Université paris 8

PARIS 8 UNIVERSITE VINCENNES - SAINT-DENIS

list cea tech

MÉMOIRE

pour obtenir le grade de Maîtrise

Université Paris 8 Vincennes à Saint Denis

Mention *Informatique*

présenté et soutenu publiquement par

Adel MANSOUR

le 02 juin 2021

Détéction d'aonomalies dans la blockchain

Directeur de mémoire : Rakia DJAZIRI

Maître d'apprentissage : Onder GURCAN

Alternance effectuée à : CEA list
2 Boulevard Thomas Gobert, 91120 Palaiseau

COMUE Paris Lumières
Laboratoire d'Informatique Avancée de Saint Denis
Laboratoire Paragraphe

MASTER

Sommaire

Introduction

Les activités suspectes dans les structures de réseau sont aussi anciennes que l'invention de la structure de réseau elle-même. Les sujets ou leurs activités qui ont tendance à se comporter anormalement dans le système (réseaux) sont appelés anomalies. La détection d'anomalies est largement utilisée dans diverses applications de cybersécurité pour cibler l'identification de la fraude financière, la détection d'invasion de réseau, la lutte contre le blanchiment d'argent, la détection de virus, etc.

L'objectif commun de ces réseaux est de détecter ces anomalies et de prévenir ces activités illégales à l'avenir. Avec les progrès de la technologie, la blockchain joue un rôle essentiel dans la sécurisation de ces structures de réseau. La blockchain est une base de données distribuée décentralisée qui maintient des enregistrements toujours croissants et assure que les enregistrements frauduleux n'en feront jamais partie mais aussi que les enregistrements précédemment ajoutés restent inchangés. Cependant, les participants au réseau blockchain peuvent essayer de se livrer à des activités illégales et, dans certains cas, réussir à tromper le système en leur faveur.

Les méthodes actuelles de détection des anomalies sont conçues et mises en œuvre avec des systèmes centralisés. Avec l'avènement de la technologie blockchain, cela impliquait également le besoin de processus de détection d'anomalies dans ces systèmes.

Dans ce mémoire, nous extrayons et analysons les données d'une blockchain accessible au public appelée bitcoin. Selon Nakamoto[8], le bitcoin est une monnaie numérique peer-to-peer(pair-à-pair) blockchain, à travers laquelle les utilisateurs peuvent envoyer et recevoir de manière anonyme une forme de crypto-monnaie sans avoir besoin d'un intermédiaire.

Mots-clés : Blockchain, Bitcoin, Apprentissage non-supervisé.

Première partie

Problématique

Sommaire

Chapitre 1

Présentation de l'entreprise

Sommaire

1.1 Présentation de l'entreprise

FIG. 1.1: Logo du CEA list

Le CEA Commissariat à l'énergie atomique et aux énergies alternatives est un organisme public à caractère de recherche scientifique.
Il est un acteur majeur dans la recherche française et à la pointe de domaine comme les énergies (nucléaires et renouvelables), la défense et la sécurité, la recherche scientifique fondamentale. Rien que pour l'année 2019 près de 670 brevets ont été déposés et plus de 7000 familles de brevets sont actifs à nos jours.

1.2 Secteur d'activité et organisation

Je travaille au sein du laboratoire systèmes d'Information de Confiance ,Intelligents et Auto - organisants (LICIA) du département d'Ingénierie des Logiciels et des Systèmes (DILS), nous travaillons sur les systèmes distribués et plus particulièrement sur la blockchain et ses domainess d'application, avec des acteurs majeurs de l'industrie en France et en europe (EDF, veolia, RTE).

Je travaille au sein d'une équipe de 4 personnes dont un chef de projet (expert en intelligence artificielle), un développeur et un doctorant(thèse sur l'intelligence artificielle). Notre mission principale est le développement d'un simulateur blockchain Multi-Agent Experimenter (MAX).

MAX est utilisé par les entreprises pour créer et simuler des blockchains dans le but de les déployer dans un futur proche dans des domaines divers allant d'agro-alimentaire aux énergies.

Notre plus grand colaborateur industriel est EDF et notre partenartiat a été mis en place dans l'optique du déploiement d'une blockchain dans le but de révolutionner le monde de l'énergie dans lequel les acteurs majeurs en France sont le CEA et EDF et cela pour répondre à beaucoup de problématiques telles que :

- La gestion de transactions et la tenue d'un registre
- Certifier l'origine de l'énergie
- Suivi de la consommation en temps réel
- Pouvoir mettre en place des boucles d'énergie au niveau local
- Faciliter les paiements, les facturations et les rémunérations

Chapitre 2

Le problème à résoudre

Sommaire

2.1 Introduction

L'utilisation de registre en finance est aussi ancien que la monnaie elle-même. De l'argile et des pierres au papyrus et au papier en passant par les feuilles de calcul à l'ère numérique, les registres ont été le fondement de la comptabilité. Les premiers grands registres numériques n'imitent que des approches de comptabilité en partie double sur papier.

Cependant, il y avait encore une grande lacune dans le consensus car ces registres augmentent en volume et en espace. L'augmentation de la puissance de calcul et les percées en cryptographie, ainsi que la découverte de nouveaux algorithmes sophistiqués ont conduit à des registres distribués. Ainsi avec le fait que les données soient immuables dans ces registres si une transaction ou un block est frauduleux les conséquences seront irréversibles, d'où un cas de figure comme celui-ci est critique. Il faut donc penser à les détecter avant qu'ils soient chaîner et pour cela il faut commencer par étudier tout cas de fraude existant pour mieux comprendre le phénomène.

2.2 Blockchain

Une blockchain utilise généralement un réseau peer-to-peer pour une communication transparente entre les nœuds et la validation de nouveaux blocs. Une fois que les données de transaction sont ajoutées au bloc, elles ne peuvent pas être modifiées sans modifier tous les blocs suivants. Pour modifier un enregistrement, il faut un consensus de la majorité des nœuds du réseau, ce qui est très difficile à obtenir. Bien que les enregistrements d'une blockchain ne soient pas immuables, ils peuvent cependant être considérés comme une conception plus sécurisée . Elle illustre également un système informatique distribué avec une tolérance aux pannes byzantins élevées. La décentralisation et les problèmes byzantins seront expliqués plus loin dans ce chapitre.

2.3 Sécurité de la blockchain

L'idée centrale de la sécurité des blockchains est de permettre aux personnes, en particulier celles qui ne se font pas confiance, de partager des informations précieuses de manière sécurisée et inviolable. Des algorithmes sophistiqués de stockage décentralisés et de consensus rendent extrêmement difficile pour les attaquants de

manipuler le système. Le cryptage joue un rôle essentiel dans l'ensemble du processus et valide les canaux de communication du réseau. Un autre problème critique que la blockchain prétend résoudre est l'accord byzantin. Il définit la fiabilité du système en cas de défaillance.

