is divided with expected values from each equation.

$$A_1 = \frac{\sum \underset{n}{max}\left(\frac{d(c,t)}{arg\,max(d(c,i \in C_t)}\right)}{n} \tag{2}$$

$$A_2 = \frac{\sum \underset{n}{max}\left(\frac{d(c,i)}{arg\,max(d(c,i \in C_i)}\right)}{n} \tag{3}$$

$$mVE = \frac{A_1}{A_2}\ ; 0 \leq mVE \leq 1\ ; A_1 > 0\ ;\ A_2 > 0 \tag{4}$$

Dual evaluation calculates on an assumption that suspicious transactions are created by suspicious users. Therefore, anomaly detection on wallet addresses and transactions should be intersected. Dual evaluation is determined by the number of outlier wallets that intersect outlier transactions. Dual evaluation is scaled by visualization evaluation where a higher score indicates a better relation in detection.

Dual evaluation is calculated from an average of two intersections. The first intersection is demonstrated by Equation (5), where *n* represents the number of outlier users. Equation (5) returns an intersection ratio of detected user outliers in top *n* detected transaction outliers. On the other hand, Equation (6) finds a ratio of detected

transaction outliers in top *m* detected user outliers. The average of both transactions is the final score for this evaluation as shown in Equation (7).

$$A_1 = \frac{|X_n \cap top\ X_n\ transaction\ outlier|}{|X_n|} \quad (5)$$

$$A_2 = \frac{|Y_m \cap top\ Y_m\ user\ outlier|}{|Y_m|} \quad (6)$$

$$mDE = \frac{A_1 + A_2}{2}\ ;\ mDE\ \in [0,1] \quad (7)$$

The known-thieve evaluation aims to observe real-world thieves. The Elliptic dataset, provided by (Weber et al., 2019), is a labeled dataset that has been exploited in previous studies (Phillips & Wilder, 2020) and (Vassallo, Vella, & Ellul, 2021). However, the Elliptic dataset does not include transaction hash, which is mandatory for gathering information. Hence, a Deanonymized Elliptic dataset (Benzik, 2000) is utilized instead.

The Deanonymized Elliptic dataset contains 202,804 transactions, of which 4,545 transactions are illicit. Thus, all wallet addresses that participated in those illicit transactions are assumed to be illicit as well. The total number of illicit wallet addresses extracted from the dataset is 14,266.

The known-thieve score is a percentage of a model's outlier prediction from the dataset as shown in Equation (8), where *a* is the number of predicted anomalous and *t* is the total number of thieves, which is 14,266 in this case. Known-thieves score is like previous evaluations where a higher score indicates better detection.

$$\frac{a}{t} \quad (8)$$

Three different evaluations serve different purposes for measurement. Visualization evaluation is used to measure how well a model can detect far instances.

Dual evaluation is applied to measure how consistent a model is between detecting suspicious transactions and suspicious wallet addresses. Known-thieves evaluation is exercised to verify the accuracy of a model. In summary, these evaluations measure different angles of an algorithm, therefore all of them are suitable for the study.

Preliminary

The anomaly detection on wallet addresses served as an early implementation showcase of the study. Both algorithms are implemented with default parameters and evaluated by visualization evaluation. Twenty thousand wallets are used for this preliminary. Both basic statistics and extra statistics are then applied to the original data (All feature distribution is in the appendices).

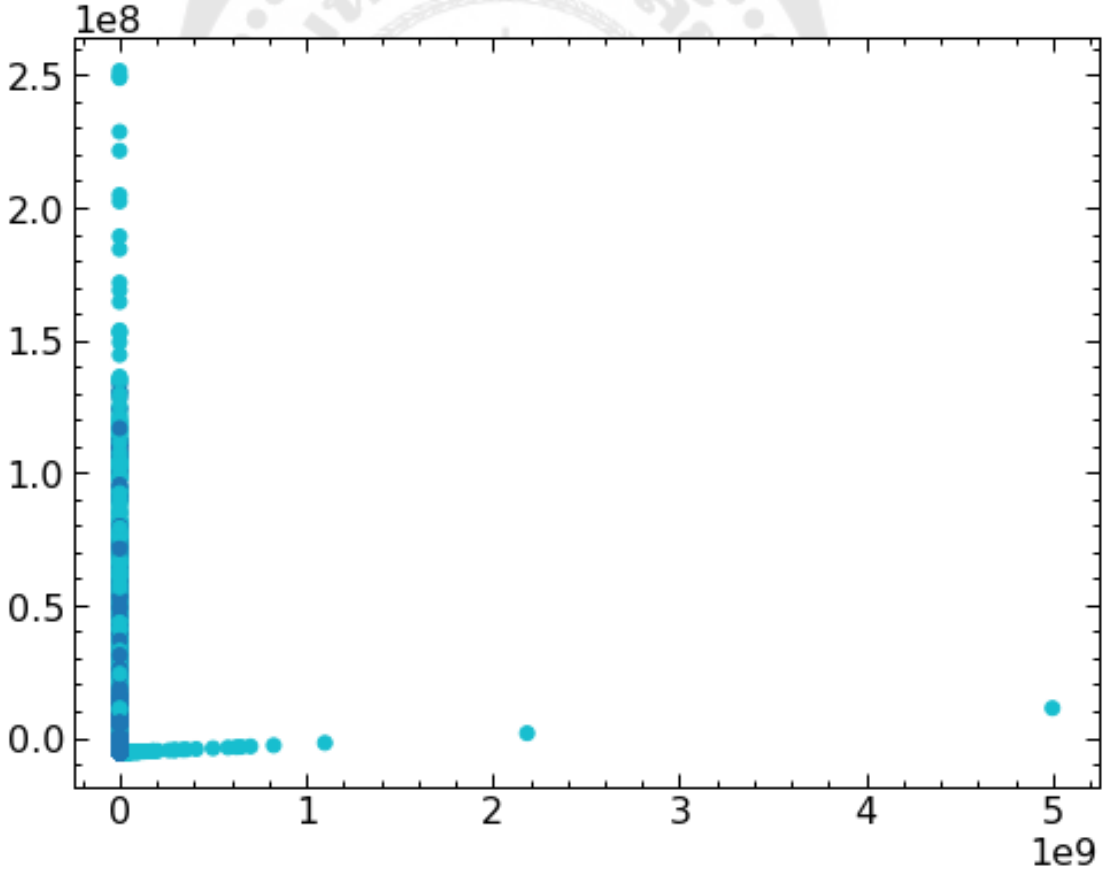


Figure 8 Predicted values of IForest.

Isolation Forest (IForest) and Histogram-based Outlier Score (HBOS) are trained on 23 features for the preliminary of the study. Then the PCA algorithm is utilized to

reduce the number of dimensions in favor of visualization. Figure 8 and Figure 9 show that IForest predicts more instances to be outliers than HBOS. HBOS seems to be biased on some features because it does not label any instance that has a value more than 0 on the x-axis.

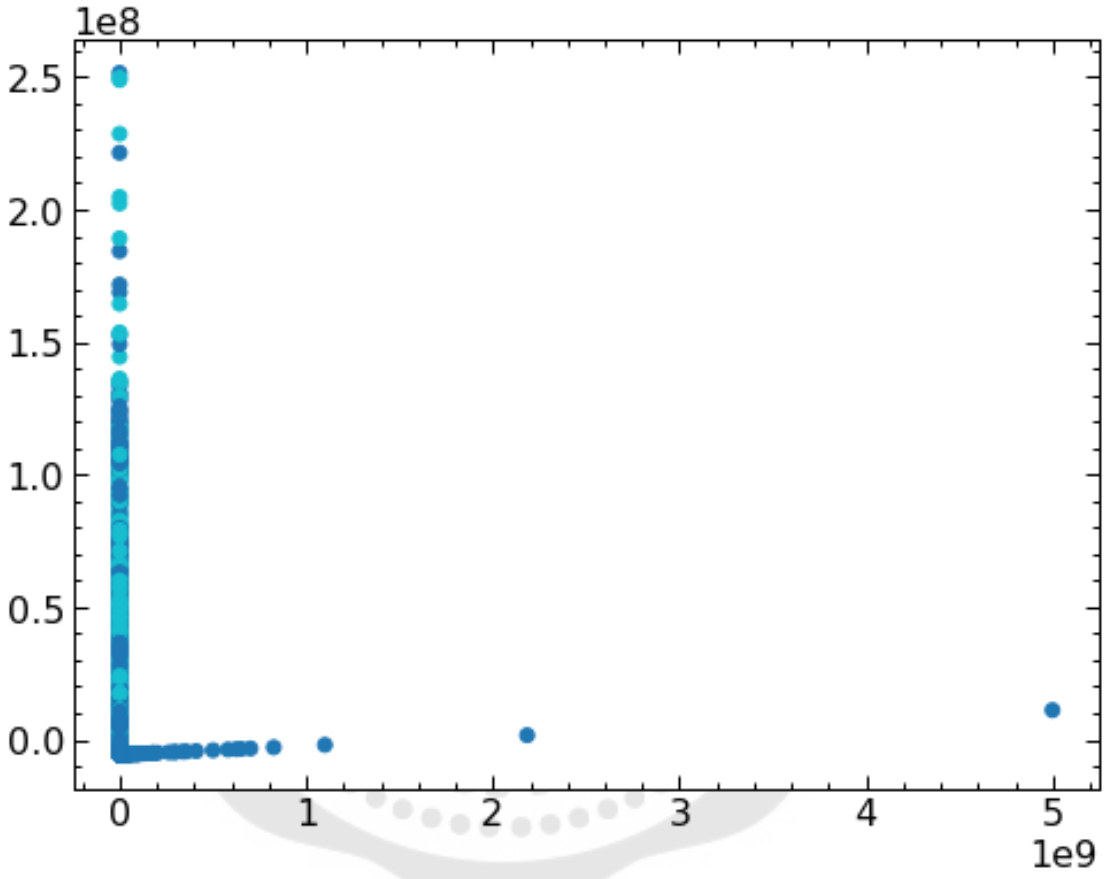


Figure 9 Predicted values of HBOS.

