University of Dublin

TRINITY COLLEGE

MCS DISSERTATION

Bitcoin Blockchain Fraud Detection with Unsupervised Learning

Author:

Ki Pan Yim

Supervisor:

Donal O’Mahony

A dissertation submitted in fulfillment of the requirements for the degree:

***Integrated Masters In Computer Science** at the*

School Of Computer Science & Statistics

Trinity Term (May 2018)

Declaration Of Authorship

I, Ki Pan Yim, declare that the following dissertation, except where otherwise stated, is entirely my own work; that it has not previously been submitted as an exercise for a degree, either in Trinity College Dublin, or in any other University; and that the library may lend or copy it or any part thereof on request.

Signature :

____

Date:

____

Summary

Currency is the cornerstone of human civilization. The form of currency has changed over time, transitioning from metal, silver and gold in the early stage to paper-money today, and possibly, in the near future, digital currency. The concept of digital cash was first introduced by David Chaum in 1983, in his research paper. In 2008, a revolutionary digital payment system named Bitcoin was released by Satoshi Nakamoto. Bitcoin has a number features that significantly weakened government’s control over the currency system. This brought it popularity, against the background of the financial crisis in 2008 where government credit was being questioned. As digital payment systems such as Bitcoin starts to dominate the world economy, Bitcoin financial fraud is becoming a critical issue to our society.

This dissertation aims to construct a Bitcoin fraud detection system that uses machine learning algorithms to detect Bitcoin fraud. The system extracts meaningful features from Bitcoin ledger and fits the feature data to the fraud detection models. Unsupervised learning algorithms are implemented to construct the fraud detection models. To address this, the selection of promising features that are relevant or correlated to Bitcoin fraud is important. A number of recent Bitcoin fraud are analyzed and studied. Based on their characteritics, a number of promising features are extract and selected for the model. These features are combined with unsupervised learning algoriths to construct the fraud detection models. The unsupervised learning algorithms we have implemented are K-means clustering and one-class SVM. K-means clustering is a method used for clustering analysis and one-class SVM is an algorithm for anomaly detection. The models have successfully recognized some patterns for the identification of Bitcoin fraud and have successfully identified a previously unobserved fraud.

TRINITY COLLEGE DUBLIN

Abstract

Computer Science Department School Of Computer Science & Statistics

Integrated Masters In Computer Science

Bitcoin Fraud Detection with Unsupervised Learning

by Ki Pan Yim

Currency is the cornerstone of human civilization. The form of currency has changed over time, transitioning from metal, silver and gold in the early stage to paper-money today, and possibly, in the near future, digital currency. The concept of digital cash was first introduced by David Chaum in 1983, in his research paper. In 2008, a revolutionary digital payment system named Bitcoin was released by Satoshi Nakamoto. Bitcoin has a number features that significantly weakened government's control over the currency system. This brought it popularity, against the background of the financial crisis in 2008 where government credit was being questioned. As digital payment systems such as Bitcoin starts to dominate the world economy, Bitcoin or digital currency financial fraud is becoming a critical issue to our society.

This dissertation aims to construct a Bitcoin fraud detection system that uses machine learning algorithms to detect Bitcoin fraud. The system extracts meaningful features from Bitcoin ledger and fits the feature data to the fraud detection model. Unsupervised learning algorithms are implemented to construct the fraud detection model. The unsupervised learning algorithms we have implemented are K-means clustering and one-class SVM. K-means clustering is a method used for clustering analysis and one-class SVM is an algorithm for anomaly detection. The models have successfully recognized some patterns for the identification of Bitcoin fraud and have successfully identified a previously unobserved fraud.

Acknowledgments

*I would like to thank my supervisor Donal O'*Mahony for his invaluable guidance and support. I would also like to thank my family and friends for their encouragement and support during my master.

Table of Contents

Table of Figure

Chapter 1

Introduction

1.1 Motivation

Currency is the cornerstone of human civilization. The form of currency has changed over time, transitioning from metal, silver and gold in the early stage to paper-money today, and possibly, in the near future, digital currency.

The concept of digital cash was first introduced by David Chaum in 1983, in his research paper. Since then, researchers and entrepreneurs flooded into the investigation and study of digital currency. In 2008, a revolutionary digital payment system named Bitcoin was released by Satoshi Nakamoto, in his research paper ‘Bitcoin: A Peer-to-Peer Electronic Cash System’(Nakamoto, 2009).

Unlike traditional paper-currency, where government asset backs the credit, the credit of Bitcoin was backed by mathematical natures and all the players involved in the Bitcoin network, under a decentralized system. For the generation of every new Bitcoin, a number of mathematical conditions need to be satisfied. The transaction information of Bitcoin was transparent and accessible to every player within the system. This significantly weakened government’s control over the currency system, which brought it popularity, against the background of the financial crisis in 2008 where government credit was being questioned. Today, there are on average 350,000 Bitcoin transactions per day worldwide.

For any modern payment system, financial fraud always remained to be one of most concerning issue. In 2017, it was believed that the UK had lost over 20 million per days due to the financial fraud. [1] As a digital payment system, financial fraud/anonymity was always remained to be a major topic for Bitcoin. As digital payment system such as

Bitcoin starts to dominate the world economy over time, digital currency or Bitcoin financial fraud is becoming a critical issue to our society. Apart from that, machine learning had become a hot topic over the last decade, many researchers have solved many problems of their research fields with machine learning algorithms. Hence, in this research, we will have an in-depth investigation into the fraud detection of Bitcoin and will attempt to use machine learning algorithms to construct a fraud detection system for Bitcoin.

1.2 Aim

Since the invention of Bitcoin in 2009, by Satoshi Nakamoto, a number of phenomenal papers had been published in the field of Bitcoin fraud/anonymity. For the current state-of-the-art, unsupervised learning algorithms were applied by the researchers to detect the pattern of Bitcoin fraud.

The methodology of the current state-of-the-art can be summarized as feature extraction and modeling. Feature extraction refers to the use of domain knowledge to extract meaningful features for the machine learning algorithms. From the reading of the state-of-the-art articles, we discovered some issues regarding the features extracted for the machine learning model. In fact, from a domain knowledge perspective, some features employed by the state-of-the-art articles may not be appropriate or enough for the detection of Bitcoin fraud. This is, in fact, mainly due to the limitation of data.

Hence, this research aims to study and investigate some of the most recent Bitcoin fraud and to create new features or to justify old features, based on the characteristics of these frauds. These features will then be combined with the state-of-the-art methodology to construct a fraud detection system for Bitcoin.

From a learning perspective, I am aiming to gain a good understanding of knowledge of Bitcoin, machine learning and fraud detection, through practically designing and constructing this state-of-the-art Bitcoin fraud detection system,

Chapter 2

State-Of-The-Art

In the previous chapter, we have provided an overview of the objective and motivation behind this research. In this chapter, we will give an in-depth overview of all the prior-knowledge needed for this research. These include concepts of Bitcoin, concepts of machine learning, the current state-of-the-art of Bitcoin fraud detection and more. The two unsupervised learning algorithms implemented for this research will also be explored deeply in this chapter.

2.1 Bitcoin

Bitcoin is a digital currency that was published in 2009 by Satoshi Nakamoto, in his paper, 'Bitcoin: A Peer-to-Peer Electronic Cash System.' [1] It is a revolutionary digital payment system with innovative features that made it secure and reliable. The two major components that comprise the system of Bitcoin are the Bitcoin transaction system and Bitcoin Blockchain.

2.11 Bitcoin Transaction System

2.111 Bitcoin Address

As mentioned, Bitcoin is a cryptocurrency. Cryptography is used in many places within the Bitcoin system to ensure its' reliability and security. Cryptography is used in the

Bitcoin transaction system for the identification of the owner of the Bitcoin. We can consider the Bitcoin transaction system as a continuously growing ledger, composed of timestamp-ordered transactions, that is public and is accessible for everyone. [3] Unlike the real-world paper money, where each person has paper notes with fixed amounts on his hand, Bitcoin doesn't have paper money for the user. Instead, each Bitcoin owners owns a cryptographic private key. For each output of previously transferred transactions, there is a public key. This public key can only be 'unlocked' by the private key held by the owner of this transaction output. (Will discuss more in the later sections) We call this public key Bitcoin address. With the private key, the owner can create a new transaction, and transfer the output to a new receiver. The outputs of this new transaction will also have a public key and the private key is held by the new receiver.

2.112 Unspent Transaction Output(UTXO)

It is important to note that, a single transaction output can only be used once. For the example presented in the previous section, when the user transferred the output, this output can never be used again. This rule is set to avoid the double spending of the same Bitcoin transaction output. For those transaction outputs that have never been transferred, we call it Unspent Transaction Output(UTXO). In this case, the transferred transaction output is no longer a UTXO and the output of new transaction created will become a new UXTO. The UTXOs in the Bitcoin ledger can be considered as all the Bitcoin money exists in the entire currency system.

2.113 Bitcoin Transaction

A transaction is created when a Bitcoin owner transfers his UTXO to a receiver. In general, a transaction is composed of:

- A Transaction Identifier/Hash(TXID): An unique identity assigned to each transaction

- Inputs
  - This is the data that references the UTXOs that are owned and intended to be used by the sender
  - There can be multiple inputs, as a single UTXO may not be enough for the payment
  - Each input contains:
    - Previous TXID: Transaction identity of the UTXO(s) owned by the sender. The sender is trying to transfer the Bitcoin in this UTXO to the receiver.
    - Output-Index: This is the index of the output of the UTXOs. Since the unspent transaction can have multiple outputs and each of these outputs becomes a single UTXO. The index tells the system which UTXO is intended to be spent.
    - Signature Script: As mentioned, to “unlock” or use a UTXO, owner of Bitcoin needs to prove his identity with the corresponding private key. Bitcoin proves the identity of the sender through an asymmetric cryptography algorithm called ECDSA. Each UTXO in a transaction points to a Bitcoin address (public key). For the sender to spend the Bitcoin in this output, he needs to provide a signature encrypted with the corresponding private key. The encrypted signature can then be verified with the address of the UTXO.
- Outputs
  - Similarly to the inputs, there can also be multiple outputs
  - Each output contains
    - Amount of Bitcoins in this output
    - Public key script: This is the Bitcoin address of this transaction output

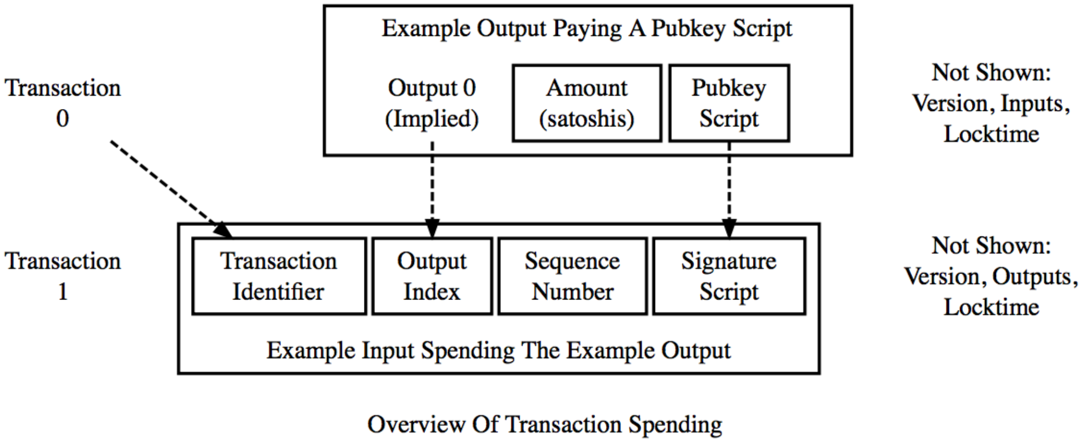

Example Output Paying A Pubkey Script
Transaction 0
Output 0 (Implied)
Amount (satoshis)
Pubkey Script
Not Shown: Version, Inputs, Locktime
Transaction 1
Transaction Identifier
Output Index
Sequence Number
Signature Script
Not Shown: Version, Outputs, Locktime
Example Input Spending The Example Output
Overview Of Transaction Spending

Figure 2.1 Bitcoin Transaction Structure, from Bitcoin.org

From the figure above, we can easily see that the payment of Bitcoin is essentially pointing the UTXO to a new transaction created by the sender.

One of the issues of Bitcoin payment is changing. The UTXO that is being spent by the sender is like a currency note. As what mentioned, a single UTXO can only be spent once to avoid the double spending which means the Bitcoin in a UTXO needs to be transferred within a single transaction. It is impossible to split the UTXO and to pay only part of it to the receiver. In fact, if the sum of outputs is smaller then the sum of inputs, the remaining Bitcoins will be given to the Bitcoin miner as the **transaction fees.**

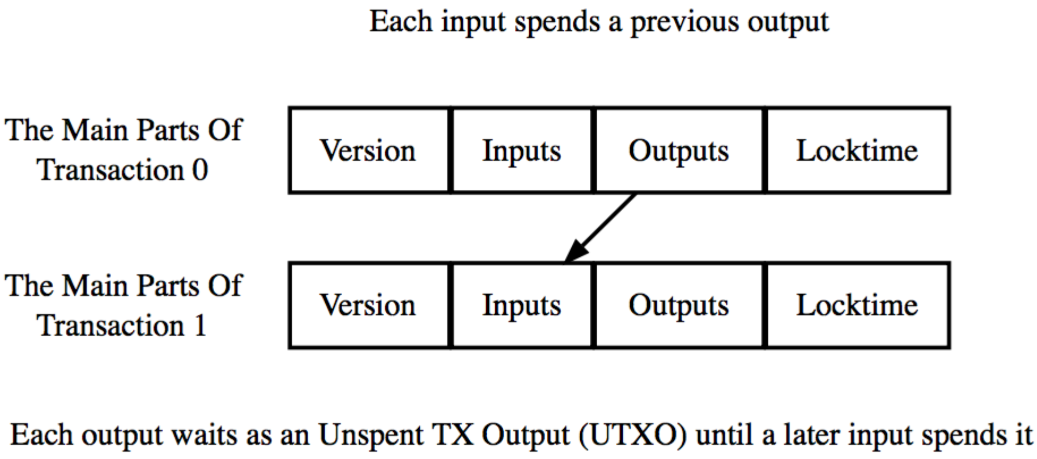

Each input spends a previous output
The Main Parts Of Transaction 0
Version
Inputs
Outputs
Locktime
The Main Parts Of Transaction 1
Version
Inputs
Outputs
Locktime
Each output waits as an Unspent TX Output (UTXO) until a later input spends it

Figure 2.112. How UTXO is spent by the sender, from Bitcoin.org

For the sender to prevent his Bitcoin from being taken by the miner as transaction fees, the sender will need to collect the remaining Bitcoin with his Bitcoin address. For example, if Peter wants to pay John 7 Bitcoins while he only has a UTXO of 10 Bitcoins, he will need to create a new public key(address) and send 7 Bitcoins to the address provided by John and 3 Bitcoins to the address owned by him.

