Université d'enseignement

Université Lumière Lyon 2

Structure d'accueil

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

RAPPORT DE STAGE

BOUBACAR SIDIGUI SOW

M2 INFORMATIQUE ET STATISTIQUE PARCOURS DATA MINING

Utilisation de techniques de machine learning pour la détection de multiples adresses dans les réseaux de transactions en crypto-monnaies

Tuteur Académique
Professeur Julien JACQUES

Tuteur d'entreprise
Maître de conférence Rémy CAZABET

20 Août 2019

Remerciements

Je présente mes chaleureuses remerciements à mon tuteur professionnel, Maître de Conférence Rémy CAZABET, pour la chance d'apprendre qu'il m'a offerte, la confiance qu'il m'a accordée, mais surtout pour le temps qu'il a sacrifié, les critiques qu'il m'a apportées au cours de ce stage et toutes les connaissances qu'il a partagées avec moi.

Je tiens à remercier aussi mon tuteur académique, le Professeur Julien JACQUES pour sa disponibilité et son encadrement. Je le remercie d'avance pour sa lecture de ce rapport de stage et pour les critiques qu'il va m'apporter.

Enfin je remercie l'équipe des étudiants qui ont participé à la collecte de données utilisées dans ce rapport,et toutes les personnes qui ont contribué au succès de ce stage.

Résumé

Ce papier est un rapport de stage. Il a été réalisé au Laboratoire LIRIS. Il s'est déroulé du 01 mars 2019 au 31 Juillet 2019. Au cours de ce stage, nous avons étudié les transactions de la Blockchain de Bitcoin de la période de Janvier 2009 à Septembre 2016. Nous avons d'abord cherché à associer toutes les adresses de la blockchain à un acteur. Nous expliquons comment nous avons fait ce groupement d'adresses et nous décrivons les clusters obtenus. Par la suite nous construisons et étudions le graphe qui combine les utilisateurs, les adresses et les transactions.

Table des matières

1 Introduction

263 352 035 050 dollars, ce chiffre est le total de la capitalisation boursière des 2475 cryptomonnaies existantes à ce jour, d'après les calculs de CoinMarketCap au 23 Août 2019. Tandis que la première cryptomonnaie, Bitcoin, a vu naissance le 03 Janvier 2009, jour lequel la première transaction a été réalisée par Satoshi Nakamoto [**?**] son créateur. D'ailleurs bitcoin est la plus importante. En moins de 10 ans, le nombre de transactions en bitcoin a grimpé à 447 millions (dont 310166 par jour en moyenne) , le prix de vente d'un bitcoin à 10 000 dollars, la capitalisation de boursières de bitcoin représente 68% du total de toutes les crypto-monnaies. Ces chiffres montrent l'importance qu'a pris les crypto-monnaies, particulièrement bitcoin.

L'histoire de bitcoin a commencé en 2008, au moment de la crise financière, quand le mysterieux Satoshi Nakamoto a publié son article *Bitcoin : A Peer-to-Peer Electronic Cash System* en ligne. Dans cet article il montre qu'il a réussi à mettre en place un système de payement électronique en ligne de pair à pair sans la nécessité d'un intermédiaire financier pour vérifier et valider les payements. Un système qui résout le problème de la dépense double (*double-spending*). La solution qu'il propose s'appelle la blockchain. Brièvement, la blockchain est une base de donnée qui enregistre toutes les transactions effectuées pair-à-pair, et immunisée de toute falsification grâce à des fonctions cryptographiques. Cette base de donnée est accessible à tout individu. En outre, tous les individus de la blockchain sont anonymes. Bitcoin est la crypto- monnaie utilisée dans les transactions de ce système.

Vu l'intérêt que suscite les crypto-monnaies, particulièrement bitcoin, d'un côté, de nombreuses critiques négatives ont été soulevées au sujet de la nature des transactions, de l'anonymat dans la blockchain de bitcoin. Les auteurs de ces critiques accusent la blockchain d'être un abri d'activités illégales (blanchiment d'argent, financement de terrorisme, ...). De l'autre côté, de nombreuses critiques sont positives, pour ces derniers bitcoin est une source d'épargne, bitcoin procure une liberté envers le système financier traditionnel, il permet de se prémunir contre l'inflation et l'instabilité financière.

Étant donné que les données des transactions sont disponibles, on peut les analyser et faire la part des choses. On peut catégoriser et quantifier la nature des transactions, on peut identifier les acteurs impliqués dans des activités illégales, on peut connaître la distribution de richesses et identifier les acteurs les plus riches dans la blockchain de bitcoin, etc. On peut aussi tester jusqu'à quel point les acteurs de la blockchain sont anonymes. Dans la littérature, de nombreux auteurs tentent d'apporter des éléments de réponses à ces questions.

Contexte du stage

Ce stage est un stage de recherche. Il est le point de départ d'un projet de recherche intitulé BITUNAM, dirigé par le Maître de conférence CAZABET Rémy. BITUNAM est financé par l'Agence Nationale de Recherche (ANR) et se déroule en

France. Plus précisément, c'est un projet de thèse qui a pour ambition de développer des méthodes de machine learning combiné à l'analyses de graphes pour comprendre la nature des activités dans les crypto-monnaies.

Entreprise d'accueil

Ce stage s'est déroulé au Laboratoire d'InfoRmatique en Image et Systèmes d'information (LIRIS) à Lyon. Le LIRIS est une "Unite mixte de recherche (UMR 5205), il est porté par le CNRS, l'INSA de Lyon, l'Université Claude Bernard Lyon 1, l'Université Lumière Lyon 2 et l'Ecole Centrale de Lyon. Il compte 330 membres, et a pour principal champ scientifique l'Informatique et plus généralement les Sciences et les Technologies de l'Information." Ce Stage a été supervisé par le Maître CAZABET Rémy qui fait partie de l'équipe Data Mining et Machine Learning (DM2L).

Plan du rapport de stage

L'étude que nous avons réalisée se divise en trois parties. Dans la première partie nous expliquons le fonctionnement de la blockchain de bitcoin, nous décrivons ses acteurs majeurs et d'autres notions nécessaires à la compréhension de l'analyse des données que nous avons effectuée telles que les heuristiques. Dans la deuxième, nous présentons un état de l'art sur le data mining des données de bitcoins. Notre objectif de cette revue de littérature est de montrer au lecteur que l'on peut extraire une variété d'informations dans les données de la blockchain. La troisième et dernière partie présente l'analyse de données que nous avons accomplie. Dans cette partie nous décrivons les données à notre disposition, nous expliquons la méthodologie que nous avons adopté pour réaliser le groupement d'adresses et nous présentons les résultats obtenus. Nous expliquons également comment nous avons construit le graphe des transactions, des adresses et des utilisateurs de bitcoins, et nous expliquons comment en soustraire des informations sur les activités de la blockchain.

2 La Blockchain de Bitcoin

Dans cette section, nous expliquons et définissons les éléments importants de la blockchain. Toutefois, nous ne rentrons pas dans les détails informatiques et cryptographiques, nous essayons seulement de fournir les informations qui nous semblent essentielles pour comprendre le fonctionnement du réseau bitcoin et la blockchain dans le but d'aider le lecteur à mieux comprendre le reste de ce rapport de stage. Pour commencer nous faisons la distinction entre blockchain et bitcoin.

2.1 Le Bitcoin

Bitcoin n'a pas de définition conventionnelle. Certains considèrent bitcoin comme une monnaie virtuelle (ou monnaie digitale ou crypto-monnaie), il peut être utilisé comme moyen d'échange contre des biens et de services. Mais il n'a pas de cours légal. Bitcoin est aussi considéré comme un actif numérique, qui peut être vendu ou acheté, il fait l'objet de spéculation financière dans de nombreuses places boursières dédiées aux cryptos-monnaies et il est réputé être très volatile. De même que la monnaie fiduciaire tel que l'euro, bitcoin est divisible, la plus petite unité de bitcoin est le *satoshi*. $1btc = 10^8$ satoshi. Mais au contraire, la création de nouveaux bitcoins n'est pas assurée par une autorité centrale, elle s'effectue grâce à l'activité de minage. En Avril 2019, la masse de bitcoins en circulation était d'environ 17 850 000 [2]. Par design, le nombre total de bitcoin ne pourra jamais dépasser 21 000 000 de Bitcoins. Le protocole de bitcoin est construit sur une blockchain. Sans l'existence de cette dernière, bitcoin n'a aucune valeur et aucune utilité. C'est la technologie blockchain qui est le garant de la confiance dans le réseau bitcoin. Elle permet et facilite les payements instantanés en bitcoin entre ses utilisateurs en toute confiance.

2.2 La Blockchain

Comme son nom l'indique (en anglais), la blockchain est une chaîne de blocs. Dans le cas du réseau de bitcoin, chaque bloc est un ensemble de transactions en bitcoins effectuées par les membres du réseau. Autrement dit, c'est une base de donnée digitale qui contient toutes les transactions électroniques effectuées entre ses utilisateurs. Elle est convoitée pour les propriétés suivantes :

- La blockchain est *transparente*, l'historique de toutes les transactions est accessible à toute personne, à tout moment.
- Elle est *décentralisée*, toutes les transactions dans la base de donnée sont vérifiées et validées sans l'implication d'un intermédiaire de confiance. C'est le premier système de payement décentralisé de pair-à-pair qui est capable de fonctionner sans le contrôle d'une autorité centrale. La blockchain n'appartient à aucune personne et n'appartient à aucune entité.
- Elle est *sécurisée* grâce à la cryptographie. La blockchain est infalsifiable et indestructible. Les transactions validées ne peuvent plus être invalidées, ni effacées de la base de donnée. En effet, plusieurs nœuds [1] du réseau détiennent une copie de toutes les transactions. La blockchain résout le problème de la double dépense ("double -spending"), c'est-à-dire, aucun acteur dans la blockchain ne peut effectuer plusieurs payements avec les mêmes bitcoins.
- Elle est *anonyme*, les acteurs dans la blockchain de bitcoin ne sont pas identifiés et ne peuvent pas être identifiés par leurs *personnalités juridiques*, mais plutôt par des adresses privées et publiques. Ces adresses sont des suites de

1. On appelle nœud, est un ordinateur qui détient une copie de toutes les transactions de la blockchain et est connecté au réseau de bitcoin.

caractères alpha numériques. Nous fournissons des détails sur ces adresses plus loin.

Notons que depuis la création de bitcoin, plusieurs monnaies virtuelles et plusieurs blockchains différentes ont vu naissance. Aujourd'hui il existe plus de 2000 monnaies digitales, un grand nombre parmi elles dispose de leur propre blockchain avec ses propres spécificités. Certaines blockchains sont privées et d'autres publiques. Grâce à ces qualités, décentralisée, sécurisée, anonyme, infalsifiable et transparente, la blockchain est vue comme une révolution technologique comparable à Internet. La technologie est adaptée dans plusieurs domaines autres que les monnaies digitales, notamment dans le domaine de l'agriculture, de la banque et finance, du vote, de la santé, le commerce, et même dans les médias [15]. Dans ce rapport, nous nous concentrons exclusivement sur la blockchain de bitcoin.

2.3 Fonctionnement, minage et création monétaire

Sans entrer dans les détails techniques, dans cette section nous résumons la construction de la chaîne de blocs dans le réseau de bitcoin.

Les mineurs et le minage

On sait qu'une transaction entre deux utilisateurs de bitcoins est valide grâce aux mineurs. Les mineurs sont les acteurs de la blockchain qui font du minage. N'importe quel acteur dans le réseau de bitcoin peut faire du minage. Le minage consiste à utiliser les processeurs d'une machine [2] pour effectuer des calculs mathématiques complexes afin de vérifier et de confirmer la validité des transactions entre les utilisateurs dans le réseau de bitcoin. Grâce à l'activité de minage, on est confiant qu'il n'y a pas de double dépense, et les signatures sont valides. En contrepartie de l'activité de minage, les mineurs sont rémunérés en bitcoins. D'ailleurs c'est la seule manière d'acquérir des bitcoins par essence (sans en acheter).

