



HAL
open science

Vers une meilleure identification d'acteurs de Bitcoin par apprentissage supervisé

Rafael Ramos Tubino, Rémy Cazabet, Céline Robardet

► To cite this version:

Rafael Ramos Tubino, Rémy Cazabet, Céline Robardet. Vers une meilleure identification d'acteurs de Bitcoin par apprentissage supervisé. Conférence francophone sur l'Extraction et la Gestion des Connaissances (EGC 2022), Jan 2022, Blois, France. pp.171-182. hal-04193274

HAL Id: hal-04193274

<https://hal.science/hal-04193274>

Submitted on 1 Sep 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Vers une meilleure identification d'acteurs de Bitcoin par apprentissage supervisé

Rafael Ramos Tubino*, Rémy Cazabet*
Céline Robardet**

*Univ Lyon, Université Lyon 1, CNRS, LIRIS UMR5205, F-69622 France

**Univ Lyon, INSA Lyon, CNRS, LIRIS UMR5205, F-69621 France
prenom.nom@liris.cnrs.fr

Résumé. Bitcoin est la crypto-monnaie la plus largement répandue et la plus étudiée. De par sa nature décentralisée, les données de transactions sont librement accessibles et peuvent être analysées. La première étape de la plupart des analyses consiste à regrouper les adresses anonymes en agrégats supposés correspondre à des acteurs. Dans cet article, nous proposons une nouvelle méthode pour réaliser ces agrégats à base d'apprentissage automatique. Notre approche repose sur la construction d'un jeu de données d'apprentissage dont la variable de classe est obtenue par une vérité de terrain calculée a posteriori. Ce jeu de données est utilisé pour identifier les adresses de change des transactions, adresses appartenant au donneur d'ordre de la transaction. Cela nous permet d'augmenter le nombre d'adresses découvertes appartenant à un même acteur. Nous montrons expérimentalement la pertinence de cette méthode en comparaison des heuristiques habituellement utilisées à l'aide d'un critère de validation externe.

1 Introduction

Présentée en 2008 et créée l'année suivante, Bitcoin (BTC) est une monnaie numérique basée sur le principe de la chaîne de blocs (Blockchain). Sa spécificité tient à plusieurs caractéristiques. Tout d'abord, Bitcoin est une monnaie **décentralisée**, c'est-à-dire que le système fonctionne sans autorité centrale, ni administrateur unique. Elle est gérée par un réseau de nœuds validateurs ouvert à tous, ordinateurs mettant à contribution leur puissance de calcul informatique afin de vérifier, de sécuriser et d'inscrire les transactions dans la blockchain. Ces nœuds assurent **la fiabilité** du système, puisque chaque transaction doit être validée par la majorité des nœuds validateurs pour être enregistrée. Enfin, **l'anonymat** est garanti par le chiffrement des identités des bénéficiaires et des donneurs d'ordre, même si toutes les transactions effectuées sont enregistrées dans un registre public. C'est ce dernier point qui rend la blockchain pseudo-anonyme, car les bénéficiaires et donneurs d'ordre des transactions sont identifiés par leur adresse publique. Il est donc possible de tracer toutes les transactions impliquant une même adresse. Afin d'accroître l'anonymat, un même acteur change donc régulièrement d'adresse. Nous définissons un **acteur** comme une personne, un groupe de personnes, une entreprise, ou une quelconque entité qui détient un ensemble de clés privées, et donc un ensemble

de clés publiques – adresses Bitcoin – correspondantes. Préalablement à l'analyse de l'activité d'un acteur, il est nécessaire d'identifier l'ensemble d'adresses lui appartenant. Dans la suite, nous nommons un ensemble d'adresses regroupées dans l'objectif d'identifier un acteur un **agrégat d'adresses**.

Précisons que l'identification d'acteurs telle que nous l'avons définie se différencie du problème de la désanonymisation de comptes Bitcoin – un problème plus étudié dans la littérature – qui vise à déterminer l'identité réelle d'un acteur déterminé par un agrégat d'adresses, ou pour le moins son domaine d'activité (portefeuille, jeux d'argent, minage, etc.). Une **transaction** consiste à débiter certains comptes (les donneurs d'ordre) pour en créditer d'autres (les bénéficiaires). A l'exception des transactions de minage (génération de Bitcoins), toutes les transactions possèdent une ou plusieurs entrées et une ou plusieurs sorties (voir Fig. 1a). La somme des valeurs des entrées est égale à la valeur totale de la transaction. Symétriquement, cette somme est égale à la somme des sorties plus les éventuels frais de réalisation de la transaction. Toute entrée doit être la sortie d'une transaction précédente, et la valeur reçue doit être utilisée dans sa totalité. Ainsi, comme les montants ne peuvent pas être changés entre deux transactions, il est très courant que dans une transaction, l'une des adresses de sortie appartienne au même propriétaire que les adresses d'entrée, afin de recevoir l'excédent payé, c'est-à-dire le change de la transaction. Par exemple, dans la figure 1b, l'adresse @c appartient au même acteur que les adresses @a. L'adresse correspondant au change est appelée **adresse de change**. La méthode que nous proposons a pour objectif de détecter automatiquement si une adresse en sortie est un paiement ou une adresse de change. Si c'est le cas, cette adresse appartient également à l'acteur à l'origine de la transaction, et nous pouvons l'ajouter à l'agrégat d'adresses de cet acteur.