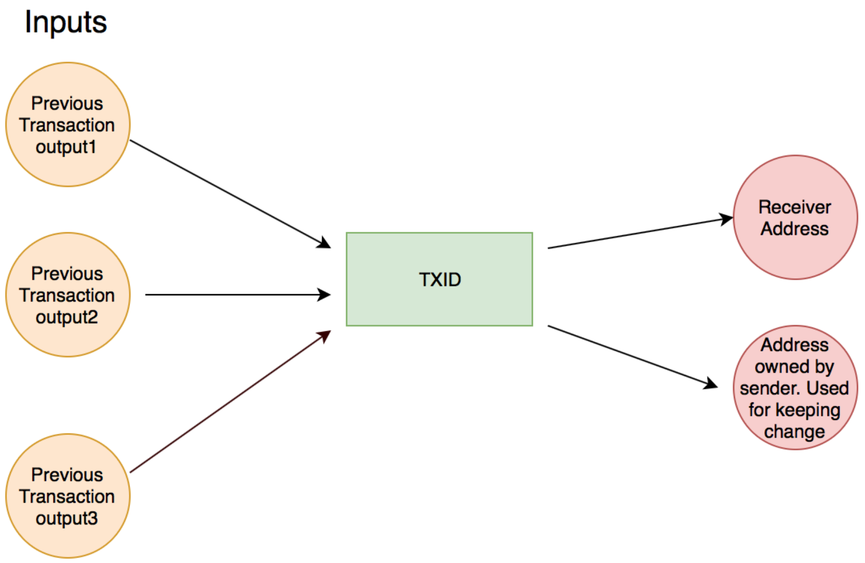

Inputs
Previous Transaction output1
Previous Transaction output2
Previous Transaction output3
TXID
Receiver Address
Address owned by sender. Used for keeping change

Figure 2.2 Bitcoin transaction diagram, how to keep the remaining change.

2.12 Blockchain - A Decentralized and Distributed Data Structure

The blockchain is a public, decentralized, distributed and peer-to-peer data storage technique introduced by Satoshi Nakamoto, along with his published paper of Bitcoin, as the storage technique for the Bitcoin ledger. [3]

2.121 Blockchain

The Bitcoin Blockchain stores the digital ledger of Bitcoin in an ordered and timestamped form. This system is used to protect against double spending and modification of previous transaction records.

Blockchain has a linked-list-liked data structure. It is composed of many blocks. The blocks in the blockchain are distinguished by the block header, this is the unique identity of the block. Each block in blockchain will contain the hash of the header of the previous block. This allows us to know the order of the blocks and hence joins them into a chain

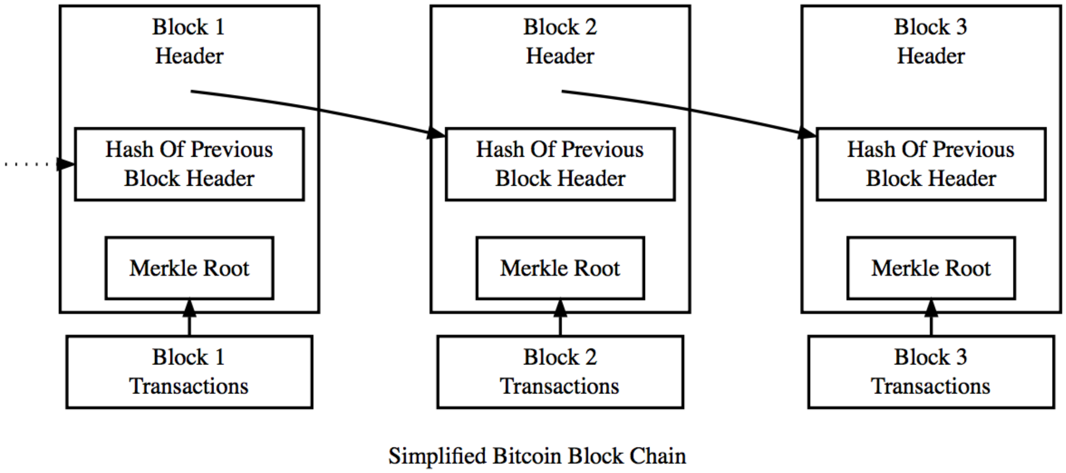


Figure 2.3 Bitcoin Blockchain, from Bitcoin.org

As mentioned, the task of Blockchain is to store the transaction data of Bitcoin. [3] Each block in the Blockchain contains many transactions, and all these transaction data are stored with a data structure called a Merkle Tree.

The Merkle Tree is constructed by pairing each TXID with one other TXID and then hashing them together. If there are an odd number of transactions within a block, the TXID without a partner will be hashed with a copy of itself. The TXID at the top of the Merkle Tree is called Merkle Root, it is often the coin-base transaction. This is the transaction that contains the new Bitcoin mined by the miner of the block. A simple Merkle Tree is presented in figure 2.22. There are four transactions stored in the Merkle tree. The TXID of transaction A and transaction B are hashed and are joined together. The joined hash of transaction A and B are then joined with the hash of transaction C and

D, where transaction C and D are hashed in the same way as transaction A and B. This structure helps us to identify the order of the transactions. (e.g. the transaction A is positioned before transaction B). This is extremely useful for the Peer-to-Peer communication of Bitcoin. For example, when a new Peer is trying to sync the transaction data from multiple peers, it can download these transactions from multiple peers. The transactions from different peers can be sorted based on the order of the TXIDs in the Merkle Tree.

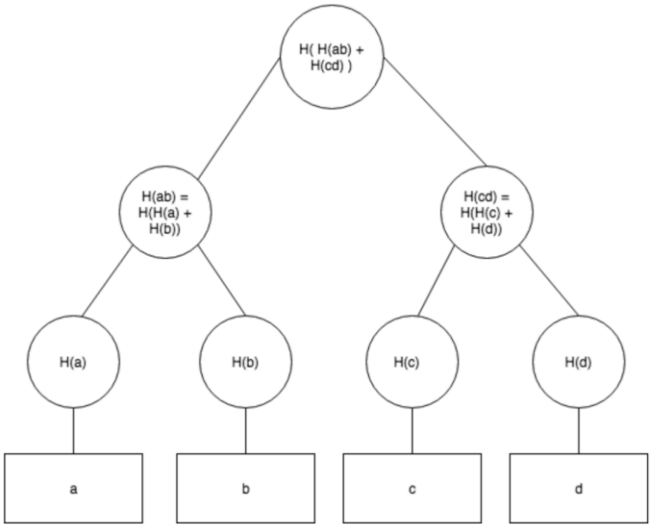


Figure 2.4 A Simple Merkle tree, from https://medium.com/@evankozliner/merkle-tree-introduction-4c44250e2da7

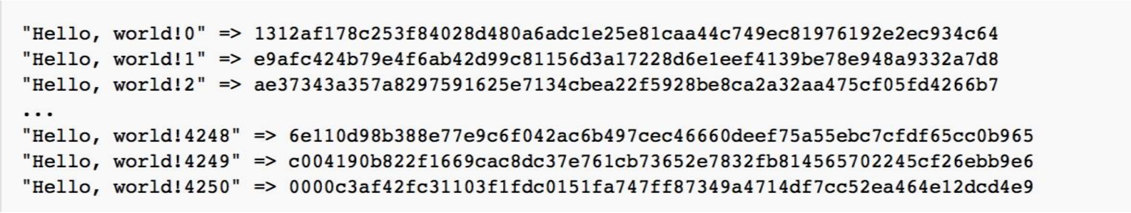
2.122 Bitcoin Mining

Unlike traditional banking or currency system, Bitcoin is a decentralized currency system, and there is no currency center or bank in the Bitcoin system so that Bitcoin can’t be created through money printing. The only way to create new Bitcoin currency is through mining.

As a distributed currency system, Bitcoin system requires anonymous peers on the network to engage in the maintenance of the system. This engagement is called Mining. As mentioned, for the sender to transfer the Bitcoin to a receiver, he will need to create a new transaction. This transaction is not moved into the ledger immediately. Instead, verification is needed to move the transaction into the Bitcoin ledger. In general, the verification involves the following steps:

- For each input in the new transaction: [6]
    - If the referenced UTXO is not in ledger, return an error.
    - If the provided signature does not match the owner of the UTXO, return an error.
- If sum of inputs is less than sum of outputs, return an error
- Return true

During Mining, a Bitcoin miner verifies a fixed number of transactions through the verification process listed above. The verified transactions will be compressed into a block. For the block to be accepted by the entire Bitcoin network, a proof of work is needed. A proof of work is a piece of data which is difficult or time-consuming to produce but easy for others to verify. [5] Producing a proof of work can be a random process with low probability so that a lot of trial and error is required on average before a valid proof of work is generated. This is the process to prove some extra work is done for the creation of a block. Bitcoin uses a Hash-cash proof of work system. Figure 2.1221 is an example of the Hash-cash proof. The miners are required to find a variable of “Hello, world!” so that the SHA-256 hash of this variable begins with ‘000’. The only possible way to solve the problem is through brute force computation. In fact, the difficulty of this work is adjusted so that a new block can be generated in every 10 minutes.



```
"Hello, world!0" => 1312af178c253f84028d480a6adc1e25e81caa44c749ec81976192e2ec934c64
"Hello, world!1" => e9afc424b79e4f6ab42d99c81156d3a17228d6e1eef4139be78e948a9332a7d8
"Hello, world!2" => ae37343a357a8297591625e7134cbea22f5928be8ca2a32aa475cf05fd4266b7
...
"Hello, world!4248" => 6e110d98b388e77e9c6f042ac6b497cec46660deef75a55ebc7cfdf65cc0b965
"Hello, world!4249" => c004190b822f1669cac8dc37e761cb73652e7832fb814565702245cf26ebb9e6
"Hello, world!4250" => 0000c3af42fc31103f1fdc0151fa747ff87349a4714df7cc52ea464e12dcd4e9
```

Figure 2.5 Example of Proof of Work

2.2 Machine learning

Machine learning has become a hot topic over the last decade. It is the concept which teaches machines to detect different patterns and to adapt to new circumstances. [3] As a consequence of this increasing popularity, many people brought machine learning techniques to the field of financial fraud detection.
Machine Learning systems can be classified according to the amount and type of supervision they get during the training. The main categories of machine learning are supervised learning and unsupervised learning. [5]

2.21 Supervised Learning

In supervised learning problem, the training data are composed of features and label/Target. For example, if I am trying to predict the salary of a person with his age and years of education. The age and year of education data will be the feature of the dataset, and the salary will be the target of the dataset. With the supervised learning algorithm, we are trying to identify the patterns between the features and the labels so that the value of the label can be predicted based on the values of the features. The two types of supervised learning algorithms are regression and classification.

2.211 Classification

Classification refers to the use of a machine learning algorithm to classify the discrete classes of instances or data. A good example of classification will be spam classification. With spam classification, we are constructing a model to predict if an email is a spam or not. A number of emails are collected and are used as the dataset. The characteristics of these emails are then extracted from the features of the dataset. The process of extracting these features is called feature extraction. It refers to the process of transforming input data into a set of features. An example of features for the spam detection could be a number of carbon copy(cc). If the sender is sending this email to too many people, then the email may be spam. [6] Each instance in the dataset are labeled so that if the email is Spam, we label it true or else we label it false. We can then fit a machine learning algorithm to the dataset to create a model. If the model is generalized (has a good ability to perform well on previously unobserved data [7]), we should be able to predict the class of a previously unobserved email (e.g., is it spam or not).

2.2111 SVM(Support Vector Machine) Classification

The classification algorithm relevant to this research is Support Vector Machine Classifier. The SVM or support vector machine classifier is a promising classification technique proposed by V.Vapnik and his co-worker in 1990s. [11] Fig 2.211 present an SVM model for the spam classification problem. The red instances are the spam email, and green cluster is the non-spam emails. The support vector machine intends to find a hyper-parameter(line) that maximize the margin(distance) between itself and the support vectors. The support vectors can be considered as the outliers of each class in the dataset. [12] In the right-hand-side plot of the figure, the support vectors of the green class are the two leftmost instances. They are the instances that are closest to the red class instances. To determine the hyperplane of the SVM model, the model intends to find a hyperplane that minimizes the cost function:

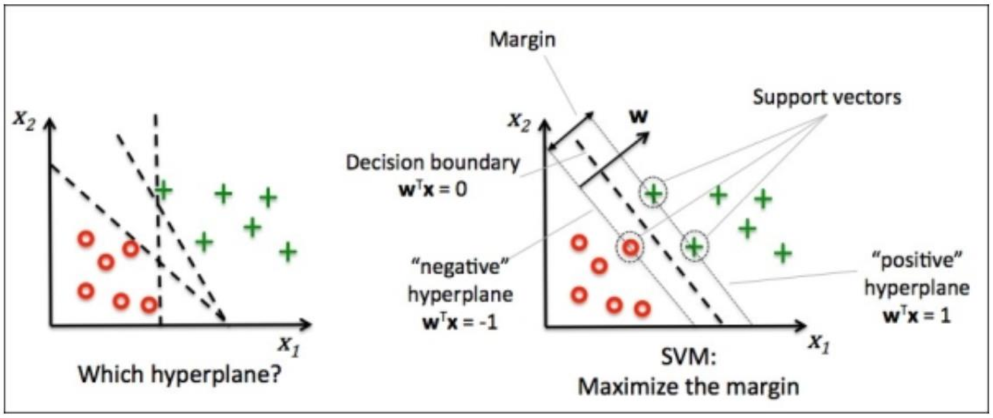


Figure 2.6 Support Vector Machine, from Deeper Insights into Machine Learning by John Hearty, David Julian, Sebastian Raschka

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\max(0, 1 - y^{(i)}\theta^{T}x^{(i)}) + \lambda\theta^{T}\theta$$

where:

m is the number of instances in the dataset

y_i is the value of the target of the ith instance

x_i is the values of the features of the ith instance

θ is the vector of weights of each feature, e.g. the hyperplane is essentially an equation such as $w_1*x_1 + w_2*x_2 + \ldots + w_n*x_n$. W are the weights of the corresponding features stored in the vector θ.

λ is the penalty applied to the hyperplane

This cost function equation estimates the average distance between the hyperplane and each of the support vectors so that we can maximize the margin by minimizing the cost function. The equation also includes a penalty for the SVM model.