Comment ça marche

Toute transaction effectuée dans le réseau de bitcoin est diffusée à l'ensemble du réseau. Avant d'être confirmées, toutes les transactions non validées sont regroupées dans ce que l'on appelle *mempool* ou *memory pool*. C'est l'espace d'attente des transactions non confirmées, on peut voir l'arrivée de toutes les nouvelles transactions non confirmées en direct en suivant ce lien). Les mineurs choisissent parmi ces transactions, celles qu'ils souhaitent miner, le premier venu est le premier servi. Par design, la taille de l'ensemble des transactions choisies par un mineur doit former un bloc. Un *bloc* obéit à des règles cryptographiques strictes et ne peut pas dépasser 1MB de transactions. Le mineur valide le bloc qu'il a formé et informe les autres membres

2. Aujourd'hui il existe même des processeurs spécifiques au minage.

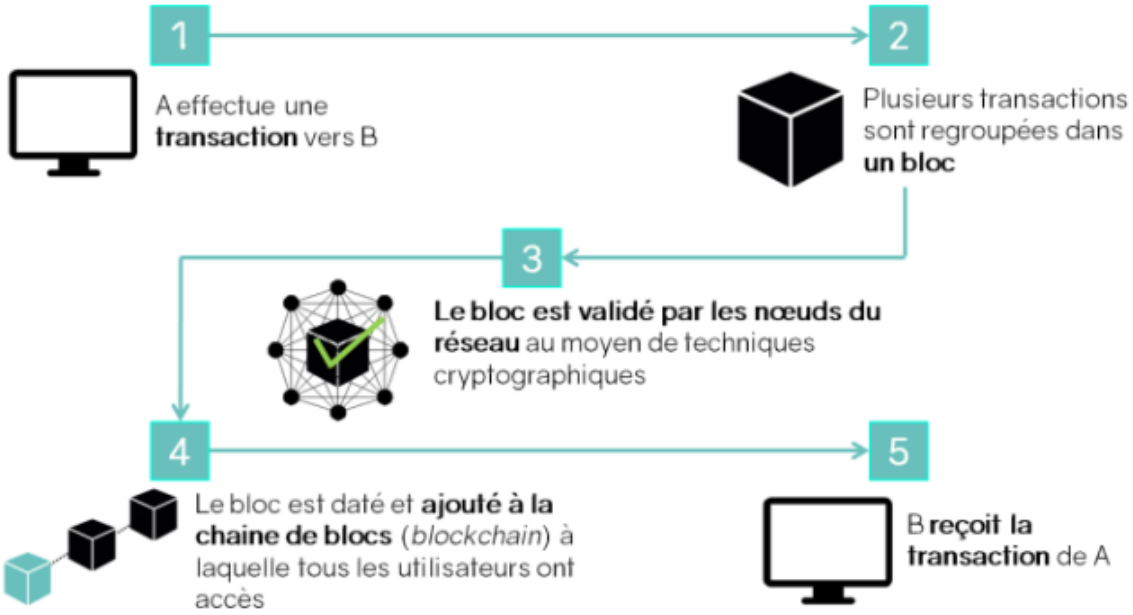

Schéma simplifié de la formation des chaînes de blocs (blockchain)Blockchainfrance

du réseau qui à leur tour vérifient la validité du bloc. Cette technique de vérification est appelée *proof of work*.

Valider un bloc consiste à résoudre un problème mathématique que l'on appelle *hash*. En effet, les mineurs utilisent un algorithme de hash, qui attribut le bon hash à toutes les transactions du bloc et lie ce bloc aux précédents blocs. Ils trouvent ce hash de manière aléatoire, plus le processeur de la machine d'un mineur est puissant en calculs plus sa probabilité de valider le bloc rapidement est grande. Si plusieurs mineurs, tentent de valider un même bloc ou des transactions communes au même moment, c'est le premier à trouver le bon hash qui gagne. De plus, le bloc nouvellement validé doit être lié à la chaîne la plus longue de la blockchain pour que les transactions qu'il contient soient définitivement valides et inscrites dans la blockchain. En réalité, un bloc peut être validé et ne pas figurer dans la blockchain s'il n'est pas associé à la plus longue chaîne. Il est estimé qu'après 6 nouveaux blocs, on peut être sûre qu'un bloc est vraiment validé par le réseau. En moyenne, on estime entre 10 et 20 minutes le temps de validité d'un bloc et donc d'une transaction. Pour pouvoir falsifier une transaction dans un bloc de la chaîne, il faudrait être capable de retrouver tous les bons hashs des blocs suivants. Sur la figure **??** est décrit un schéma simplifié du fonctionnemnt de la blokchain.

Rémunération des mineurs et création monétaire

Le choix des transactions à miner par les mineurs est déterminé par la rémunération qu'ils reçoivent dans les frais de transactions. Ils sont rémunérés de deux manières pour chaque bloc miné. La première façon est dictée par le design de la

blockchain, les mineurs sont récompensés pour le travail qu'ils font par une valeur de *nouveaux bitcoins* [3] générée et fixée par le protocole informatique de bitcoin. Cette valeur est directement envoyée au mineur après chaque validation d'un nouveau bloc. La valeur de rémunération baisse de moitié après 210 000 blocs minés, soit environ chaque 4 ans. Ainsi, au début du fonctionnement de la blockchain, entre 2009 et 2011, la valeur de la rémunération du minage d'un bloc était de 50 BTC. Après cette période, la récompense est devenue 25 BTC. Depuis 2016, elle est redescendue à 12 .5 BTC et elle restera ainsi jusqu'en 2020 puis passera à 6.25 BTC en 2021 [1]. Cette forme de rémunération constitue la "création monétaire" dans le réseau de bitcoin. Elle a pour conséquence une limitation du nombre total de bitcoin possible à 21 000 000.

Le deuxième type de rémunération des mineurs est les frais de transactions. Lors d'une transaction, l'utilisateur est libre de proposer le montant de rémunération qu'il souhaite au mineur. Ce montant peut être supérieur ou égale à 0 BTC. A cause du volume de transactions toujours croissant dans la blockchain, dans une transaction donnée, plus les frais de transaction sont élevés, plus vite la transaction sera choisie par un mineur et validée. A l'inverse, plus les frais de transactions sont très faibles, la transaction restera plus longtemps en attente de confirmation, elle risque même de ne jamais être traitée et jamais validée.

2.4 Transactions et Adresses dans la blockchain de bitcoin

Les adresses et l'anonymat

SzavMBLoXU6xDrqtUVmffv et 1g2YTUzygLMEC1jZkn3poypqZCbBXQAiR sont respectivement une adresse privée (au format réduit) et une adresse publique de bitcoin. Elles ont aussi la dénomination de clé privée et de clé publique. Les adresses privées permettent de gérer et d'avoir accès aux adresses publiques. Il est impossible que deux utilisateurs aient la même adresse. Les Bitcoins sont conservés dans les adresses publiques et les transferts de Bitcoins se font entre elles. Si un utilisateur perd sa clé privé, il perd l'accès à ses bitcoins de la clé publique associée. La clé publique est la seule identité d'un utilisateur dans la blockchain.

Tout utilisateur peut générer autant de pairs adresses privées/adresses publiques qu'il le désire. En conséquence chaque utilisateur peut disposer autant d'identités anonymes. L'anonymat que procure ces clés est très limité, certes il est possible de ne pas savoir à qui une clé appartient, mais on peut absolument identifier toutes les transactions effectuées par une adresse, connaître les montants des bitcoins qu'elle a envoyé, ou reçu. A titre d'illustration, c'est comme si on pouvait connaître l'infor-

3. On peut faire une comparaison entre la création monétaire de bitcoin et la création monétaire fiduciaire. La similarité des deux est le faite que la nouvelle monnaie est créée à partir de rien. Leur différence, est que pour Bitcoin cette création est fixée dans le temps, et n'est pas influencée par des facteurs externes. Tandis que la création monétaire fiduciaire est assurée par les banques commerciales et limitée par la Banque centrale.

mation contenu par le compte bancaire de quelqu'un, les euros qu'il possède, son relevé bancaire, etc, mais sans connaître le nom de la personne. Pour renforcer son anonymat dans la blockchain, il est recommandé d'éviter la réutilisation d'une même adresse dans plusieurs transactions. Plus loin, nous comprendrons la raison.

A titre d'information, aujourd'hui il existe 3 formats de clés publiques en utilisation. Le format P2PKH, P2SH et Bech32 [1]. Ces formats se distinguent respectivement par un 1, 3 et bc1 au début de l'adresse. Chacun de ces formats de clé est une chaîne comprise entre 26 et 35 caractères alphanumériques.

Une transaction

On parle de transaction dans la blockchain de bitcoin, lorsqu'un utilisateur envoie (ou reçoit) un ou plusieurs bitcoins à un ou plusieurs autres utilisateurs de la blockchain. Une transaction bitcoin est irréversible, c'est à dire, lorsqu'un payement est effectué, il n'y a pas de possibilité de restitution des bitcoins émis ni d'annulation de la transaction. Par principe, une transaction ne peut être émise que par un seul acteur, il contrôle toutes les clés publiques qui contiennent les bitcoins qu'il envoie.

Contrairement aux systèmes traditionnels de payements, tel que le compte bancaire, les bitcoins d'un utilisateur ne sont pas stockés au même endroit (dans le même compte). Mais plutôt dans une ou plusieurs adresses publiques. Par design, dans chaque transaction, tous les bitcoins de l'adresse de l'émetteur doivent être entièrement dépensés. Lors d'une transaction, si le dépensier n'a pas la somme nécessaire dans une seule de ses clés, il est obligé d'additionner les autres bitcoins de ses autres clés pour effectuer le payement qu'il doit. De plus, s'il y'a une différence entre les bitcoins qu'il a réunis et les bitcoins qu'il envoie, dans la même transaction il doit se renvoyer cette différence vers une de ses adresses ou vers une nouvelle adresse. On parle alors d'adresse de change ("change adresse"). Par définition, on appelle adresse de change, la nouvelle adresse à laquelle l'expéditeur des bitcoins se renvoie la différence entre les bitcoins en entrée et les bitcoins en sortie de la transaction.

Pour mieux comprendre le fonctionnement d'une transaction, voyons l'exemple de figure 1. Pour cet exemple, servons nous des noms Bob et Alice. Mais il faut savoir que dans la blockchain, à la place de ces noms nous avons des clés publiques.

Example 1. *Dans la **TX A**, Bob est le dépensier, il paye 0.5 BTC à Alice. Pour ce faire, Bob doit d'abord disposer d'une adresse publique qui contient suffisamment de bitcoins pour pouvoir payer Alice, mais aussi connaître l'adresse publique de Alice qui va recevoir les Bitcoins. Dans l'illustration on voit qu'il dispose de 50 BTC qu'il avait obtenu grâce à un investissement, contenu dans une adresse publique qui lui appartient. Il s'en sert pour payer Alice. Malgré qu'il n'effectue qu'un payement de 0.5 BTC, il est obligé de dépenser tous les 50 BTC de l'adresse qu'il utilise. On observe aussi que Bob se renvoie à lui même le reste des 49.5 BTC de la transaction, dans une nouvelle adresse (ou à la même adresse). On dit qu'il effectue un changement d'adresse. La valeur 50 BTC et l'adresse qui la contient sont respectivement appelées valeur en entrée et adresse en entrée ("input") de la transaction **TX A**, et les montants*

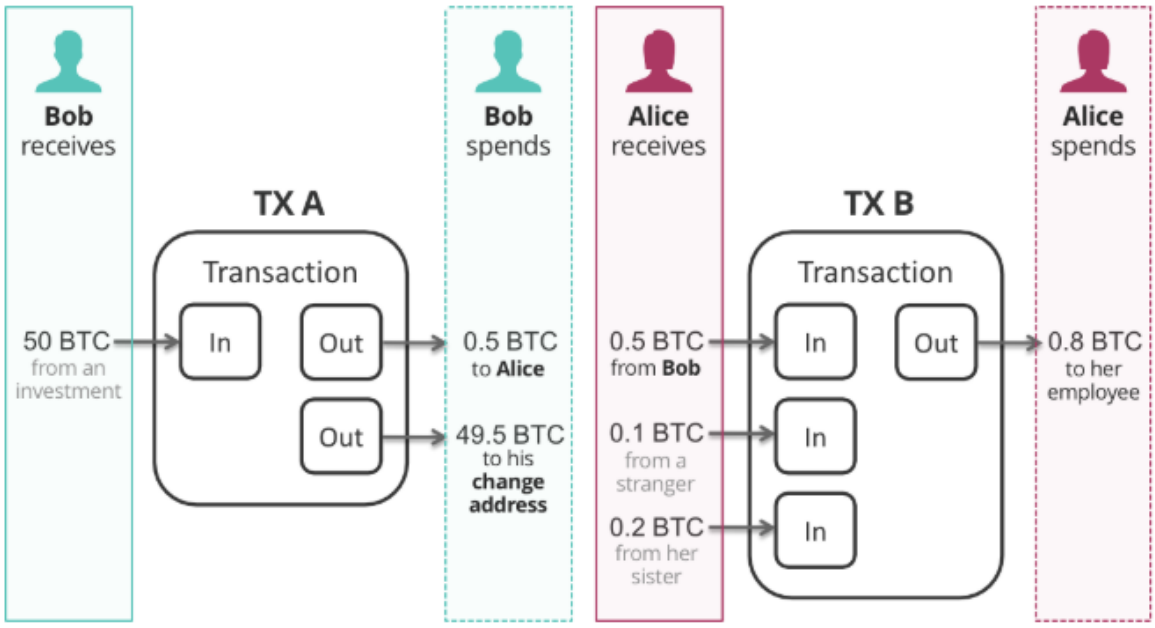

FIGURE 1 – Schéma simplifié d'une transaction de bitcoin.
source : freedomnode.com

0.5 et 49.5 sont les valeurs de sorties, les adresses qui les reçoivent sont les adresses de sorties ("output"). Dans la deuxième transaction ***TX B****, en entrée de la transaction on voit que Alice rassemble trois montants de bitcoins à partir de 3 clés publiques différentes pour avoir le montant nécessaire et effectuer un payement à son employé. Dans cette transaction, on a trois valeurs et trois adresses en entrée, une adresse et une valeur en sortie.*

La figure 2 nous montre en détails les données d'une transaction. On observe qu'une transaction bitcoin est caractérisée par son identifiant unique (*txid*), par une marque temporelle (*timestamp*), un taux d'échange (*exchage_rate*), une valeur total de bitcoin (*total_ value*), puis des entrées (*tx_ins*) et des sorties (*tx_outs*). L'identifiant est un *hash* la la transaction, cette information est unique pour chaque transaction. Le timestamp est l'heure à laquelle la transaction a été effectuée. L'input (ou l'entrée) est l'ensemble des adresses et valeurs en bitcoin de l'expéditeur. Dans la blockchain, les valeurs échangées dans une transaction sont exprimées en *satoshis*. Le dépensier gère toutes clés privées des adresses publiques qui sont en entrée de la transaction. L'output (ou la sortie) est l'ensemble des adresses publiques et des valeurs en bitcoin des destinataires de la transaction.