2.4 Problème byzantin

Le problème byzantin, tel qu'expliqué par Lamport, Shostak et Pease [**lamport2019byzantin**], stipule que les ordinateurs d'un réseau décentralisé ne peuvent pas être entièrement et irréfutablement garantis qu'ils affichent les mêmes données. Compte tenu du manque de fiabilité du réseau, les nœuds ne peuvent jamais être sûrs que les données qu'ils ont communiquées ont atteint leur destination. Par essence, le problème byzantin consiste à obtenir un consensus au sein d'un réseau distribué de nœuds, alors que quelques nœuds sur le réseau peuvent être éventuellement défectueux et peuvent également tenter d'attaquer le réseau. Un nœud byzantin peut induire en erreur et fournir des informations corrompues aux autres nœuds impliqués dans le processus de consensus. La blockchain doit fonctionner avec robustesse dans une telle situation et parvenir à un consensus malgré l'interférence de ces nœuds byzantin. Même si le problème de byzantin ne peut pas être entièrement résolu, différentes chaînes de blocs et différents registres distribués utilisent divers mécanismes de consensus comme la preuve de travail et la preuve d'enjeu pour surmonter le problème d'une manière probabiliste.

2.5 Bitcoin

La première conception connue d'un enchaînement cryptographique de données liées en chaînes et en blocs ont été réaliser dans les années 1990.Cependant, la toute première conceptualisation d'une blockchain a été réalisée et introduite par une personne portant le pseudonyme de Nakamoto [8]. Par conséquent, elle a donné naissance au premier système de blockchain. Ce système de blockchain a été baptisé "Bitcoin" et est conçu comme un système d'argent de pair à pair, autrement dit une crypto-monnaie. Bitcoin sert de registre public pour toutes les transactions du réseau et est capable de représenter numériquement la monnaie et fonctionne donc comme une monnaie électronique. La représentation numérique d'un actif peut entraîner de multiples problèmes. Bitcoin a été développé pour résoudre certains problèmes importants.

Premièrement, en théorie, l'actif représenté numériquement peut techniquement être dupliqué, ce qui n'est pas acceptable. En effet, cette reproduction d'un actif numérique peut confondre la propriété réelle de l'actif. Deuxièmement, l'actif répliqué peut être dépensé plusieurs fois, ce qui entraîne un chaos dans le système. Ce problème est également connu sous le nom de problème de double dépense. Le bitcoin résout ce problème en utilisant l'algorithme de preuve de travail (POW).

En utilisant le mécanisme de preuve de travail, les mineurs vérifient l'authenticité des transactions et sont récompensés par des bitcoins nouvellement générés.

De nombreux éléments sont impliqués lorsque les participants au réseau transmettent des bitcoins. Cependant, nous pouvons diviser les activités en deux groupes : les activités de l'utilisateur et les activités de transaction. Dans ce mémoire, notre intérêt principal est dans les activités de transaction et son analyse pour détecter les comportements suspects. Les transactions sur la blockchain bitcoin sont relativement plus complexes en profondeur. Le but de ce mémoire est d'analyser les données transactionnelles bitcoin. Ce mémoire met en lumière la perspective analytique des données bitcoin.

2.6 Objectif

La détection d'anomalies est l'identification d'éléments, d'observations ou d'événements uniques qui éveillent les soupçons en étant significativement différents de la majorité des données. Habituellement, les éléments anormaux sont interprétés comme des sortes d'anomalies, comme une fraude à la carte de crédit, un défaut structurel ou une erreur dans le texte. Les anomalies peuvent également être décrites comme du bruit, des valeurs aberrantes et des nouveautés. Cependant, il existe une différence significative entre la détection des aberrations et la détection des nouveautés. Dans le contexte de la détection des nouveautés, les données de formation ne sont pas polluées par des valeurs aberrantes, nous nous sommes intéressés à détecter si une observation nouvellement introduite est une anomalie pour le système ou non. Alors que, dans le cas de la détection des aberrations, les données d'apprentissage contiennent des valeurs aberrantes qui sont définies comme des observations qui sont éloignées des autres.

Ce mémoire se concentre principalement sur la détection des anomalies dans la structure des données de la blockchain. Indépendamment du domaine de la blockchain, on peut appliquer des techniques spécifiques à cette structure de données et tenter d'identifier les anomalies. Il s'agit d'un problème d'apprentissage non

supervisé et les données pour les cas de détection d'anomalies sont généralement très déséquilibrées. En raison de la disponibilité des données et de la littérature académique, ce mémoire a l'intention d'utiliser une blockchain publique financière connue sous le nom de bitcoin. Les données transactionnelles de bitcoin passent par une phase de prétraitement intense, puis diverses méthodes de modélisation sont appliquées afin d'identifier les transactions qui sont liées à des vols, des casses ou des blanchiments d'argent. Un processus d'évaluation complet est réalisé afin de déterminer la solution optimale après avoir comparé les modèles.

2.7 Organisation du mémoire

Dans le chapitre 1 j'ai présenté l'entreprise et son organisation,

Dans le chapitre 2, j'ai présenté le Domaine d'activité, la problématique et le but du projet que je mène. Le reste de ce mémoire est structuré comme suit.

Dans le chapitre 3, je présenterai un état de l'art des techniques de detection des anomalies(transactions frauduleuses) dans les réseaux blockchain.

Ceci est suivi du chapitre 4, qui fournit une description des données utilisées et les prétraitements que j'aurais appliqués. Dans le chapitre 5 ma contribution est abordée avec les techniques que j'ai appliquées ainsi leurs résultats. Je présenterai et discuterai les résultats dans le chapitre 6. Enfin, quelques conclusions seront fournies.

Deuxième partie

État de l'art

Sommaire

Chapitre 3

État de l'art des techniques

Sommaire

3.1 Introduction

La détection d'anomalies est largement utilisée dans de nombreuses applications, telles que la détection de défaut, la détection d'intrusion, la détection de fraude et bien d'autres. L'objectif principal de ce travail est de détecter les fraudes. La détection de la fraude est un domaine bien étudié et peut être divisé en plusieurs domaines, tels que la fraude par carte de crédit, l'intrusion de virus, la fraude à l'assurance, etc. Cette recherche utilise généralement diverses techniques pour effectuer ses analyses, telles que des algorithmes d'apprentissage automatique et des méthodes d'analyse de réseau. Développements récents dans le domaine et recherches antérieures Ahmed et al . montrent que les algorithmes d'apprentissage automatique non supervisés se comportent plus spécifiquement vis-à-vis des anomalies. La plupart des études de détection de fraude utilisent des techniques d'apprentissage automatique supervisé et se concentrent sur le développement d'un modèle complet pour apprendre les modèles d'anomalies dans les données de formation. Les applications des technologies de registres distribués étant relativement nouvelles, la recherche sur la détection de la fraude dans ce domaine est toujours en cours. La technologie Blockchain est basée sur des réseaux décentralisés. Des études antérieures ont montré que dans la plupart des cas de recherche sur la blockchain, les chercheurs abordent le problème soit en termes de données réseau, soit en termes de données brutes. Ce chapitre explique comment les recherches précédentes ont abordé le problème de la détection des anomalies de la blockchain et quelles approches et quels résultats elles ont pu en tirer.[1]