If the model is applied to a previously unobserved data, the model will predict it to be spam if it is at the red side or vice versa. Since the hyperplane of the model is linear, the SVM model presented is also called Linear SVM Classification. By implementing kernel method, we can construct non-linear SVM model. Kernel method can be thought of as an instance-based learning method. A weighting function K(x,z) is defined to maximize x when x equals to z and decays when the distance between x and z increases. In the cost function, $\theta^{T}x$ represents the linear hyperplane where theta is a vector contains the weight of each feature and x_i is a vector contains the value of each feature of the ith instance. By replacing $\theta^{T}x$ with a kernel function, for example, Gaussian kernel, we will be able to generate a non-linear hyperplane (fig 2.2112):

$$J(\alpha,\theta) = \frac{1}{m}\sum_{i=1}^{m} \max(0, 1 - y^{(i)} \sum_{j=1}^{m} \alpha_j y^{(j)} K(x^{(i)}, x^{(j)})) + \lambda\theta^T\theta$$

where:

m is the number of instances in the dataset

y_i is the value of the target of the ith instance

x_i is the values of the features of the ith instance

θ is the vector of weights of each feature

λ is the penalty applied to the hyperplane

$K(x_i, y_i)$ is the kernel function

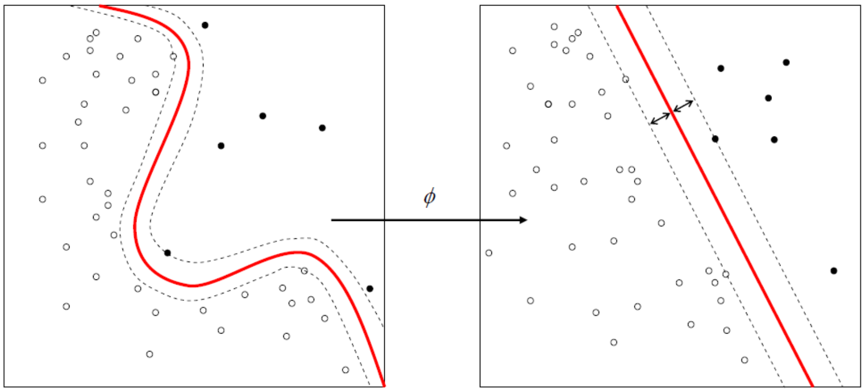


Figure 2.7 Kernel SVM, from Wikipedia support vector machine

2.212 Regression

Regression refers to the use of a machine learning algorithm to predict the one or more continuous variable. An example of regression is the prediction of salary. Let's assume there is a dataset consist of age attribute and salary attribute. We are attempted to predict salary of a person based on his age. A linear regression model is constructed.(fig2.212) This is simply a line of best fit for the dataset, determined through the sum of square error, the cost function of the model.

$$RSS = \sum_{i=1}^{n}(y_i - f(x_i))^2$$

where:

y_i is the value of the target of the ith instance
x_i is the values of the features of the ith instance
f is the function of the regression model

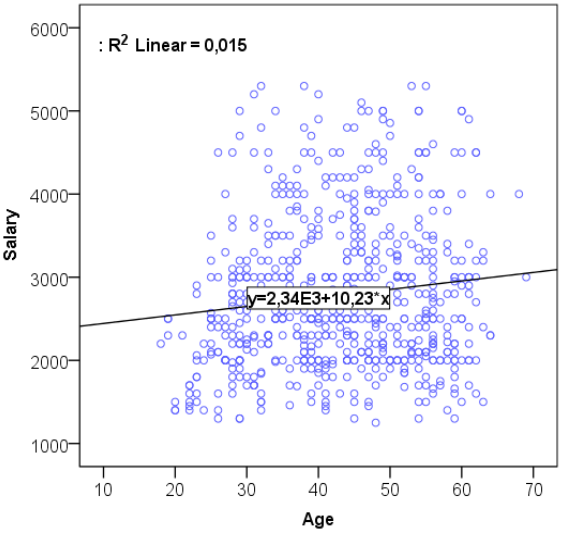


Figure 2.8 Linear regression, age against salary, from https://blogs.helsinki.fi/quantitative-communication/data-analysis/visualising-data/

With the trained model, we can predict the salary of a previously unobserved person with his age.

2.213 Performance Measurement

The performance of supervised learning models can be evaluated with a number of metrics. In general, the dataset will be split into a training dataset and testing dataset. The model will be constructed with the training dataset. Features in the testing dataset will be fitted into the trained model to obtain predicted values. The predicted values will be compared with the actual value, through a selected metric, to obtain the performance of the model. This is called cross-validation. For regression, a frequently used metric is Sum-Of-Square Error. This is to sum the square of all the difference between the actual value and the predicted value:

$$RSS = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

where:

y_i is the value of the target of the ith instance
x_i is the values of the features of the ith instance
f is the function of the regression model

For classification, a frequently used metric is accuracy. The accuracy of a classification model is calculated from the Confusion matrix measure:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$$

confusion matrix can be defined as:

- Ture Positive/TP: Actual class is true and predicted class is true
- True Negative/TN: Actual class is false and predicted class is false
- False Positive/FP: Actual class is true and predicted class is false
- False Negative/FN: Actual class is false and predicted class is true

2.22 Unsupervised Learning

In unsupervised learning, the training data is unlabeled. This means we will not be able to use the existing data to predict the outcome of unobserved data. Instead of making a prediction, unsupervised learning is used to uncover the ‘hidden structure’ of the unlabeled data.

Despite machine learning has been found to be extremely powerful in the application of many fields, fraud detection is still being regarded as one of the challenging problems. The biggest issues of fraud detection, especially Bitcoin fraud detection, are the lack of labeled data and the rarity of fraudulent transactions. There are no organizations that collect and track Bitcoin fraud. When a Bitcoin user suffers from fraud, the only thing he can do is to publish fraudulent Bitcoin address on social media. It is also important to notice that fraudulent transactions are rare in any financial system, including Bitcoin, in comparison with the massive number of transactions. Both of these two factors have combined making the labeling of the Bitcoin transaction fraud extremely difficult. Another problem with fraud detection is that the tactics of the fraudster are constantly changing. It may not be a good practice to try to predict a new type of fraud with a model that is based on old types of fraud. In this respect, unsupervised learning techniques have become widely used for the Bitcoin fraud detection. The advantage of unsupervised learning is that it helps us to identify the pattern in the dataset.

2.221 Clustering – K-means Clustering

K-means clustering is a well-known clustering algorithm created by J. MacQueen in 1967. It is commonly selected as a clustering method because of its simplicity and applicability. [13] The algorithm classifies datasets based on attributes value into K groups.

2.2211 Algorithm

The methodology of classical K-means Clustering algorithm is the following: [14]

1. A hyper-parameter K is initialized by the user. A hyper-parameter is a parameter whose value is set before the learning process begins. In other word, it needs to be set by the user manually. The hyper-parameter K refers to the number of clusters to group data into.
2. The position of the first K clusters needs to be initialized, it can either be initialized by:
    - the user manually, selecting k instances from the dataset as the initial centroids of the clusters
    - letting the application taking k elements as the initial centroid randomly
3. Calculate the arithmetic means of each cluster formed in the dataset. This means for each cluster, the cluster centroid is replaced with the mean of all the instances belonging to the corresponding cluster.
4. Each instance or record is assigned to the nearest cluster based on the measure of distance. In here the Euclidean distance is used to measure the distance. It is the cost function of the model.

$$\frac{1}{n}\sum_{i=1}^{n}\left[\min_j \ d^2(x_i, m_j)\right]$$

where:

n is number of instances

x_t is the position of the instance

m_j is the position of the cluster centroid of the instance

d is the distance function for the two points

5. Step 3 and Step 4 are repeated until the model converges (the old clusters are the same as new)

2.2212 Tuning/Hyperparameter Optimization for k-means clustering

Hyperparameters refer to the parameters whose value is set by the user before the learning process begins. [15] By contrast, the values of other parameters are derived via training. Hyperparameter needs to be set manually by the user. This process is called tuning or hyperparameter optimization. [16]

For k-means clustering, the value of the hyperparameter K can be determined through elbow method. [14]Elbow method is a method which looks at the percentage of variance. The method exists upon the idea that a value should be chosen for k so that if we increase the value of k, the percentage of variance of the clusters doesn’t getting much better. The methodology of elbow method is the following:

1. Start from k* = 1
2. Increment the value of k*
3. Implement the clustering algorithm with k = k*. Measure the cost of the cluster model where k = k*. In this research, Sum-Of-Square Error is used for the cost measure. This is the sum of the squares of the distances between each instance and the centroid of the corresponding cluster:

   $$SSE = \sum_{i=1}^{n} (x_i - \bar{x})^2$$

   where:
   n is number of instances
   x_i is the position of the instance
   x-hat$_i$ is the position of the cluster centroid of the instance
4. Repeat step 2 and step 3. If at some point where the value of k at which improvement in the quality measure declines the most(e.g. the increase of k no longer dramatically decrease the sum of square error of the model) , the value of k is the elbow, at which we should stop dividing the data into further clusters. [15]
5. End

From figure 2.212, the elbow point or K is 3. This is when the increase of k no longer dramatically decreases the sum-of-square error of the model, and the graph starts to become gently sloped.

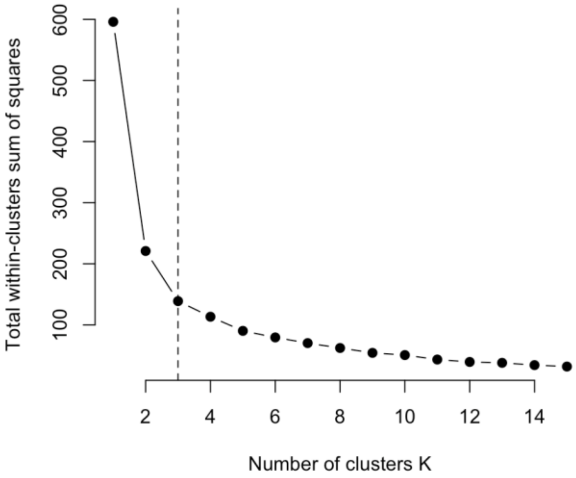


Figure 2.9 elbow method, sum-of-square error against k, from http://www.sthda.com/english/wiki/print.php?id=239

2.222 Anomaly Detection – One-class SVM

2.2221 Algorithm

Since fraudulent transactions are rare, they can be identified as an anomaly or outlier of the dataset. This is called **Anomaly detection**. One of the most popular anomaly detection models is unsupervised SVM or one-class SVM. It is an unsupervised learning algorithm based on the Support Vector Machine. [19] There are a number of types of one-class SVM. The one-class SVM algorithm implemented in this research is the v-SVM algorithm by Schölkopf. [20]

Two-class SVM classifiers model are presented in the supervised learning section where the model is intended to analyze the data and recognize the patterns of the data by identifying a hyperplane that separates the training data into the class of spam or the class of non-spam. With one-class SVM, there is only one class, which is the normal class. As shown in figure 2.2221, the model is intended to find a hyperplane that maximizes its distance from the origin. The data apart from the origin are classified as normal or +1 class. For those data that are outside the hyperplane, we classify them as the outliers, anomalies or -1 class. [21]

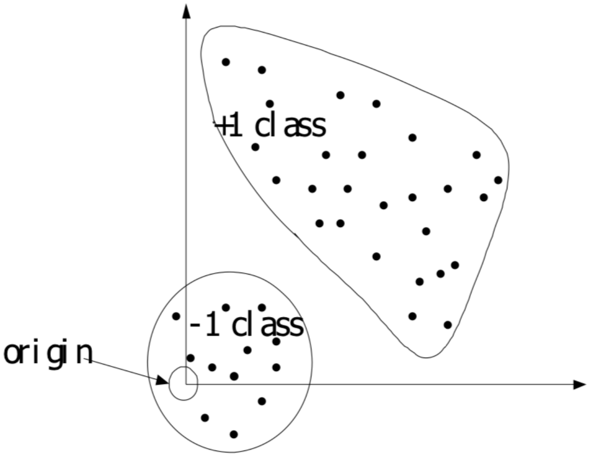


Figure 2.10 an example of one-class SVM, from [21]

In general, the hyperplane of one-class SVM model can be found through solving the following minimization quadratic programming problem:

$$\arg\min_{\mathbf{w},\xi,\rho} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n} \xi_i - \rho, \qquad (1)$$

$$\text{subject to} \quad \begin{cases} \langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle \geq \rho - \xi_i \\ \xi_i \geq 0 \end{cases}, \qquad (2)$$

[19]

“where n is the number of training sample, $\xi = [\xi 1 \; . \; . \; . \; \xi n]$ and $\Phi(\cdot)$ is a kernel function. In the equation (1), w represents the normal vector and ρ is the offset of the desired hyperplane in the feature space. The slack variable ξ_i measures the degree of misclassification of the data. The trade-off parameter $\nu \in [0,1]$ is an upper bound on the fraction of training samples outside the decision boundaries and a lower bound on the fraction of support vectors (i.e. the data points that cannot be discarded in simplifying the SVM solution).” [19]

Simply speaking, the equation tries to find the smallest possible hyper-ball(non-linear hyperplane) in the high-dimensional feature space that captures the majority of the observations(data). The data outside the hyper-ball are the anomalies. The parameter v ∈ [0,1] can be thought as the percentage of data points within the hyperplane. This is usually a hyperparameter that needs to be determined by the user. Apart from v, the kernel function $\Phi(\cdot)$ is also a hyperparameter that needs to be chosen by the user. This is achieved through tuning and we will discuss it in the next section.

2.2222 Tuning

In general, if the one-class SVM model is for classification problem, the hyperparameter of one-class SVM is tuned through cross-validation. This means a labeled test-set is used to measure the performance of the one-class SVM model so that the kernel function, the gamma of the kernel function and hyper-parameter v will be adjusted until the model generalizes (work well with previously unobserved data).

However, since our model is for unsupervised learning, we do not have the labeled test-set to tune the hyperparameter. Hence, we follow the approach taken by Steven and his colleague, where they employed one-class SVM for the detection of Bitcoin fraud. [20] Following their approach, we choose the Radial basis function kernel for the model, where the value of gamma is 1/feature size. For the hyperparameter v, we will determine it through Dual Evaluation. This means we can test for our methods’ consistency by checking if detected suspicious users own detected suspicious transactions. Specifically, with the user graph we can get the top N user outliers and with the transaction graph we can get the top M transaction outliers. Since we are only testing 1000 instances for this experiment, we choose N = M = 10. We then determine X_N - the set of transactions corresponding to the top N node outliers and Y_M - the set of users corresponding to the top M transaction outliers defined above. We define:

$$A1 = |X_N \cap \text{top } X_N \text{ transaction outliers}|\ |X_N\ |$$

and

$$A2 = |Y_M \cap \text{top } Y_M \text{ user outliers}| \,.\, |Y_M|$$

Finally, we define the Dual Evaluation Metric m_{de} by:

$$m_{de} = (A1 + A2)/2 \,.$$

Note that $m_{de} \in [0, 1]$, and the bigger it is, the more accurate our method is.

2.223 Performance Measure

Unlike supervised learning where you can cross-validate the model with metrics such as the sum of square error and confusion metric, you are not able to measure the performance of unsupervised learning since there is no target/label for you to compare. Hence, the performance of unsupervised learning algorithms can only be measured through visualization. A group of experts from a variety of backgrounds will be brought together to determine the performance of the unsupervised learning model through visualizing the pattern recognized by the model. [9] This is called visualization evaluation.

2.3 State of the art of Financial Fraud Detection

To fully understand the current state of the art of the Bitcoin fraud detection, it is essential for us to first explore the state-of-the-art techniques of financial fraud detection, the Bitcoin fraud detection in the industry and the Bitcoin fraud detection in the Academia. Through the comparison, the limitations and the reason behind the limitations of the current state-of-the-art of Bitcoin fraud detection can be understood.

2.31 Financial fraud detection

Over the last decades, financial fraud has become one of the biggest threat to the stability and security of the human society. Researchers and entrepreneurs had put an enormous amount of work on detection and prevention of financial fraud. As a consequence of the increasing popularity of machine learning, researchers started to bring machine learning techniques to the field of financial fraud detection.