Visualization evaluation requires K-Means clustering algorithms in order to calculate the distance between outliers and respective centroids. The elbow method is explored for the optimal number of clusters. The number 4 is found to be an optimal number in this case, as shown in Figure 10. Hence, it is used to train a K-Means algorithm. Instances are then labeled to cluster and derive the distance from their respective centroid. Then, the longest distance of each centroid is determined for each cluster. The distance of each instance is divided by its longest distance in the centroid,

to derive a distance ratio. Finally, predicted anomalous records are sorted by distance in descending order to find the top 100 longest distances. The top 100 distance ratios are summed up as a final score for the visualization evaluation. The scores of IForest and HBOS without standardization (Equation (2)) are 0.358 and 0.218 respectively.

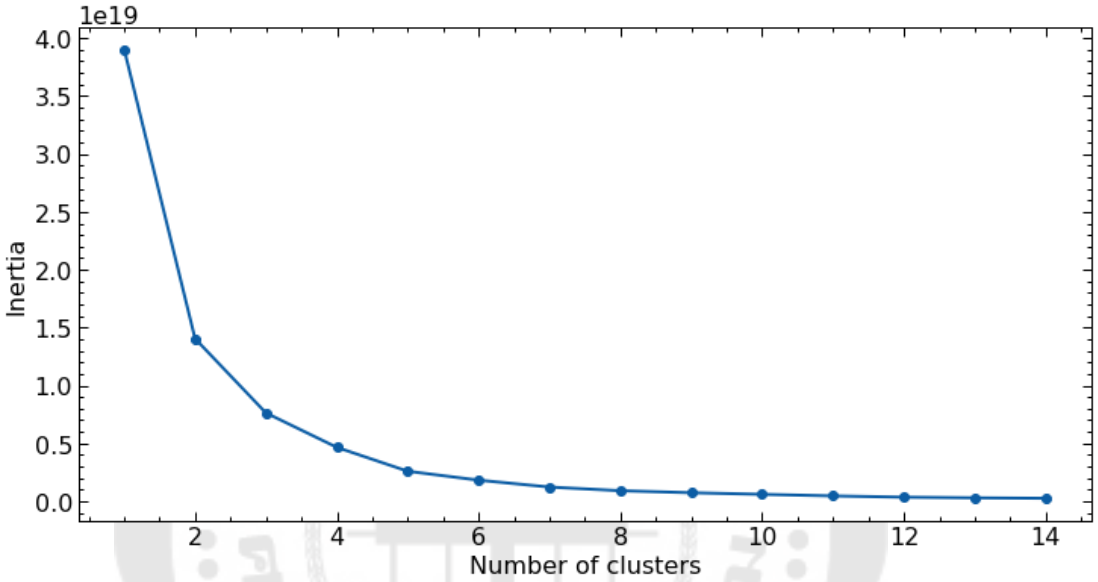


Figure 10 Finding an optimal number of k.

Principal Component Analysis (PCA) is applied to visualize the distribution of clusters. As shown in Figure 11, the distribution of clusters is highly imbalanced. There are two large groups close to the left of the figure, one having low values on the y-axis and another having high values. Two groups that have higher values on the x-axis are small, especially the group with the highest value on the x-axis.

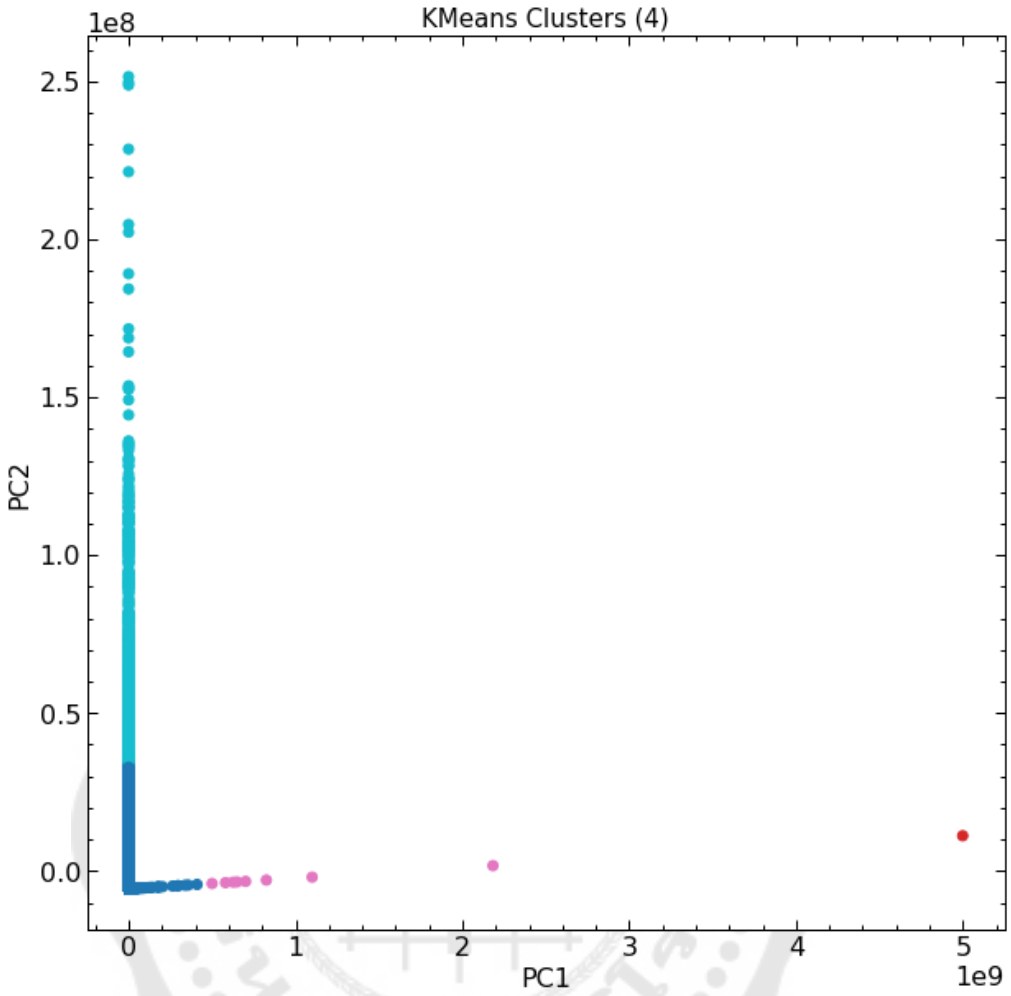


Figure 11 K-Means clustered on PCA.

CHAPTER 4

RESULT

Two algorithms are trained and evaluated. One is a Histogram-based Outlier Score (HBOS), and another is an Isolation Forest (IForest). Both models are trained under the same condition of contamination rate, which is restricted by real-world observation as mentioned in the literature review chapter. Therefore, the contamination parameter is fixed to 0.02 (2%). Moreover, a selected hyperparameter is explored in each model, i.e. the number of bins for HBOS and the number of estimators for IForest. The evaluations are discussed in the following order:

1. Evaluation of Histogram-based Outlier Score (HBOS)
2. Evaluation of Isolation Forest (IForest)
3. Comparison between HBOS and IForest
4. Feature importance
5. Comparison of feature importance

1. Evaluation of Histogram-based Outlier Score (HBOS)

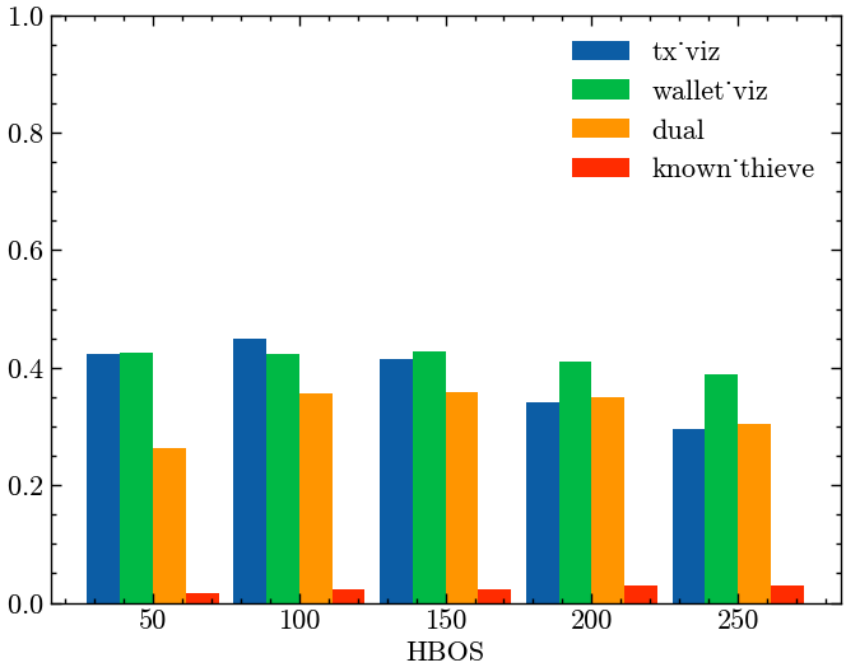


Figure 12 Evaluation of HBOS

According to Figure 12, HBOS has been trained 5 times with 50, 100, 150, 200 and 250 bins. The 50-bin model has a score of 0.423 for transaction visualization, 0.426 for wallet visualization, 0.263 for dual, and 0.017 for known-thieves evaluations. The 100-bin model scores 0.449 on transaction visualization, 0.423 on wallet visualization, 0.356 on dual, and 0.023 on known-thieves. The 150-bin has scores of 0.415 for visualization, 0.428 for wallet visualization, 0.358 for dual, and 0.023 for known-thieves. The 200-bin model has a score of 0.341 for transaction visualization, 0.410 for wallet visualization, 0.349 for dual, and 0.029 for known-thieves. The 250-bin model scores 0.296 for transaction visualization, 0.388 for wallet visualization, 0.305 for dual, and 0.029 for known-thieves.