3.2 Travaux basés sur les réseaux

Huang et coll. [4] ont abordé le problème à partir d'un modèle de comportement du réseau. Les données utilisées dans cette étude n'ont pas été divulguées et, selon les auteurs, cette méthode peut être utilisée car elle est répandue dans toutes les blockchains. Cependant, l'idée principale est de trouver et de classer les modèles de comportement dans le réseau de blockchain chinois à l'aide d'une nouvelle méthode de regroupement comportemental (BPC). La valeur des transactions qui changent au fil du temps est classée en groupes qui se répartissent en plusieurs catégories. Ces séquences sont mesurées à l'aide d'un algorithme de mesure de similarité. Cette étude a examiné la distance euclidienne, Warping Time Dynamics (DTW), Real Sequence Distance (EDR) et la plus longue sous-séquence (LCSS). Les auteurs de cette étude ont quitté EDR et LCSS et ont choisi DTW comme référence pour se concentrer sur le traitement de données silencieuses et bruyantes sur la blockchain car cette dernière ne contient pas beaucoup de données compromises . La version modifiée des k-means 3.4.5 est utilisée pour identifier des modèles dans

le réseau connus sous le nom de grappes de modèles comportementaux (BPC) et pour comparer les résultats de BPC à l'aide de la classification hiérarchique (HIC) et de la dépendance à la densité (DBSCAN). Le résultat final montre que l'algorithme BPC est plus efficace que la méthode conventionnelle. Cette étude explore les problèmes des algorithmes de perturbation du réseau blockchain. [11]

D'autres études ont converti les données Bitcoin en une structure de type réseau, qui est ensuite divisée en deux graphes primaires, le graphe des utilisateurs et le graphe des transactions.
Le graphe des utilisateurs extrait est utilisé pour tenter de détecter les utilisateurs suspects, tandis que le graphe des transactions est utilisé pour tenter de découvrir les transactions suspectes. En utilisant les deux graphes, on essaie non seulement de détecter les utilisateurs et les activités anormales, mais aussi d'établir un lien entre les deux. Il en résulte un système qui peut également associer des utilisateurs suspects à des transactions inhabituelles.

Les métadonnées des deux graphes sont extraites pour créer un nouvel ensemble de données qui n'a pas été rendu public. Elles contiennent des caractéristiques telles que le degré d'entrée, le degré de sortie, la quantité d'équilibre des nœuds du graphique et de nombreuses autres variables vitales. Les méthodes utilisées pour l'analyse des anomalies de réseau dans le réseau bitcoin sont le clustering K-means motivé par Power Degree & Densification Laws et Local Outlier Factor (LOF). Le clustering K-means est utilisé en combinaison avec le Local Outlier Factor (LOF). Pour réduire la liste des k-nearest-neighbour dans le LOF, des indices sont calculés à partir du clustering K-means. La liste des nœuds de chaque groupe est obtenue à partir de la méthode K-means 3.4.5, puis une recherche de voisins les plus proches(k-nearest-neighbour) est effectuée pour gagner du temps. En raison des limites de calcul des expériences, seul un petit sous-ensemble de caractéristiques extraites a été traité. Un cas d'anomalie a été détecté avec succès sur les 30 cas connus.

Les défis rencontrés par les chercheurs dans cette étude étaient principalement liés à l'évaluation et à la validation. En référence à cette étude, LOF semble être plus performant que les autres pour les données du réseau bitcoin. En outre, cette recherche ne parle pas des cas de faux positifs et de vrais négatifs.[7]

3.3 Etudes basées sur les données brutes

Pham et Lee [9] ont publié une étude dans laquelle ils ont fait des recherches sur la détection d'anomalies dans les données bitcoin en utilisant des méthodes d'apprentissage non supervisées. Cette étude a utilisé le même ensemble de données qu'une autre de leur étude mais a abordé le problème d'un point de vue basé sur l'apprentissage automatique. Le jeu de données, composé de métadonnées du graphe des utilisateurs de bitcoins et du graphe des transactions, est prétraité pour extraire un sous-ensemble de caractéristiques. Les algorithmes d'apprentissage automatique non supervisés choisis pour cette étude sont le regroupement K-means, la méthode basée sur la distance de Mahalanobis et les machines à vecteurs de support non supervisées (SVM). L'étude a permis de détecter deux cas connus de vol et un cas connu de perte sur 30 cas connus. En outre, elle ne traite pas les cas de faux positifs et de vrais négatifs.

D'autres études ont cherché à faciliter l'exploration des données bitcoin, en tentant de détecter le blanchiment d'argent sur le réseau bitcoin à l'aide de services de mélange et en suivant les sorties des services de mélange jusqu'à leurs entrées. Les services de mélange regroupent des fonds provenant de différentes sources et les mélangent dans le but de brouiller les pistes jusqu'à la source originale des fonds. Cette étude utilise également le même ensemble de données du réseau bitcoin à l'Université de l'Illinois. Les techniques de prétraitement ont été utilisées pour associer les clés publiques aux utilisateurs afin d'établir une relation logique au sein des données. L'algorithme K-mean de James MacQueen et al.[6] a été utilisé en combinaison avec l'algorithme Role Extraction (RolX) d'Henderson de manière non supervisée pour l'analyse. Cette étude adopte une approche unique de restructuration de l'ensemble des données pour découvrir les anomalies de blanchiment. Elle a introduit un concept de hubs dans lequel les utilisateurs ayant un nombre élevé de transactions ont été filtrés pour rendre l'ensemble des données plus efficace pour les algorithmes non supervisés. Cependant, les statistiques des résultats n'ont pas été partagées dans cette recherche.

En plus de l'analyse de données hétérogènes, l'analyse visuelle a également atteint sa popularité. Des travaux ont cherché à comprendre et à analyser la détection des anomalies dans les blockchains à l'aide de caractéristiques visualisées. Ces recherche ont non seulement proposé une approche théorique, mais également mis en œuvre de manière pratique une solution pour la soutenir. La solution proposée exécute une application décentralisée au-dessus de la blockchain Ethereum et collecte les statistiques de la blockchain pour les stocker dans une base de données NoSQL qui fournit ensuite des visualisations et des tableaux de bord

personnalisables. Un noyau d'apprentissage automatique sophistiqué opère sur les données stockées pour créer une analyse visuelle et détecter les anomalies en ligne, les résultats sont représentés visuellement sur des tableaux de bord. Une telle approche optimise l'interprétabilité et la visualisation des algorithmes d'apprentissage automatique. Les études sur la détection des anomalies dans le bitcoin se sont principalement concentrées sur la structure des transactions du réseau bitcoin. Cependant, la blockchain ethereum, avec ses contrats intelligents, produit un contexte complexe pour l'analyse ; c'est pourquoi cette recherche complète la stratégie d'analyse du graphe des transactions des chercheurs précédents.[12]

3.4 Méthodes de détection des anomalies

La détection des anomalies est classée en trois grands types de méthodes d'apprentissage : les méthodes d'apprentissage supervisé, les méthodes d'apprentissage semi-supervisé et les méthodes d'apprentissage non supervisé. La majorité des problèmes de détection d'anomalies dans le monde réel relèvent du domaine de l'apprentissage non supervisé, comme la détection des fraudes par carte de crédit, la détection des intrusions dans les réseaux et la détection des transactions frauduleuses dans notre blockchain. Ce mémoire considère que les méthodes d'apprentissage supervisé et semi-supervisé sont hors de portée et vise à se concentrer entièrement sur les méthodes d'apprentissage non supervisé.

L'apprentissage non supervisé est le type le plus flexible, car il ne nécessite pas d'étiquettes de classe pour la formation. Il évalue les données en fonction de leurs attributs intrinsèques, tels que la distance, la densité, etc. Ils tentent de distinguer les anomalies des données standards sur la base d'une estimation. La majorité des méthodes de détection d'anomalies non supervisées ne nécessitent pas d'étiquettes de classe pour la formation.

Ce mémoire met en œuvre de manière sélective un ensemble de méthodes de détection d'anomalies non supervisées capables de traiter une grande quantité de données et de calculer des résultats dans un temps raisonnable. Les sous-sections suivantes de ce chapitre expliquent ces algorithmes plus en profondeur.