2.5 Quelques acteurs importants dans la blockchain

Dans cette partie nous mentionnons quelques acteurs qui sont importants à connaître pour mieux comprendre le fonctionnement de bitcoin. Nous avons déjà parlé des

```
tx_example = {
    "exchange_rate": 19.28,
    "timestamp": 1308025372,
    "total_value": 6000000,
    "tx_ins": [
        {
            "address": "1DRLxQ3ZpNwYxXygHxRPQ7jfs8fUjNV6kr",
            "hashPrevOut": "f80d836aeea612edda0069dd38d17841b95e7f6e64ace9aa2647b55f064a5497",
            "indexPrevOut": 0,
            "value": 1000000,
        },
        {
            "address": "1HkAx1YYmHPPoFtxrPbGHCdAfs6Vfs2kt1",
            "hashPrevOut": "a26d776cf11690b645f7b410f0ef5fca92ef5b21caffe288a19752c2cff5e5f7",
            "indexPrevOut": 1,
            "value": 5000000,
        },
        {
            "address": "1G8dGQoiWABcWMf6dYqrwdgmYxwqMGPzXF",
            "hashPrevOut": "65ecab9dee83ec80359518c2b2dc55e067611f3b7318dbb4f70e5678bd9d14b8",
            "indexPrevOut": 1,
            "value": 1000000,
        },
    ],
    "tx_outs": [
        {
            "address": "1CJpSaWmgSLSRQ3CUSXMkuYfZgMhWN1v2g",
            "indexOut": 0,
            "value": 1000000,
        },
        {
            "address": "1NHwJncVGz6nnWWRP5RAYXGg87KZAp5mxo",
            "indexOut": 1,
            "value": 5000000,
        },
    ],
    "txid": "2e7357eeaa6ec14b939d009bf10bdda2958a177be149fa19d0d8d1010934581f",
}
```


FIGURE 2 – Données d'une transaction effectuée le 14 Juin 2011 à 06h 22 min 52s.

mineurs, et des utilisateurs lambda. Maintenant nous allons brièvement présenter les wallets, les services de mixage et les plateformes boursières (marchés du bitcoin). Ces acteurs sont considérés comme les plus importants dans l'écosystème de bitcoin. Ces acteurs proposent des services externes à la blockchain, ils ne font pas partie du protocole de bitcoin. Ils ne sont que des exploitants de la blockchain.

Les Wallets

Rappelons que les bitcoins d'un utilisateur ne peuvent être stockés que dans une ou plusieurs clés publiques. Pour beaucoup, gérer ces clés privées/publiques individuellement est pénible et risqué. A l'image du compte bancaire, des portefeuilles bitcoins (ou portefeuilles digitales) ont vu la naissance afin de faciliter aux utilisateurs désireux la gestion de leurs Bitcoins. C'est la manière la plus simple d'utiliser bitcoin, mais aussi d'en acquérir sans faire du minage en achetant des bitcoins. Cette manière ne nécessite que l'installation d'une application sur son téléphone ou dans son ordinateur. Les wallets fournissent aux utilisateurs plusieurs fonctionnalités, notamment le calcul de frais de transactions adaptées, la création d'adresses de change, l'information du cours du bitcoin etc. Les utilisateurs doivent mettre à la disposition des wallets leurs vrais identités(noms de famille, adresse mail, carte bancaire, etc). Donc l'utilisateur doit faire confiance au wallet pour protéger ces informations.

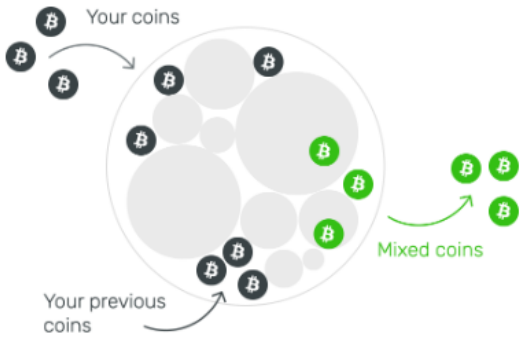

FIGURE 3 – Illustration simplifiée du fonctionnement du mixage de bitcion.
source :CryptoMixer

Les services de mixage (“CoinJoin”)

Le but du mixage est d’améliorer l’anonymat des utilisateurs de la blockchain. En effet, comme toutes les transactions sont accessibles dans la blockchain, à l’aide de techniques d’analyses de données on peut suivre toutes les transactions effectuées par une adresse, et savoir avec quel autre utilisateur elle a effectué des échanges de bitcoin. L’utilisation d’un service de mixage rend compliqué ce traçage. De manière très simplifiée, le mixage fonctionne comme suit, l’acteur qui souhaite mixer ses bitcoins les envoie vers l’adresse fournie par le service de mixing, et ce dernier lui renvoie les bitcoins vers une nouvelle adresse. En entrée d’une transaction de mixing, il y’a plusieurs adresses, et en sortie il y’a plusieurs adresses. Les bitcoins en entrée appartiennent à différents acteurs, les bitcoins en sorties sont envoyés dans de nouvelles adresses vers d’autres acteurs. Mais on ne peut plus (ou presque pas) associer les nouvelles adresses aux anciennes adresses.

La figure 3 est une illustration du fonctionnement du mixage. La figure illustre bien que l’utilisateur du service de mixage ne reçoit pas les mêmes coins de départ. L’activité de mixage est aussi appelée blanchiment de bitcoins. On peut remarquer que, avant de faire recours à ce type de service, il faut absolument que l’acteur qui mixte ces bitcoins ait confiance au service de mixing. Confiance que ce dernier va lui rendre ses bitcoins. Notons aussi que plusieurs services de mixages existent avec des niveaux de sécurité différents.

Platformes d’échanges

En général les plateformes d’échanges sont intermédiaires d’échange qui mettent en lien les vendeurs et les acheteurs de bitcoins (mais aussi d’autres cryptos-monnaies). Ce sont des places de marchés digitales où l’on achète ou vend des bitcoins contre de la monnaie fiduciaire ou contre d’autres cryptos-monnaies. Comme pour les wallets, pour avoir accès aux plateformes d’échanges, l’utilisateur doit s’inscrire et mettre à leur disposition ses informations personnelles.

Tous les services cités ci-dessous ne sont pas gratuits. Si un utilisateur achète des bitcoins à ces services, il les reçoit dans une adresse bitcoin, s'il en vend, il les envoie dans une adresse. Toutes ces transactions réalisées sont enregistrées dans la blockchain.

2.6 Heuristiques utilisées pour faire du Clustering

Le regroupement d'adresses ou clustering consiste à collectionner toutes adresses publiques d'un utilisateur de la blockchain. Comment trouver les adresses d'utilisateurs alors qu'ils sont anonymes dans la blockchain ? C'est possible grâce à ce que l'on appelle heuristique. Les heuristiques sont des méthodes intuitives qui peuvent trouver les adresses des utilisateurs. Celles qui sont les plus utilisées sont présentées ci-dessous accompagnées de simples exemples. Il n'existe pas de consensus dans la numérotation des heuristiques.

Heuristique 1 (h1). *The common-senders/input heuristic*. Toutes les adresses en entrée d'une transaction appartiennent au même acteur. De plus, dans toutes les transactions de la blockchain, s'il y a une intersection non nulle entre les adresses en entrée de deux ou plusieurs transactions, elle suppose que la totalité des adresses en entrée de ces transactions concernées appartiennent au même acteur (individu). En d'autres mots, si deux transactions A et B ont au moins une adresse en commun de leurs entrées, toutes les adresses en entrée de A, et toutes les adresses en entrée de B appartiennent au même acteur. L'heuristique 1 est considérée comme étant l'heuristique la plus sûre. Néanmoins, faisons remarquer que cette heuristique ne peut pas grouper les adresses d'un utilisateur qui utilise une nouvelle adresse à chaque transaction ; elle ne prend pas en compte les adresses en sorties des transactions. De plus, elle peut regrouper des adresses de différents acteurs si ces derniers font recours à un service de mixage.

Heuristique 2 (h2). Dans toutes les transactions contenant deux adresses (donc deux valeurs de bitcoins) en sortie,

(a) si l'une des deux valeurs en sortie est un nombre décimal qui à 4 chiffres ou moins après la virgule,

(b) et que la seconde valeur est aussi un nombre décimal, mais qui a 3 chiffres de plus après la virgule que la première valeur,

(c) on déduit que l'adresse qui a reçue les bitcoins avec le plus grand nombre de décimales est l'adresse de change.

(d) Et on suppose que cette adresse appartient à l'acteur qui contrôle les adresses en entrée.

Cette heuristique à été proposée par [8]. Reprenant l'exemple des auteurs, si on a ces deux types de valeurs en sortie *0.02* et *0.01615265*, on déduit que l'adresse qui

contient la dernière valeur est l'adresse de change. La raison avancée par les auteurs de cette heuristique est qu'il est improbable qu'un utilisateur de bitcoin effectue un transfert à un autre utilisateur d'un montant d'une valeur décimale avec un grand nombre de chiffre après la virgule.

Pour que cette heuristique soit vraie il faudrait que tous les utilisateurs fassent leurs transactions en bitcoins, c'est à dire pensent en terme de BTC. Cependant, depuis que bitcoin est devenue populaire, de nombreux utilisateurs effectuent leurs transactions en dollars ou euros, notamment les utilisateurs qui ont recours au wallets ou aux plateformes d'échanges. Sur les interfaces de leurs applications ils font transactions en dollars, ce sont les applications qui convertissent les valeurs en satoshis qui apparaissent dans la blockchain. Donc il est possible de voir une transaction qui a des valeurs de sorties de *0.02* et *0.01615265* et que *0.01615265* ne soit pas la valeur de change.

Heuristique 3 (h3). Dans toute transaction, si une adresse de change unique existe, on suppose qu'elle appartient à l'acteur qui contrôle les adresses en entrée. [19] considèrent qu'une adresse de change doit respecter les conditions suivantes :

(a) elle apparaît dans une transaction pour la première fois dans la blockchain.

(b) la transactions n'est pas une génération de bitcoin (c-à- d. pas du minage),

(c) il n'y a pas une adresse identique qui apparaît en entrée et en sortie de la même transaction (donc pas de retour de bitcoin vers une adresse qui est apparue en entrée),

(d) enfin, dans une transaction donnée, parmi les adresses en sortie, la condition a) ne doit être respectée que par une seule adresse [4].

Heuristique 4 (h4).(*CoinJoin and Mixing Transactions*). Cette heuristique a été egalement proposée par [8]. Dans une transaction donnée, s'il y'a 4 adresses en entrée et 4 adresses en sortie, on considère cette transaction comme étant du "mixage" et toute transaction de ce type est exclue des autres heuristiques [5].

3 État de l'art

Certains résultats présentés dans cet état de l'art n'ont pas un rapport direct avec nos résultats figurant dans ce rapport de stage, toutefois il est important de montrer au lecteur, les possibilités, et les types d'informations que l'on peut extraire dans la blockchain de bitcoin. La revue de littérature sera utile dans la continuité de ce travail.

4. Notons que cette heuristique sera ensuite modifiée.

5. [8], les auteurs ont trouvé 14 957 194 d'adresses qui sont impliquées au moins dans une transaction de mixage et 66 160 456 d'adresses apparaissent au moins une fois dans une transaction *qui n'est pas* du mixing

Athey et al. [8], cet article se divise en deux principaux travaux empiriques. Dans le premier, les auteurs proposent un modèle théorique des taux d'échanges de Bitcoins. Ils montrent que les taux d'échange sont consistants avec les fondamentaux, les prix du bitcoin sont déterminés par l'offre et la demande. Le second travail qu'ils font est d'étudier les données de la blockchain. Ils groupent les adresses de chaque utilisateur, en se servant des heuristiques (h1, h2 et h3) présentées ci-haut. Ils trouvent un nombre total d'entités égal à 27 474 538 parmi lesquelles 19 654 960 sont des adresses uniques. Les auteurs montrent aussi que le nombre d'utilisateurs avec une adresse est passé de moins de 20% en juillet 2012, à plus de 40% en Mai 2015.