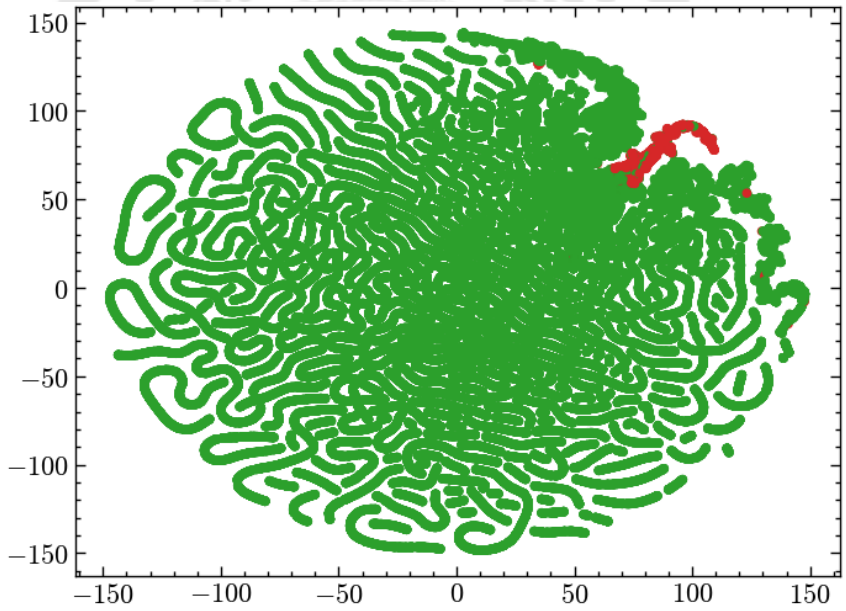


Figure 13 HBOS predictions on transactions.

HBOS predictions on transactions form a solid group of outliers, as shown in Figure 13. There are few predicted outliers at the right of Figure 13, however, it is crowded by inliers and difficult to notice. Moreover, there is an outlier instance on the top of the figure too. These indicate that data in the top right are on the edge of decision boundaries.

2. Evaluation of Isolation Forest (IForest)

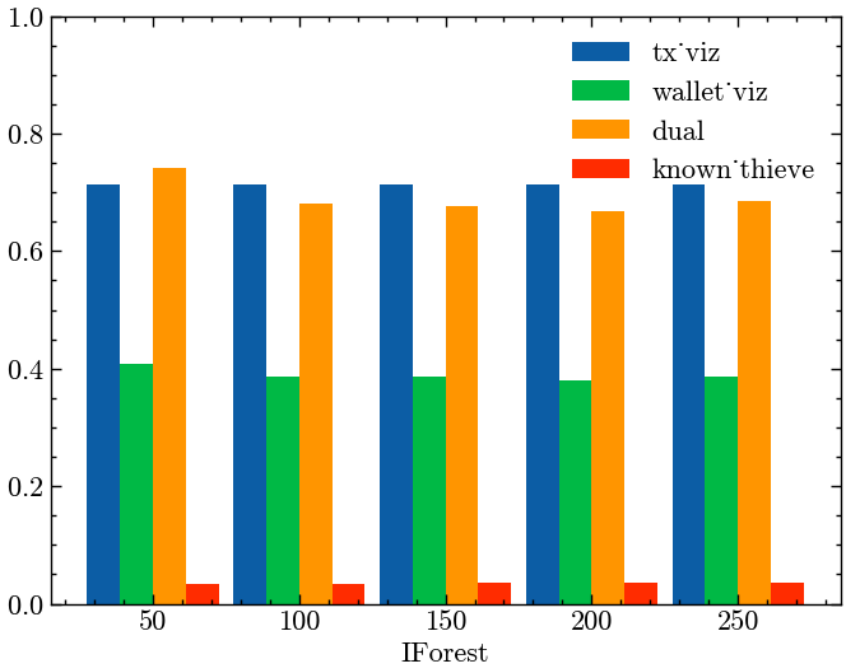


Figure 14 Evaluation of IForest.

According to Figure 14, IForest has been trained 5 times with 50, 100, 150, 200, and 250 estimators. The 50-estimator model has a score of 0.713 for transaction visualization, 0.408 for wallet visualization, 0.742 for dual, and 0.035 for known-thieves evaluations. The 100-estimator model scores 0.713 on transaction visualization, 0.387 on wallet visualization, 0.681 on dual, and 0.035 on known-thieves. The 150-estimator has a score of 0.713 for visualization, 0.387 for wallet visualization, 0.676 for dual, and 0.035 for known-thieves. The 200-estimator model has a score of 0.713 for transaction visualization, 0.379 for wallet visualization, 0.668 for dual, and 0.036 for known-thieves. The 250-estimator model scores 0.713 for transaction visualization, 0.386 for wallet visualization, 0.685 for dual, and 0.036 for known-thieves.

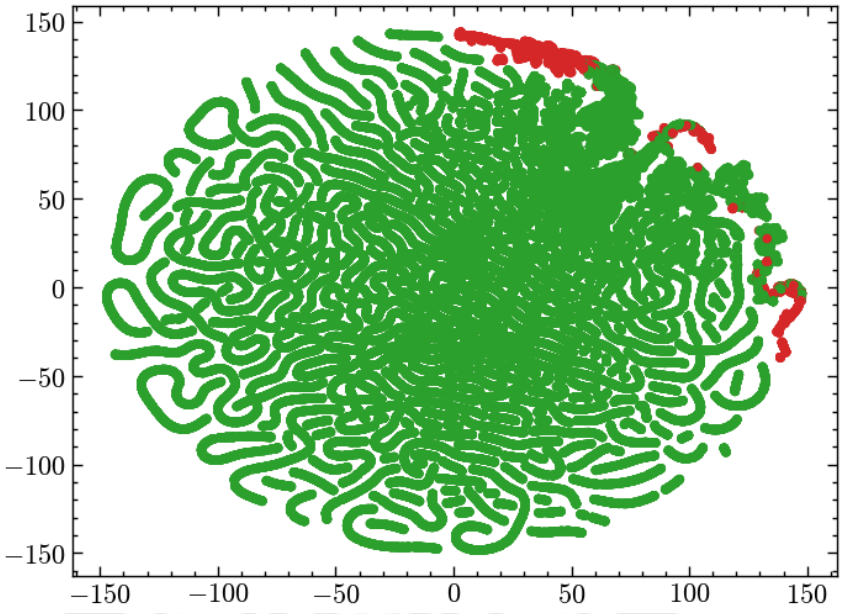


Figure 15 IForest predictions on transaction.

Figure 15 demonstrates how IForest predicted outliers. Three groups are being assumed outliers by IForest. The group on top of the figure is the largest. Also, there are smaller groups on the top right and the right of the figure.

3. Comparison between HBOS and IForest

According to Figure 16, the convention of the model name is based on the algorithm, contamination rate, and hyperparameter, for example, HBOS-0.02-50 represents an HBOS algorithm with a contamination rate set to 0.02 and the number of bins is 50. Similarly, IF-0.02-100 represents the IForest algorithm with a 0.02 contamination rate along with 100 estimators.