3.4.1 Isolation forest :

Liu, Ting et Zhou [5] ont publié un algorithme appelé "isolation forest" pour la détection non supervisée des anomalies, qui a connu une grande popularité ces dernières années. L'idée derrière la forêt d'isolement est qu'il est comparativement plus simple d'isoler les anomalies des données que de construire un modèle qui pourrait estimer le comportement normal.

Pour isoler une observation anormale, l'ensemble de données est divisé de manière aléatoire et récursive jusqu'à ce qu'il ne reste plus qu'un seul point de données qui soit l'observation comme le montre la figure 3.1. Une structure arborescente est utilisée pour représenter la partition récursive. La base de l'algorithme de la forêt d'isolation l'ensemble des arbres d'isolement. ils sont construits pour estimer la mesure de la normalité et de l'anormalité des observations dans l'ensemble de données.

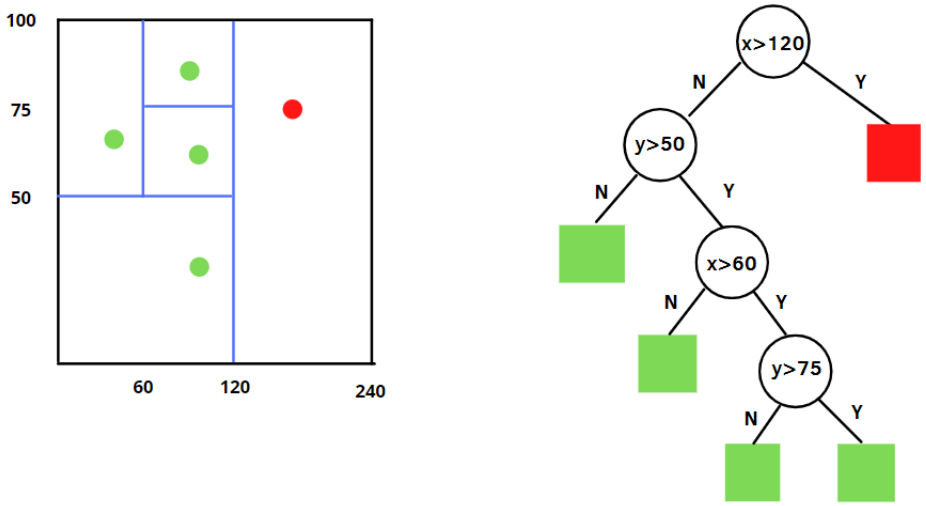

FIG. 3.1: Illustration de l'algorithme isolation Forest

Une forêt d'isolement est une combinaison d'arbres d'isolement binaires dans lesquels chaque nœud a soit zéro, soit deux nœuds descendants. Soit N un nœud qui est soit un nœud feuille sans enfant, soit un nœud parent avec deux nœuds descendants N_l et N_r . Pour décider quels nœuds descendants vont à quel nœud parent, on associe un test au nœud N. Ce test consiste à sélectionner une caractéristique aléatoire q dans les données et un test d'hypothèse. Ce test implique la sélection d'une caractéristique aléatoire q dans les données et d'un point de sépa-

ration aléatoire p, de telle sorte que les nœuds N point de séparation aléatoire p tel que les nœuds $p < q$ sont sous N_l et $p \geq q$ sont sousN_r.

Étant donné un ensemble de données $X = x1, x2, ..., xn$ contenant n observations, des arbres d'isolation récursifs aléatoires sont construits jusqu'à ce que chaque arbre atteigne la limite supérieure de $|X| = 1$ ou que toutes les données de X aient la même valeur. Si l'on suppose que la majorité des données de X sont distinctes et que l'arbre d'isolation a atteint sa taille maximale, chaque observation est isolée à un noeud feuille. L'arbre aura alors n feuilles et $n - 1$ nœuds internes, soit au total $2n - 1$ nœuds. La longueur du chemin d'une observation x peut être calculée par l'heuristique $h(x)$ qui est une mesure du nombre d'arêtes qu'il faut traverser du nœud racine au nœud feuille. Afin d'utiliser la forêt d'isolation comme technique de détection d'anomalies, elle doit générer des scores d'anomalies comparables à $h(x)$. Car $h(x)$ ne peut pas être utilisé directement comme un score d'anomalie.En effet, alors que la hauteur maximale d'un arbre est de l'ordre de n, la hauteur moyenne est de l'ordre de $log\ n$.

Le score d'anomalies est calculé suivant l'équation suivante :

$$c(n) = 2H(n-1) - (2(n-1)/n)$$

Lors de l'apprentissage d'un algorithme de forêt d'isolement, les arbres d'isolement sont créés de manière aléatoire. L'apprentissage nécessite deux paramètres : le nombre d'arbres t et la taille du sous-échantillonnage. La limite de la hauteur des arbres l est calculée en fonction de la taille du sous-échantillonnage. Le processus de sous-échantillonnage contrôle la taille des données pour la formation et constitue une variable essentielle car elle peut affecter directement les performances du classificateur. Comme un échantillon différent est choisi pour créer un arbre d'isolation à chaque fois, cela peut aider à isoler les anomalies d'une meilleure manière. Quant à la la prédiction des anomalies à partir du modèle entraîné est utilisée pour calculer un score d'anomalie s comme expliqué ci-dessus. Ce score d'anomalie s est compris entre les valeurs de [0,1], les valeurs les plus proches de 1 étant les plus susceptibles d'être des anomalies.

3.4.2 Histogram-based Outlier Score (HBOS) :

Goldstein et Dengel [2] ont présenté un algorithme non supervisé pour la détection des anomalies. Le score d'aberration basé sur les histogrammes (HBOS) est une méthode statistique conçue pour traiter de grands ensembles de données. Sa

complexité de calcul est un attribut attrayant pour les chercheurs.

Un histogramme univariant est créé pour chaque caractéristique des données. Pour les données catégorielles, la fréquence des valeurs de chaque catégorie est calculée et la hauteur de l'histogramme est calculée. Pour les caractéristiques numériques, on utilise soit la technique des histogrammes de largeur de bande statique ou bien la technique de l'histogramme de largeur de binaire dynamique. La méthode de largeur de bande statique utilise k cases de largeur égale sur une plage de valeurs, et la hauteur de la case est estimée en comptant les échantillons qui tombent dans chaque case. La largeur de bac dynamique est déterminée en triant tous les points de données, puis en regroupant un nombre fixe N/k de valeurs consécutives, où N représente le nombre total d'observations et k le nombre de cases. Les deux approches, statique et dynamique, sont introduites dans cet algorithme car les données du monde réel présentent des distributions de caractéristiques très différentes, notamment lorsque les caractéristiques ont une gamme significative d'écarts entre elles. Une case fixe peut mal estimer la densité d'anomalies, ce qui entraîne l'ajout de la plupart des données à un petit nombre de bacs. Anomalie Les problèmes de détection d'anomalies présentent généralement des écarts considérables dans la plage de données, car la plupart des valeurs aberrantes sont très éloignées des observations standards. Les auteurs préfèrent généralement la méthode dynamique si les distributions des données sont inconnues. La technique utilisée pour déterminer la valeur de k bins est la racine carrée des instances N.