Ils construisent ensuite le graphe des acteurs (entités), dont les nœuds sont les acteurs et les liens sont les transactions effectuées entre eux. Ils trouvent que le réseau n'est pas très connecté "sparse" : moins de 40% des acteurs ont moins de 3 liens et seulement 10% ont 7 liens ou plus. Grâce à ce graphe, ils étudient le comportement de ces acteurs et ont extrait un grand nombre d'informations sur l'adoption et l'usage de bitcoin. Ils montrent que la bitcoin est utilisée pour acheter des substances illégales (drogues, armes à feu...), pour financer des activités illégales, pour effectuer des payements internationaux, ils trouvent que bitcoin est plus fréquemment utilisée comme investissement (un actif de spéculation) et comme épargne.

D'autre part, les auteurs montrent que les utilisateurs qui sont susceptibles d'être impliqués dans des activités illégales ont une probabilité plus élevée de faire appel à du mixage par rapport aux autres membres de la blockchain. Sur les échanges entre acteurs, les auteurs distinguent plusieurs catégories d'échanges, ils trouvent que le plus grand nombre de transaction se fait entre les acteurs connus, ensuite suivent les jeux, les transactions de contrebande, et enfin les transactions de minage. Les auteurs étudient aussi la manière dont bitcoin est utilisée par différents secteurs à l'international. Ils ne trouvent pas de différence statistique dans les régions. La plupart des transactions est réalisée dans les places boursières "exchanges", et par des entités non identifiées.

Ron and Shamir [23], est l'une des premières études à analyser le comportement des utilisateurs dans la blockchain de bitcoin. Les auteurs étudient les données de transactions de la période de Janvier 2009 au 13 Mai 2013. Ils construisent le graphe des acteurs (ou entités). Ils font le clustering des adresses (l'identification des acteurs) grâce à l'heuristique 1, d'ailleurs ces auteurs sont cités comme étant les premiers utilisateurs de cette heuristique. Ensuite ils extraient des statistiques telles que la distribution du nombre d'adresses par entité (ou acteur), la quantité de bitcoins dépensés et les quantités conservés. Ils trouvent que la plupart des bitcoins sont dans des adresses dormantes, soit 51% des bitcoins. Ils calculent aussi la distribution du nombre de bitcoins reçu par chaque acteur, le niveau d'activité (qu'ils associent au nombre de transactions) de chaque adresse et de chaque acteur, ils déterminent la distribution de la balance courante par acteur et par adresse, les entités les plus actives. Ensuite ils analysent le sous graphe des transactions majeures, dans lesquelles les montants échangés sont supérieurs à 50 000 BTC, pour les données de l'époque, au total ils trouvent 364 transactions. Toujours dans ce sous graphe,

les auteurs montrent que les utilisateurs ont souvent des transactions consécutives longues, une grande quantité de bitcoins est transférée d'une adresse vers une autre via plusieurs adresses alternatives, et de nombreux bitcoins ne circulent pas dans la blockchain.

Parino et al. [21] Utilisent des données de transactions de bitcoin de la période du 09 Janvier 2009 au 25-Février-2016 et des sources de données externes , pour décrire l'adoption de bitcoin par pays. Ils étudient les facteurs qui expliquent cette adoption. Les données externes qu'ils utilisent sont les adresses IP des utilisateurs, les données de téléchargement de Bitcoin client (Bitcoin core), des données de google trends et des indexes socio-économiques des pays. Dans la première partie de leur papier, ils montrent que l'adoption de bitcoin s'est accrue entre 2015 et 2017 principalement dans les pays en développement. L'adoption de bitcoin est fortement corrélée aux indexes, PIB, à la liberté de commerce, et à l'indexe de pénétration d'internet. Dans la seconde partie du papier, les auteurs ont tenté de déterminer les variables socio-économiques clées corrélées aux transferts de bitcoins au niveau international. Pour cela, ils font d'abord un clustering des utilisateurs de la blockchain grâce à l'heuristique 1 et 2.1, ensuite ils associent chaque utilisateur à son pays (par géolocalisation de l'adresse IP). Ils trouvent que les variables PIB, liberté d'échange et la taille de la population sont des variables clés qui expliquent les transferts de bitcoins au niveau international.

Kondor et al. [17] réalisent une étude empirique du graphe de transaction de bitcoins de la période de Janvier 2009 au 7 Mai 2013. Les nœuds de ce graphe sont les adresses de Bitcoins et les liens sont les transactions qui lient deux adresses. Ils font une analyse de l'évolution du réseau et la dynamique des transactions en place, ils étudient les flux et l'accumulation de bitcoins entre les utilisateurs. Ils fournissent des mesures statistiques de la richesse des individus. Ils analysent le coefficient de Gini des degrés entrant et des degrés sortant et de leurs distributions, ils analysent le coefficient de clustering. Ils déduisent que les transactions "normales" sont caractérisées par une forte hétérogénéité entre les distributions des dégrés entrants et les degrés sortants, ils trouvent aussi que le graphe est "disassortatif", une tendance des noeuds similaires à être associés. Ils trouvent que le degré de distribution et le coefficient de clustering étaient très élevés au début du lancement de bitcoin par rapport à un graphe aléatoire, et après 2010, ils atteignent des niveaux stables. Ils montrent que l'attachement préférentiel est un important facteur façonnant la distribution du degré et de la richesse. Ils montrent aussi que dans le cas de bitcoin, la richesse des nœuds des plus riches s'accroît plus vite que le reste des nœuds. Ils trouvent qu'il existe une corrélation positive entre la richesse et le degré d'un nœud. Sur les dynamiques des transactions, ils mesurent le coefficient de gini dans le temps, ils trouvent que sa valeur est très élevée mais devient stationnaire vers la fin ; indiquant une hétérogénéité dans la distribution de la richesse. Enfin ils montrent qu'ils existent un lien entre les évolutions du réseau de transactions et la distribution de richesse.

Bartoletti et al. [10] utilisent plusieurs techniques de classifications supervisées pour automatiquement détecter des pyramides de ponzi. Pour cela, ils construisent

leur propre base de donnée, ils font un clustering des adresses de bitcoins en utilisant l'heuristique 1, ensuite ils font une fouille de données de clés publiques et noms de leurs détenteurs dans des forums [6] de discussion à propos de bitcoin. ils collectent les adresses des individus qui proposent des investissements à des gains très élevé (des pyramides de ponzi). A cela, ils rajoutent un certain nombre de variables exogènes qui expliquent la base de donnée ainsi obtenue est ensuite scindée en deux pour obtenir un ensemble d'entraînement et un ensemble de test. Le meilleur classifieur qu'ils obtiennent avait 1% de faux positifs. Les auteurs trouvent que, les acteurs qui font recours aux pyramides de ponzi utilisent plusieurs adresses, 19 sur 32 acteurs utilisent plus d'une adresse pour un total de 1211 adresses, en dollars, le montant est d'environ 10 millions. En plus de ces résultats, les auteurs fournissent des statistiques descriptives telles que le nombre d'adresses détenues par chaque acteur ponzi, les montants résultants en dollar et en bitcoin, le nombre de transactions effectuées, etc. Les auteurs mettent à disposition leur base de donnée contenant des adresses des pyramides de ponzi et leur noms ainsi que les variables explicatives significatives permettant de classifier un acteur comme chaîne de ponzi (ces variable sont le coefficient de Gini des valeurs sortantes, le ratio entre les transactions entrantes et le nombre total de transactions, la moyenne et l'écart type des valeurs sortantes, le nombre différent d'adresses de celui qui a fait le transfert de bitcoin au cluster, et le nombre total d'activités).

Meiklejohn et al. [19] Tentent de caractériser le réseau de bitcoin et de déanonimiser (trouver les "vrais identités") des utilisateurs de la blockchain de bitcoin, ils travaillent avec les données de la blockchain de Janvier 2009 au 13 Avril 2013. Ils font du clustering des utilisateurs en se servant de l'heuristique 1 et l'heuristique 2.0 (et une version améliorée de cette dernière), puis fournissent des statistiques détaillées de ces clusters (taille, addresse par cluster, etc...). Ils identifient les "vrais acteurs" à qui appartiennent ces adresses, premièrement en effectuant directement des transactions avec plusieurs type d'acteurs (les mineurs, les "wallets" portefeuilles, les plateformes d'échanges, les vendors, les joueurs et d'autres acteurs) ; deuxièment en faisant du web scrapping sur les forums de discussions à propos de bitcoins. Ils construisent deux graphes dirigées. Le graphe des transactions et le graphe des adresses publiques. Ils montrent que l'on peut se focaliser sur différents acteurs (particulièrement les acteurs populaires) et étudier la nature de leurs transactions. Ils analysent les comportements de quelques principaux acteurs des plates-formes de jeux, des places boursières et les acteurs impliqués dans des activités illicites.

Di Francesco Maesa et al. [13] étudient le graphe des utilisateurs bitcoin. Ce graphe contient des données de la blockchain de bitcoin de la période du Janvier 2009 au 23 Décembre 2015. Ces auteurs font une analyse sur les mesures classiques des propriétés d'un graphe (la densité, le coefficient de clustering, et plusieurs mesures de centralité). Ils font des comparaisons de ces mesures à d'autres graphes complexes. Les auteurs montrent que le phénomène, "rich-get-richer" est présent

6. bitcointalk.org, reddit et aussi sur internet archive

dans la blockchain de bitcoin, les nœuds centraux se comportent comme des hubs dans plusieurs endroits du graphe et correspondent aux acteurs les plus populaires. Ils montrent également que les nœuds dans le *strongly connected component* (SCC) s'accroissent plus vite.

Dans un différent papier [18], les mêmes auteurs étudient la structure "bow tie" du graphe des utilisateurs de bitcoin pour en extraire des informations de l'activité économique des différents composants de ce graphe. L'analyse se base sur des données de la blockchain à partir de la période du Janvier 2009 au 4 Décembre 2015.Ils se servent uniquement de l'heuristique 1 pour grouper les adresses. Ils distinguent 6 types de nœuds, chaque type est associé à un composant du graphe. Ces types sont les nœuds qui sont présents dans le SCC (Strongly connected component). Les nœuds qui ne sont pas connectés au SCC (appelés Disconnected). Les nœuds qui ne sont pas dans le SCC mais qui sont connectés aux nœuds du SCC (IN). Les nœuds qui ne sont pas dans le SCC mais reçoivent des bitcoins provenant des nœuds du SCC (appelés OUT). Les nœuds qui ne sont pas dans l'une des catégories citées mais sont liés à au moins un nœud de IN ou OUT (appelé TENDRIL). Les nœuds qui ne font pas partie de SCC mais qui sont liés à au moins un nœud de OUT, et peuvent être atteints par au moins un nœud de IN (TUBE). Enfin les noeuds qui ne font partie d'aucune des catégories citées ci-hauts. L'analyse des auteurs montrent que les échanges économiques sont effectuées par les clusters qui appartiennent au SCC du graphe. La plupart des mineurs sont dans IN et reçoivent de fortes rémunérations par rapport aux autres mineurs du SCC. Ils font aussi une étude temporelle pour comprendre comment les différents composants du graphe et les activités économiques évoluent au cours du temps.

4 Analyse des données de la Blockchain de bitcoin

Dans cette partie, nous donnons des explications détaillées sur ce qui a été fait dans l'analyse des données. Nous commençons d'abord par donner quelques informations sur les outils utilisés.

4.1 Outils utilisés

Cette étude à été réalisée essentiellement grâce au serveur du laboratoire LIRIS, de Python et Neo4j. Le serveur de LIRIS dispose de 64G de mémoire RAM et de 20 Processeurs. Il fonctionne sous le système d'exploitation Linux, il nous a fourni la puissance de calcul nécessaire pour la réalisation de notre analyse. Je dispose d'un ordinateur qui fonctionne sous Windows. Pour accéder au serveur [7], nous nous sommes servis de PuTTY et de FileZilla. PuTTY est un terminal virtuel sur lequel

7. L'accès au serveur nécessite une connexion VPN. Grâce à Autoconnect, nous avons pu avoir accès au VPN à partir d'un ordinateur qui fonctionne sous Windows.

on peut exécuter des commandes Linux sur un serveur à partir de Windows. FileZilla facilite l'accès et la gestion de fichiers sur le serveur, on peut aisément copier ou déplacer des fichiers entre le serveur et mon ordinateur personnel.

Python nous a permis de traiter les données et de les analyser. Nous avons fait recours à ses librairies standards. Nous avons utilisé ses "built-in" (des listes, des tuples, des dictionnaires et des sets) pour transformer nos données d'origines de la blockchain qui étaient sous le format JSON. Grâce à Neo4j (associé à python), nous avons réussi à faire le groupement des adresses de chaque utilisateur, et construire le graphe du réseau de bitcoin. Neo4j est un gestionnaire de base de données basé sur les graphes. Il permet de stocker et gérer des données. On peut y construire d'énormes graphes et y extraire les informations que l'on souhaite. On s'est servi aussi de Python et des librairies dédiées, pour faire des requêtes sur Neo4j.