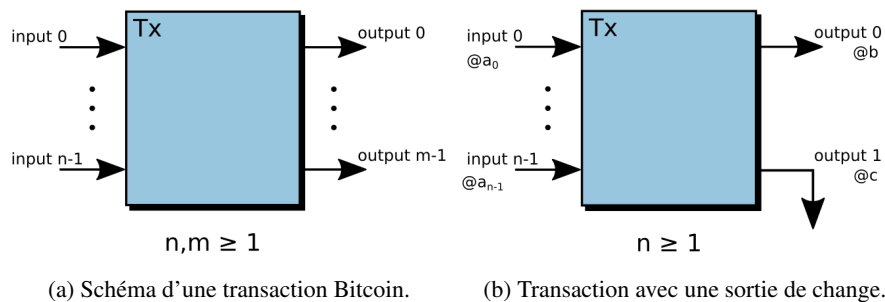


FIG. 1: Illustration de transactions Bitcoin.

L'article est organisé comme suit : dans la section 2, nous introduisons les travaux de l'état de l'art liés à notre contribution. La section 3 présente la construction des jeux de données. La section 4 introduit notre méthode pour découvrir des agrégats d'adresses. Enfin, notre évaluation empirique est présentée dans la section 5. Nous concluons et présentons des perspectives à ce travail dans la section 6.

2 État de l’art

L’identification d’acteurs de Bitcoin par découverte d’agrégats d’adresses est une tâche clé nécessaire à la plupart des travaux d’analyse des activités liées à cette crypto-monnaie, dès que l’on souhaite aller au-delà de statistiques macroscopiques (nombre et fréquence des transactions, quantités moyennes échangées, etc.). Plusieurs méthodes ont déjà été proposées dans la littérature.

2.1 Méthodes à base d’heuristiques

Historiquement, cette tâche a d’abord été faite à l’aide d’heuristiques, dont la logique repose sur le protocole de Bitcoin ou sur des observations de comportement. L’heuristique probablement la plus répandue et la plus largement acceptée dans la littérature (e.g., Meiklejohn et al. (2013); Harrigan et Fretter (2016)), que nous nommerons ici heuristique **H1**, définit que toutes les adresses en entrée d’une même transaction appartiennent à un même acteur (le donneur d’ordre). Cela permet de découvrir par transitivité de larges agrégats d’adresses (plusieurs millions pour de gros acteurs). La pertinence de cette heuristique a régulièrement été démontrée, notamment dans (Harrigan et Fretter, 2016).

Une autre heuristique fréquemment trouvée dans la littérature, sous différentes variantes (e.g., Androulaki et al. (2013); Meiklejohn et al. (2013)) est l’heuristique de sortie de change **H2** qui consiste à identifier l’adresse de change en utilisant un système de règles ad-hoc : par exemple, SI il y a deux sorties ET que l’une des adresses en sortie n’a jamais été vue auparavant ET que l’autre adresse a déjà été vue auparavant ALORS on peut supposer que la sortie qui n’a jamais été vue correspond à une adresse de change (e.g., (Meiklejohn et al., 2013)).

L’heuristique H1 a l’avantage d’offrir un résultat avec une précision proche de 1 : toutes les adresses qu’elle associe dans un agrégat appartiennent réellement au même acteur, en l’absence de – rares – méthodes d’obfuscation spécifiques (cf. CoinJoin, Sec. 3.2). En revanche, on sait qu’elle manque de nombreuses associations : des agrégats d’adresses appartenant à un même acteur sont détectés comme distincts. H2 a été proposée pour pallier cette faiblesse. Cependant son approche déterministe – un seul exemple correspondant aux règles établies suffit à considérer deux agrégats comme en formant un seul – est clairement insatisfaisante et conduit à un effondrement de la précision (Cazabet et al., 2018). Lorsqu’elle est utilisée en pratique pour analyser les acteurs de Bitcoin, des opérations de correction doivent être effectuées manuellement (Meiklejohn et al., 2013). Une étude empirique comparant l’efficacité de ces deux heuristiques et de leurs variantes a été réalisée dans (Nick, 2015).