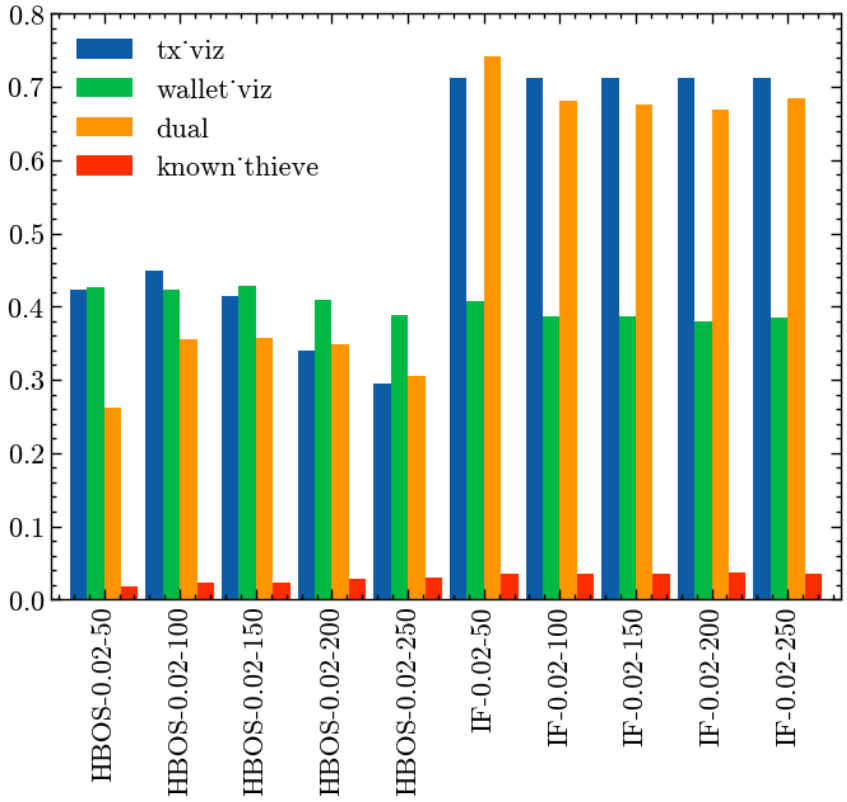


Figure 16 Model Comparison

HBOSs are displayed from the left to IForest on the right with an increasing number of hyperparameters. As observed from Figure 16, the bars on the right are taller than the bars on the left. This implies that IForest scores are higher in most measurements.

Many evaluations are being applied to the models, which increases the complexity of selecting the best-fit model. The visualization evaluation is based on the distance of the predicted data point to the centroid of the cluster. This implies that the higher the score is the predicted data is further from the centroid. On the other hand, the Dual evaluation is based on the duality between suspicious transactions and wallets. It makes the model more meaningful as it aligns with the hypothesis that suspicious wallets are involved in suspicious transactions. Therefore, models with higher dual scores are more convincing.

HBOS and Isolation Forest performances are on a similar scale for wallet detection. However, when considering transaction and dual evaluations, Isolation Forest

outperformed. Moreover, Isolation Forest detects more thieves in the dataset than HBOS.

4. Feature importance

Feature importance is a technique identifying which features play a significant role in a model's decision. IForest can determine important features based on impurity. In contrast, HBOS requires an extra step to determine its importance. SHAP (Lundberg & Lee, 2017) is utilized for HBOS in this regard.

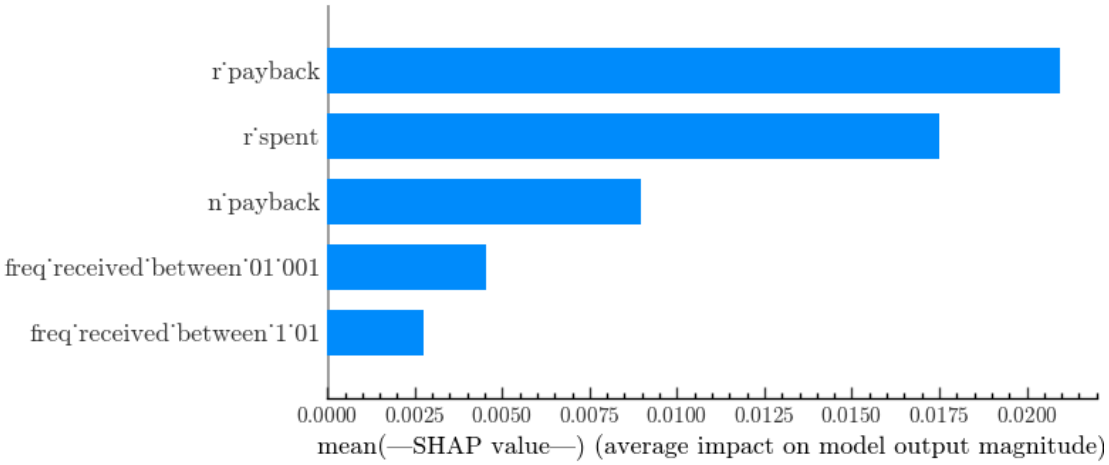


Figure 17 HBOS feature important on wallet task

According to Figure 17, features that highly impact HBOS decisions on the wallet task are r_payback, r_spent, n_payback, freq_received_between_01_001, and freq_received_between_1_01. The mean SHAP values are 0.021, 0.017, 0.009, 0.004, and 0.003, respectively.

On the transaction task, HBOS considers input_count, time, and output_count to be the most important features, as shown in Figure 18. The input_count is the most important feature which is 3 times higher in magnitude compared to time and ouput_count. Time and output_count considerably equally affect HBOS decisions. The mean SHAP values for input_count, time, and ouput_count are 0.037, 0.010, and 0.010, respectively.

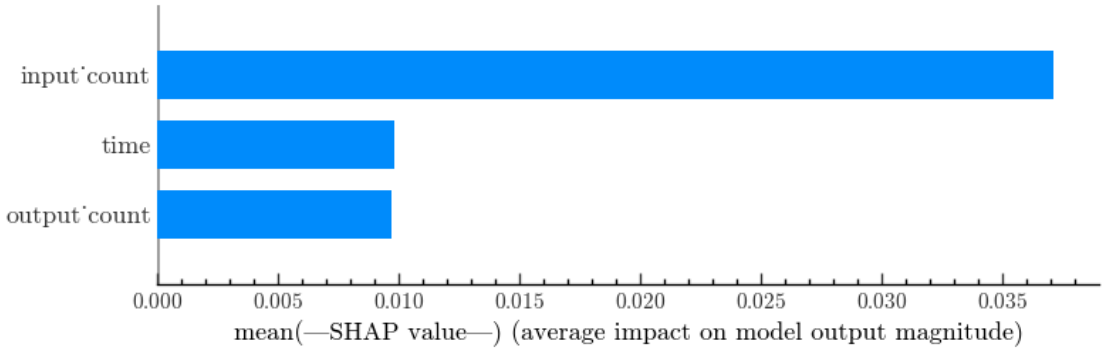


Figure 18 HBOS feature important on transaction task

While SHAP operates on permutations, the tree-based algorithms have built-in impurity-based feature importances. Another difference is that SHAP is calculated from the sample, while tree-based impurity is calculated from the training dataset. Hence, mean SHAP values cannot be referred to as a contribution because it is not calculated from a whole dataset. That is why the mean SHAP value does not add up to 1.

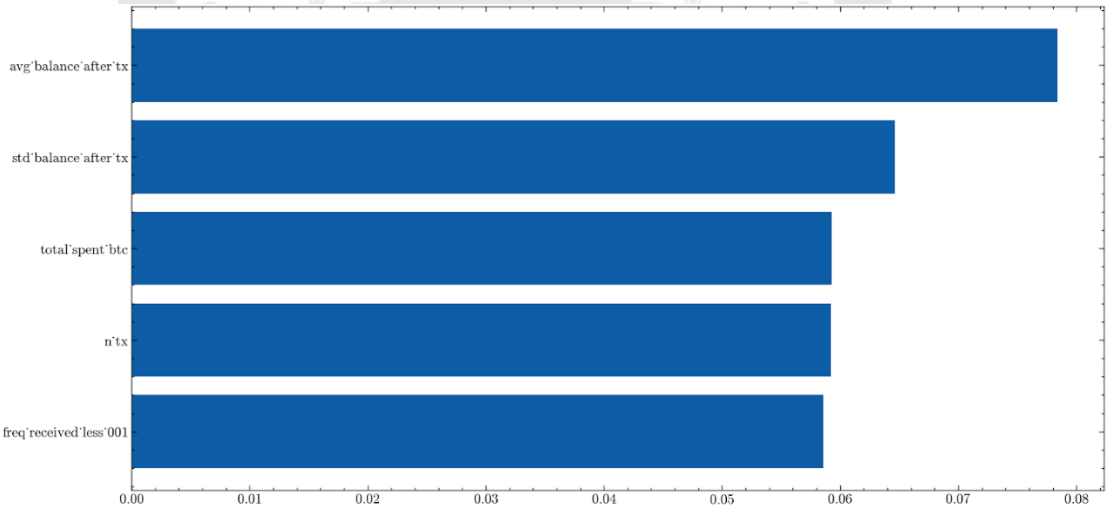


Figure 19 The IForest feature important for wallet task

According to Figure 19, avg_balance_after_tx, std_balance_after_tx, total_spent_btc, n_tx, and freq_received_less_001 are the top 5 features that affect IForest decision on the wallet task, with feature important scores 0.078, 0.065, 0.059, 0.059, and 0.059, accordingly.