4.2 Description des Données

Nous disposons des données de la blockchain du premier jour 03 janvier 2009 au 09 septembre 2016. La première transaction de la blockchin a été réalisée le 03 Janvier 2009. Au total nous disposons exactement de 152 673 373 de transactions reparties en 200 fichiers au format JSON. Une ligne d'un fichier correspond à une transaction. L'exemple de la figure 2 correspond à une ligne dans le fichier qui le contient. L'ensemble de tout les fichiers de données fait un poids de 115GB.

Les fichiers sont nommés de part-00000 à part-199. Nous avons trouvés des erreurs dans les fichiers part-40 à part-44. Nous avons remarqué que la dernière ligne de chacun de ces fichiers est inachevée. Nous n'avons pas pu identifier les raisons de cette erreur, mais on sait qu'elle signifie que nous avons des transactions manquantes dans les fichiers concernés. On n'est sûr que nous avons au minimum 5 transactions non prises en compte. Nous suspectons qu'il peut y en avoir plus. En effet, sans prendre en compte les fichiers avec erreur, en moyenne, nous disposons de 773788 par fichier ; le fichier qui contient le nombre de lignes minimum a 555237, celui qui contient le maximum a 1141965 lignes. Tandis que dans chacun des fichiers avec erreur nous avons respectivement 667727, 524924, 332451, 138489 et 120935 lignes.

4.3 Groupement des adresses ("clustering")

Pour faire le clustering nous avons appliqué les deux premières heuristiques présentées plus haut. Dans cette partie nous expliquons en détails comment nous avons implémenté ces heuristiques et les résultats obtenus. Mais d'abord nous expliquons comment nous avons pu gérer 115 G de données sachant les outils qui étaient à notre disposition.

Le problème le plus important que nous avons rencontré dans l'analyse des transactions est leur taille (ou leur poids). Il est impossible de charger en mémoire 115 GB de données dans une machine de 64 G de RAM en une fois. En conséquence, nous avons eu l'idée d'analyser les données ligne par ligne. C'est à dire d'extraire

FIGURE 4 – Identification des transactions d'un même utilisateur

les informations dont nous avions besoin dans une transaction à la fois. Et de garder chaque information dans un des built-ins de Python, soit dans un tuple, un dictionnaire, une liste ou un set. Et nous travaillons avec les données extraites.

Application de l'heuristique 1

Rappelons que l'heuristique 1 dit que toutes les adresses en entrée d'une transaction appartiennent au même utilisateur et de même que l'union de toutes les intersections des adresses en entrée. Cette heuristique peut être appliquée de plusieurs manières, nous l'avons appliqué en suivant les étapes ci-dessous.

1. D'abord nous avons collecté toutes les adresses en entrée de toutes les transactions dans différentes listes. Et nous avons mis toutes ces listes dans une seule liste.
2. Ensuite nous avons vérifié si une *intersection* existe entre deux listes ou plusieurs listes. Si c'est le cas, on fait *l'union* des listes concernées. On déduit que toutes les adresses contenues dans ces listes appartiennent au même utilisateur.
3. On répète 2 jusqu'à ce qu'il n'existe plus d'intersection entre les listes d'adresses restantes.

L'exemple suivant explique concrètement cela.

Example 2. *Considérons l'illustration de la figure 4, supposons que* @1,... @666, *sont des adresses bitcoins au format chaîne de caractères alphanumériques. (1) Pour chaque transaction Tx 5, Tx 109, Tx 314, si on collecte les adresses en entrée de chaque transaction dans une différente liste, on obtient respectivement les listes d'adresses suivantes :* $liste_1 = [@1, @13]$, $liste_2 = [@33, @11, @78]$ *et* $liste_3 = [@1, @33]$. *(2) Ensuite cherchons les intersections entre ses listes. On observe qu'il existe une intersection entre les listes 2 et 3. Nous déduisons donc que les adresses de l'union des deux listes, soit la* $liste_{23} = [@1, @33, @11, @78]$, *appartient au même utilisateur. (3) Il existe une intersection entre* $liste_1$ *et* $liste_{23}$, *donc les adresses de l'union entre les deux listes, soit* $liste_{123}$ *appartient au même utilisateur. Au final, on trouve que les adresses de ses trois transactions appartiennent au même utilisateur, on déduit aussi que toutes ces transactions ont été effectuées par le même utilisateur.*

Nous avons appliqué cette méthode dans l'ensemble des transactions de la blockchain. Après l'étape 1, on avait 152 millions de différentes listes sur lesquelles il fallait trouver des intersections puis faire des unions. Nous avons d'abord tenté d'accomplir cette tâche sur Python uniquement. Nous avons regroupé toutes ces listes

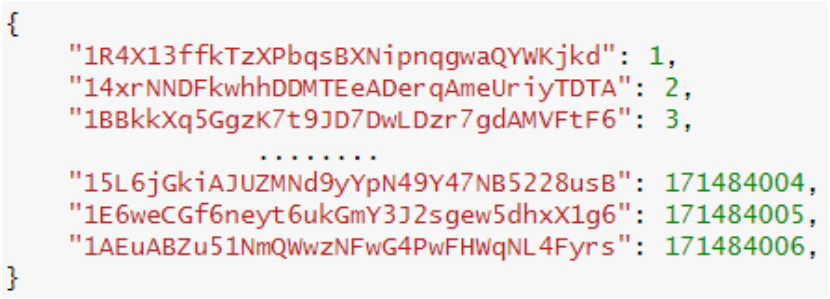

FIGURE 5 – Extrait du dictionnaire des adresses

dans une seule liste. En d'autres termes c'est la liste de toutes les adresses en entrée des transactions de la blockchain.

$$LL_{caractres} = [[@1, @13], [@33, @11, @78], ..., [@17, @150]] \quad (1)$$

On s'est vite rendu compte cela était presque impossible, pour des raisons de mémoire RAM, et de temps de calculs. En effet, la liste de listes $LL_{caractres}$ faisait 15G.

Pour remédier au problème de mémoire, au lieu de travailler avec les adresses au format chaînes de caractères, c'est à dire de cette forme : "1R4X13ffkTzXPbqsBXNi pnqgwaQYWKjkd", nous avons attribué à chaque adresse une valeur numérique. La raison est que les 26 ou 35 caractères alphanumériques de l'adresse prennent plus d'espace dans la mémoire qu'une valeur numérique. Nous avons construit un dictionnaire qui contient toutes les adresses uniques dans la blockchain, de sorte que chaque adresse unique soit associée à un chiffre unique. La *clé* du dictionnaire est l'adresse en chaîne de caractère et sa *valeur* est un nombre entier. Par exemple, nous considérons que :

$$1R4X13ffkTzXPbqsBXNipnqgwaQYWKjkd \equiv 1$$

Nous présentons un extrait du dictionnaire sur la figure 5. Après conversion, $LL_{caractres}$ de 1, dévient une liste de listes d'entiers naturels qui est de la forme :

$$LL_{convertie} = [[1, 33], [33, 11, 78], ..., [255, ..., 171484006]] \quad (2)$$

La liste convertie en nombre entier fait environ 4G sur le disque dur. Le problème de mémoire était résolu, mais il reste le problème de temps de calculs. Nous n'avons pas pu faire une fonction efficiente sur Python pour trouver toutes les intersections possibles 152 millions de listes dans un temps raisonnable.

Pour résoudre ce dernier problème, au lieu de chercher des intersections et des unions sur Python, nous avons fait recours à Neo4j et à la théorie des graphes. Notre idée à été de construire un graphe des adresses en entrée. Nous appelons ce graphe, *Graphe du Clustering (GC)*. Nous nous servons toujours de la liste de listes d'adresses en entrée des transactions de la blockchain convertie en valeurs numériques $LL_{convertie}$.

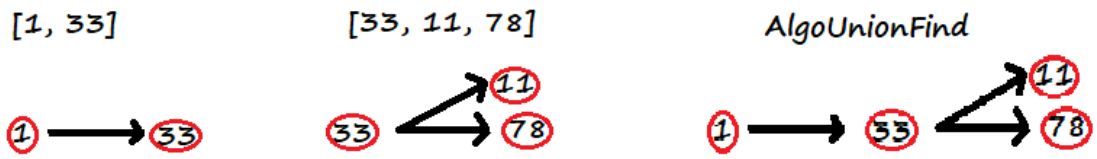

FIGURE 6 – Clustering par l'heuristique 1 en utilisant la théorie des graphes

Nœuds de GC . Les nœuds du graphe sont toutes les adresses en entrée de chaque transaction. En pratique, pour nous ce sont les adresses (en valeurs numériques) se trouvant dans liste de listes. A titre d'exemple, dans liste présentée en 2, 1 est un nœud, 33 est un nœud, de même pour toutes les valeurs de chaque liste.

Les liens de GC . Nous avons crée des liens entre toutes les adresses d'une même sous liste. En d'autres termes, nous créons des liens entre toutes les adresses en entrée d'une même transaction, mais de manière efficiente, c-à-d. nous prenons la première adresse et la lions aux restes des adresses de la liste (voir l'exemple suivant).

Après avoir construit ce graphe, nous trouvons les composantes connexes du graphe grâce à l'algorithme *UnionFind*. La composante connexe du graphe est l'ensemble de tous les nœuds qui sont liés. Toutes les adresses (nœuds) liées forment une partition du graphe, et on suppose que chaque partition appartient au même utilisateur (selon l'heuristique 1).

Example 3. *Considérant toujours notre exemple précédent, nous faisons un lien entre l'adresse 1 et 33 dans la liste 1, puis des liens entre 33 et le reste des éléments de la liste 2. Ensuite, l'application de l'algorithme UnionFing connecte tous les nœuds qui sont liés. La partition résultante est aussi appelée cluster.*

La figure 6 montre un extrait du résultat obtenu dans Neo4j, on observe 4 composantes connexes. Toutes les adresses qui sont connectées appartiennent au même acteur. On appelle acteur ou entité, un cluster d'adresses. En dessous de la figure, on observe les propriétés du nœud 2097156, on voit que cette adresse (et toutes ces connexions) appartient à la partition 1029212. Après l'application de l'algorithme *UnionFind* Neo4j marque dans les propriétés de tous les nœuds, le numéro de la partition au quel ils appartiennent. Il faut noter que Neo4j permet d'ajouter des informations supplémentaires sur les nœuds et les liens que l'on appelle propriétés. Cette fonctionnalité est très importante car elle nous permet d'effectuer des requêtes en fonction de ces propriétés. Nous le verrons dans la construction du graphe des utilisateurs et des transactions, grâce à la possibilité d'avoir des propriétés riches, nous pourrons rajouter quasiment toutes les informations disponibles dans la blockchain.

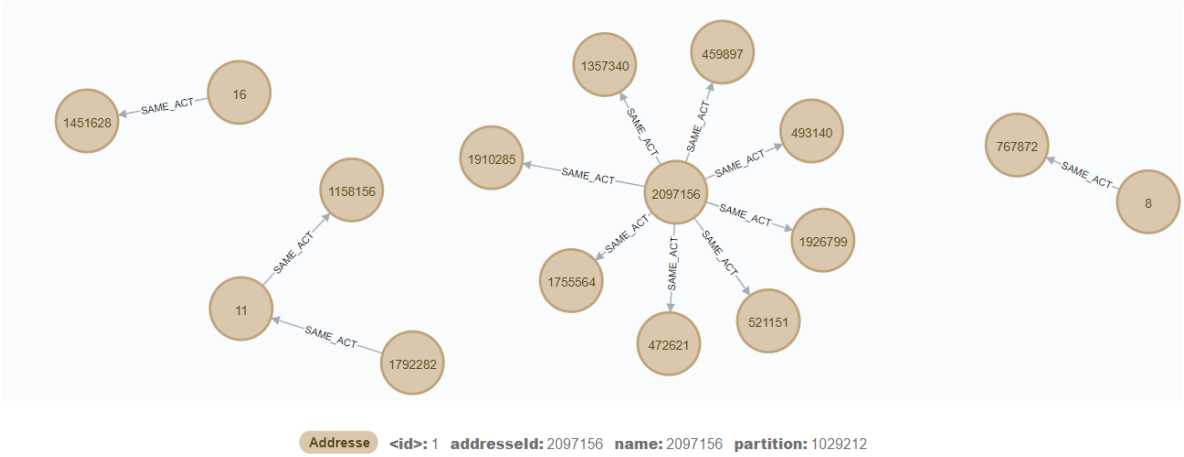

FIGURE 7 – Extrait des composantes connexes (Clustering graph).

Application de l'heuristique 2

Nous avons appliqué l'Heuristique 2 de la même manière. Dans l'exemple de la figure 4, supposons que la transaction Tx 5 respecte les critères de l'heuristique 2, et que l'adresse 666 est l'adresse de change. Nous collectons une seule adresse en entrée et l'adresse de change dans une liste $[1, 666]$. Nous rajoutons toutes les listes ainsi obtenues aux listes obtenues par l'heuristique 1. Nous créons le Graphe du Clustering de la même manière que précédemment, comme indiqué dans l'exemple 3.