Un histogramme particulier est calculé pour chaque dimension d il est ensuite normalisé de sorte à ce que la densité maximale soit 1.0. Ce faisant, on s'assure que chaque caractéristique est pondérée de manière égale pour le score d'anomalie. Le HBOS pour chaque observation p est calculé à l'aide de l'équation suivante :

$$HBSO\,(p) = \sum_{i=0}^{d} \log \frac{1}{hist_i\,(p)} \tag{3.1}$$

Ce score peut également être représenté comme un inverse de Naive Bayes discret où au lieu d'une multiplication, une somme de logarithmes est utilisée, en utilisant le fait que $\log(a - b) = \log(a) + \log(b)$. La raison de cette astuce est de rendre l'algorithme HBOS moins sensible aux erreurs dues à la précision en virgule flottante, qui peut entraîner des scores d'anomalie très élevés pour des distributions extrêmement déséquilibrées.

3.4.3 Cluster-Based Local Outlier Factor (CBLOF) :

He, Xu et Deng [3] ont proposé un algorithme appelé CBLOF (cluster-based local outlier factor) qui possède les propriétés d'un algorithme basé sur les clusters ainsi que sur l'algorithme LOF (local outlier factor) . L'idée de la détection des anomalies dans cet algorithme tourne autour du regroupement des données qui sont ensuite utilisées pour calculer un score d'anomalie de manière similaire à LOF. Tout algorithme de clustering arbitraire peut être utilisé dans l'étape de clustering. Cependant, la qualité des résultats de l'algorithme de clustering influence directement la qualité des résultats de CBLOF.

L'algorithme procède de la manière suivante : il classe un ensemble de données D à l'aide d'un algorithme de classification arbitraire et attribue chaque observation de D à une classification. Leurs tailles respectives trient les clusters de telle sorte que $|C_1| \geq |C_2| \geq --- \geq |C_k|$ où $C_1, C_2, ..., C_k$ représentent tous les clusters avec k comme nombre de clusters. L'intersection de toute paire de clusters devrait être un ensemble vide, tandis que l'union de tous les clusters devrait représenter toutes les observations de l'ensemble de données D. L'étape suivante consiste à rechercher une valeur d'indice de limite qui sépare les petits groupes (Small Clusters) des grands clusters (Large Clusters).

3.4.4 Principal Component Analysis (PCA)

L'analyse en composantes principales (ACP) est généralement considérée comme un algorithme de réduction de la dimensionnalité. Elle peut manifester la variance-covariance des caractéristiques d'un ensemble de données pour créer de nouvelles variables qui sont représentées par des fonctions de variables originales appelées composantes principales. Ces composantes principales sont des combinaisons linéaires distinctes d'un nombre p de variables aléatoires $X_1, X_2, ..., X_p$. Elles possèdent des propriétés uniques telles que la non-corrélation entre elles, la variance des composantes décroît par ordre décroissant, la première composante ayant la variance la plus élevée, elle diminuant avec chaque composante suivante. Cependant, la somme de la variation totale des caractéristiques originales $X_1, X_2, ..., X_p$ est toujours égale à la variation totale de toutes les composantes principales combinées. Les composantes principales sont faciles à calculer en utilisant l'analyse propre de la matrice de corrélation ou de covariance des caractéristiques des données.

Bien que l'ACP soit généralement utilisée pour la réduction de la dimensionnalité, Shyu et al [10] ont proposé une approche permettant d'utiliser cet algorithme

pour la détection des anomalies. Si toutes les variables de l'ensemble de données suivent la distribution normale, les composantes principales (PC) que nous calculons devraient être normalement distribuées de manière indépendante, avec une variance égale aux valeurs propres. Les valeurs propres que nous pouvons obtenir à partir de la décomposition en valeur unique. Avec ceci l'algorithme indique que $\sum_{i=0}^{p} \frac{PC_i^2}{\lambda_i}$ suit la distribution du chi- carré avec p degrés de liberté. Pour détecter les anomalies à l'aide de l'algorithme PCA, toutes les valeurs aberrantes évidentes sont éliminées à l'aide de la distance de Mahalanobis. Cette étape est effectuée plusieurs fois afin d'éliminer tous les points de données présentant une distance de Mahalanobis très élevée. Avec S comme matrice de covariance, x i comme observation de la i-ième caractéristique dans les données et x comme moyenne de toutes les caractéristiques, l'algorithme de Mahalanobis peut être utilisé pour déterminer la distance de Mahalanobis. la moyenne de toutes les caractéristiques, la distance de Mahalanobis D est défini selon l'équation suivante :

$$D = \sqrt{(x_i - \bar{x})^T S^{-1} (x_i - \bar{x})}$$

Ensuite, une analyse en composantes principales est appliquée à l'ensemble de données d'apprentissage et les composantes principales correspondantes sont calculées pour l'ensemble de données de test. Les scores respectifs des anomalies majeures et mineures sont calculés à l'aide des équations suivantes respectivement .

$$AS_1 = \sum_{j=1}^{q} \frac{PC_j^2}{\lambda_j}$$

$$AS_2 = \sum_{j=p-r+1}^{p} \frac{PC_j^2}{\lambda_j}$$

Les prédictions sont effectuées à l'aide des scores d'anomalie calculés. Si $AS1 \geq c1$ ou $AS2 \geq c2$, l'observation est considérée comme une anomalie. Alors que si $AS1 < c1$ ou $AS2 < c2$ l'observation est étiquetée comme un point de données normal.

3.4.5 K-means

Dans les algorithmes de clustering, K-means est l'un des choix les plus populaires pour la détection des anomalies. Il a été présenté comme un algorithme d'apprentissage non supervisé par J. MacQueen [6]. L'algorithme vise à partitionner les données en k groupes différents où chaque échantillon appartient au groupe dont la moyenne est la plus proche. Chaque groupe est représenté par un centroïde c

qui est la moyenne des observations de ce groupe. La mesure de similarité utilisée lors de la comparaison des observations à la grappe est la distance euclidienne au carré, qui peut être calculée à l'aide de l'équation où x i représente les observations et c i est le centroïdes et n représente le nombre d'observations.

$$d(x, c)^2 = \sum_{i=1}^{n}(x_i - c_i)^2$$

La première étape du processus de l'algorithme K-means consiste à initialiser les k centroïdes. Ensuite, chaque observation est comparée de manière itérative à chaque centroïde, elle est ensuite attribuée en fonction de la plus petite somme des carrés des grappes à l'un d'eux. Ces observations assignées forment finalement des clusters autour des centroïdes spécifiques. Ce processus est identique à l'affectation de chaque observation à son centroïde le plus proche en utilisant la distance euclidienne ou la distance euclidienne au carré, car la valeur minimale des deux est la même. Cette étape de l'algorithme est également connue sous le nom d'étape d'attente.

Une fois que chaque observation est assignée à son centroïde le plus proche et que les premières grappes sont formées, les centroïdes sont mis à jour sur la base de la nouvelle moyenne de l'échantillon pour chaque grappe. Les centroïdes peuvent être n'importe quel point de donnée arbitraire dans le cluster représentant la moyenne du cluster ; il n'est pas nécessaire qu'il s'agisse de l'une des observations. La moyenne arithmétique consiste également à minimiser la somme des carrés au sein des grappes. Cette étape de mise à jour des centroïdes est également connue sous le nom d'étape de maximisation.