Two recent credit card fraud detection papers are “Credit card fraud detection using machine learning techniques: A comparative analysis”, published by John O’Awoyemi and colleague in October 2017 [23] and “Using deep networks for fraud detection in the credit card transactions”, published by Zahra Kazemi and his colleague in December 2017. In Credit card fraud detection using machine learning techniques: A comparative analysis”, John and his colleague fitted their dataset to supervised learning algorithms such as k-nearest neighbor, logistics regression and more. And in “Using deep networks for fraud detection in the credit card transactions”, supervised artificial neural network has been employed for the detection of financial fraud.

Both papers employed supervised learning algorithms for the construction of the model. It is, in fact, much easier to the collect labeled credit card fraud than Bitcoin fraud as the banks act as centers which control and monitor credit card fraud as well as the information of the users. This is different for Bitcoin, due to the decentralization nature and anonymous nature of the Bitcoin system, there aren't any organizations that track and monitor fraudulent transactions in Bitcoin and it is extremely difficult to track the information of users. This makes it extremely difficult for us to collect labeled data for Bitcoin fraud.

2.32 Bitcoin fraud detection in the industry - Coinbase

Coinbase is a secure online platform for buying, selling, transferring, and storing digital currency founded in 2012. [19] It has over 10 million users and over 50 billion trades since it was established. Fraud is one of the biggest security concerns for digital currency platforms such as Coinbase.

As mentioned, the state-of-the-art techniques used by industries such as Coinbase is much more advanced in comparison with those who do academic research. This essentially due to three reasons:

- Data-richness: Companies such as Coinbase know much more about the digital currency market in comparison with the academic researchers. Bitcoin is an anonymous digital payment system which means that no identification is needed. A person can hold many bitcoin addresses and no one will know these addresses are belonging to the same person. This makes it extremely difficult to track the behavior of a user. This vulnerability is being blocked in the ecosystem of Coinbase, each user will need to provide his identity when he registers so that it is extremely difficult for a user to create a new account.
- Better feedback mechanisms: As mentioned previously, one of the biggest problems with Bitcoin is that there are no organizations that collect and track Bitcoin fraud. As a consequence of that, there is no labeled data for Bitcoin fraud. It is very difficult for us to judge if a transaction is fraudulent or not. Hence, the only option we can use is unsupervised learning. This vulnerability is also resolved by Coinbase, through an efficient feedback mechanism. Coinbase users are encouraged to report the fraud immediately after the occurrence. This provides a labeled dataset with good quality. In fact, Coinbase detects fraud with supervised learning.
- Less responsibility: There are over a hundred of a thousand of Bitcoin transactions every day. Only part of these transactions is transferred through Coinbase. The fraud detection system constructed by Coinbase is only responsible for these transactions, and Coinbase has an abundance of information for all these transactions due to the two factors mentioned previously. [20]

The entire picture of the techniques behind the Coinbase fraud detection system is unknown. However, they have presented some fragments of them in a talk. (https://www.youtube.com/watch?v=tCNWnrvIoJo)

2.33 Bitcoin fraud detection in Academia

Unlike industrial actors such as Coinbase, academic researchers do not have a platform or product that helps them to actively and efficiently collect useful data for the detection of Bitcoin fraud. In fact, the only data that can be easily accessed by the academic researchers is the public ledger data of Bitcoin stored in Bitcoin Blockchain.

As Bitcoin was invented in 2009, the study of Bitcoin fraud detection is a relatively new research field. In recent years, due to the popularity of Machine learning, many researchers started to apply machine learning techniques into their research field. As a consequence, researchers also started to bring machine learning into the field of Bitcoin fraud detection. One of the earliest research papers in the field is the “Analysis of Bitcoin Network Dataset for Fraud” [25], published by Deepak Zambre and Ajey Shah in 2013. The study was based on another research paper “An Analysis of Anonymity in the Bitcoin System”, [26] published by Fergal Reid and Martin Harrigan in 2011. Fergal and Martin were intended to analyze the anonymity of Bitcoin. To address this, they analyzed the structure of the transaction network and the user (Bitcoin anonymous address) network of the Bitcoin system. Fergal and Martin’s research has been developed upon, by Deepak and Ajey, to the research of the Bitcoin fraud detection. Deepak and Ajey made use of the transaction network data and user network data constructed by Fergal and Martin to analyze Bitcoin fraud. They have extracted a number of features from the transaction network and user network and model them with unsupervised learning algorithms. The algorithm they have implemented is k-means clustering algorithm. Deepak and Ajey have successfully separated the data of the two datasets into a number of clusters. In 2014, Philip Thai Pham and Steven Lee published their research paper “Anomaly Detection in the Bitcoin System - A Network Perspective”. [27] In this research paper, they have made a significant improvement to Deepak and Ajey’s work. They have experimented with new algorithms including Power Degree & Densification laws, K-means clustering and Local Outlier Factor to the transaction network dataset and user network dataset. With these three algorithms, Philip and Steven have also successfully grouped the data into a number of clusters. In 2016, Patrick Monamo and his colleague published the research paper “Unsupervised learning for robust Bitcoin fraud detection”. The research paper experimented “trimmed K-means” and “K-means clustering” to the transaction network dataset and user network dataset and intended to see if the two outcomes of the two algorithms would have any disparities. The experiment had shown some disparities between the two clustering algorithms in the presence of outliers. [28] In 2017, Thai T. Pham and his colleague published “Anomaly Detection in Bitcoin Network Using Unsupervised Learning Methods” published. [23]. The research experimented with a number of unsupervised learning algorithms including Mahalanobis Distance-Based Method, K-means clustering and one-class SVM. The experiment successfully grouped data into clusters with clustering methods as well as detected anomalies of the data with anomaly detection method such as one-class SVM. All the papers mentioned above made use of the dataset housed by the Laboratory for Computational Biology at the University of Illinois. The datasets contains the transaction network data and user network data of the Bitcoin network, starting from the genesis block to block 230686 dated 7 April 2013. In general, their work can be summarized as Feature Extraction and Modelling.

2.331 Feature Extraction

Feature extraction refers to the process of transforming input data into a set of features [6] For Bitcoin fraud detection, feature extraction is used to extract meaningful information from the Bitcoin ledger. The extracted features data will be fitted to the algorithm to create the unsupervised model. The three articles mentioned above all created features based on graph theory. Features are extracted from Bitcoin network. There are two types of network in Bitcoin – the user network and the transaction network. The user network refers to the connections between each unique Bitcoin address. The transaction network refers to the connection between transactions.

2.3311 Transaction Network Features

The features extracted from the transaction network can be summarized as:

- In-degree: In graph theory [24], in-degree refers to the number of arcs going into a node. Regarding the transaction network graph, this is the number of inputs of a transaction. E.g., As mentioned, a sender may combine multiple UTXOs to pay the receiver(as one UTXO may not be enough), the number of UTXO(s) is the in-degree.
- Out-degree: In graph theory [24], in-degree refers to the number of arcs out of a node. Regarding transaction network graph, this is the number of outputs of a transaction. E.g., As mentioned, a sender can send the Bitcoins to multiple addresses through one transaction. The number of addresses it is sending to is the out-degree
- The total value of transaction: This is the sum of bitcoin being sent from the sender to the receiver.

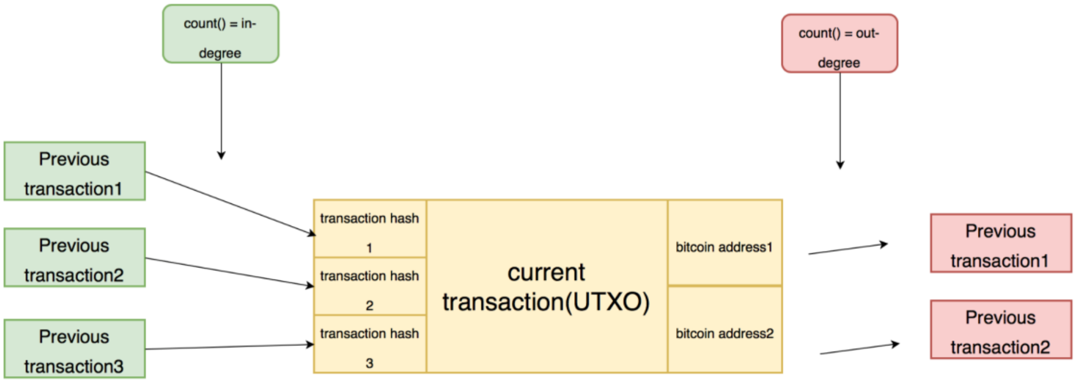


Figure 2.11 Bitcoin Transaction network

2.3312 User Network Features

The features extracted from the transaction network can be summarized as:

- In-degree: The in-degree of a Bitcoin address or user is the number of addresses that have ever send Bitcoin to it.

- Out-degree: The out-degree of a Bitcoin address or user is the number of addresses that have ever received Bitcoin from this user/address.
- Mean incoming transaction value: This is the mean of Bitcoin received by this address
- Mean outgoing transaction value: This is the mean of Bitcoin sent from this address
- Mean time interval: This is the mean time interval between transactions
- Clustering coefficient: [25] Clustering coefficient in graph theory refers to the likelihood that nodes that share a common neighbor are neighbors themselves. For the Bitcoin user network, it is a measure of connectivity amongst neighbors of a given user.

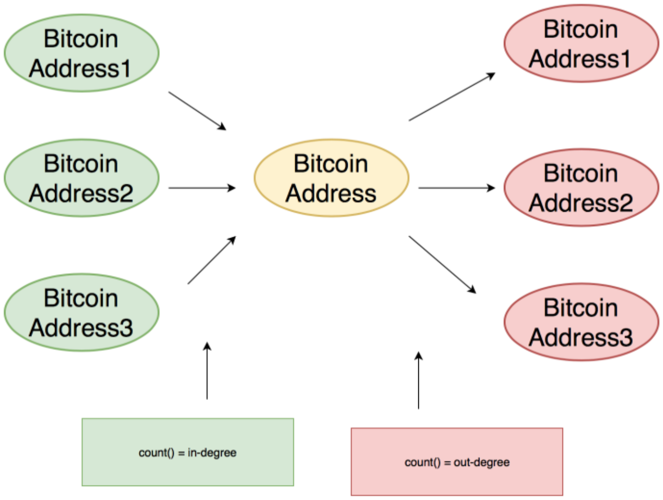


Figure 2.12 Bitcoin User network

2.3313 Dispute on the state-of-the-art feature extraction for Bitcoin fraud

As mentioned, the main goal of this research is to study some of the recent Bitcoin fraud case-study so that new features may be created based on the characteristics of these frauds. In fact, from a domain-knowledge perspective, some of the features employed by the recently published papers may not be appropriate for the detection of Bitcoin fraud.

For the transaction network, features that are being extracted are sum of value, in-degree, and out-degree. It may not be a good practice to construct the model with the value of a transaction. This is because the value of Bitcoin is constantly fluctuating. Hence, the real-world value of a transaction in 2016 can be twice bigger than the real world value of a transaction in 2014, despite the sum of value in Bitcoin are the same. In-degree may also be not appropriate for the detection of Bitcoin fraud. This is because it only tells you how many UTXO the user used for the payment. Indeed, if the payer is trying to pay a lot of Bitcoin he will need many UTXO. However, this also applies to the fact that the value of Bitcoin is constantly fluctuating so that it is very hard to tell if the transaction is fraud based on it. Hence, these two features may not be appropriate for the pattern recognition of Bitcoin fraud. Similarly, the out-degree of a transaction only tells us the number of Bitcoin address it is sending to so that it is extremely difficult to distinguish fraudulent transactions and regular transactions with it. In fact, from this research, it is found that the out-degree of a transaction can be a sign of money laundering. Hence, out-degree can be an appropriate feature. This will be discussed with more detailed in the Design chapter. For the user network, features being extracted are in-degree, out-degree, and more. Similarly to the features of the transaction network, any features that are associated with value of Bitcoin may not be appropriate for the detection of Bitcoin fraud. However, in-degree may be good features for the fraud detection of Bitcoin. Since for those addresses with high in-degree, the user must use this address to collect Bitcoin from many different people. However, we may need other features to separate the trickster from regular business since business may also need to collect Bitcoins from many different people.

2.332 Modeling

The algorithms implemented in some of the research paper have been listed in the previous section. It is important to notice that all these algorithms are graph-based algorithms. This means the clustering of data and detection of anomalies are all based on relative distances of the instances on graphs or in multi-dimensional feature space. The result of the research papers indicated that the model had recognized some patterns from the dataset successfully. However, as discussed in the feature extraction section, it is very difficult to tell if the clusters or the anomalies recognized are really fraudulent since from a domain knowledge perspective, it is very difficult to tell if there is any correlations between the features listed and Bitcoin fraud. For example, the result of the unsupervised SVM model on the transaction network dataset is presented in figure 2.32211. The blue dots on the graph or feature space are the anomaly data detected by the model. The characteristic of these anomalies is that they all have either extremely high out-degree/in-degree, extremely low out-degree/in-degree or both. It is difficult to tell if these features are relevant to Bitcoin fraud. Hence, the goal of this research will be to study some of recent fraud case study and attempt to create some new features or to justify some of the old features, based on the characteristics of these frauds.

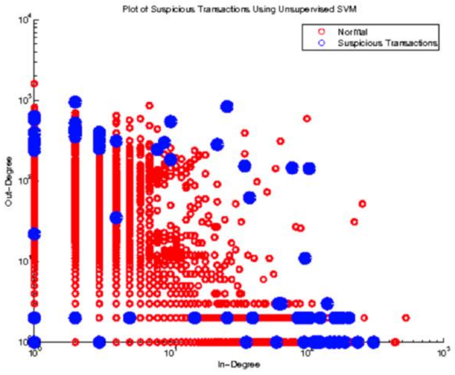


Figure 6. Anomaly Detection using Unsupervised SVM: Transaction Graph

Figure 2.13 Unsupervised SVM model, by Anomaly Detection in Bitcoin Network, Using Unsupervised Learning method

Chapter 3

Design

In the previous chapter, we have provided the knowledge needed for this research by presenting the current state of the art of the Bitcoin fraud detection as well as the fundamental knowledge of Bitcoin. This included the background, system and the architecture of Bitcoin, machine learning algorithms used for the detection and the feature extraction of the models.

In this chapter, we will discuss the design of our approach. A detailed description will be provided to help the reader to understand how the research or the experiment will be carried out and why they are carried out in such ways. This will include the approaches of the research, the architecture of the application as well as the technology implemented and the improvement we are going to make upon the current state of the art.

3.1 Approach

To achieve the objective of the research, a number of approaches need to be carried out. In general, the approaches of the experiment can be summarized as Data Gathering, Data Parsing, Feature Extraction, Data Handling, Modeling and Evaluation.