L'intérêt d'appliquer l'heuristique 2 est de pouvoir classer des adresses qui n'ont pas pu être classé par l'heuristique 1, lier des groupes qui n'ont pas pu être liés par l'heuristique 1 et surtout de pouvoir classer les adresses en sorties qui n'apparaissent pas en entrées des transactions.

Graphe du Clustering

Nous avons construit GC sans prendre en compte les adresses uniques en entrée, elle n'auraient pas été utiles pour le clustering parce que de toute façon elles ne seraient liées à aucune autre adresse. Le graphe [8] ainsi obtenu avait 119 612 744 de noeuds et 125 919 589 de lien.

Avant de présenter la description des clusters, nous avons cherché à extraire des informations de la blockchain dans la liste de listes des adresses en entrées des transactions. Nous avons construit une deuxième liste pareil pour les adresses en sorties [9]. Les résultats que nous avons obtenus sont présentés dans la section suivante.

8. Dans ce graphe nous avons seulement utilisé les données extraites par l'heuristique 1. Après avoir comparer les résultats obtenus par les deux heuristiques, nous n'avons retenu que l'heuristique 1. Nous verrons les justifications dans les sections suivantes

9. Je n'ai pas réussi à comprendre pourquoi une liste qui prenait 4 GB dans la mémoire disque

couples	counts	%p
(1, 2)	76123308	50.534426
(2, 2)	24972576	16.578034
(1, 1)	10803499	7.171898
(3, 2)	6958355	4.619301
(4, 2)	4170316	2.768463
(1, 3)	4069532	2.701557
(2, 1)	3230527	2.144584
(5, 2)	1942319	1.289408
(2, 3)	1623128	1.077513
(6, 2)	1243521	0.825511

TABLE 1 – Couples nombre d'adresses en entrée et en sortie. Top 10 des couples fréquents.

Extraction d'informations dans les listes de listes d'adresses en entrée et en sortie

On peut extraire d'importantes informations dans des listes de listes des adresses en entrée et sortie que nous avons construit pour faire le clustering.

Nombre total de transactions. On peut retrouver le nombre total de transactions dont nous disposons en trouvant le nombre d'éléments de la liste de listes LL (des adresses en entrée ou en sortie). Il suffit de trouver le nombre total de sous listes. Ainsi, nous retrouvons que nous avons 152 Millions de transactions.

Couples des nombres d'adresses en entrée et sortie de chaque transaction. Cette information répond à la question, combien d'adresse est utilisée en entrée d'une transaction, et combien apparaît en sortie. Le résultat est présenté dans le tableau 1. La colonne *couples* contient les couples (nb. entrée, nb. sorties), la colonne *counts* contient le comptage de chaque couple, la colonne $\%p$ contient le pourcentage du nombre total de transactions. On remarque que les transactions avec une adresse en entrée et 2 en sortie, sont les plus fréquentes. Autrement dit, dans la blockchain, 50% des transactions ont 1 adresse en entrée et 2 en sorties. 16% des transactions ont 2 adresses entrées et 2 en sorties. Notons que certaines adresses peuvent être répétées en entrée ou en sortie [10].

Statistiques des adresses. Le nombre total d'adresses dans la blockchain est de 171

dur prenait 30GB en mémoire RAM dans le serveur. J'ai converti la liste format JSON, pickle, hdf5 et numpy array et l'ai enregistré dans le disque dur la taille change dans ce dernier, mais dès que je charge le fichier en mémoire RAM, la taille passe à 30 GB.

10. Nous avons vérifié aussi qu'en ne prenant que les adresses uniques dans les entrées et les sorties, les résultats du top10 sont les mêmes. Il peut sembler étrange d'avoir la même adresse en entrée plusieurs fois ou la même adresse en sortie plusieurs fois. Mais il en existe et c'est liée au design de bitcoin, voici des exemples en) sortie et entrée.

addresses_ids	name	counts
147182193	1NxaBCFQwejSZbQfWcYNwgqML5wWoE3rK4	1853678
40532287	1dice8EMZmqKvrGE4Qc9bUFf9PX3xaYDp	1596535
154453850	1LuckyR1fFHEsXYyx5QK4UFzv3PEAepPMK	1138721
157094167	1dice97ECuByXAvqXpaYzSaQuPVvrtmz6	1101320
56059225	1VayNert3x1KzbpzMGt2qdqrAThiRovi8	782378
170331201	1dice9wcMu5hLF4g81u8nioL5mmSHTApw	593252
3515159	1dice7fUkz5h4z2wPc1wLMPWgB5mDwKDx	436171
138134635	1MPxhNkSzeTNTHSZAibMaS8HS1esmUL1ne	405098
69900983	1dice7W2AicHosf5EL3GFDUVga7TgtPFn	393468
168876290	3HNSiAq7wFDaPsYDcUxNSRMD78qVcYKicw	364310

TABLE 2 – Les adresses les plus fréquentes en *sortie* des transactions

484 006. Des extraits des adresses les plus utilisées en entrée et sortie des transactions sont respectivement présentés dans les tableaux 2 et 3. La colonne *addresses_ids* contient les correspondances numériques des adresses contenues dans la colonne *name*; la colonne *counts* contient les comptages du nombre de fois qu'une adresse a été utilisée dans la blockchain. Dans les tables on observe que ce sont les mêmes adresses qui apparaissent majoritairement en entrée et en sortie. L'adresse numéro 147 182 193 apparaît le plus dans la blockchain.

Le minage. L'adresse qui a le nom 0 (id : 23357234) dans le tableau 3 est l'adresse en entrée dans la transaction de rémunération du minage. En réalité, le *id 23357234* (nom 0) n'est pas une adresse, il n'appartient pas à un acteur. Par design, un mineur reçoit une récompense de minage dans une transaction ayant 0 comme adresse en entrée si c'est une génération de nouveaux bitcoins (et pas de frais de transactions). Cet id apparaît dans 423348 transactions dans la blockchain, ce résultat veut dire que nous avons eu 423348 minages entre 2009 et 2016. C'est aussi le nombre de fois qu'il a eu de création monétaire dans la blockchain durant cette période.

Il est possible de savoir à quels acteurs appartiennent les adresses les plus fréquentes, les valeurs en bitcoins qu'elles ont reçues ou envoyées. Nous répondrons à ces questions dans les sections suivantes après la construction du graphe des transactions et des acteurs.

4.3.1 Description des clusters

Dans les lignes qui suivent nous référons au groupement par application de l'heuristique 1 à *h1* et au groupement par combinaison des deux heuristiques à *h1-h2*. Le premier du résultat du clustering est présenté dans le tableau 4. Le nombre de clusters (acteurs) trouvé uniquement grâce à l'heuristique 1 est de 74 264 935. Combiné à l'heuristique 2, le nombre de clusters passe à 55 951 492. Dans la première

addresses_ids	name	counts
147182193	1NxaBCFQwejSZbQfWcYNwgqML5wWoE3rK4	1852778
40532287	1dice8EMZmqKvrGE4Qc9bUFf9PX3xaYDp	1595655
154453850	1LuckyR1fFHEsXYyx5QK4UFzv3PEAepPMK	1136530
157094167	1dice97ECuByXAvqXpaYzSaQuPVvrtmz6	1100444
56059225	1VayNert3x1KzbpzMGt2qdqrAThiRovi8	781771
170331201	1dice9wcMu5hLF4g81u8nioL5mmSHTApw	592591
3515159	1dice7fUkz5h4z2wPc1wLMPWgB5mDwKDx	435419
23357234	0	423348
69900983	1dice7W2AicHosf5EL3GFDUVga7TgtPFn	392892
168876290	3HNSiAq7wFDaPsYDcUxNSRMD78qVcYKicw	364311

TABLE 3 – Les adresses les plus fréquentes en *entrée* des transactions

ligne du tableau, on observe que le nombre total d'acteurs ayant au minimum deux adresses est de 18 389 763 avec les deux heuristiques et de 22 393 672 pour h1. Parmi les clusters de taille 1, c-à-d ne contenant qu'une seule adresse, 30 millions font partie des adresses en entrée pour h1-h2 contre 44 millions pour h1. De même, il y a 7,6 millions de clusters contenant une adresse obtenue avec h1, dont les adresses sont en sortie de transaction ; et un plus de 6 millions et demi pour h1-h2. Une autre chose importante à remarquer ici est que h1-h2 a permis de classer 853681 [11] nouvelles adresses de sortie et près de 15 millions d'adresses en entrée de transaction qui n'étaient pas classés par h1.

Dans le tableau 5, nous présentons les 10 plus gros clusters et nous comparons le groupement par h1 et h1-h2. Dans la colonne *Cluster_ID* figure numéro du cluster. Rappelons que pour nous la dénomination de cluster est équivalente à acteur ou entité. La colonne *size(nb. d'adresses)* comporte la taille du cluster. C'est le nombre d'adresses que le cluster contient. Nous observons les résultats suivants :

- Le plus gros cluster obtenu par l'heuristique 1 (le cluster 0) détient 11 421 397 d'adresses publiques et si on rajoute l'heuristique 2 le nombre d'adresses devient 56 225 527. Les deuxièmes acteurs ont respectivement 2 095 653 et 69 845 de clés publiques selon les clusterings par h1 et h1-h2.
- Nous remarquons que lorsque que l'on incorpore l'heuristique 2 dans le clustering on obtient un super cluster. Ce constat nous pousse à nous interroger sur la validité du clustering et l'intérêt d'utiliser h2.
- L'effet attendu par l'utilisation de h2 semble être exacerbé. Il est peu probable qu'une seule entité (ou seul utilisateur) détienne autant d'adresses. Il est évident que l'application de h2 a rassemblé tous les gros clusters obtenus uniquement par h1.

Dans le tableau 6 nous présentons le type de cluster le plus fréquent parmi les

11. différence entre 7679445-6825764.

Taille des clusters	H1	H1 and H2
> 2	22 393 672	18 389 763
$= 1$ (ads. en Entrée)	44 191 818	30 735 965
$= 1$ (ads. Sortie)	7 679 445	6 825 764
Nombre total de clusters	74 264 935	55 951 492

TABLE 4 – Statistiques des Clusters

clusters. La colonne *cluster size* contient le nombre d'adresses que détient le cluster. La colonne *counts* contient le comptage du nombre de fois que ce type est présent parmi les clusters. On observe que les groupes avec une seule adresse sont les nombreux. Nous avons obtenu 53 500 662 clusters de taille 1 (qui ne contiennent qu'une seule adresse) avec l'heuristique 1 et 39 012 729 avec l'heuristique 2. De manière générale, on remarque que les clusters ayant peu d'adresses sont les plus nombreux. 72.04% des clusters obtenus avec h1 ont une seule adresse contre 69.7% pour ceux obtenus en associant h1 et h2. Ce résultat peut signifier plusieurs choses :

- Un grand nombre d'adresses n'a pas pu être classé dans un groupe par les deux heuristiques. On peut expliquer cela par des facteurs tels que, l'inefficacité des heuristiques. Ou que de nombreux utilisateurs de la blockchain sont très prudents et ne réutilisent pas (ou peu) les mêmes clés publiques, ou font beaucoup recours au mixage.
- La similarité des résultats obtenus par h1 et h1-h2 (toujours dans le tableau 6) prouve que l'heuristique 2 a été inefficace. En effet l'application de h2 n'a classifié que peu de nouvelles adresses. On comprend que, peu d'adresses du cluster gigantesque qu'il a trouvé (voir tableau 5) ne proviennent des clusters de taille 1 ou 2.

Une première raison que l'on peut avancer pour expliquer le mauvais résultat obtenu par h1-h2 est le fait que nous avons des données manquantes. Une deuxième raison est que l'on considère que les individus échangent uniquement en satoshis. On sait par exemple que beaucoup d'individus qui utilisent des wallets, vont faire des transactions en dollar, et que ces valeurs une fois converties en satoshis dans la blockchain, vont satisfaire la condition de requise par h2 sans qu'elles ne soient des valeurs de change (l'inverse peut être vrai aussi).

4.3.2 Identification des vrais acteurs

Il est possible de connaître les *vrais identités* de certains acteurs à partir de leurs adresses publiques. En effet, de nombreux acteurs sont obligés de faire connaître leurs clés publiques par internet ou d'autres moyens. C'est le cas des vendeurs, des services de mixage, des demandeurs de dons, les auteurs de pyramides de Ponzi, etc. D'autres utilisateurs, publient leurs adresses volontairement ou par manque de

	H1	H1 and H2
Cluster_ID	size(nb of addresses)	size(nb of addresses)
0	11421397	56225527
1	2095653	69845
2	2005880	52710
3	849725	49462
4	825954	45793
5	585058	45647
6	578719	42411
7	489411	40221
8	475985	39558
9	473446	36084

TABLE 5 – Fréquence du nombre d'adresse dans chaque cluster (top 10). Comparaison des heuristiques (h1 et h2).