2.2 Identification d’agrégats d’acteurs par apprentissage supervisé et non supervisé

Peu de méthodes ont été proposées dans la littérature pour la découverte d’acteurs en utilisant des méthodes d’apprentissage machine. Deux articles (Shao et al., 2018; Liu et al., 2020) ont proposé d’utiliser des réseaux de neurones pour résoudre un problème de classification : en présentant deux adresses caractérisées par un ensemble d’attributs descriptifs, l’algorithme apprend à les classer comme appartenant à un même acteur ou non. A partir de cet apprentissage, un plongement (*embedding*) est effectué, ce qui permet soit de ré-identifier le propriétaire d’une

adresse particulière via ses k plus proches voisins, soit de découvrir des agrégats d'adresses par clustering non ou semi-supervisé. Une méthode non supervisée, basée sur la détection de communautés (clustering de graphe) dans un graphe de signaux faibles, a été proposée dans (Cazabet et al., 2018).

Une faiblesse de ces approches est que leur efficacité n'a pas pu être évaluée quantitativement à grande échelle, faute de vérité de terrain satisfaisante. Par exemple, Cazabet et al. (2018) n'utilise que 776 adresses identifiées pour valider la qualité de son clustering, tandis que Shao et al. (2018) en utilise 8986. Par comparaison, notre approche nous permet d'avoir plus de 500 000 adresses dans notre vérité de terrain.

Enfin, un article en preprint (encore non publié) proposant une approche relativement similaire à la nôtre a été déposé sur ArXiv très récemment (Möser et Narayanan, 2021). Comme le nôtre, il propose d'utiliser des méthodes de machine learning pour identifier l'adresse de change. Il résout également le problème de la quantité de données d'apprentissage en utilisant l'heuristique H1 pour obtenir par analyse *a posteriori* des exemples d'entraînement (cf Section 3). Au-delà des différences d'implémentation (variables considérées, algorithme d'apprentissage), notre travail se différencie sur plusieurs aspects importants : 1) Notre vérité de terrain est plus robuste, grâce à l'usage d'une vérité externe, comme expliqué dans la section 3.1. 2) Dans notre méthode, nous divisons le jeu de données temporellement (voir section 3.1). Au contraire, dans (Möser et Narayanan, 2021), les auteurs considèrent une seule période pour l'apprentissage et l'évaluation. Ils font leur apprentissage en se basant sur un sous-ensemble des acteurs, et testent sur un autre sous-ensemble, ce qui impose de ne travailler que sur un échantillon des acteurs (impossibilité de rassembler un acteur du jeu d'entraînement et du jeu de test). 3) Les résultats de la méthode sont évalués quantitativement uniquement en terme d'identification positive ou erronée d'une adresse de change. Nous pensons que cela est biaisé, car une seule identification peut avoir un effet majeur (fusion de deux grands agrégats appartenant à des acteurs différents), là où plusieurs erreurs peuvent avoir un faible impact (mauvaise identification d'adresses appartenant à des agrégats de petite taille). Les auteurs observent d'ailleurs un problème d'*agrégat géant* dans leurs résultats.

3 Jeux de données

La totalité des données de la blockchain de Bitcoin étant accessibles à tous, nous avons collecté les données de transactions via un nœud Bitcoin, en suivant une procédure standard équivalente à (Emery et Latapy, 2021). Les transactions utilisées sont celles entre le bloc 0 (du 3 janvier 2009, date de la mise en marche de Bitcoin) et le bloc 667542 (25 janvier 2021, date de l'obtention des données), soit environ 600 millions de transactions. Pour chaque transaction, nous disposons des adresses en entrée (donneur d'ordre) et en sortie (bénéficiaires), des montants concernés pour chacune des adresses de la transaction, des frais payés, et du timestamp du bloc auquel appartient la transaction.

Nous avons séparé cet ensemble de transactions en deux parties (voir Fig. 2) : un **jeu de données d'étude (D-2017)** constitué des données jusqu'au bloc 501950 (31/12/2017), et un jeu de données pour calculer la vérité de terrain *a posteriori*, contenant l'ensemble des données (**D-2021**). Nous avons choisi d'arrêter le jeu de données d'étude en 2017 car les acteurs identifiés dans la source externe que nous utilisons dans notre vérité de terrain sont connus pour être identifiés de manière robuste jusqu'à cette période.

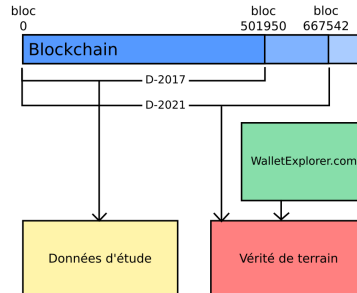


FIG. 2: Génération des données d'étude (D-2017) et des données pour le calcul de la vérité terrain (D-2021). Les données de la vérité de terrain sont qualifiées grâce à la source de données externe WalletExplorer.com).

3.1 Jeu de données d'apprentissage et de test

Notre méthode nécessite d'apprendre de manière supervisée à reconnaître des adresses de change. Nous proposons donc de constituer un corpus d'adresses de change connues par analyse a posteriori.