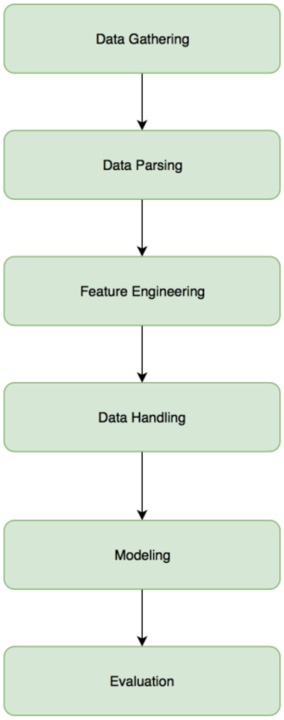


Figure 3.1 Approaches of the research

3.11 Data Gathering

Data collection is critical to the success of this project. When collecting the data, three key concerns arise:

- What data need to be obtained
- Where the data will be obtained
- How the data will be stored

Similarly to other recent, our experiment will also be based on the public ledger data of Bitcoin. The ledger data will be obtained directly from the Bitcoin P2P network Blockchain. We will achieve this with Bitcoin Core, a Bitcoin wallet application. With Bitcoin Core, we become part of the Bitcoin P2P network. The application automatically syncs the local Blockchain ledger with the latest blockchain ledger from other peers in the network. The data will then be stored in a temporal data store as binary files. These binary files will then be transferred to the Hadoop Server, which is deployed on Google Cloud Platform.

3.12 Data Parsing

The Bitcoin ledger data is stored Bitcoin blockchain as binary files. The transaction data of the Bitcoin ledger is in fact encrypted, compressed before being stored in these files as array of bytes. It is impossible to process this array of bytes as they are not readable and understandable. Hence, the next stage of the research will be to parse these binary files so that the transaction data will be decompressed and decrypted from the binary files. The transaction data will then be converted into a table form. Every single transaction in the Bitcoin ledger will be stored as a row in the table. The table will be stored back into the Hadoop server for future processing.

3.13 Feature Extraction

It is impossible for any existing machine learning algorithms and framework to process a transaction ledger data. Hence, meaningful and measurable features need to be extracted from the Bitcoin ledger. These features are then fitted into the machine learning algorithms to create the model.

As discussed in Chapter 2, one of the objectives of this research is to analyze and study some of the recent Bitcoin frauds. From these analyses and studies, some useful characteristics of Bitcoin fraud may be identified so that new features that are useful for the detection of Bitcoin fraud may be created. All these features will be experimented with in this research.

The case study investigated in this research are WannaCry ransomware attack and Blackmail fraud.

3.131 WannaCry Case Study

The WannaCry ransomware attack was a May 2017 worldwide cyberattack by the WannaCry ransomware crypto-worm, which targeted computers running the Microsoft Windows operating system by encrypting hard-disk data. The hacker demands the victims pay a ransom payment in the Bitcoin cryptocurrency. [14] Three Bitcoin addresses were published to the social media by the hacker. Victims were required to transfer Bitcoins to these three addresses [18] so that the hacked operating system will be unlocked.

The study is carried out by investigating the flow of the 'black Bitcoin' gained by the Wannacry hackers. By tracking the transaction network of the three Bitcoin addresses, evidence of money laundering has been found.

The flow of the 'black Bitcoin' can be summarized as follows:

1. The hacker used his private key to transfer the UTXO sent by the Victim to his Bitcoin address.
2. The new UTXOs owned by hacker are transferred to other new addresses. It is believed that these new addresses are also owned by the hacker.
3. The UTXOs in the new addresses are being sent to other new addresses and this happens repetitively. By doing so, the "black money/Bitcoin" is being "spread" over the entire Bitcoin ledger or the Bitcoin transaction network.
4. Some of the transactions have extremely high out-degree, the UTXO is split and sent to a large number of different addresses.

To make it easier for the reader to understand how the money is being laundered, we can look at the diagram presented in Figure 3.11. The figure presents part of the transaction network of one of the Bitcoin addresses published by the Wannacry hacker. The leftmost node in the diagram is the transaction node that holds the Bitcoin sent by one of the victims (e.g., in the predecessor of this transaction node, the victim points the Bitcoin owned by him to the output of a new transaction. He specifies the public key of this output to be the public address published by the hacker. The hacker then creates a new transaction where the input is the output of the transaction from the Victim and the signature is created by the private key of the address published. The leftmost transaction node represents this new transaction created by the hacker)

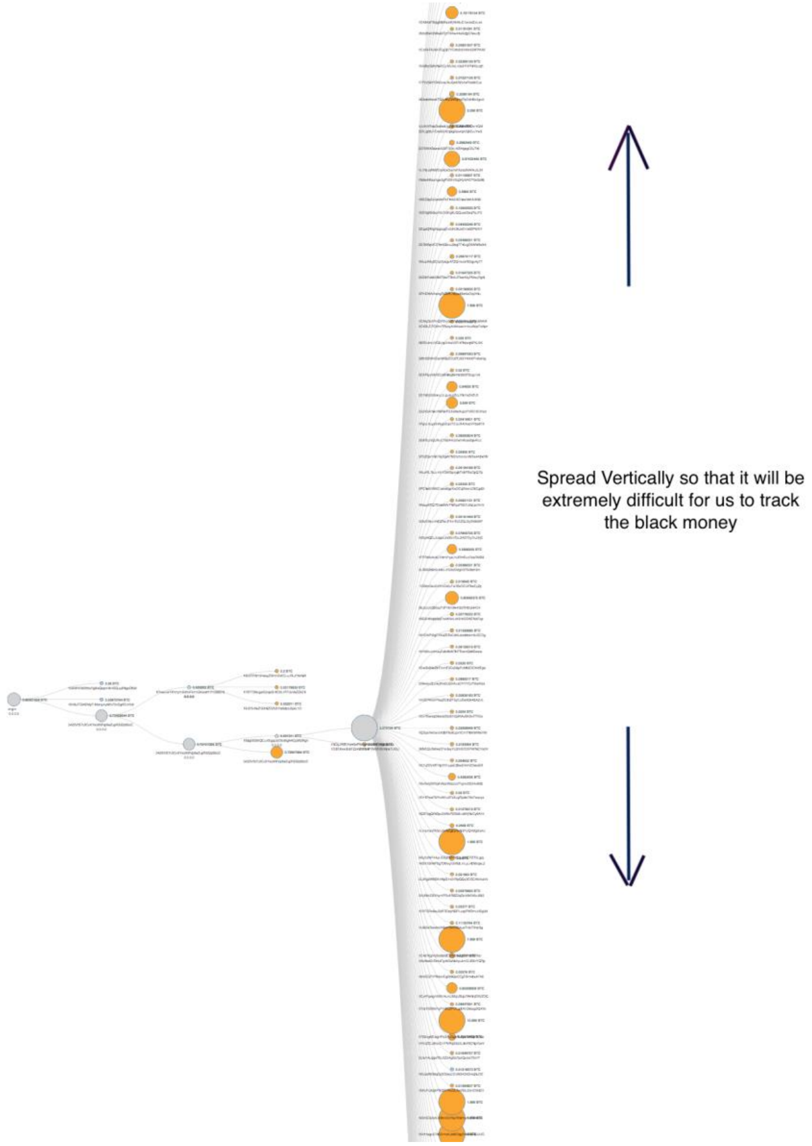

Spread Vertically so that it will be extremely difficult for us to track the black money

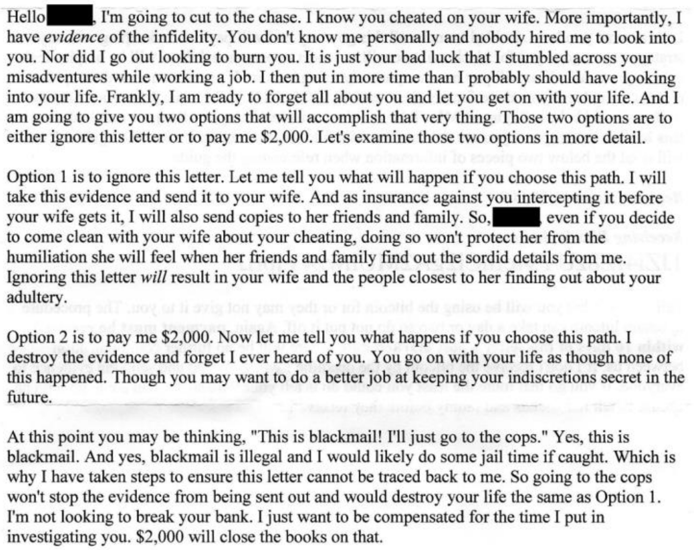
Figure 3.2 WannaCry Money Laundering – transaction network

The spreading of “black Bitcoins” makes it extremely difficult for us to track the flow of the “black Bitcoins”. However, if the transaction network only spends horizontally (e.g.

each node only sends the Bitcoins to a small number of nodes), the tracking will be much easier. This is because all we need to do is to track the transaction nodes with a large amount of Bitcoin. It is also extremely inefficient for the hacker as creating Bitcoin transaction can be slow.(All the transaction needs to be validated). To solve this problem, instead of spreading horizontally only, the hacker also spreads the “Black Bitcoins” vertically. For example, we can see from the rightmost part of the figure that the transaction or UTXO is being sent to an extremely large number of successors. The radius of the circle represents the relative amount of Bitcoin hold by the nodes of the same level. We can spot that some of the nodes hold a large amount of Bitcoin while some only hold a small amount of Bitcoin. It makes it much difficult for us to track as some nodes can be ‘smokes and mirrors’. (e.g. the hacker is giving up some of the nodes and is trying to bring out rest of the Bitcoins)

3.132 Blackmail Case Study

Apart from the WannaCry ransomware attack, other types of Bitcoin fraud are also being studied and analyzed. We picked Blackmail fraud as another case study for this research and intended to find out some characteristics that are different from the characteristics of the Wannacry ransomware attack.



Hello [redacted], I'm going to cut to the chase. I know you cheated on your wife. More importantly, I have *evidence* of the infidelity. You don't know me personally and nobody hired me to look into you. Nor did I go out looking to burn you. It is just your bad luck that I stumbled across your misadventures while working a job. I then put in more time than I probably should have looking into your life. Frankly, I am ready to forget all about you and let you get on with your life. And I am going to give you two options that will accomplish that very thing. Those two options are to either ignore this letter or to pay me $2,000. Let's examine those two options in more detail.

Option 1 is to ignore this letter. Let me tell you what will happen if you choose this path. I will take this evidence and send it to your wife. And as insurance against you intercepting it before your wife gets it, I will also send copies to her friends and family. So, [redacted], even if you decide to come clean with your wife about your cheating, doing so won't protect her from the humiliation she will feel when her friends and family find out the sordid details from me. Ignoring this letter *will* result in your wife and the people closest to her finding out about your adultery.

Option 2 is to pay me $2,000. Now let me tell you what happens if you choose this path. I destroy the evidence and forget I ever heard of you. You go on with your life as though none of this happened. Though you may want to do a better job at keeping your indiscretions secret in the future.

At this point you may be thinking, "This is blackmail! I'll just go to the cops." Yes, this is blackmail. And yes, blackmail is illegal and I would likely do some jail time if caught. Which is why I have taken steps to ensure this letter cannot be traced back to me. So going to the cops won't stop the evidence from being sent out and would destroy your life the same as Option 1. I'm not looking to break your bank. I just want to be compensated for the time I put in investigating you. $2,000 will close the books on that.

Figure 3.3 Hacker Black Email [33]

Figure 3.32 presents an example of blackmail emails collected by CNBC news. The email is broadcasted by the trickster and the victim is asked to transfer the trickster 2000 euro in Bitcoins or otherwise the hacker will send victim’s sensitive information to his wife. According to CNBC, the attacker is hoping that they might get lucky with someone who actually … [has] some infidelity there. And if they hit that target, that's a person who's probably willing to pay." [33] We have found a Blackmail Bitcoin address from a forum - 1CvwXG7AMgunAHqU8UVvAQkpEAS3VsVtnU. [34] By analyzing the transaction flows of the Bitcoin address, we found that this address has received transactions from two addresses. We also found evidence of money laundering as the hacker was also attempting to launder the Bitcoin by spreading the Bitcoin across the transaction network. (fig 3.31322)

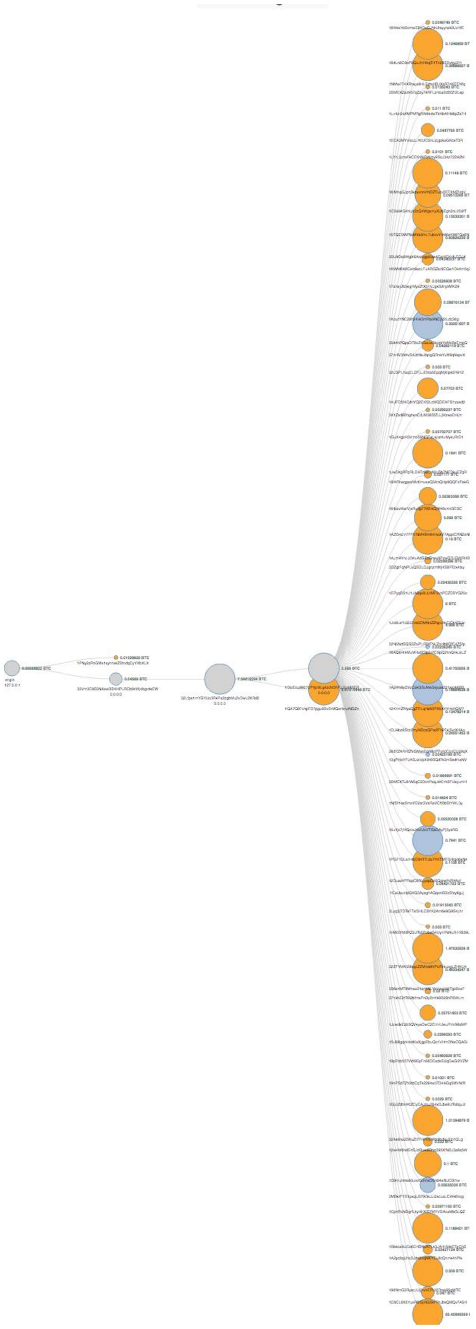


Figure 3.4 Blackmail Money Laundering – transaction network [34]

3.133 Characteristics of fraud

From the two case studies presented, we have noticed some characteristics of Bitcoin fraud, and we are using these characteristics as assumptions for modeling of this research:

1. Limited Communication: From both two case studies, we have noticed that the trickster is always trying to minimize the communication between himself and the public. In the Wannacry case study, the hacker only communicated with the public for one time. This is when the hacker installed the malware to the victim's hard disk. The victim will not be able to log in the system and the Bitcoin address will be displayed on the screen. In the blackmail case study, the communication between the trickster and the victim is also limited. They only communicate when the trickster sends this email to the victim. The victims are asked to pay the Bitcoin to the given address and no communication will happen between the trickster and victim.
2. Limited number of Bitcoin addresses: For both case studies, the hacker only provides a few Bitcoin addresses to the victim. The victims are asked to create new transactions in the Bitcoin blockchain and use the Bitcoin address given as the output public key. The hacker does this to limit the communication needed as well as to save the amount of works needed as the hacker will need to deal with many victims.
3. High Successors Out-degree for money laundering: In both case studies, we have seems the hacker laundered the "black Bitcoin" by spreading them vertically and horizontally across the transaction network. In the current state of the art, out-degree, which is the number of outputs in a transaction, is being used as one of the features in the model. As discussed in Chapter 2, the out-degree may not be appropriate or enough for the detection of Bitcoin fraud as it only represents the number of receivers of the transaction. In both case study, as the victim are required to transfer the Bitcoin to one of the addresses, the transaction is likely to have one to two outputs. (One output for the trickster and one output for changing.) The out-degree of the transaction is in fact not enough to separate the transaction from regular payment. However, from the Wannacry case study, we have noticed that the out-degree of the successors may be a good evidence of

money laundering as the hacker or trickster needs to spread the Bitcoin vertically to efficiently confuse the tracker and hence buys time for himself.