	H1		H1 and H2	
cluster size	counts	%p	counts	%p
1	53500662	72.040273	39012729	69.725985
2	14780139	19.901908	10400586	18.588577
3	2859909	3.850955	2306626	4.122546
4	1058389	1.425153	1632923	2.918462
5	540161	0.727343	621854	1.111416
6	288561	0.388556	470917	0.841652
7	181757	0.244741	237936	0.425254
8	129165	0.173925	191667	0.342559
9	100293	0.135048	126987	0.226959

TABLE 6 – Tailles des clusters les plus fréquents (top 10). Comparaisons des heuristique 1 et heuristique 2.

FIGURE 8 – Dictionnaire des vrais noms des acteurs (extrait)

prudence, dans les réseaux sociaux, dans des forums de discussion bitcoin, etc. En concéquence certaines adresses et les noms de leurs propriétaires peuvent être collectionné sur internet en faisant du web scrapping. Aussi, on peut tout simplement effectuer des transactions bitcoin avec plusieurs acteurs de la blockchain dont on connaît le nom dans le monde physique pour connaître leurs adresses.

Des auteurs tels que [19] et [11], ont utilisé certaines de ces techniques dans leurs papiers. Nous avons réutilisé leurs jeux de données, puis des données de web scrapping qui ont été réalisé par des étudiants de mon Superviseur de stage. L'ensemble des données dont nous disposons fait une base de données de 30083 d'adresses d'utilisateurs labellisées. Nous présentons sur la figure 8 un extrait du dictionnaire que nous avons construit pour déterminer l'appartenance de chaque adresse. Les clés du dictionnaire (les valeurs numériques) sont les adresses, et les valeurs sont les labels. Sur la figure, on voit que l'adresse numéro 35468562 appartient à CryptoTorLocker2015. Comme on peut le voir, nous exagérons un peu sur l'utilisation du terme *vraie identité* de l'utilisateur, le terme label est plus approprié. Associer un label à un utilisateur n'est pas *absolument* équivalent à connaître sa personnalité juridique. Dans certains cas ce l'est, d'autres pas. Pour les clés publiques qui collectent des dons [12] et qui ne sont pas des arnaques, tels que des ONG (l'UNICEF, Save the Children,...), le financement de projets open Sources, etc, on connaît leur identité juridique (noms, date de naissance...). C'est le cas pour les wallets, places boursières, certains services de mixage et certains vendeurs. Au contraire, pour des noms tels que *7ev3vn* ou *Zwai12*, on ne peut rien déduire de leurs vrais identités. Cependant l'Etat le pourrait s'il le souhaite.

Remarquons que le nombre d'identités trouvé (30 083) est très faible par rapport au nombre total d'adresses dans la blockchain (171 Millions), et par rapport au nombre total de clusters (74 Millions avec). Toutefois, il est possible d'identifier quelques milliers d'acteurs. Car l'association de l'identité d'un utilisateur à une

12. Quelques organisations qui acceptent des dons en bitcoins ICI

Cluster_0	Cluster_1	Cluster_3	Cluster_5	Cluster_6	Cluster_7	Cluster _8	Cluster_9
zakyzakj1	dipkiss	gudstuff	naiyana	booyakamix	clamander	luwei1984	kamalosman
chazley	YanNaing	che rosnani	Aekthada	az09za90za		Rojana Deesean	Deemo
starik69	SolarNick	sakdmmm	Luckydayman	bmbmb		obsarita	liburu
songyuanwei	kyawhlaing	Limtriang	pennee	1badbullitt		vpwdle23	pneumatic5
NxtStakeAssets	icash	LennonNZ	reallyfunny	Keitaru84		jiaxuan1023	Adam3mmm
Light	shawwal	Chris7sa	say pisit	meisenst			
Dexon	kavenlim	dongluo	gegist21	powderfan			vellfire
chenhui1	Kosterc	sergeyvademan	thainaraty	Chainsaw			houshan
firejuan	heinmyatsoe	piansmall	panida	Templeton			Zwai12
cook7096@gmail.com	mooen	kimheng	Boonlert	septarious			denis pevtsov
...	...	...	...	...	...	...	...
143	454	443	732	31	1	5	19

TABLE 7 – Identification des acteurs obtenus par le clustering h1

adresse implique que, le cluster dans lequel l'adresse se trouve appartient à cet utilisateur.

Dans les tables 7 et 8, nous présentons les dix plus gros clusters que nous avons réussi à *identifier*. Les colonnes du tableau contiennent les clusters (0 à 9, c'est les plus gros cluster en termes d'adresses), chacun contient les labels qui lui sont associés. Le tableau 7 contient les clusters obtenus par l'heuristique 1, parmi les 30 083 labels que nous avons collectés, 143 appartient à l'acteur qui a le plus d'adresses dans la blockchain, le cluster 0. Il a les noms suivants *zakyzakj1, starik69, cook7096@gmail.com,....* 454 noms appartiennent au deuxième plus gros groupe d'adresses, le Cluster 1. On remarque que dans ce top10 les clusters 2 et 4 sont absents, c'est lié au fait que nous n'avons trouvé aucun label qui appartenait à ces clusters. Dans le tableau 8 nous présentons les mêmes résultats pour les clusters obtenus par combinaison de l'heuristique 1 et l'heuristique 2. 16 291 labels appartiennent au plus gros clusters, le cluster 0 ; et 1 seul label pour le cluster 5. Le reste des clusters du top 10 ont 0 label associé. Ce résultat prouve encore une fois l'inefficacité de l'heuristique 2.

Idée d'amélioration du clustering

Il est possible d'améliorer le clustering que nous avons réalisé avec l'heuristique 1. Si deux ou plusieurs clusters ont des labels en commun on peut déduire que tous ces clusters appartiennent au même individu **si** aucun de ces clusters n'est impliqué dans du mixage. De cette manière, on pourra rassembler des clusters isolés qui appartiennent au même acteur. Mais pour que l'idée marche, il faut être sûr que les labels que nous avons recueilli sont exacts. Il faudrait aussi pouvoir caractériser et isoler les transactions qui impliquent du mixage. Il est aussi fort probable que l'heuristique 2 marche mieux si on arrive à isoler les transactions de mixage, de ventes ou d'achats de bitcoins, et si on arrive à déterminer sur quels unités monétaires (bitcoin ou monnaie fiduciaires) les utilisateurs échangent. C'est-à-dire, savoir si les utilisateurs pensent et échangent en termes de dollars ou satoshis.

Cluster_0	Cluster_5
wangtao13904039526	xiaoyuyu111
ShrektZombie	
indra leksana	
Zhang Tingting	
azhenmmm	
leoheart0924	
che rosnani	
LennonNZ	
chazley	
ipankkerz	
...	...
16291	1

TABLE 8 – Identification des acteurs obtenus par le clustering h1-h2

4.4 Construction du graphe des acteurs

Grâce à Neo4j, nous avons aussi construit le graphe des transactions et des acteurs. Non seulement ce graphe nous permettra de mieux comprendre ce qui se passe dans le réseau bitcoin, il représente également une base de données plus accessible de toutes les transactions de la blockchain. Dans sa construction, nous n'avons utilisé que les acteurs obtenus grâce à l'heuristique 1 uniquement. Nous avons décidé de ne pas prendre en compte l'heuristique 2 pour les raisons expliquées plus haut. Nous avons réussi à construire un graphe détaillé et aisé d'utilisation. Rappelons que sur Neo4j, il est possible d'ajouter des propriétés aux nœuds, et aux liens. Nous avons profité de cet avantage pour enrichir notre graphe.

Les Nœuds du Graphe

Le graphe dispose de trois catégories de nœuds que l'on appelle labels dans le langage de Neo4j. Le nombre total d'adresses uniques, le nombre total d'acteurs et le nombre total de transactions.

Addresses Les nœuds des adresses sont l'ensemble de toutes les adresses de la blockchain. Chaque nœud est une adresse qui a pour propriété le nom en string de l'adresse, le nom de l'identification à l'acteur auquel il appartient, et sa correspondance en valeur numérique.

Actors Les nœuds des acteurs sont l'ensemble des clusters, un cluster correspond à un nœud. Chaque nœud des acteurs a pour propriété, le numéro du cluster et le nombre total d'adresses que ce dernier contient. Nous avons fait en sorte que les numéros de clusters commençant par 0 (c-à-d. cluster 0, cluster 1, ...) soient

(a) Un noeud Adresse

(b) Un noeud Acteur (cluster)

(c) Un noeud Transaction

FIGURE 9 – Illustration des noeuds et leurs propriétés

les plus gros clusters par leur taille. Ainsi, on sait que le cluster 0 est celui qui a le plus d'adresses, le cluster 1 est le deuxième plus gros des clusters.

Transactions Les nœuds des transactions sont l'ensemble des transactions. Chaque transaction constitue un nœud qui a les propriétés suivantes, l'identifiant de la transaction, le timestamp, le taux d'échange et la valeur totale de bitcoin échangée dans la transaction. La transaction 1 est le nœud nommé Tx 1 dans le graphe. C'est la première transaction de l'histoire de la blockchain de bitcoin.

Dans le figure 10 nous présentons une illustration des 3 types de nœuds que notre graphe contient et leurs propriétés. Nous voyons que l'adresse qui est présentée contient le numéro d'adresse 71842, son adresse bitcoin (*bitcoin_adresse*) et le label associé (*actor_identity : alenasultan28*). Le nœud de l'acteur (cluster) qui est présenté contient le numéro du cluster et l'information de la taille du cluster *cluster_size*, ce cluster ne contient qu'une seule adresse. Le noeud qui s'appelle transaction de cet exemple, est la transaction 333 344 366.

Les Liens du Graphe

Nous avons créés deux types de liens entre les nœuds du graphes. Des liens entre les adresses et les transactions, des liens entre les adresses et clusters (ou acteurs), que nous appelons respectivement *IS_IN_TRANSACTION* et *BELONG_TO_ACTOR* dans Neo4j. Et dans chaque lien nous ajoutons dans les propriétés des informations supplémentaires qui nous aiderons plutard à réaliser facilement des requêtes.

IS_IN_TRANSACTION Nous créons un lien dirigé des adresses vers les transactions, si l'adresse apparait en entrée ou en sortie de la transaction. Dans les propriétés du lien nous précisions si l'adresse est en entrée ou en sortie de la transaction, la valeur en bitcoin associée (à l'entrée ou la sortie) et leurs identifiants uniques (le *hashPrevOut* ou *indexOut*).

BELONG_TO_ACTOR Nous créons un lien dirigé des adresses vers les acteurs (clusters), si l'adresse appartient à l'acteur. Autrement dit, si l'adresse fait partie des adresses du cluster. On remarque bien que, autant un cluster a d'adresses autant il a de liens qui pointent vers lui. A titre d'exemple, le plus gros cluster a environ 11 millions d'adresses, en conséquence il a aussi 11 millions de liens

(a) Lien transaction-adresse

(b) Lien acteur–adresse

FIGURE 10 – Illustration des liens et leurs propriétés

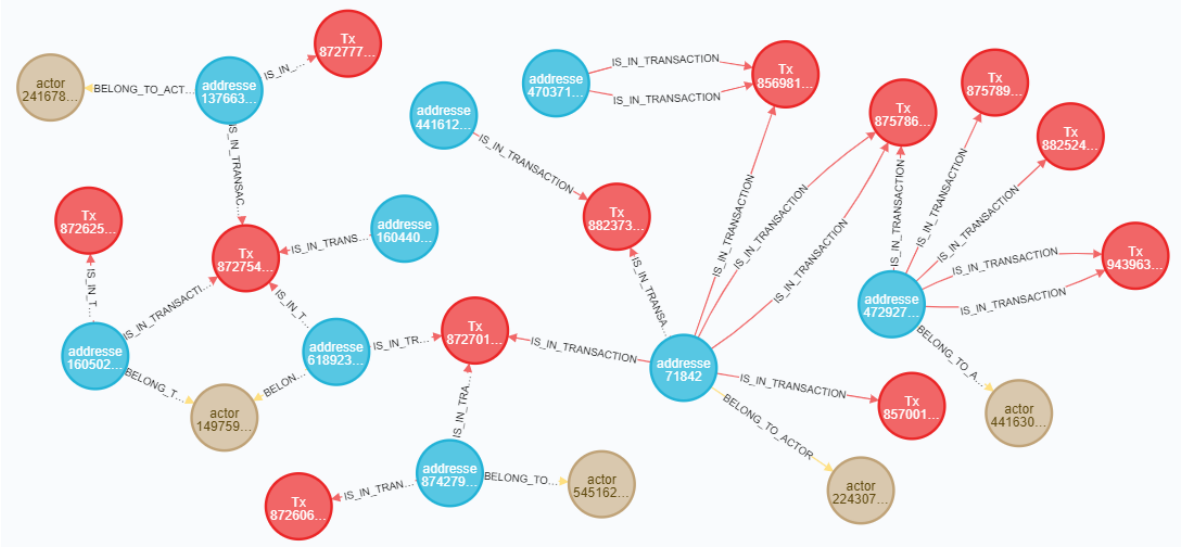

FIGURE 11 – Extrait du graphe du réseau de bitcoin que nous avons construit.

avec les nœuds de ses adresses. Les propriétés des liens sont uniquement les noms identifiés des propriétaires des adresses.