Création de la variable de classe à l'aide des agrégats H1 a posteriori (H1-AP). Pour construire la variable de classe, nous procédons comme suit. Nous calculons les agrégats à l'aide de l'heuristique H1 sur l'ensemble des données (D-2021), puis nous créons, à partir de ces agrégats a posteriori, des agrégats sur la période D-2017 tels que deux adresses présentes dans D-2017 appartiennent au même agrégat si et seulement si elles appartiennent à un même agrégat calculé sur D-2021. Les agrégats obtenus ainsi sur D-2017 sont appelés **H1-2017** et ceux créés à partir de D-2021 sont appelés **H1-AP** (A Posteriori).

Nous utilisons cette information pour créer la variable de classe sur la période d'étude D-2017 : si l'une des adresses de sortie appartient au même agrégat que les adresses d'entrée de la même transaction, il s'agit d'une adresse de change.

Création des jeux de données d'apprentissage et de test. Chaque adresse de sortie constitue un exemple de nos jeux de données. Comme nous cherchons à ré-identifier des adresses de change mal identifiées, nous affectons les exemples dans la base d'apprentissage ou de test en fonction du nombre d'adresses de change associées à une même transaction. Nous nous limitons aux transactions avec deux adresses en sortie, qui représentent le cas le plus fréquent et a priori le plus interprétable (une sortie de paiement et une de change). Nous pouvons donc séparer les transactions de la période d'étude en trois groupes : les transactions contenant zéro, une ou deux adresses de change, d'après l'heuristique H1. Nous nommons ces données les datasets 00, 01 et 11 respectivement.

- Les transactions avec zéro adresses de change d'après H1-2017 (00) constituent notre jeu de données de **test**. Nous savons que l'adresse de change n'a pas été encore identifiée et nous cherchons à la prédire.

Vers une meilleure identification d'acteurs de Bitcoin par apprentissage supervisé

- Les transactions à une seule adresse de change d'après H1-2017 (01) seront utilisées lors de l'étape d'apprentissage (données **train**). Elles peuvent contenir des adresses de change non reconnues, mais notre priorité étant d'éviter les faux positifs, les faux négatifs sont des erreurs acceptables.
- Les transactions à deux adresses de change (11) sont reconnues comme étant des adresses de change bien reconnues. Cependant elles ne seront pas utilisées lors de la phase d'apprentissage pour avoir un même nombre d'exemples labélisés vrai ou faux dans le jeu de données d'apprentissage.

Variables descriptives des exemples. Chaque exemple est décrit à l'aide de 16 variables construites à partir des informations disponibles dans la blockchain :

- la valeur totale en entrée de la transaction (en satoshi ¹);
- la valeur totale en sortie de la transaction (en satoshi);
- la valeur de la sortie (en satoshi);
- le nombre de chiffres différents de zéro parmi les huit derniers chiffres de la valeur de la sortie (en satoshi) (l'idée est ici de quantifier si la valeur de sortie est une valeur arrondie);
-
- le nombre de valeurs décimales de la valeur en Bitcoin;
- le pourcentage de la valeur de sortie par rapport à la valeur totale de la transaction;
- l'indice de la sortie;
- le nombre d'entrées;
- le jour de la semaine;
- l'année;
- le mois;
- le jour du mois;
- les frais payés dans la transaction;
- le nombre de fois où l'adresse de sortie est déjà apparue avant dans la blockchain;
- un booléen indiquant si la valeur de la sortie est supérieure à la somme des frais de la transaction et la plus grande valeur parmi les entrées.

3.2 Vérité de terrain externe pour l'évaluation

Dans les travaux de l'état de l'art, la vérité de terrain est constituée 1) soit d'un petit nombre d'adresses dont les acteurs sont connus (Cazabet et al., 2018; Shao et al., 2018), 2) soit par la méthode **H1 a posteriori** (Möser et Narayanan, 2021) elle-même.

Ceci peut être problématique pour plusieurs raisons : 1) Ce que nous considérons comme la vérité terrain correspond en fait à des informations qui auraient toujours pu être obtenues par H1, et non à la découverte d'informations véritablement nouvelles; 2) Au contraire, un résultat négatif peut en fait correspondre à deux agrégats connus comme appartenant au même acteur par une source externe, donc à une information réellement nouvelle et plus pertinente que H1; 3) La méthode est sensible aux faux positifs de H1, possibles par une méthode de mixing connue sous le nom de CoinJoin (Maurer, 2016). Les auteurs de (Möser et Narayanan,

1. satoshi : la plus petite unité du système Bitcoin. Équivaut à 0,00000001 BTC

2021) reconnaissent ce problème comme une limite de leur travail, même s'ils ont retiré de leur jeu de données quelques exemples problématiques évidents.