Au moment de la fin de l'algorithme K-means, les résultats avec les centroïdes finaux et l'affectation des clusters de toutes les observations sont fournis. L'algorithme se termine lorsque les centroïdes ont convergé, ce qui signifie que la valeur des centroïdes cesse de s'actualiser au fil des itérations. Il n'y a pas de garantie que k-means atteigne l'optimum global avant la fin de l'algorithme, c'est pourquoi il est courant d'exécuter l'algorithme plusieurs fois sur les données et de choisir un modèle avec la somme des distances quadratiques à l'intérieur du cluster le plus faible. Comme la moyenne de chaque cluster peut être n'importe quel point de données arbitraire, l'exécution de l'algorithme à chaque fois peut produire des solutions différentes qui ont convergé différemment, mais qui pourraient être quelque peu similaires. Si les centroïdes sont initialisés près de l'optimum local, l'algorithme est susceptible de converger plus rapidement. K-means est une variante d'une approche de regroupement plus généralisée. L'Expectation-Maximization

(EM), l'algorithme EM a une approche plus probabiliste du clustering et crée des soft-clusters contrairement aux hard-clusters créés par K-means.

3.5 Conclusion

Dans ce chapitre j'ai survolé les principales techniques et les principales approches sur la détection d'anomalies dans les réseaux blockchains, dans le chapitre qui suit je vais me pencher sur les données, leurs caractéristiques, sur la façon avec laquelle j'ai fait des prétraitements des données.

Chapitre 4

État de l'art des données

Sommaire

4.1 Données

Pour ce mémoire les données utilisées sont tirées de la blockchain bitcoin, elles ont ensuite été concaténées. Sachant que pour chaque transaction, les données utilisées dans datent plus précisément de l'époque entre 2011 et 2013. Il peut y avoir plusieurs nombres d'adresses d'expéditeurs et de destinataires. En outre, un seul utilisateur peut posséder plusieurs adresses, chaque utilisateur est anonyme car aucune information personnelle n'est associée à l'une de ces adresses.

En 2014 le Bitcoin Forum a référencé toutes les anomalies liées aux transactions dans la blockchain bitcoin qu'il s'agisse du vol, du piratage ou de la perte de bitcoins. Ces données ont ensuite été concaténées puis croisées avec celles de la blockchain bitcoin pour labelliser les transactions frauduleuses. Cela se fait en comparant le hash des transactions, ensuite toutes les transactions filles ont été labellisées comme frauduleuses.

Au final un fichier (.csv) a été créé à partir de ces deux bases de données où chaque ligne représente une transaction avec le nombres d'entrées et de sorties ainsi que le montant transféré en BTC (bitcoins), Les variables qui y sont présentes sont :

- **In-degree :** Nombre de transactions desquelles une transaction donnée reçoit de l'argent
- **Out-degree :** Nombre de transactions desquelles une transaction donnée envoie de l'argent
- **In-BTC :** Nombre de bitcoins sur chaque arête entrante d'une transaction donnée
- **Out-BTC :** Nombre de bitcoins sur chaque arête sortante d'une transaction donnée
- **Total-BTC :** Nombre net de bitcoins pour une transaction donnée en tenant compte de tous les bords rentrants et sortants de cette transaction.
- **Average in-BTC :** Nombre moyen de bitcoins sur chaque bord entrant d'une transaction donnée
- **Average Out-BTC :** Nombre moyen de bitcoins sur chaque bord sortant d'une transaction donnée
- **Is-Anomaly :** Statut d'une transaction donnée, si elle est malveillante ou non

Le jeu de données final comprenait environ 10 millions de transactions normales non malveillantes et 82 transactions malveillantes. Il est à noter que les données sont fortement déséquilibrées et qu'il est difficile de créer un modèle robuste pour la détection des anomalies. Tous les algorithmes de détection d'anomalies ne sont pas capables de traiter un tel volume de données. Ainsi, pour certains algorithmes, un nombre spécifique de sous-échantillons a été modélisé et comparé pour trouver le modèle le mieux adapté.

4.2 Prétraitement des données

L'analyse exploratoire des données comprend diverses visualisations qui aident à comprendre les données et la relation entre leurs dimensions de manière plus approfondie. Elle a également permis de détecter le déséquilibre des classes dans les données ainsi que la corrélation entre les caractéristiques. L'analyse exploratoire a également permis de comprendre quel type de transformation peut être appliqué aux données pour les préparer avant la modélisation.

L'analyse exploratoire des données a révélé certaines informations sur les données. Comme je l'ai également mentionné dans la section précédente, la distribution des classes de l'ensemble de données est fortement déséquilibrée avec 10 247 944 points de données non malveillants et seulement 82 transactions malveillantes. Les transactions malveillantes ne représentent qu'environ 0,0008% de l'ensemble des données, comme le montre la figure 4.1 .

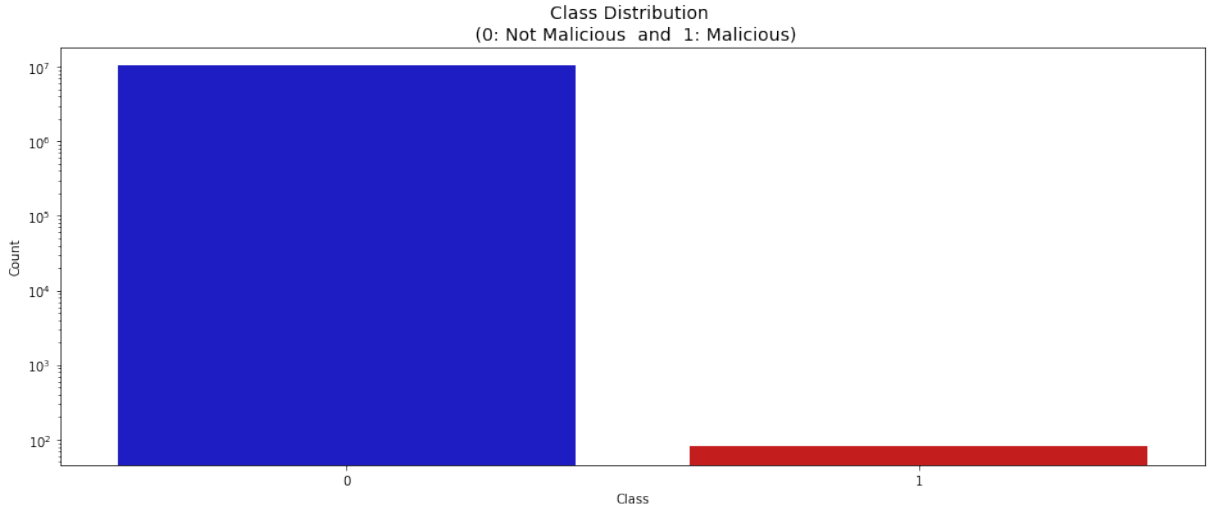

FIG. 4.1: Distribution des données frauduleuses et non-frauduleuses

J'ai ensuite tracé les distributions de toutes les variables pour savoir si elles sont asymétriques et si elles nécessitent une transformation ou une normalisation. Par la suite, j'ai illustré les données dans une grille d'histogrammes de distribution dans la figure 4.2. Chaque histogramme représente une seule caractéristique de l'ensemble de données et sa distribution de données respective. Pour la représentation visuelle, l'axe des x et l'axe des y ont été mis à l'échelle logarithmique.