Dans la figure 10 nous présentons une illustration des 2 types de liens que nous avons créés et des informations qu'ils contiennent. Pour le lien entre l'adresse et la transaction, ce lien nous dit que l'adresse qui est associée est en sortie de la transaction (*Tx_INS_or_OUTS : out*). Dans les tableaux et , nous présentons le nombre total de noeuds et de liens.

Types	Nombres
IS_IN_TRANSACTION	829981034
BELONG_TO_ACTOR	171484007

TABLE 9 – Liens de chaque type.

Labels	Nombres
Addresses	171484007
Actors	74264935
Transactions	152673373

TABLE 10 – Noeuds de chaque catégorie.

Dans la figure 11 nous présentons un extrait graphe que nous avons construit. Les nœuds en couleur bleue sont les adresses, en couleur jaunes ce sont les clusters que

nous appelons acteurs, en couleur rouge ce sont les noeuds des transactions. Pour chaque transaction, pour savoir si une adresse est en entrée ou sortie et la valeur de bitcoin qu'elle stocke, il faut regarder dans les propriétés des liens de couleur rouge. Un noeud adresse qui est liée à un noeud acteur signifie que cette adresse appartient à cet acteur. Sur la figure, on peut voir que l'adresse 71842 appartient à l'acteur 224307..., elle participe dans 3 transactions (en sortie ou en entrée). Pour savoir s'il y a eu une transaction entre deux acteurs, il faut voir si les adresses qui sont liées à ces acteurs sont liée à une même transaction. Ainsi sur la figure, on voit que l'adresse 71842 qui appartient à l'acteur 224307... est connectée à la transaction Tx 872701... de même l'adresse 874279... qui appartient à l'acteur 545162... est connectée à la même Tx 872701..., donc il y'a eu une transaction entre les acteurs 224307... et 545162... .

4.4.1 Quelques résultats que l'on peut extraire de ce graphe

Les résultats obtenus à partir des listes avaient été entièrement réalisés sur Python à l'exception du clustring, nous pouvons retrouver les mêmes résultats sur le graphe que nous avons construit, notamment le nombre d'adresses total, les adresses plus utilisées dans la blockchain, le type de transactions le plus fréquent, etc. On peut trouver plusieurs autres informations de la blockchain dans ce graphe. A titre d'illustration, nous avons cherché les informations suivantes dans le graphe, mais nous n'avons pas eu le temps de présenter les résultats dans ce rapport.

- Nous avons calculé des statistiques sur les adresses :
  - Est-ce que les adresses les plus fréquentes dans la blockchain sont les plus riches en bitcoins ;
  - Les adresses les plus fréquentes appartiennent à quels acteurs ? au plus gros acteurs ?
- Sur les acteurs nous avons cherché :
  - Qui sont les acteurs les plus riches,
  - Quelles sont les sont les acteurs qui font le plus de transactions ? sont ils les plus riches ?
  - Quels sont les acteurs qui font le plus d'échanges entre eux.
- Statistiques sur les valeurs :
  - Quelles sont les valeurs les plus fréquentes dans la blockchain.
  - Les valeurs les plus élevées ont été réçues/envoyées par quelles adresses, dans quelles transactions ?
- Statistiques sur le minage :
  - On peut retrouver le nombre de minage dans la blockchain.
  - On peut étudier les frais de transactions, leur évolutions dans le temps.
  - Qui sont les mineurs les plus riches ?

On peut croiser tous ces résultats dans Neo4j, on peut choisir d'étudier une seule ou plusieurs adresses, on peut analyser un seul ou plusieurs acteurs dans la blockchain, on peut étudier des transactions, etc. Nous pouvons déterminer la part des

types transaction (part d'activités illégales, part de mixing, ...). Notons que nous avons rencontré aussi un problème de mémoire RAM avec ce graphe et certains calculs prennent plus temps. Une solution que l'on peut suggérer pour résoudre ce problème est d'éliminer les noeuds isolés qui ne sont connectés à aucun autre noeud (sauf si on veut les étudier bien sûr). Aussi pour certaines analyses on peut juste extraire le sous graphe qui nous intéresse à étudier.

5 Conclusion

Au cours de ce stage nous avons essentiellement réussi à traiter les données d'origines de la blockchain et grouper les adresses de chaque utilisateur, et nous avons aussi construit le graphe des utilisateurs.

Nous avons commencé ce rapport de stage par présenter la blockchain de bitcoin. Nous avons tenté de faire une présentation de sorte que le lecteur non initié puisse comprendre. Nous avons tenté de faire comprendre ce que c'était une adresse bitcoin, une transaction, comment la blockchain fonctionne et qui sont les acteurs majeurs. Toute suite après, nous avons effectué une revue de littérature de papier clés sur l'analyse des données de la blockchain. Le but de cette revue de littérature a été de faire savoir au lecteur qu'il existe une quantité énorme d'informations que l'on peut extraire dans la blockchain. Par exemple malgré l'anonymat qu'on attribue à la blockchain on peut identifier plusieurs acteurs et la nature de leurs activités.

Ensuite, nous avons tenté d'analyser les transactions de la blockchain de bitcoin entre la période de 2009 et 2016. Nous avons expliqué comment nous avons réussi à grouper les adresses des utilisateurs en nous servant des heuristiques h1 et h2. Nous avons montré que le résultat que l'on obtient n'était pas satisfaisant si on combine les deux heuristiques, car nous obtenons un super cluster. Le résultat obtenu uniquement avec l'heuristique 1 nous a montré que, 72% des groupes ne contenaient qu'une seule adresse. Ce qui veut dire, entre autres que, pour de nombreuses adresses nous n'avons pas trouvé son utilisateur ou que peu d'utilisateurs réutilisent les mêmes adresses. Il faut considérer ces résultats en prenant en compte qu'il y'avait quelques données manquantes dans les données que nous avons étudiées.

Nous avons aussi montré que l'on pouvait trouver les vrais identités de certains acteurs qui laissent leurs informations personnelles sur internet et leurs adresses publiques. Ensuite, nous avons construit le graphe qui combine les transactions, les utilisateurs et les adresses publiques de bitcoin sur Neo4j. Nous avons présenté en détails le graphe, et nous avons donné des exemples sur la manière de l'utiliser. Nous avons montré que grâce a ce graphe, on pouvait facilement et rapidement étudier un acteur ou une adresse dans la blockchain et extraire des données en faisant des requêtes dans le graphe.

Dans la suite de ce projet, nous pensons qu'il serait intéressant de développer plus d'heuristiques pour appuyer l'heuristique 1. Avec le graphe que nous disposons, on peut créer des profils types pour certaines catégories d'utilisateurs en cher-

chant des généralités dans leurs comportements. Par exemple, c'est possible de faire cela pour le mixage. On peut collecter plusieurs adresses de services de mixage sur internet, on peut aussi interagir avec ces services pour connaître leurs adresses, ensuite on peut tenter de comprendre ce qu'il y a de commun entre les comportements des acteurs à qui ces adresses appartiennent.

Compétences acquises dans ce stage

Durant ce stage j'ai eu la chance d'acquérir des compétences et d'améliorer ma compréhension dans plusieurs domaines. Notamment, j'ai acquis :

1. Une bonne compréhension du fonctionnement de bitcoin et de sa blockchain.
2. Des notions de base des théories des graphes à travers,
   — Les cours en lignes de mon Superviseur de stage Maître Cazabet, et un cours de Coursera "social and economic networks".
   — Analyse des graphes sur Python avec Networkx et Gephi.
3. Une meilleure compréhension de Python et certaines librairies d'analyse de données telles que (Pandas, Numpy,...)
4. Construire des graphes sur Neo4j et le langage Cypher.
5. Une meilleure compréhension des systèmes d'exploitation Windows et Linux.
6. Des compétences en analyse de grosses bases de données.

Références

[1] Bitcoin Wiki, . URL https://en.bitcoin.it/wiki/Main_Page.

[2] Blockchain.com, . URL https://www.blockchain.com.

[3] Blockchain for Advertising & Media - IBM Blockchain, . URL https://www.ibm.com/blockchain/industries/advertising-media.

[4] Cryptocurrency Market Capitalizations, . URL https://coinmarketcap.com/.

[5] How Bitcoin Works - freedomnode.com, September 2016. URL https://freedomnode.com/guides/17/how-bitcoin-works.

[6] Israa Alqassem, Iyad Rahwan, and Davor Svetinovic. The Anti-Social System Properties : Bitcoin Network Data Analysis. *IEEE Transactions on Systems, Man, and Cybernetics : Systems*, pages 1–11, 2018. ISSN 2168-2216, 2168-2232.

[7] Elli Androulaki, Ghassan O. Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. Evaluating User Privacy in Bitcoin. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell,

Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, and Ahmad-Reza Sadeghi, editors, *Financial Cryptography and Data Security*, volume 7859, pages 34–51. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-39883-4 978-3-642-39884-1. doi : 10.1007/978-3-642-39884-1_4.

[8] Susan Athey, Ivo Parashkevov, Vishnu Sarukkai, and Jing Xia. Bitcoin pricing, adoption, and usage : Theory and evidence. 2016.

[9] Massimo Bartoletti, Andrea Bracciali, Stefano Lande, and Livio Pompianu. A general framework for blockchain analytics. *arXiv :1707.01021 [cs]*, July 2017. arXiv : 1707.01021.

[10] Massimo Bartoletti, Tiziana Cimoli, Livio Pompianu, and Sergio Serusi. Blockchain for social good : a quantitative analysis. *Proceedings of the 4th EAI International Conference on Smart Objects and Technologies for Social Good - Goodtechs '18*, pages 37–42, 2018. arXiv : 1811.03424.

[11] Massimo Bartoletti, Barbara Pes, and Sergio Serusi. Data mining for detecting Bitcoin Ponzi schemes. March 2018. arXiv : 1803.00646.

[12] Annika Baumann, Benjamin Fabian, and Matthias Lischke. Exploring the Bitcoin Network. In *WEBIST (1)*, pages 369–374, 2014.

[13] Damiano Di Francesco Maesa, Andrea Marino, and Laura Ricci. Data-driven analysis of Bitcoin properties : exploiting the users graph. *International Journal of Data Science and Analytics*, 6(1) :63–80, August 2018. ISSN 2364-415X, 2364-4168.

[14] Damiano Di Francesco Maesa, Andrea Marino, and Laura Ricci. The Graph Structure of Bitcoin. In Luca Maria Aiello, Chantal Cherifi, Hocine Cherifi, Renaud Lambiotte, Pietro Lió, and Luis M. Rocha, editors, *Complex Networks and Their Applications VII*, volume 813, pages 547–558. Springer International Publishing, Cham, 2019. doi : 10.1007/978-3-030-05414-4_44.

[15] Doug Galen, Nikki Brand, Lyndsey Boucherle, Rose Davis, Natalie Do, Ben El-Baz, Isadora Kimura, Kate Wharton, and Jay Lee. Blockchain for Social Impact : Moving Beyond the Hype. *Center for Social Innovation*, 2018.

[16] Angela S.M. Irwin and Adam B. Turner. Illicit Bitcoin transactions : challenges in getting to the who, what, when and where. *Journal of Money Laundering Control*, 21(3) :297–313, July 2018.

[17] Dániel Kondor, Márton Pósfai, István Csabai, and Gábor Vattay. Do the rich get richer ? An empirical analysis of the BitCoin transaction network. *PLoS ONE*, 9 (2) :e86197, February 2014. arXiv : 1308.3892.

[18] Damiano Di Francesco Maesa, Andrea Marino, and Laura Ricci. The bow tie structure of the Bitcoin users graph. *Applied Network Science*, 4(1), December 2019. ISSN 2364-8228. doi : 10.1007/s41109-019-0163-y.

[19] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage. A fistful of bitcoins : characterizing payments among men with no names. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 127–140. ACM, 2013.

[20] Satoshi Nakamoto. Bitcoin : A peer-to-peer electronic cash system. 2008.

[21] Francesco Parino, Mariano G. Beiró, and Laetitia Gauvin. Analysis of the Bitcoin blockchain : socio-economic factors behind the adoption. *EPJ Data Science*, 7(1), December 2018. ISSN 2193-1127.

[22] Deepa Pavithran and Rajesh Thomas. A Survey on Analyzing Bitcoin Transactions. In *2018 Fifth HCT Information Technology Trends (ITT)*, pages 227–231, Dubai, United Arab Emirates, November 2018. IEEE. ISBN 978-1-5386-7147-4.

[23] Dorit Ron and Adi Shamir. Quantitative Analysis of the Full Bitcoin Transaction Graph. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, and Ahmad-Reza Sadeghi, editors, *Financial Cryptography and Data Security*, volume 7859, pages 6–24. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[24] Wei Shao, Hang Li, Mengqi Chen, Chunfu Jia, Chunbo Liu, and Zhi Wang. Identifying Bitcoin Users Using Deep Neural Network. In Jaideep Vaidya and Jin Li, editors, *Algorithms and Architectures for Parallel Processing*, volume 11337, pages 178–192. Springer International Publishing, Cham, 2018.