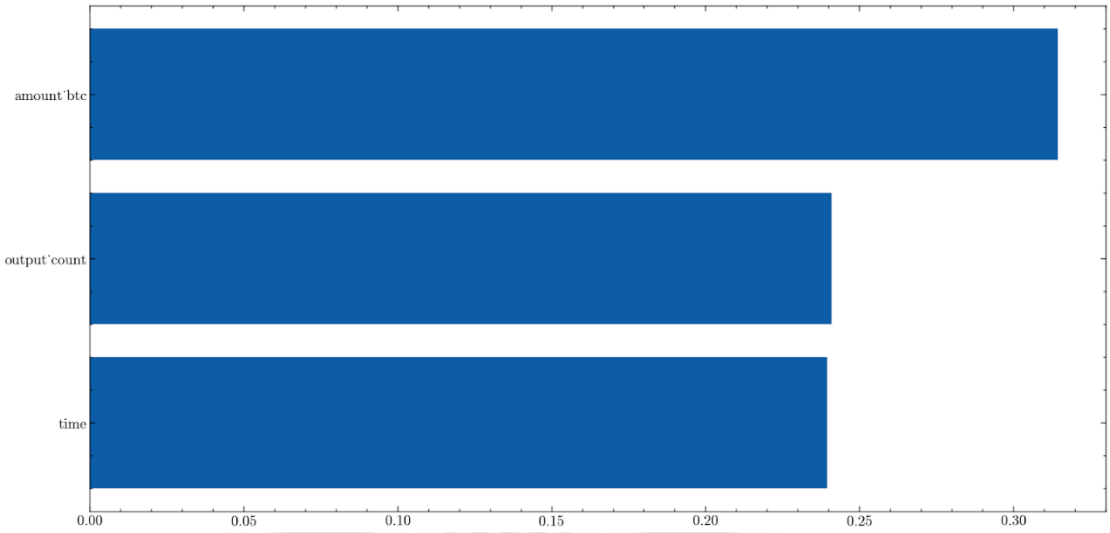


Figure 20 The IForest feature important in transaction task

On the transaction task, amount_btc, output_count, and time are the top 3 features that contributed to the IForest decision, as shown in Figure 20. The important feature scores are 0.314, 0.241, and 0.240, respectively.

5. **Comparison of feature importance**

This section aims to compare features that highly impact algorithms' decisions and discuss them based on human intuition. It is important to note that both algorithms applied different techniques for calculating feature importance. Hence, the score of both algorithms cannot be compared. They are on a different scale. Still, the order in which feature importances are being presented is valuable and beneficial for discussion.

On the wallet task, HBOS is influenced heavily by basic statistics. There are 4 basic statistics features in the top 5 feature importances, which are the ratio of payback and the ratio of spent, the frequency of receiving between 0.01 to 0.1 BTC, and the frequency of receiving between 0.1 to 1 BTC. Only 1 extra statistic that appeared in third place is the number of paybacks.

The ratio of paybacks can promote a couple of characteristics of the wallet. A high number is expected for a normal wallet, as it is the nature of transactions to receive changes when spending. In contrast, a very low number can indicate that a wallet

usually spends all of its money in a transaction, which is abnormal in nature. The ratio of spending can infer a few characteristics. A very low number can indicate a saving wallet where the number of receiving transactions is greater than the number of spending. In contrast, a very high number can suggest an active wallet, as it rarely receives but spends frequently. This scenario is obvious in a fiat financial system, where an account receives transactions monthly, and spending daily. Another interesting ratio is 0.5, where receiving and spending transactions occur on the same frequency. This number can happen to a normal wallet, but likely it is not. Wallets participating in the layering stage of money laundering are likely to end up with 0.5 on the ratio of spending. The number of paybacks is similar to the ratio of paybacks. While the ratio indicates how often a wallet receives transactions compared to its own transactions, the number does not recognize its self-compared. Hence its comparison is made globally to the dataset. A low number can tell that a wallet rarely receives paybacks, but does not include the number of transactions. Therefore, rarely-spent wallets and frequently-spent wallets may share the same number if the frequently-spent wallets rarely receive changes.

Combining the top 3 features creates an endless number of possible characteristics of a wallet. There are a few types of wallets that can be derived from these 3 features. For example, a single-used criminal wallet may comprise characteristics of 0 or 1 for the ratio of payback, 0.5 for the ratio of spending, and 0 or 1 for the number of payback transactions. A multiple-used layering wallet may comprise a low ratio of paybacks, a 0.5 ratio of spending, and a low number of receiving payback transactions.

On the other hand, IForest is impacted by a combination of extra statistics and raw features. The top 2 features are the average balance after transactions and the standard deviation after transactions, which belong to extra statistics. The third and fourth places are from raw features, which are the total spent in BTC and the number of transactions. There is only one basic statistic that made it to the top 5, which is the frequency of receiving less than 0.001 BTC. Surprisingly, there is no overlapping feature in the top 5 features for the wallet task.

The average balance after transactions can assume a couple of characteristics of a wallet. A very high average balance can be considered a rich wallet. Also true for the opposite, a very low average balance can be assumed for a poor or seldom-used wallet. In addition, a very low average balance can be a characteristic of a layering wallet. The standard deviation suggests the stability of a wallet for each transaction. A high number is desired for a normal wallet as the remaining balance goes high and low from time to time. While, a low number indicates a stable balance of wallets, which can be suspected of abnormality. The total spent BTC introduced similar characteristics to the average balance feature, where a high number indicates wealthy wallets and vice versa.

The top 3 features of IForest can combine to form traits of rich, poor, as well as anomalies. High values in average balance, standard deviation of balance, and total spent BTC indicate a wealthy wallet. On the other side, a poor wallet would have a low value in average balance, a medium-to-high value in the standard deviation of balance, and a low value in total BTC spent. Anomalous wallets can inferred from a very low stand deviation of balance as it indicates consistency of remaining balance, especially, with an average balance close to zero would make it even more suspect. Total spent BTC can help to identify single-used and multiple-used wallets. A single-used layering wallet should have a lower value compared to a multiple-used layering wallet.

On the transaction task, HBOS is affected by the number of inputs, time, and the number of outputs. While IForest is powered by the amount of BTC, time, and the number of outputs. Due to a very low number of features, there are overlapping features in the second and the third places, which are time and the number of outputs. Yet, the arrangement is different for each algorithm.

HBOS and IForest have prioritized a different feature for their most important one. The number of inputs is the most important feature for HBOS, while the total amount of BTC is the most significant feature for IForest. However, both features are related to spending in transactions. HBOS considers the number of coins used in a transaction, whereas IForest considers the total amount. Consider the scenario of two transactions

transferring 1 BTC. One transaction used 1 coin, another used 10 coins. HBOS gives priority to the number of coins used in a transaction, hence it may label one transaction as normal and another as an anomaly. In contrast, IForest is likely to label these two transactions with the same label. The time feature indicates the hour that the transaction is being made. Though it seems less important because the Bitcoin network is used globally, it has peak hours. The number of outputs can be considered as the number of receivers; senders are included when change is applicable.

There are many scenarios in the Bitcoin transactions. For example, many coins are summed to 0.1 BTC being transferred to a single receiver with a change to the sender, or a single coin of 1 BTC is being divided and delivered to multiple wallets with no change to the sender. In the context of transactions, it is difficult to determine whether a big or small transaction is a rare event. However, it is notable that both algorithms place a higher priority on the sender than the receiver.

All in all, both algorithms have a different set of feature importances. Both feature sets present possibilities to detect different types of anomalous wallets and transactions that align with human intuition. While validating different types of anomalous wallets required further study, this study yielded an interesting understanding of anomalous wallets. However, the transaction task is not as compelling as the wallet task due to its low dimensions making it difficult to interpret.

CHAPTER 5
SUMMARY DISCUSSION AND SUGGESTION

This study has aimed to detect illicit activities on the Bitcoin network by utilizing raw data from the network with statistical features and unsupervised learning methods. Two algorithms are evaluated, compared, and summarized as follows:

1. Summary
2. Discussion
3. Suggestion

1. Summary

Bitcoin, like other financial systems, is inevitable to be fraudulent. The prevention of fraud must be actively studied because fraud itself evolves over time making it harder to prevent. Therefore, this study attempts to improve fraudulent detection techniques, using the Bitcoin network as a case study, by implementing unsupervised learning algorithms.

This study collected data from the Bitcoin network via API from a blockchain explorer website, called blockchain.info, to train artificial intelligence models. Histogram-based Outlier Score (HBOS) and Isolation Forest (IForest) are selected as guidelines. Two statistical techniques are applied, which are categorized as basic and advanced. Three evaluations are being undertaken for model comparison, which are visualization, dual, and known-thieves.

2. Discussion

The best performer of the Histogram-based Outlier Score (HBOS) is a model with 100-bin. Even though the 50-bin model has similar scores on transaction and wallet visualizations, its dual score is lower than the 100-bin. Also, an increasing number of bins has shown a decline in scores, as evident in Figure 12. Additionally, the 150-bin

has slightly better wallet visualization and dual scores, but a moderate drop in transaction visualization. Hence, it is the 2nd best performer for this model.

Similarly, the best performer for Isolation Forest (IForest) is a 100-estimator model. The 50-estimator has higher scores in wallet visualization and dual evaluations, yet it is not chosen to be the best performer for IForest. Since the 50 and 100 estimator models shared the same score on known-thieves, it is suspected that the 50-estimator model is overfitting. Furthermore, the scores of 100 and 250 estimators are stable across every evaluation. In summary, the 100-estimator is the best performer because it utilizes the optimal number of estimators.

This study has found that Isolation Forest (IForest) outperforms Histogram-based Outlier Score (HBOS). IForest scores are almost 2 times higher than HBOS on transaction visualization, with scores of 0.713 and 0.449 respectively. On the other hand, HBOS has a slightly higher score on wallet visualization. The difference is subtle between both algorithms, HBOS's score is 0.423, while IForest's is 0.387. However, HBOS is unable to compete on dual evaluation, with only 0.356, while IForest scores at 0.681. In addition, IForest performs slightly better on known-thieves evaluations, with 0.035 against 0.023 respectively. The summary of the score is elaborated in Table 6.