Pour résoudre ces problèmes, nous nous sommes basés sur le site WalletExplorer², fréquemment utilisé dans la littérature pour reconnaître des acteurs (Sun et al., 2019; Ermilov et al., 2017; Möser et Narayanan, 2021). On y trouve, pour un certain nombre d'acteurs connus de Bitcoin, un ensemble sélectionné d'agrégats d'adresses. Bien que les détails précis de la constitution de ces agrégats ne soient pas publiés, il s'agit de la source la plus fiable publiquement accessible. Les agrégats d'adresses fournis correspondent, d'après la documentation, au résultat de l'heuristique H1. À un agrégat d'adresses est attribué le nom d'un acteur par collection manuelle : si une des adresses de l'agrégat est connue comme appartenant à l'acteur A_1 , alors toutes les adresses de l'agrégat appartiennent à A_1 .

Pour de nombreux acteurs, WalletExplorer associe donc plusieurs agrégats H1. Notre vérité de terrain consiste donc à 1) ne garder qu'un sous-ensemble des agrégats H1 a posteriori obtenus, ceux qui sont validés par WalletExplorer, afin d'éviter d'introduire des faux positifs (CoinJoin) et des faux négatifs (agrégats multiples pour un acteur) dans notre vérité de terrain, et 2) regrouper les agrégats H1 connus comme appartenant à un même acteur, afin de corriger des faux négatifs dans notre vérité de terrain.

A notre connaissance, c'est la première fois que cette approche est utilisée dans la littérature pour obtenir une vérité de terrain robuste, différente de l'heuristique H1 elle-même. Nous pouvons représenter cette vérité de terrain sous la forme d'un ensemble de triplets $G = \{(a_i, c_i, o_i)\}$ avec : a, l'adresse Bitcoin; c, un numéro identifiant l'agrégat obtenu par application de l'heuristique H1 *a posteriori*; et o, l'acteur possédant cette adresse d'après notre source externe. La vérité de terrain H1 a posteriori (**H1-AP**), donnée pour référence, est définie comme $\{H_1, \dots, H_N\}$ avec $H_n = \{a_i \mid \exists (a_i, c_i, o_i) \in G, c_i = n\}$, tandis que la vérité de terrain Acteurs d'après WalletExplorer (**A-WE**) est définie comme $\{G_1, \dots, G_L\}$ avec $G_\ell = \{a_i \mid \exists (a_i, c_i, o_i) \in G, o_i = \ell\}$.

Pour des raisons de performance et de fiabilité, nous nous sommes concentrés dans cette étude sur un sous-ensemble comprenant les adresses et transactions de 6 acteurs de WalletExplorer (Bter.com, PrimeDice.com, BitcoinVideoCasino.com, FaucetBOX.com, BTCCPool et BitZino.com). Ces acteurs ont été choisis pour leur taille, leur diversité (Minage, *Exchanges*, Jeu d'argent), et le fait que leurs données étaient riches sur la période d'étude. Avoir 6 acteurs seulement est une limite, mais nous avons pris le parti de préférer des données complètes et de qualité sur un faible nombre d'acteurs, plutôt que parcellaires et pouvant être biaisées sur un plus grand nombre d'acteurs mal connus.

Nous disposons finalement de 520 578 adresses appartenant aux 6 acteurs sélectionnés, de 354 006 exemples d'entraînement d'après H1-AP (adresses de sorties, 50% d'adresses de change et 50% d'adresses de paiements), ainsi que de 2 557 002 adresses en sortie à évaluer comme étant ou non des adresses de change.

4 Méthode d'identification d'acteurs

La méthode que nous proposons consiste à identifier de manière automatique les adresses de change d'une transaction – adresse en sortie appartenant au même acteur que les adresses

2. <https://www.walletexplorer.com>

d'entrée – et d'utiliser cette information pour fusionner des agrégats d'adresses obtenus par l'heuristique H1 sur la période d'étude (H1-2017). En effet, l'heuristique H1, qui stipule que toutes les adresses en entrée d'une même transaction appartiennent à un même acteur (voir section 2), permet d'identifier des groupes d'acteurs avec pas ou très peu de faux positifs (adresses dans un même agrégat appartenant à des acteurs différents), mais avec de nombreux faux négatifs (adresses appartenant au même acteur positionnés dans des agrégats différents). L'objectif est donc de faire diminuer les faux négatifs, en évitant autant que possible d'introduire des faux positifs.

4.1 Prédire l'adresse de change par classification

Nous utilisons le jeu de données **train** pour apprendre un modèle de classification d'adresses de sortie de transactions.

Apprentissage d'un arbre de décision. Nous avons choisi d'apprendre un modèle interprétable sous forme d'un arbre de décision. Nous utilisons une méthode *Grid Search* pour optimiser les valeurs de nombre maximum de feuilles et le nombre minimum d'items par feuilles en maximisant l'*accuracy* par validation croisée. Nous utilisons les implémentations de ces méthodes de la bibliothèque scikit-learn³.