4. High User-Indegree: As suggested, the trickster tends to make victims to transfer the Bitcoins to a limited number of addresses. Hence, a successful fraudulent address will tend to have a high user in-degree. In other word, if the fraud is successful, each of the addresses will receiver Bitcoins from many different users and hence will have a high user-indegree.

3.134 Features extraction for the model

Similarly to those recently published papers, we will also construct a transaction-network model and a user-network model for this research:

User-Network Features

- User In-degree: The number of addresses that sent Bitcoin to the current address.
- User Out-degree: This is the number of addresses that received Bitcoin from the current address. As discussed, if the fraud is successful, a great number of UTXOs will be created in the blockchain and they can only be ‘unlocked’ or transferred by the private key owned by the trickster. The trickster can either transfer all these UTXOs to one or many new Bitcoin addresses. This will affect the value of user out-degree.
- User Money-Laundering-Index/: From the two case studies, we have found that the hacker launders the Bitcoin by spreading the Bitcoin across the entire transaction network. This is being used as evidence of money laundering. To capture this feature, for each Bitcoin user, we count the number of UTXO received by the hacker that has a successor with extremely high out-degree.

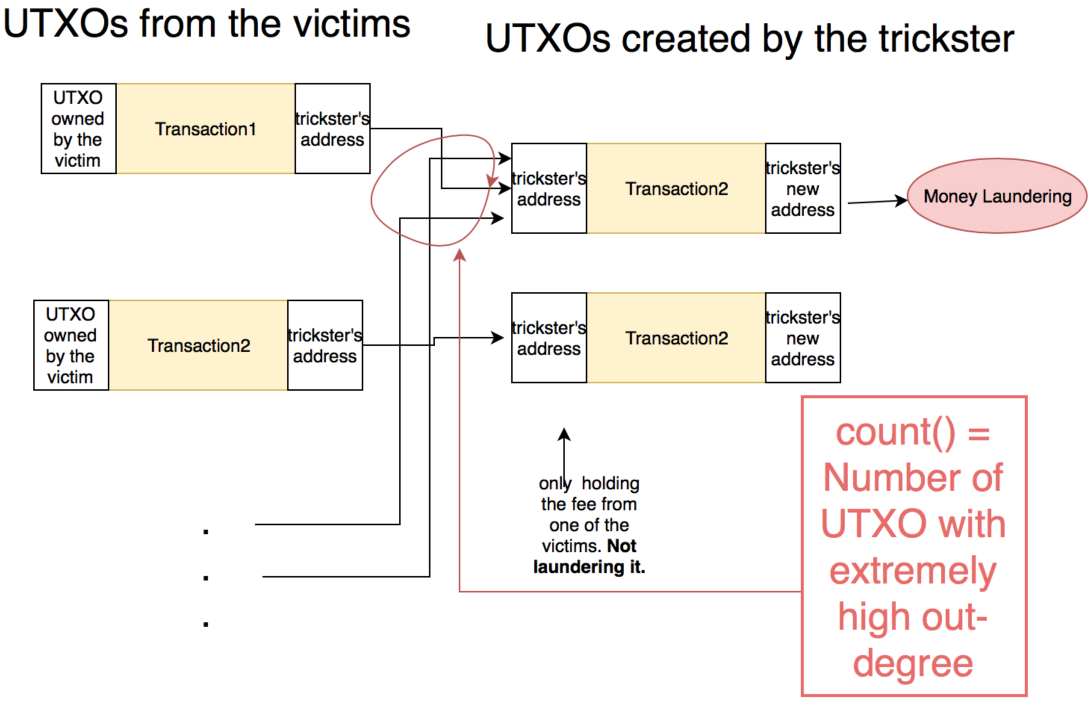


Figure 3.5 How the number of transactions, received by the trickster's address, with extremely high out-degree is counted

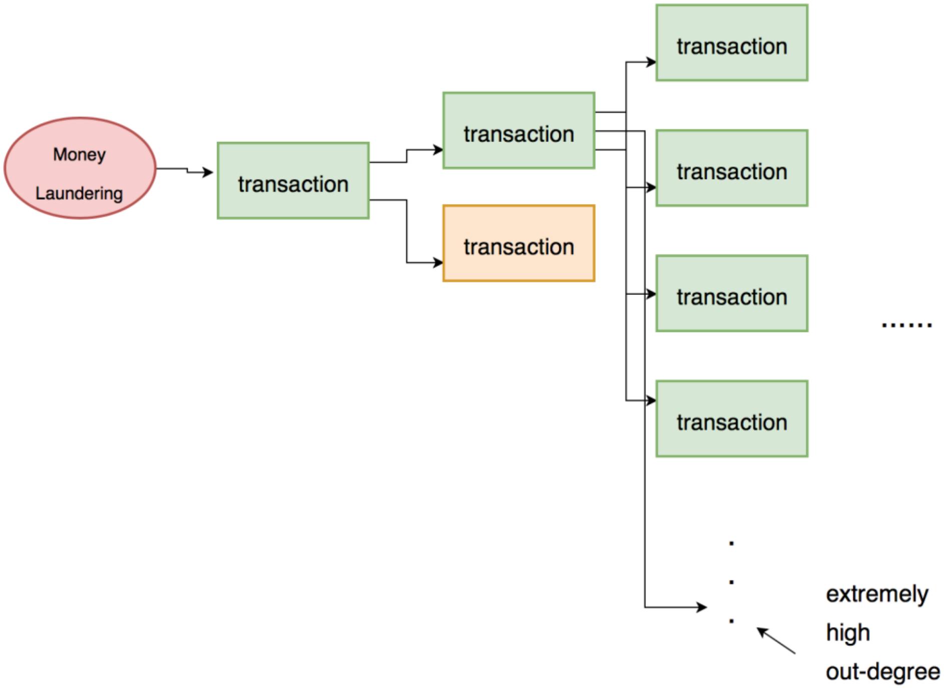


Figure 3.6 How Wannacry Money Laundering works

Transaction-Network Features

- Transaction In-degree: The number of inputs a transaction has. From a domain-knowledge perspective, it is difficult to tell if this feature is correlated to Bitcoin fraud. In general, if the victim needs to pay a lot of Bitcoins to the hacker, he may need to pay with multiple addresses. On the other hand, if the victim has never used Bitcoin before, he may only pay with one address(he buys Bitcoin from other users and hold all these Bitcoin with only one address)
- Transaction Out-degree: The number of outputs a transaction has. Similarly to transaction in-degree, it is difficult to tell if this feature is correlated to Bitcoin fraud. In general, if the hacker may try to launder the Bitcoin by having extremely high out-degree.
- Transaction Money-laundering-index/Number of immediate successor transactions that have successor transactions with extremely high out-degree: Similarly to user-network feature, this feature is also inspired by the money laundering evidence of both case studies. As mentioned, the hacker laundered the Bitcoins by having extremely high out-degree in the successor transactions. This feature is intended to extract this characteristic by counting the number of immediate successor transactions(the transactions that receive Bitcoin from the current transaction) that have successors with extremely high out-degree. In here, we also set 1.5 times of the average out-degree as the threshold.

3.14 Data Preprocessing

With feature extraction, meaningful information or features are being extracted from the Bitcoin Ledger Data. Such feature data may need to be preprocessed depending on the machine learning algorithms into which it fits.

For this research, k-means clustering and unsupervised SVM are the two algorithms we are implementing. Normalization is required for both algorithms. It refers to the adjusting of the value of each feature, which is on different scales, to a notionally common scale. [15] For example, if the scale of feature1 is much larger then feature2, for distance-based algorithms such as clustering and Euclidian distance, the feature with the greater scale will dominate the model. To avoid this, we need to make the features to have the same scale. In this research, we normalize the data with min-max scaling:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

where

x is the value of the current instance
x_{norm} is the normalized value of the current instance
x_{max} is the value of the largest instance
x_{min} is the value of the smallest instance

3.15 Modeling

3.151 Algorithms

As mentioned, we will be using k-means clustering for the clustering analysis and one-class SVM for the anomaly detection in this research.

3.152 Hyperparameter Optimization/Tuning

Hyperparameter optimization is required for both models. For K-means clustering, the hyperparameter k will be determined through elbow method. For one-class SVM, we need to determine the hyperparameter v through dual evaluation.

3.16 Evaluation

As mentioned, unlike supervised learning where you can measure the performance of the model through metric such sum of square error and accuracy, you are not able to measure the performance of unsupervised learning since there is no target/label for you to compare. Hence, the performance of unsupervised learning algorithms can only be measured through visualization evaluation. A group of experts from a variety of backgrounds will be brought together to determine the performance of the unsupervised learning model. [9] For this research, the performance will be evaluated by me and my supervisor, Donal O'Mahony.

3.2 Architecture

3.21 Technology Overview

In this section, we will explore the technologies implemented for the construction of the Bitcoin fraud detection system.

3.211 Apache Hadoop

Apache Hadoop is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. In this research, we have constructed a Hadoop server to store our data. The data in the Hadoop cluster is being loaded by our application to perform data parsing, feature extraction, data preprocessing and modeling. [30]

3.212 Apache Spark

Apache Spark is a fast and general-purpose cluster computing system. [31] In this research, it will be used in cooperating with Apache Hadoop to perform feature extraction on the Bitcoin ledger data in parallel. The extracted features data will be stored back to Hadoop.

3.213 R

R is a software environment for statistical computing and graphics. [31]In this research, it is being used to preprocess the dataset and construct the model. [32]

3.22 Application architecture

Given the approaches described in the approach section and the technologies specified in the technology overview section, application to address the research objectives can now be designed. For each of the approach specified in the approach section, individual components need to be constructed to fulfill the corresponding task. The architecture is as follows:

- Data Gathering:
  - The blockchain ledger data will be downloaded with Bitcoin Core.
  - Bitcoin core will store the data in a temporal data store.
  - The data will then be transferred to the Hadoop server
- Data Parsing:
  - The spark application will be loaded the Bitcoin ledger data from Hadoop
  - The ledger data will be parsed so that the transaction data will be converted from binary to SQL tables.
  - The table will be stored back to the Hadoop server
- Feature Extraction:
  - The spark application will load the table from the Hadoop server

  - Feature extraction will be applied to the ledger table so that the meaningful features will be extracted from the transaction data
  - The feature table will be stored in the Hadoop server
- Pre-processing:
  - R application will load the feature data from the Hadoop server and pre-process the feature data
  - The pre-processing required is normalization
- Modeling:
  - The pre-processed data will be modelled with unsupervised learning algorithms
- Evaluation
  - The model will be visualized. Domain knowledge will be used to determine the pattern recognized that are relevant to fraud.

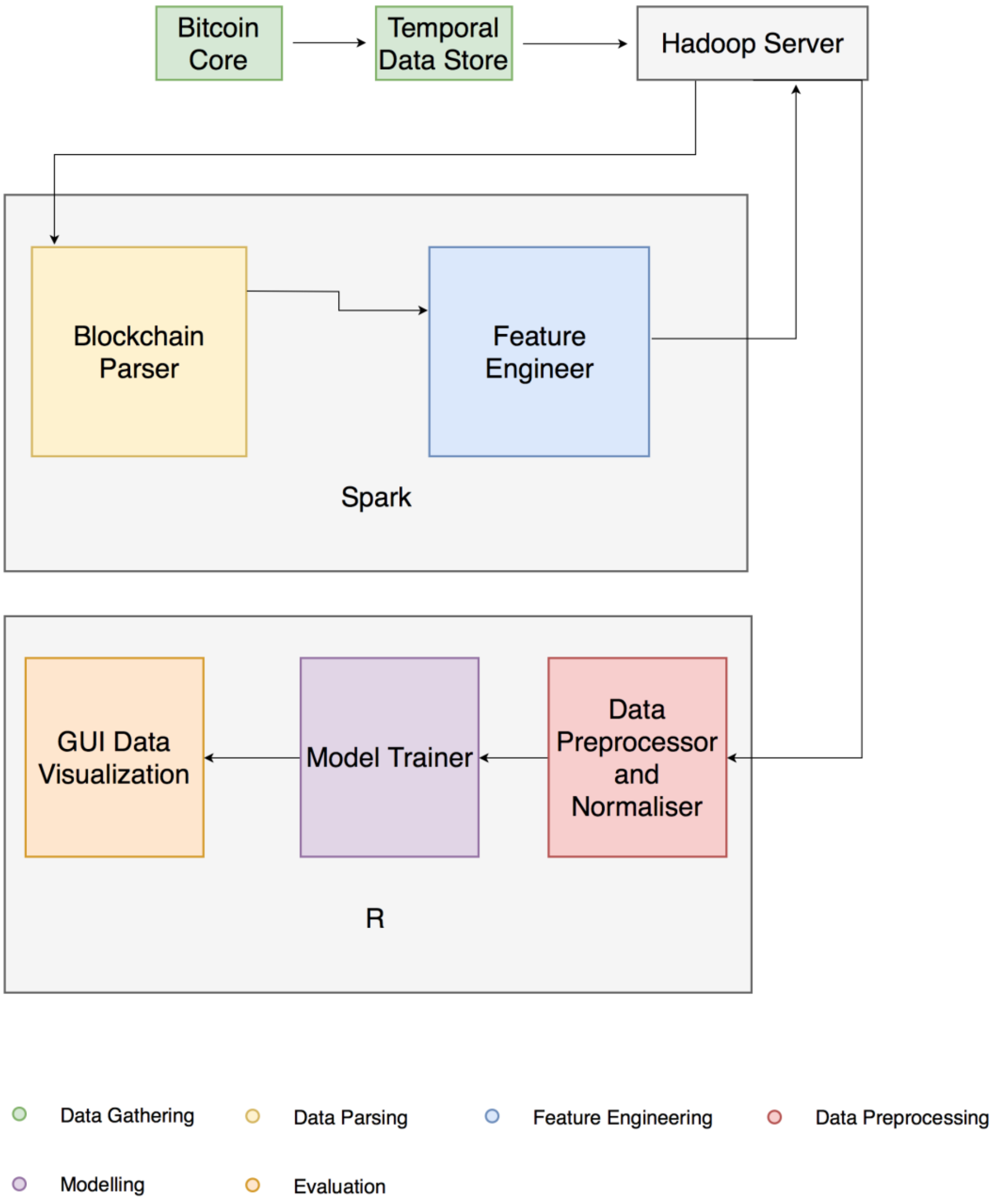


Figure 3.7 Architecture of the system

Chapter 4

Implementation

In the previous chapter, we have provided an in-depth description of the design of this research. This included the approaches of the experiment as well as the architecture of the application that will be implemented to carry out the approaches designed. A detailed description is also provided for the feature extraction that will be implemented. This included the reason behind it along with the case studies that have been studied. In this chapter, we will discuss the implementation of this research. A detailed description of how the approaches are carried out will be provided in this chapter.

4.1 Data Gathering and Parsing

4.11 Data Gathering

As discussed, Bitcoin Core is used to gather the Bitcoin ledger data. The data is stored in the temporal storage and is then transferred to Hadoop server.