Nous pouvons voir que toutes les variables des données suivent une distribution fortement inclinée vers la droite. Cette asymétrie des données peut entraîner des difficultés dans le processus de modélisation. Les transformations permettant de réduire l'asymétrie des données sont la racine carrée, la racine cubique et le logarithme. Une transformation de la fonction Log semble avoir un effet positif sur les données. Cependant, selon le tableau 4.1, nos données contiennent des valeurs nulles, et la transformation logarithmique standard ne peut pas leur être appliquée. Le tableau décrit également d'autres statistiques des données telles que la moyenne, l'écart type, la valeur minimale et la valeur maximale pour chaque caractéristique de l'ensemble de données avant la transformation.

	indegree	outdegree	in_btc	out_btc	total_btc	mean_in_btc	mean_out_btc
count	10248134.000	10248134.000	10248134.000	10248134.000	10248134.000	10248134.000	10248134.000
mean	2.006	2.113	119.640	120.209	239.849	109.326	63.254
std	7.122	4.136	2211.396	2209.615	4417.813	1639.967	1207.891
min	0.000	0.000	0.000	0.000	0.000	0.000	0.000
max	1311.000	927.000	550000.000	500020.700	1050000.000	499259.588	500000.000

TAB. 4.1: Statistiques des données brutes

Puisque la transformation logarithmique standard ne peut pas être appliquée aux données, lors de plusieurs expériences, une transformation log(x+1) a été appliquée en même temps que la transformation "Robust Scaler". L'objectif de cette transformation est de normaliser la valeur de l'échelle de mesure comme le montre la figure 4.3 qui illustre l'histogramme de chaque donnée après transformation.

Les matrices de corrélation sont l'un des facteurs essentiels de la compréhension des données. Elles peuvent nous aider à déterminer si les caractéristiques influencent fortement la malveillance d'une transaction spécifique. Une corrélation extrêmement positive ou négative peut révéler la capacité d'une caractéristique à affecter la malveillance. Pour s'assurer que le déséquilibre important des données n'affecte pas la corrélation des caractéristiques, une deuxième matrice de corrélation pour un sous-échantillon aléatoire avec des classes équilibrées est également créée et analysée. La figure 4.4 visualise la matrice de corrélation pour les données complètes, tandis que la figure illustre la matrice de corrélation pour un sous-échantillon aléatoire à classes équilibrées. Sur la base des deux matrices de corrélation, nous pouvons dire que l'indice et l'écart présentent une corrélation négative avec la malveillance. Plus ces valeurs de corrélation sont faibles, plus

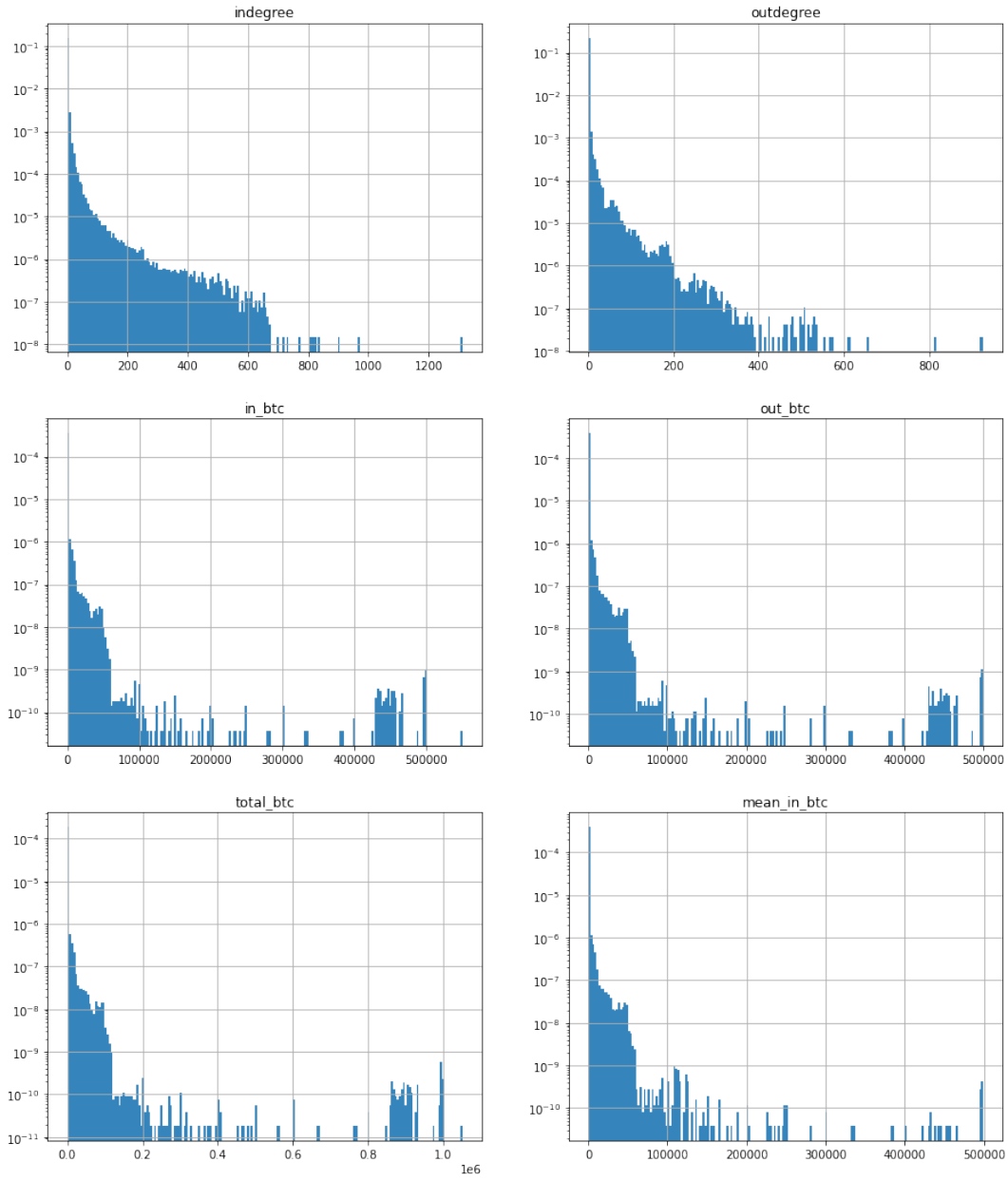

FIG. 4.2: Distribution des fréquences de chaque donnée frauduleuse et non-frauduleuse

le résultat sera probablement une transaction frauduleuse. Les caractéristiques in-btc, out-btc, mean-in-btc, mean-out-btc et total-btc quant à elles présentent une corrélation positive. Plus ces valeurs sont élevées, plus il est probable que le résultat final soit une transaction frauduleuse. Une couleur bleue forte représente