Prédiction de la classe sur les données test. En utilisant l'arbre de décision pour prédire si une adresse de sortie est une adresse de change dans la période d'étude, nous obtenons une probabilité sur laquelle nous appliquons un seuil pour décider si elle est considérée comme une adresse de change. Nous proposons dans la section suivante différentes stratégies pour fixer ce seuil. Nous fusionnons les agrégats d'adresses d'entrée et sortie d'un même exemple si la probabilité obtenue par l'arbre de décision est supérieure à ce seuil.

4.2 Fusionner les agrégats

Lorsque nous identifions qu'une adresse de sortie d'une transaction est une adresse de change, cela nous conduit à fusionner les agrégats issus de H1-2017 auxquels appartiennent respectivement les adresses en entrée de la transaction et l'adresse de change en sortie.

Comme les agrégats $\{C_1, \dots, C_K\}$ trouvés lors de l'application de l'heuristique H1 aux données apprentissage sont forcément des sous-ensembles des agrégats de notre vérité terrain A-WE, nous nous trouvons dans l'un des cas représentés dans la figure 3. Les cercles verts représentent les agrégats $\{F_1, \dots, F_m\}$ obtenus par H1 sur les données d'étude (H1-2017). Les grands cercles rouges représentent la vérité terrain H1-AP $\{C_1, \dots, C_K\}$, et les contours bleus la vérité de terrain A-WE $\{G_1, \dots, G_L\}$. Trois cas de figures sont possibles :

- La fusion de type 1 (Fig. 3 (A)) regroupe deux agrégats C_i, C_j appartenant à un même groupe de la vérité terrain H1-AP. Cette fusion qui nous rapproche de notre vérité terrain est souhaitable.
- La fusion de type 2 (Fig. 3 (B)) regroupe deux agrégats appartenant à des acteurs différents, en contradiction avec notre vérité de terrain. Cette fusion est indésirable.

3. <https://scikit-learn.org/>

- La fusion de type 3 (Fig. 3 (C)) regroupe deux agrégats appartenant au même acteur d'après notre vérité de terrain A-WE mais appartenant à des agrégats différents selon H1-AP. Ce type de fusion est souhaitable. Elle aurait été considérée comme indésirable par une évaluation classique basée uniquement sur H1-AP.

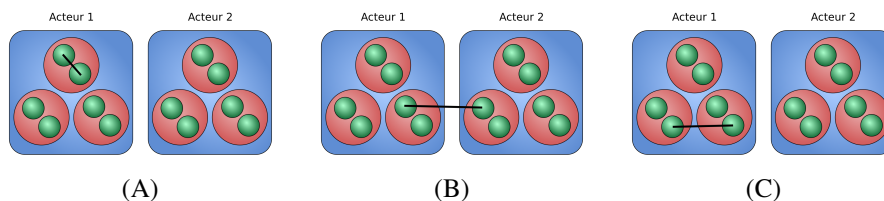


FIG. 3: (A) Fusion de deux agrégats H1-2017 appartenant à un même agrégats H1-AP. Fusion désirable. (B) Fusion de deux agrégats H1-2017 appartenant à des acteurs différents. Fusion indésirable. (C) Fusion de deux agrégats H1-2017 appartenant à des agrégats H1-AP différents d'un même acteur. Fusion désirable.

L'un des écueils majeurs de la méthode d'identification des adresses de change est qu'une seule identification erronée peut donner lieu à une *fusion catastrophique*, c'est à dire une fusion de deux agrégats contenant un grand nombre d'adresses mais appartenant à des acteurs différents. Ce risque nous conduit à proposer des variations afin de préférer quelques faux négatifs à un faux positif qui conduirait à une *fusion catastrophique*. Dans cette perspective, nous proposons trois méthodes de fusions :

Par classification simple avec seuil variable - M1. La méthode de référence consiste simplement à utiliser le résultat de l'algorithme de classification. Nous faisons varier le seuil de probabilité à partir duquel un résultat est considéré comme positif pour trouver le meilleur compromis entre diminution des faux négatifs et ajouts de faux positifs.

En imposant un seul change par transaction - M2. Avec cette méthode, nous comparons les probabilités prédites pour les deux adresses de sorties d'une même transaction. Nous savons que par définition, il est peu probable que les 2 adresses en sortie soient des adresses de change. Dans le but d'éviter les faux positifs, nous considérons donc, lorsque les deux sorties dépassent le seuil, que l'adresse de sortie ayant la plus forte probabilité est une adresse de change, l'autre étant une adresse d'un autre bénéficiaire. Si les deux probabilités sont égales (à 0.01 près), nous rejetons les deux sorties.