Table 5 HBOS and IForest Scores

Algorithm	Transaction Visualization	Wallet Visualization	Dual	Known-thieves
HBOS	0.449	0.423	0.356	0.023
IForest	0.713	0.387	0.681	0.035

While evaluations are in place to determine the best performer for anomaly detection on the Bitcoin network, feature importances help amplify the decision made from each model. HBOS considers the ratio of payback to be to most important indicator. A high ratio of payback means that a person does not spend all the value of a coin he/she received. As a result, he/she gets a payback as a change in return. In

contrast, a low ratio of payback refers to a wallet that usually spends the same amount he/she received. However, it is undeterminable whether HBOS considers a low or a high ratio of payback to be a red flag. Human intuitive would consider a low ratio of payback to be a suspicious wallet as it is aligned with the second stage of money laundering, which is a layering stage. The next important feature that HBOS considered is the ratio of spending transactions. This feature is also aligned with the layering stage, where a high ratio indicates spending more often than receiving, and vice versa. However, humans presume a 0.5 ratio to be a suspicious account, where receiving and spending occur symmetrically in pairs. The third important feature of HBOS is the number of paybacks. This is an interesting feature as it can be a complementary feature to the ratio of payback or cause confusion in the model. Because the number of paybacks can be considered a global comparison, while the ratio of paybacks can be considered a local comparison. For example, a newly created wallet may have a high ratio of payback, but a low number of paybacks as it only made a few transactions. All in all, HBOS feature importances are aligned with human intuition.

On the other hand, the average balance in BTC after each transaction contributes the most to the IForest decision on the wallet task. This can identify a few groups of wallets, such as rich wallets where the average balance is high. Also, low average balance wallets are suspicious of participating in the layering stage. The next important feature that IForest considered is the standard deviation of balance after each transaction which is considered to be a complementary feature to the average balance in BTC after each transaction as it helps affirm how consistent the wallet balance after each transaction is. From a human perspective, a low average balance along with low standard deviation wallets seem to be solid criteria for suspicious wallets. The third important feature is the total spend amount in BTC. This again may find different types of wallets, such as rich wallets. However, it can be a good indicator for wallets that participate in layering multiple times. To sum up, IForest feature importances are intuitive to human understanding, but its detection may include rich wallets.

On the transaction task, HBOS seems to rely heavily on the number of inputs. While IForest chose to rely on the amount of BTC in the transaction. In this regard, the IForest feature is more intuitive. The number of inputs reflects the number of coins used in the transaction which seems to be less meaningful compared to the amount in the transaction. For example, HBOS notices the difference in 1 BTC with 1 coin and 1 BTC with 100 coins, while IForest does not. In reality, these features slightly correlate to each other. Hence, there are overlapping predictions in the transaction task, as evident in Figure 13 and Figure 15.

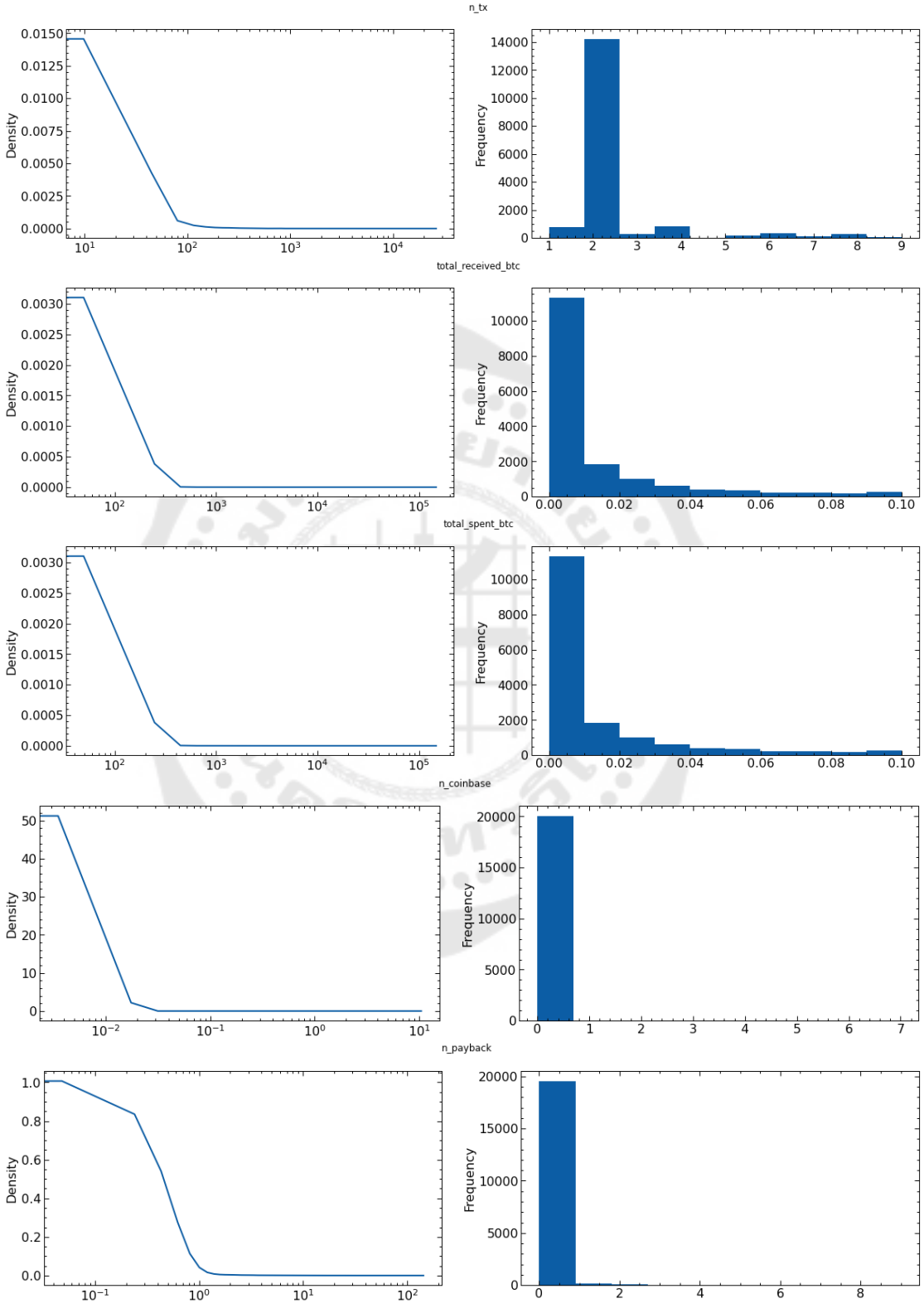
Both algorithms detect anomalous wallets by allocating their interest to a different set of features. HBOS places its attention heavily on basic statistics, while IForest distributes its focus on extra statistics and raw features. On the transaction task, both algorithms shared a similar group of features with a different feature in the leading position. IForest places the BTC amount as its greatest priority, but HBOS ignores it.