4.12 Data Parsing

The spark application then loads the blockchain ledger data from the Hadoop server and parsed it. The parsed transactions data are stored back to the Hadoop server

4.2 Feature Extraction

There are over billions of Bitcoin addresses(user) and over 300 million of Bitcoin transactions stored in the Bitcoin ledger. The feature extraction process is, in fact, very expansive. To extract the money-laundering-index feature, for each of these Bitcoin address and transactions, we will need to search repetitively and recursively(will discuss in more detail) for all the successor transactions, from over 300 million transactions. Due to the time limitation and high computational cost, it is impossible to process this data within the time given. Hence, we sampled 1000 Bitcoin addresses and 1000 Bitcoin transactions in 2016 and experimented them in this research.

4.21 User Network

4.211 User In-degree

The user in-degree data are collected through the following steps:

- For each of these Bitcoin address
  - Search for the transactions where the output public key is the current Bitcoin address.
  - For each of these transactions:
    - Gather the input transaction hash
    - Search for all the transactions with the transaction hash
    - Get the output public addresses of these transactions, this is the **in-address** of the current address
  - Count the number of in-addresses

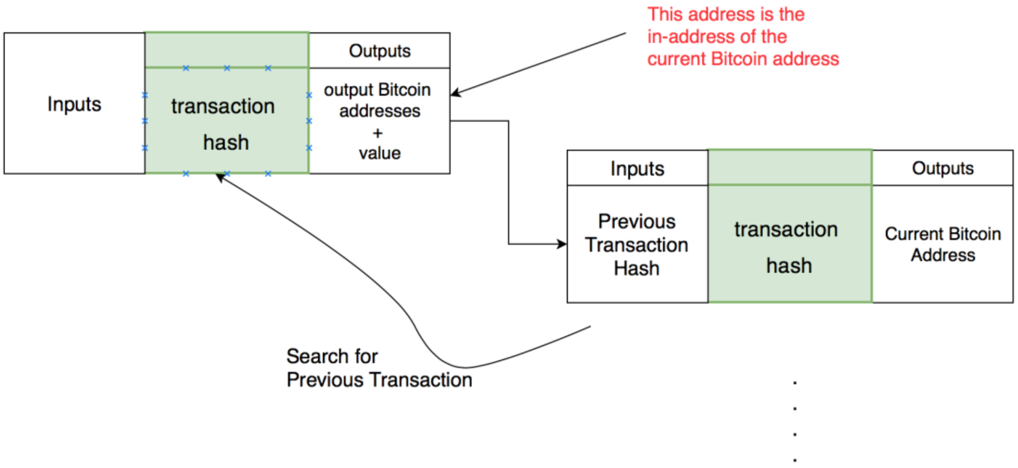


Figure 4.1 Find out in-degree of a Bitcoin address

4.212 User Out-degree

The user out-degree data are collected through the following steps:

- For each of these Bitcoin address
  - Search for the transactions where the output public key is the current Bitcoin address.
  - For each of these transactions:
    - Gather the transaction hash of the current transaction
    - Search for all the transactions where the current transaction hash is its input transaction hash
    - Get the output public addresses of these transactions, this is the **out-address** of the current address
  - Count the number of out-addresses

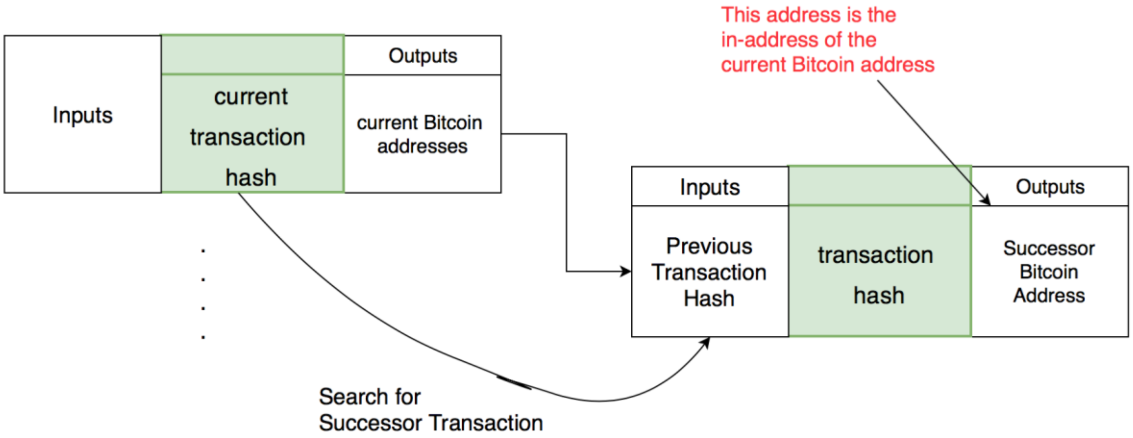


Figure 4.2 Find out in-degree of a Bitcoin address

4.213 User Money-Laundering-Index/Number of UTXO received by an address that has a successor with extremely high out-degree

The feature data are collected through the following steps:

- For each of these Bitcoin address
  - Search for the transactions where the output public key is the current Bitcoin address.
  - For each of these transactions:
    - Search all the successor transactions recursively
    - A depth is set for this recursively searching. We can treat it as a tree search. For transactions that are too deep, we can assume that they are not relevant to the current transaction or the current Bitcoin address
    - During the searching, if there is a transaction with extremely high outdegree, stop the searching and plus one to the counter of the current Bitcoin address
  - The counters of each address are the data looked for.

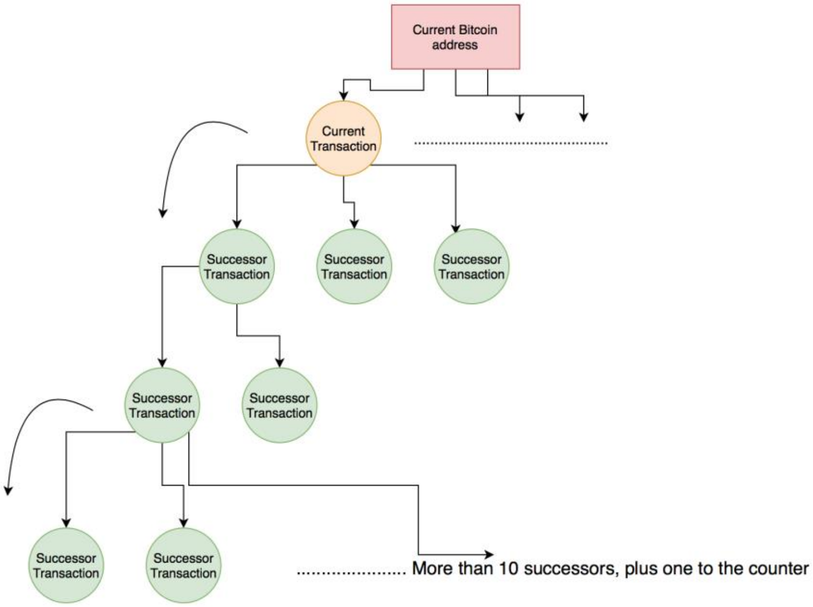


Figure 4. 3 Find out the number of UTXO received by an address that has a successor with extremely high out-degree

4.22 Transaction Network

4.221 Transaction In-degree

The transaction in-degree data are collected through the following steps:

- For each transaction:
  - Count number of inputs of the transaction

4.222 Transaction Out-degree

The transaction in-degree data are collected through the following steps:

- For each transaction:

  - Count number of outputs of the transaction

4.223 Transaction Money-Laundering-Index/Number of immediate successor transactions that have successor transactions with extremely high out-degree

The feature data are collected through the following steps:

- For each transaction
  - Search for the transactions where the inputs of transaction referenced the TXID of the current transaction. These are the immediate successor transactions.
  - For each of these immediate successor transactions:
    - Search the successor transactions recursively
    - A depth is set for this recursive searching. We can also treat it as a tree search, similarly to the approaches outlined in the user features section. For transactions that are too deep, we can assume that they are not relevant to the current transaction or the current Bitcoin address
    - During the searching, if there is a transaction with extremely high outdegree, stop the searching and plus one to the counter of the current transaction
- The counters of each transaction are the data looked for.

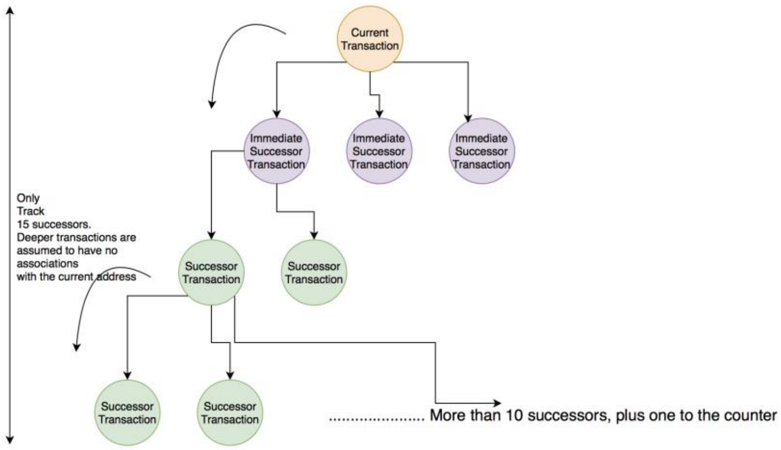

Current Transaction
Immediate Successor Transaction
Immediate Successor Transaction
Immediate Successor Transaction
Only Track 15 successors. Deeper transactions are assumed to have no associations with the current address
Successor Transaction
Successor Transaction
Successor Transaction
Successor Transaction
.... More than 10 successors, plus one to the counter

Figure 4.4 Find out the number of immediate successor transactions that has a successor with extremely high out-degree

4.3 Data Preprocessing

The feature table is being loaded from the Hadoop server and the data are being normalized.

4.4 Modeling

The normalized data are fitted to the k-means clustering model and the one-class SVM model.

4.41 K-means clustering

4.411 Hyperparameter Optimisation

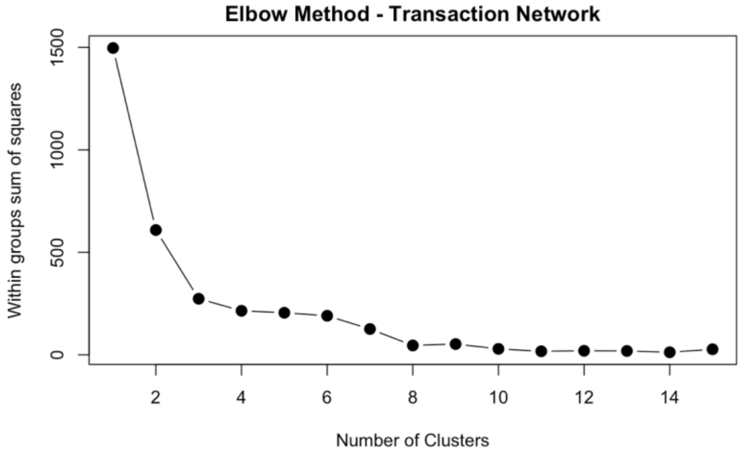


Figure 4.5 elbow method to determine the optimal k for transaction network

From the elbow method, as the sum of square error of the clustering model stops decreasing dramatically at around 4. Based on this, we determined that the optimal value of k for transaction network clustering model is 4.

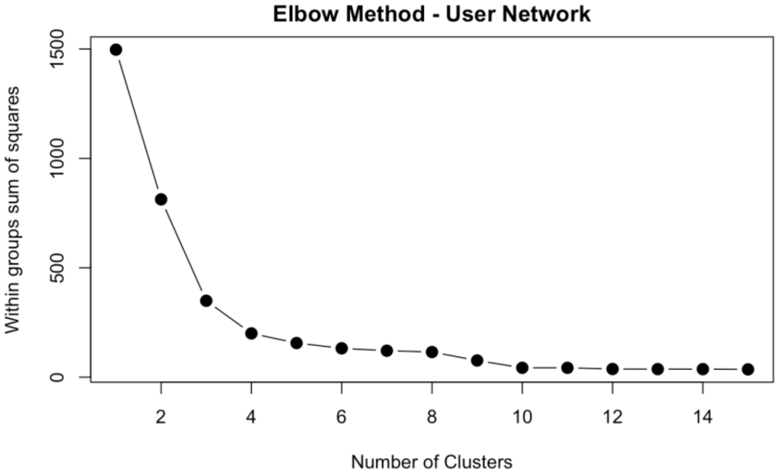


Figure 4.6 elbow method to determine the optimal k for user network

Similarly to the transaction network, from the elbow method, as the sum of square error of the clustering model stops decreasing dramatically at around 4, we also determined the optimal value of k for user network clustering model to be 4.

4.42 One-class SVM

4.421 Hyperparameter Optimization

The hyperparameters we need to determine are kernel function, gamma and v. Through tuning. They are set to be the following values:

Hyper-parameter	Kernel function	Gamma	V(nu)
value	radial basis function	1/size of dataset	0.14

Chapter 5

Evaluation

5.1 K-means clustering

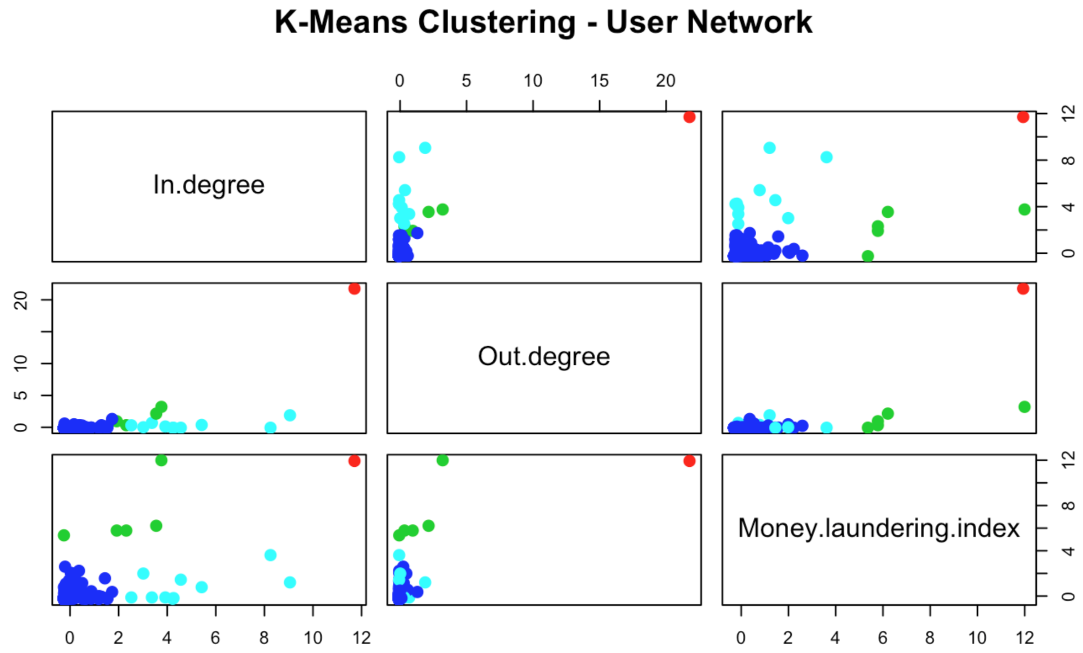


Figure 5.1 K-means Clustering – User Network

In general, the clustering model for the user network is successful. We have successfully clustered the dataset into 4 clusters. The blue cluster is a cluster that has a low in-degree, low out-degree and low money-laundering index. From a domain knowledge perspective, these are the addresses owned by regular users as the addresses received Bitcoin from a small number of addresses and transferred them to small number of addresses. The light blue cluster has a relatively high in-degree, low out-degree and low money-laundering index. From a domain knowledge perspective, these addresses could be owned by Businesses who have large number of Bitcoins from their customers. Low money-laundering-index also indicates that they were not trying to launder the Bitcoin they have. The green cluster has a medium in-degree, low out-degree and a relatively high money laundering-index. They could be the Bitcoin addresses owned by a blackmail hacker. The fraud was not successful so that the value of in-degree and out-degree are medium but there is evidence of money laundering. Finally, the red cluster is a cluster of high risk, it has a high in-degree, a high out-degree and a high money-laundering index. In fact, it was found that the red instance in the graph is one of the Bitcoin address owned by the Wannacry hacker. All the detected users will need to be confirmed and checked by human manually to determine if these users are hackers.