En requérant plusieurs changes observés pour fusionner - M3. Pour éviter qu'une seule erreur d'identification d'adresse de change ne conduise à fusionner deux agrégats d'adresses importants, nous ajoutons la contrainte d'observer plusieurs fois un résultat positif (une adresse appartenant à C_1 observée en sortie de change d'une transaction de C_2) pour fusionner deux agrégats C_1 et C_2 . Nous avons expérimenté avec plusieurs valeurs, mais nous présentons ici les résultats en fixant un seuil égal à 2, c'est-à-dire que lorsqu'un seul résultat positif indique que deux agrégats devraient être fusionnés, nous ignorons ce résultat.

5 Résultats

Pour évaluer la performance de notre méthode, nous avons comparé les agrégats d'adresses obtenus par différentes méthodes avec notre vérité de terrain. Nous avons utilisé les scores utilisés de manière classique en comparaison de *clusters* : *Homogeneity*, *Completeness*, *v-score/NMI*, *aNMI* et *Rand Index*.

Dans un premier temps, à titre de comparaison, nous présentons les scores d'*Homogeneity* obtenus en comparant avec H1-AP (Tab. 1a) et A-WE (Tab. 1b). Nous voyons que la méthode M3 obtient un score inférieur à 1.0 avec la vérité de terrain H1-AP. Cependant, comme mentionné lors de la présentation des types de fusions, une fusion de type Fig. 3 (C) implique un score inférieur en comparant avec la vérité de terrain H1-AP, ce qui ne serait pas vrai en comparant avec la vérité de terrain fusionnée A-WE. En vérifiant à l'aide du Tab. 1b, nous voyons qu'à partir d'une probabilité de 0.8, nous nous trouvons dans ce cas, c'est-à-dire, des bonnes fusions non reconnaissables en utilisant H1-AP.

Les observations sont similaires pour les autres scores, nous ne présenterons donc dans la suite que les résultats obtenus pour A-WE.

Heuristique H1	1.000		
Probabilité	M1	M2	M3
0.7	0.593	0.903	0.780
0.75	0.593	0.903	0.799
0.8	0.644	0.903	0.948
0.85	0.644	0.903	0.948
0.9	0.799	0.903	0.948
0.95	0.920	1.000	0.978

(a) Homogeneity Score - 6 acteurs - Vérité de terrain H1 a posteriori (H1-AP).

Heuristique H1	1.000		
Probabilité	M1	M2	M3
0.7	0.562	0.890	0.732
0.75	0.562	0.890	0.761
0.8	0.623	0.890	1.000
0.85	0.623	0.890	1.000
0.9	0.761	0.890	1.000
0.95	0.908	1.000	1.000

(b) Homogeneity Score - 6 acteurs - Vérité de terrain Acteurs WalletExplorer (A-WE.)

TAB. 1: Scores *Homogeneity* en utilisant H1-AP ou A-WE comme vérité de terrain.

La *completeness* permet d'évaluer le gain en terme de diminution du nombre de faux négatifs, c'est-à-dire d'agrégats H1-2017 correctement regroupés. Nous pouvons constater (tableau 2a) qu'à nouveau la méthode M3 a obtenu les meilleurs scores, en dépassant le score obtenu par l'heuristique H1. Comme la méthode de comptage a obtenu les meilleurs scores en *homogeneity* et *completeness*, elle obtient naturellement aussi les meilleurs scores au *v-score/NMI*, comme nous pouvons le constater au tableau 2b. Enfin, en utilisant les autres scores permettant d'évaluer la correspondance entre nos agrégats fusionnés et la vérité de terrain (*aNMI* (2c) et *Rand Index* (2d)), nous confirmons que la méthode M3 permet une amélioration significative par rapport aux autres approches.

6 Conclusion

Dans cet article, nous avons proposé une méthode originale d'évaluation de la qualité de l'identification d'acteurs utilisant un grand nombre d'adresses et évitant un certain nombre de limites des approches précédentes. Nous avons utilisé une vérité de terrain externe pour comparer trois approches à base d'apprentissage automatique pour identifier des acteurs, et avons

Heuristique H1	0.626		
Probabilité	M1	M2	M3
0.7	0.593	0.617	0.588
0.75	0.593	0.617	0.597
0.8	0.606	0.617	0.661
0.85	0.606	0.617	0.661
0.9	0.597	0.617	0.661
0.95	0.618	0.627	0.641

(a) Completeness Score - 6 acteurs - Vérité de terrain Acteurs WalletExplorer (A-WE).

Heuristique H1	0.770		
Probabilité	M1	M2	M3
0.7	0.577	0.729	0.652
0.75	0.577	0.729	0.669
0.8	0.614	0.729	0.796
0.85	0.614	0.729	0.796
0.9	0.669	0.729	0.796
0.95	0.736	0.770	0.781

(c) aNMI - 6 acteurs - Vérité de terrain Acteurs WalletExplorer (A-WE).