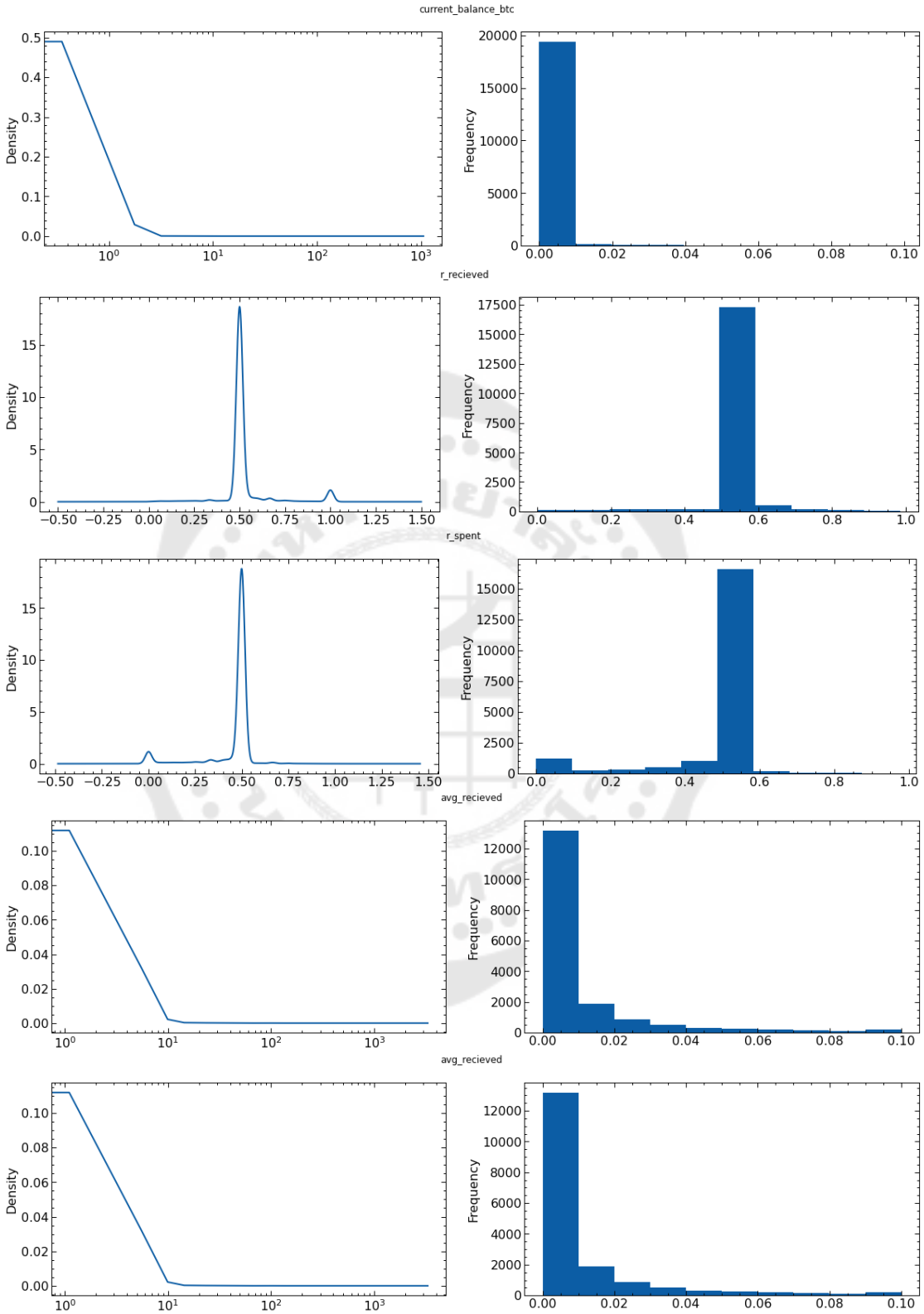
3. Suggestion

1. Data used in this study is only extracted at a single period, hence utilizing multiple periods may increase model performance.
2. This study utilized only raw data from the Bitcoin network. Additional data, such as trading data may provide additional useful features for algorithms. For example, converting BTC to USD to overcome price fluctuation.
3. This study focuses on low-complexity algorithms and utilizes only two algorithms, which are Histogram-based Outlier Score and Isolation Forest. Alternatively, higher complexity algorithms may perform better.
4. GPU computation could improve the efficiency of processing the full dataset.
5. Also, further evaluation could be conducted using different datasets or scenarios to validate the robustness of the proposed algorithms.

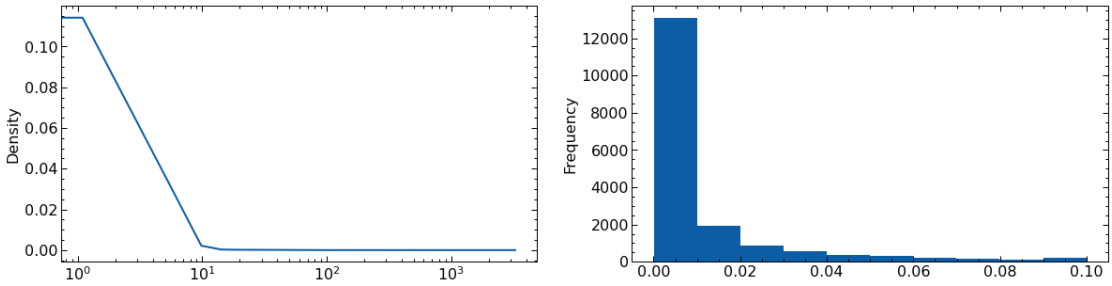
APPENDIX

The distribution of features in the dataset

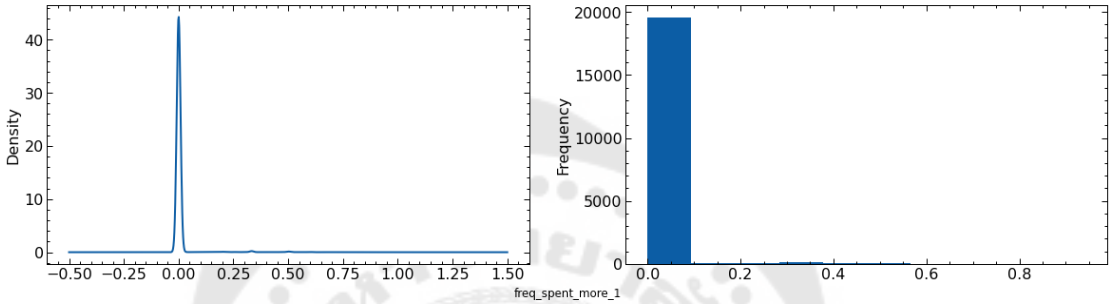



current_balance_btc
Density
Frequency
r_recieved
Density
Frequency
r_spent
Density
Frequency
avg_recieved
Density
Frequency
avg_recieved
Density
Frequency