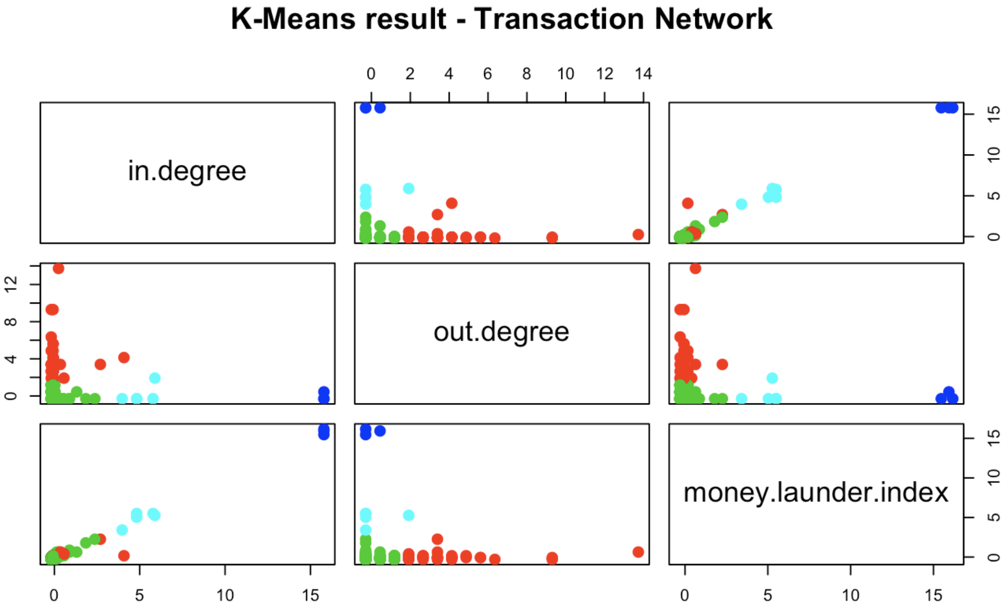


Figure 5.2 K-means Clustering – Transaction Network

It is very difficult to determine if the clustering model is successful or not. This is because the transaction network feature may not be appropriate for the detection of Bitcoin fraud. The transaction in-degree and in-degree refers to the input size and output size of a transaction. As discussed, there is no evidence to justify the assumption that a fraudulent transaction will have a high/low in-degree or out-degree. However, the money laundering index refers to number of immediate successor transactions with successor transaction that have extremely high out-degree. In order word, the number of immediate successor transactions that show evidence of money laundering. In this respect, the blue cluster may be a high-risk class as the transaction has a high money-laundering index. All the detected transactions will need to be confirmed and checked by human manually to determine if the user is a hacker as there is a great possibility that these transactions are owned or controlled by the hackers.

5.2 One-class SVM

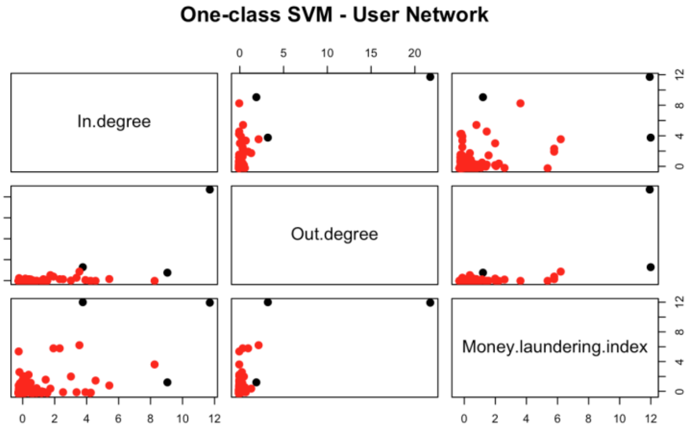


Figure 5.3 One Class SVM – User Network

In general, the anomaly detection model for the user network is successful. The red instances are belonging to the +1/normal class and the black instances are belonging to -1/anomaly class. The model successfully detected high-risk users which includes the wannacry Bitcoin address. All the detected users will need to be confirmed and checked by humans manually to determine if the user is a hacker.

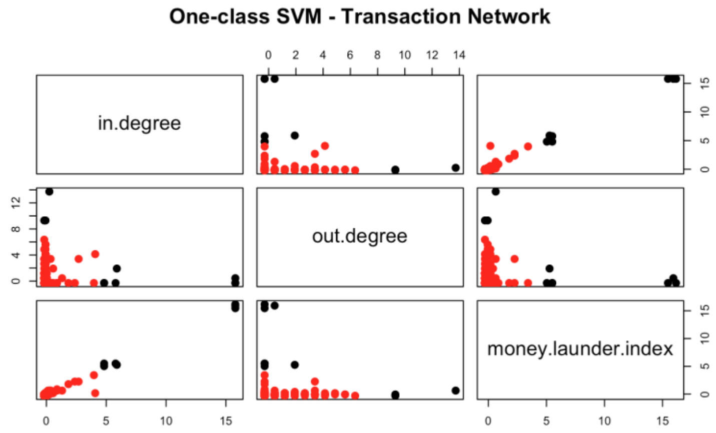


Figure 5.4 One Class SVM – Transaction Network

Similarly to the clustering model of the transaction network. It is extremely difficult to determine if the in-degree and out-degree features are appropriate for the detection of Bitcoin fraud. However, the model successfully identifies the transactions with high money laundering index. All these detected transactions will need to be confirmed and checked by humans manually to determine if the user is a hacker. There is a great possibility that these transactions are owned or controlled by the hackers.

Chapter 6

Conclusion

6.1 Project Overview

In general, we have successfully recognized some patterns for the detection of Bitcoin fraud. For the user network, with both clustering model and anomaly detection model, we have successfully identified one of the Bitcoin address owned by the Wannacry hacker, along with some other high-risk users. For any Bitcoin addresses that are belonging to this high-risk group, we should double check the address manually and act accordingly. For the transaction network, we also identified a cluster with high money-laundering-index. There is a great possibility that these transactions are owned or controlled by the hackers. Again, these transactions should be double-checked and monitored by a person manually.

Regarding the downside of this research, I have to conclude that the lack of labeled data is still a major bottleneck in the research field of Bitcoin fraud detection. It is extremely difficult for us to have a fair and measurable judgment to the performance of the model since the measurement is based on human opinion, although the person may be an expert in the field. Lack of data is also preventing us from constructing an effective fraud detection system. As discussed, academic researchers have too little information about the Bitcoin transactions and the Bitcoin users. The only accessible data for them is the Bitcoin ledger. Moreover, the anonymous nature of Bitcoin is preventing them from fully understanding the Bitcoin transaction ledger. In contrast, industrial actors such as

Coinbase have a considerable advantage in this and hence the fraud detection system constructed by them is much more effective.

From a learning perspective, I have successfully achieved the aim of this research. I have gained a great understanding of Bitcoin which include Bitcoin transaction network and Bitcoin Blockchain. For Bitcoin transaction system, I have gained an in-depth understanding of concepts including Bitcoin address, Bitcoin transaction, UTXO and more. For Bitcoin blockchain, I have learned the structure of blockchain, the use of it, how Bitcoin mining works and more. I also gained a good knowledge machine learning. This includes clustering analysis algorithms such as K-means clustering and anomaly detection algorithms such as One-class SVM. For fraud detection, I have read papers and articles of many subfields of financial fraud detection. These include the credit card fraud detection, Bitcoin fraud detection in industry and Bitcoin fraud detection in Academia. All these knowledge learned have contributed to this research.

6.2 Future Work

Many possible improvements have been left due to the lack of time. In summary, following improvement can be made:

- Experiment the system with more data: Due to the time limitation, Only 1000 transactions and users sampled and are experimented with the system constructed. In the future, we should experiment the system with more transaction and user data so that we can recognize the patterns of more transactions and users.
- Study more Bitcoin fraud case study: Due to time limitation, we only studied two Bitcoin fraud case studies in-depth. In the future, we should study more case studies and try to create more new features that are correlated or relevant to Bitcoin fraud.
- Collecting label data: The biggest problem with Bitcoin fraud detection is the lack of labeled data. In the future, if with sufficient amount of time, we can try

to gather Bitcoin fraud data so that we can construct a labeled Bitcoin transaction dataset. This will make the fraud detection system much more effective.

Bibliography

[1] "Financial fraud losses in the UK last year topped £20m a day – report," [Online]. Available: https://www.theregister.co.uk/2017/03/30/uk_financial_fraud_loses_grow/.

[2] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2009.

[3] "Bitcoin Official Guide," [Online]. Available: https://bitcoin.org/en/developer-guide.

[4] "Bitcoin Wiki," [Online]. Available: https://en.bitcoin.it/wiki/.

[5] "Ethereum White Paper," [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper.

[6] A. Géron, Hands-On Machine Learning with Scikit-Learn & TensorFlow, Aurélien Géron, 2017.

[7] A. Ng, Machine Learning and AI via Brain simulations, Andrew Ng.

[8] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, Ian Goodfellow; Yoshua Bengio; Aaron Courville, 2016.

[9] H. Drucker, D. Wu and V. Vapnik, "Support vector machines for spam categorization," 1999.

[10] Hearty, John, Julian, David, Raschka and Sebastian, Deeper Insights into Machine Learning.

[11] L. Mika and H. Yrjö, "Anomaly dRecognition of Systematic Spatial PaKt-tmerenasnisn Silicon Wafers Based on SOM and Recognition of Systematic Spatial Patterns in Silicon Wafers Based on SOM and K-means Copyright © 2018 IFAC C2o40p5yetection using baseline and K-means clustering," *IFAC*, 2016.

[12] O. J. Oyelade, O. O. Oladipupo and I. C. Obagbuwa, *Application of k-Means Clustering algorithm for prediction of Students' Academic Performance.*

[13] "Hyperparameter Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Hyperparameter_(machine_learning).

[14] "Hyperparameter Optimisation Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Hyperparameter_optimization.

[15] P. Bholowalia and A. Kumar, "EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN," *International Journal of Computer Applications ,* 2014.

[16] P. Dangeti, Statistics for Machine Learning, 2017.

[17] "Microsoft Azure - One-class SVM," 24 01 2018. [Online]. Available: https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/one-class-support-vector-machine.

[18] B. Scholkopf, R. Williamson, A. Smola, J. Shawe-Taylort and J. Platt, "Support Vector Method for Novelty Detection".

[19] L. Kunlun and T. Guifa, Unsupervised SVM Based on p-kernels for Anomaly Detection, Kunlun Li; Guifa Teng, 2006.

[20] S. Ben-David, R. Urner, a. U. v. Luxburg and M.-F. Balcan, "Foundations of Unsupervised Learning," 2016.

[21] J. O. Awoyemi, A. O. Adetunmbi and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis".

[22] "Coinbase," [Online]. Available: https://www.coinbase.com.

[23] C. Soups Ranjan, "Conf.Startup.ML: Fraud Detection in Bitcoin Payment Networks," [Online]. Available: https://www.youtube.com/watch?v=tCNWnrvIoJo.

[24] D. Zambre and A. Shah, "Analysis of Bitcoin Network Dataset for Fraud," 2013.

[25] F. Reid and M. Harrigan, "An Analysis of Anonymity in the Bitcoin System," 2011.

[26] P. T. Pham and S. Lee, "Anomaly Detection in the Bitcoin System - A Network Perspective".

[27] P. Monamo, V. Marivate and B. Twala, "Unsupervised learning for robust Bitcoin fraud detection," *Information Security for South Africa (ISSA).*

[28] T. T. Pham and S. Lee, "Anomaly Detection in Bitcoin Network Using Unsupervised Learning Methods," *arXiv,* 2017.

[29] K. Ruohonen, Graph Theory, 2013.

[30] O. Babaoglu, Network Science: Graph Theory, Dipartimento di Informatica — Scienza e Ingegneria.

[31] [Online]. Available: https://en.wikipedia.org/wiki/WannaCry_ransomware_attack.

[32] "WannaCry Addressess," [Online]. Available: https://www.reddit.com/r/Bitcoin/comments/6axuzs/wannacry_wcry_wannacrypt_bitcoin_addresses/.

[33] "'I know you cheated on your wife.' Growing blackmail scam demands payment in bitcoin," *CNBC,* 2018.

[34] "Blackmail Bitcoin address," 20 02 2018. [Online]. Available: https://myonlinesecurity.co.uk/attempted-blackmail-scam-watching-porn/.

[35] [Online]. Available: https://en.wikipedia.org/wiki/Normalization_(statistics).

[36] [Online]. Available: http://hadoop.apache.org.

[37] H. Karau, A. Konwinski, P. Wendell and M. Zaharia, O'Reilly Learning Spark, 2015.

[38] "R," [Online]. Available: https://www.r-project.org.

[39] G. Grolemund and H. Wickham, R for Data Science.

[40] V. D. S. A, Advanced support vector machines and kernel methods, V. David Sánchez A, 2003.

[41] A. Talwar and Y. Kumar, http://www.ijecs.in/issue/v2-i12/11%20ijecs.pdf, Anish Talwar; Yogesh Kumar, 2013.

[42] M. Jordan and J. B. S. Kleinberg, Pattern Recognition And Machine Learning, Jordan, M.; Kleinberg, Jlkopf, B. Scho, 2006.

[43] L. A. Garcia-Escudero and A. Gordaliza, "Robustness Properties of k Means and Trimmed k Means," 1999.

[44] "Bitcon Core," [Online]. Available: https://bitcoin.org/en/bitcoin-core/.

[45] L. C. ∗. C. C. A. Beghi ∗, M. Rampazzo, F. Simmini and G. Susto, " One-Class SVM Based Tool for Machine Learning Novelty Detection in HVAC Chiller Systems," 2014.

[46] P. M. Monamo, V. Marivate and B. Twala, "A Multifaceted Approach to Bitcoin Fraud Detection: Global and Local Outliers," *Machine Learning and Applications (ICMLA).*