Heuristique H1	0.770		
Probabilité	M1	M2	M3
0.7	0.577	0.729	0.652
0.75	0.577	0.729	0.669
0.8	0.614	0.729	0.796
0.85	0.614	0.729	0.796
0.9	0.669	0.729	0.796
0.95	0.736	0.770	0.781

(b) V-Score/NMI - 6 acteurs - Vérité de terrain Acteurs WalletExplorer (A-WE).

Heuristique H1	0.481		
Probabilité	M1	M2	M3
0.7	0.286	0.432	0.345
0.75	0.286	0.432	0.351
0.8	0.299	0.432	0.532
0.85	0.299	0.432	0.532
0.9	0.351	0.432	0.532
0.95	0.446	0.481	0.501

(d) Indice de Rand(Rand Index) - 6 acteurs - Vérité de terrain Acteurs WalletExplorer (A-WE).

TAB. 2: Scores obtenus avec les différentes méthodes proposées. Nous pouvons observer que M3 surpasse à la fois l'heuristique H1 habituelle et les autres méthodes proposées dans la plupart des contextes.

montré l'efficacité de l'approche se basant sur des observations répétées que deux agrégats H1 étaient liés par une relation d'adresse de change. On peut noter en particulier que notre méthode a été capable de trouver dans les données de 2017 des informations que l'heuristique H1 n'a pas été capable de trouver même en considérant les données jusqu'en 2021 (analyse a posteriori).

Une limite de notre travail est que nous n'avons considéré que 6 acteurs. Augmenter le nombre d'acteurs risque de diminuer la qualité des résultats et nécessiter d'améliorer les stratégies proposées. Cette limite était nécessaire pour des raisons de complexité due à la quantité de données, et de restrictions sur la qualité des données, mais nous espérons résoudre ces problèmes lors d'une prochaine étude. Nous pensons que les résultats démontrent cependant l'intérêt de l'approche comme de la méthode d'évaluation. L'étude des arbres de décision devrait permettre d'identifier de nouvelles heuristiques d'identification d'adresse de change. Celle-ci reste à réaliser dans des travaux futurs.

Références

- Androulaki, E., G. O. Karame, M. Roeschlin, T. Scherer, et S. Capkun (2013). Evaluating user privacy in bitcoin. In *Financial Cryptography and Data Security*, pp. 34–51.
- Cazabet, R., R. Baccour, et M. Latapy (2018). Tracking bitcoin users activity using community detection on a network of weak signals. In *Complex Networks & Applications*, pp. 166–177.

- Emery, J. A. et M. Latapy (2021). Full bitcoin blockchain data made easy. In *Advances in Social Networks Analysis and Mining*.
- Ermilov, D., M. Panov, et Y. Yanovich (2017). Automatic bitcoin address clustering. In *IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 461–466. IEEE.
- Harrigan, M. et C. Fretter (2016). The unreasonable effectiveness of address clustering. In *Intl IEEE Conferences on Ubiquitous Intelligence Computing*, pp. 368–373.
- Liu, T., J. Ge, Y. Wu, B. Dai, L. Li, Z. Yao, J. Wen, et H. Shi (2020). A new bitcoin address association method using a two-level learner model. In *Algo. and Arch. for Parallel Proces.*, pp. 349–364.
- Maurer, F. K. (2016). A survey on approaches to anonymity in bitcoin and other cryptocurrencies. In H. C. Mayr et M. Pinzger (Eds.), *Informatik 2016*, Bonn, pp. 2145–2150. Gesellschaft für Informatik e.V.
- Meiklejohn, S., M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, et S. Savage (2013). A fistful of bitcoins : characterizing payments among men with no names. In *Conference on Internet measurement*, pp. 127–140.
- Möser, M. et A. Narayanan (2021). Resurrecting address clustering in bitcoin. *arXiv preprint arXiv :2107.05749*.
- Nick, J. D. (2015). Data-driven de-anonymization in bitcoin. Master's thesis, ETH-Zürich.
- Shao, W., H. Li, M. Chen, C. Jia, C. Liu, et Z. Wang (2018). Identifying bitcoin users using deep neural network. In *Int. Conf. on Alg. and Arch. for Parallel Proces.*, pp. 178–192.
- Sun, Y., H. Xiong, S. M. Yiu, et K. Y. Lam (2019). Bitvis : An interactive visualization system for bitcoin accounts analysis. In *2019 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pp. 21–25. IEEE.

Summary

Bitcoin is the most used and studied cryptocurrency. Being decentralized, transaction data is freely accessible and can thus be analyzed. The first step to most analyses consists in grouping individual – anonymous – addresses in clusters, corresponding to Bitcoin actors. In this article, we propose a new method to realize these machine learning-based aggregates. Our approach is based on the construction of a training dataset whose class variable is obtained by a ground truth computed a posteriori. This dataset is used to identify the change addresses of transactions, addresses belonging to the author of the transaction. This makes it possible to increase the number of addresses discovered as belonging to the same actor. Using an external validation criterion, we experimentally show the relevance of this method in comparison with the heuristics usually used.