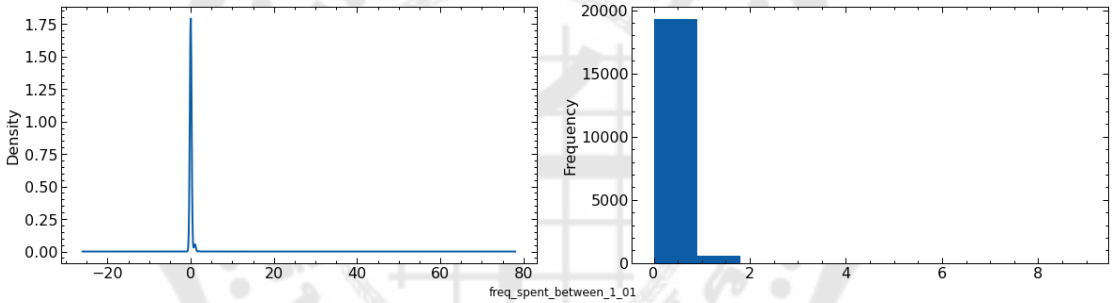
avg_spent



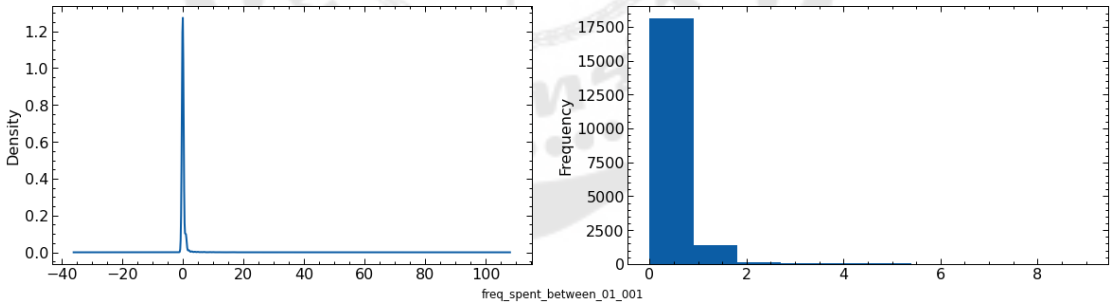
r_payback



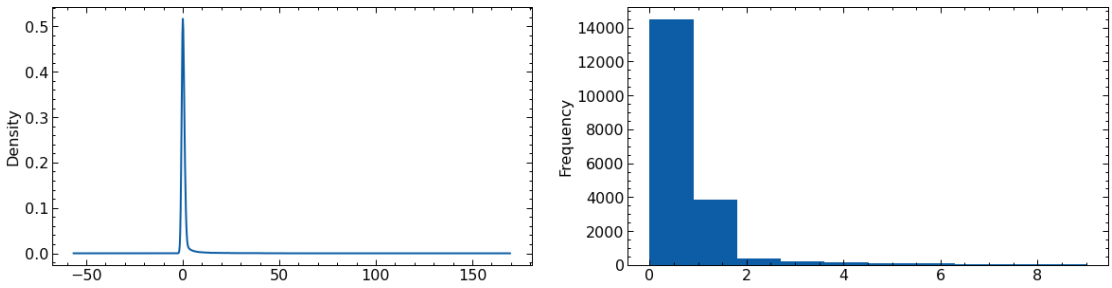
freq_spent_more_1

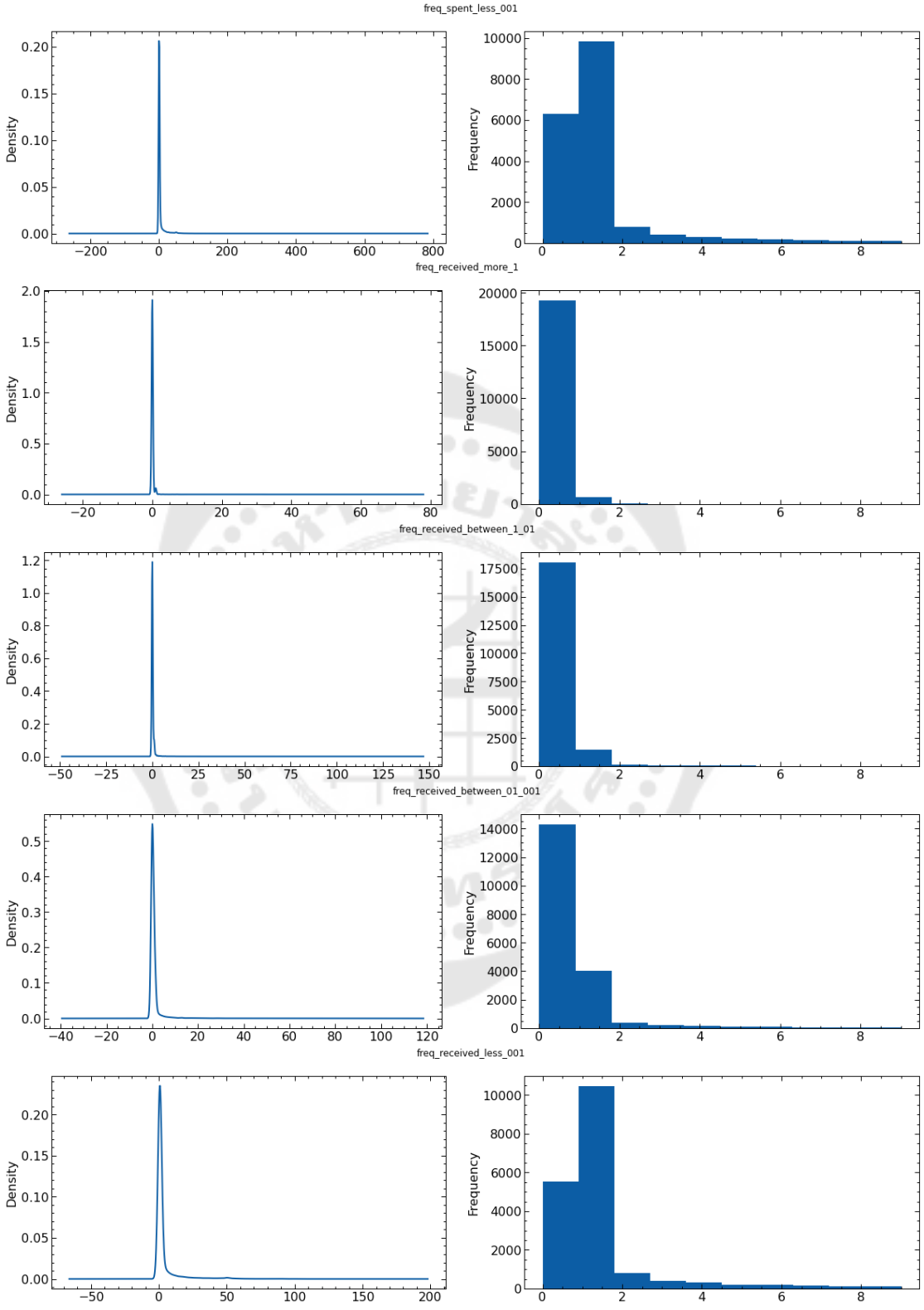


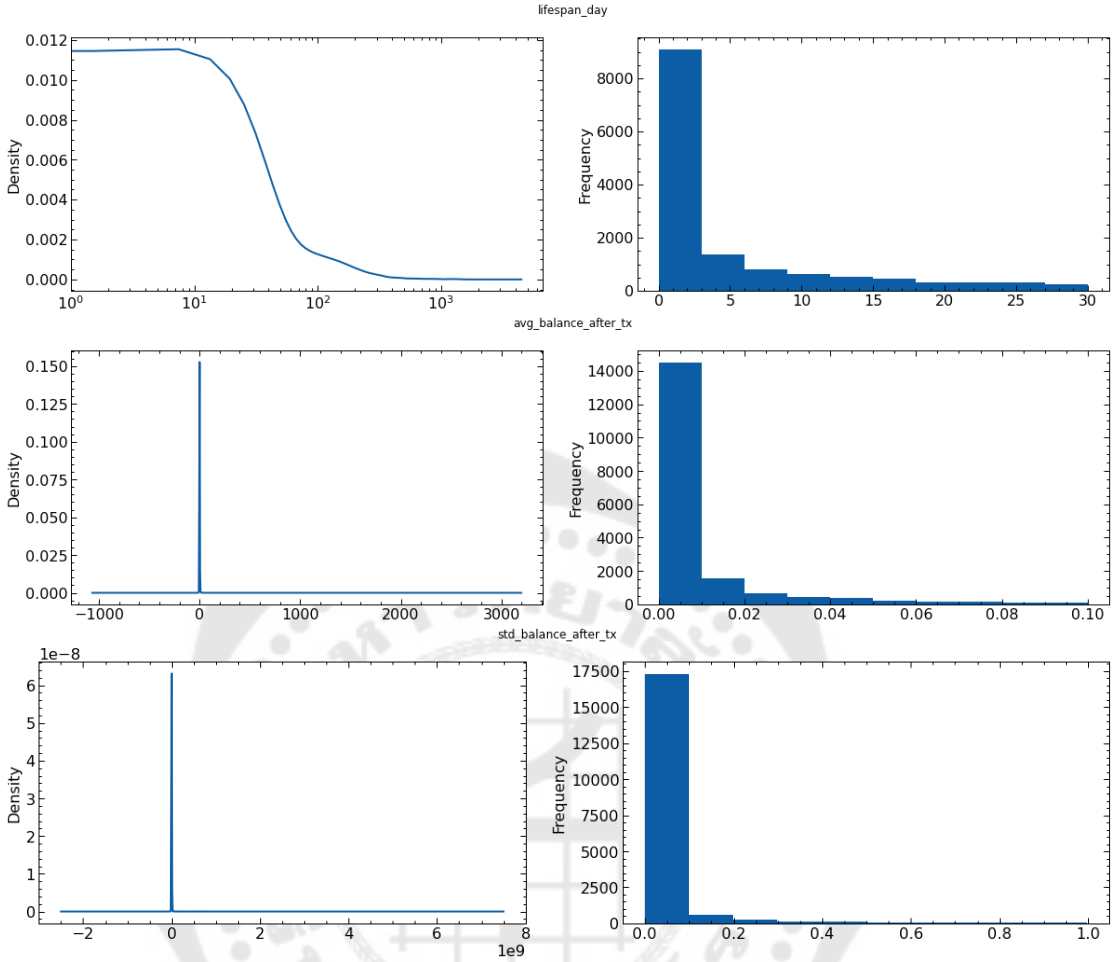
freq_spent_between_1_01



freq_spent_between_01_001




freq_spent_less_001
Density
Frequency
freq_received_more_1
Density
Frequency
freq_received_between_1_01
Density
Frequency
freq_received_between_01_001
Density
Frequency
freq_received_less_001
Density
Frequency


lifespan_day
Density
Frequency
avg_balance_after_tx
Density
Frequency
std_balance_after_tx
1e−8
Density
1e9
Frequency

REFERENCES

Awad, M., & Khanna, R. (2015). Machine Learning. In M. Awad & R. Khanna (Eds.), *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers* (pp. 1-18). Berkeley, CA: Apress.

Baek, H., Oh, J., Kim, C. Y., & Lee, K. (2019, 2-5 July 2019). *A Model for Detecting Cryptocurrency Transactions with Discernible Purpose.* Paper presented at the 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN).

Becirovic, S., Zunic, E., & Donko, D. (2020). *A Case Study of Cluster-based and Histogram-based Multivariate Anomaly Detection Approach in General Ledgers.* Paper presented at the 2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH).

Benzik. (2000). *Deanonymized 99.5 pct of Elliptic transactions*. Retrieved from: https://www.kaggle.com/datasets/alexbenzik/deanonymized-995-pct-of-elliptic-transactions

BigQuery. (2019). *Bitcoin Blockchain Historical Data*. Retrieved from: https://www.kaggle.com/datasets/bigquery/bitcoin-blockchain

Bivin S. Nair, R. K. V. (2018). Anonymity Analysis of Bitcoin Transactions Using Unsupervised Machine Learning. *International Journal of Research and Scientific Innovation (IJRSI), 5*(7), 126-129. Retrieved from https://www.rsisinternational.org/journals/ijrsi/digital-library/volume-5-issue-7/126-129.pdf

Blockchain.com. Blockchain Developer APIs. Retrieved from https://www.blockchain.com/explorer/api/blockchain_api

Goldstein, M., & Dengel, A. (2012). Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track*, 59-63.

Lin, Y.-J., Wu, P.-W., Hsu, C.-H., Tu, I.-P., & Liao, S.-w. (2019). *An evaluation of bitcoin*

address classification based on transaction history summarization. Paper presented at the 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC).

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). *Isolation forest.* Paper presented at the 2008 eighth ieee international conference on data mining.

Lundberg, S. M., & Lee, S.-I. (2017). *A unified approach to interpreting model predictions*. Paper presented at the Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA.

Mabunda, S. (2018, 6-7 Aug. 2018). *Cryptocurrency: The New Face of Cyber Money Laundering*. Paper presented at the 2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD).

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, 21260.

Pham, T., & Lee, S. (2016). Anomaly detection in bitcoin network using unsupervised learning methods. *arXiv preprint arXiv:1611.03941*.

Phillips, R., & Wilder, H. (2020, 2-6 May 2020). *Tracing Cryptocurrency Scams: Clustering Replicated Advance-Fee and Phishing Websites.* Paper presented at the 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC).

Toyoda, K., Ohtsuki, T., & Mathiopoulos, P. T. (2018). *Multi-class bitcoin-enabled service identification based on transaction history summarization.* Paper presented at the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData).

Vassallo, D., Vella, V., & Ellul, J. (2021). Application of Gradient Boosting Algorithms for Anti-money Laundering in Cryptocurrencies. *SN Computer Science, 2*(3), 143. doi:10.1007/s42979-021-00558-z

Weber, M., Domeniconi, G., Chen, J., Weidele, D. K. I., Bellei, C., Robinson, T., & Leiserson, C. E. (2019). Anti-Money Laundering in Bitcoin: Experimenting with Graph Convolutional Networks for Financial Forensics. arXiv:1908.02591.

doi:10.48550/arXiv.1908.02591

Zhao, Y., Nasrullah, Z., & Li, Z. (2019). Pyod: A python toolbox for scalable outlier detection. *arXiv preprint arXiv:1901.01588*.

VITA