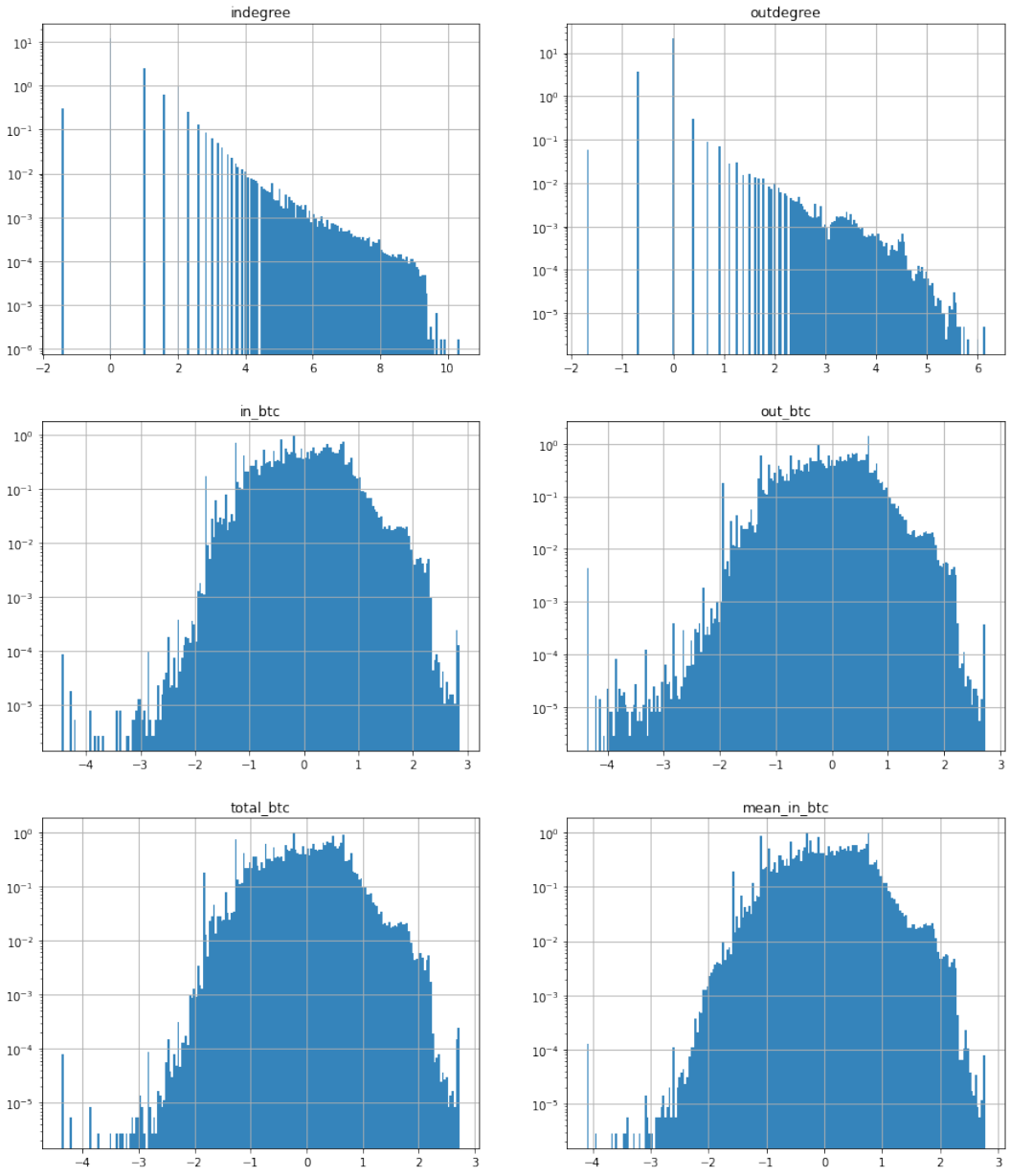

FIG. 4.3: Distribution des fréquences de chaque donnée frauduleuse et non-frauduleuse après avoir normaliser et mis les données à l'echelle

des valeurs de corrélation élevées, une couleur rouge forte représente des valeurs de corrélation faibles, toutes les nuances entre les deux montrent des valeurs variantes entre les deux.

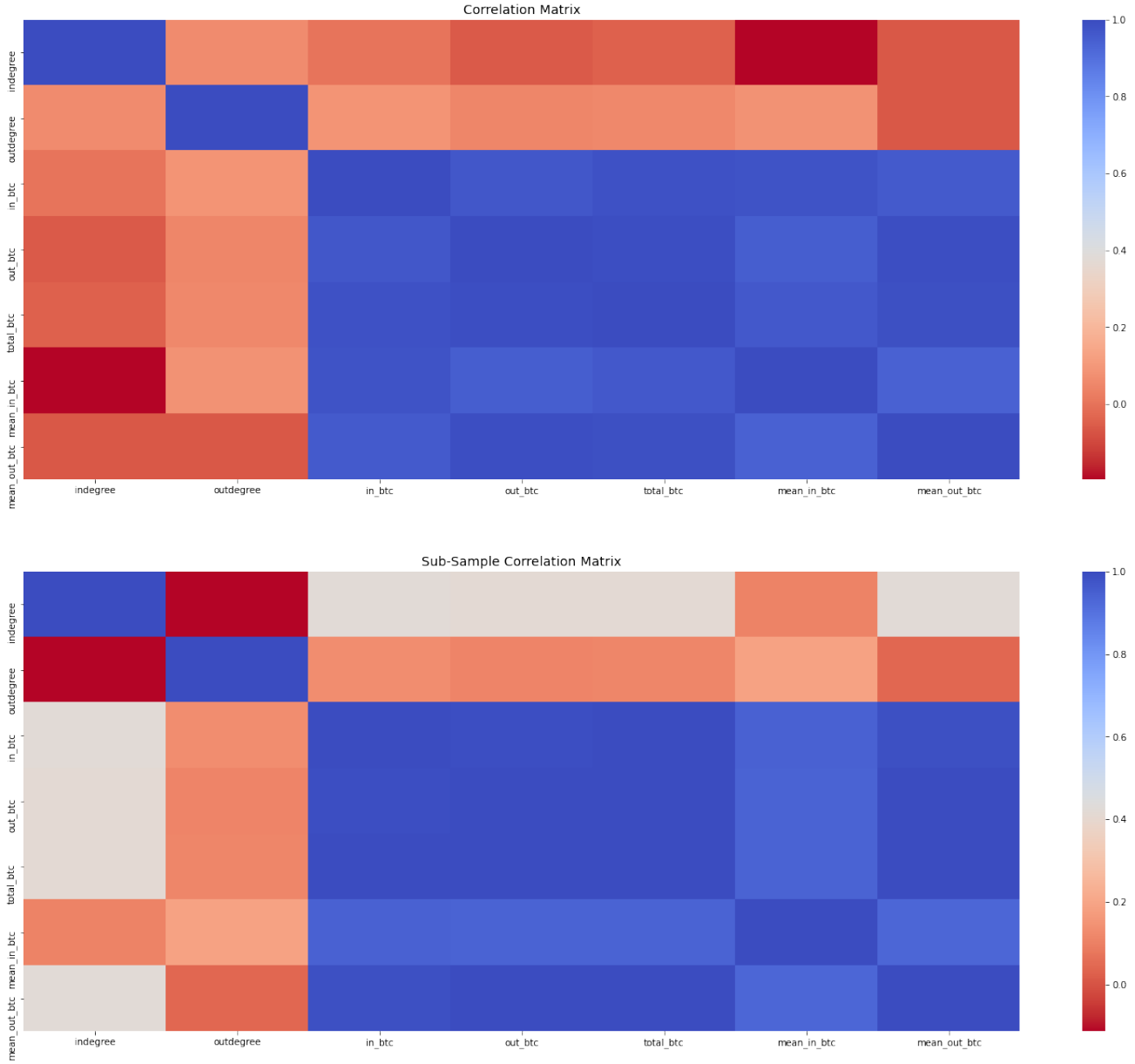

FIG. 4.4: Matrice de correlations des données origilales et d'un sous-échantillon aléatoire équilibré par classe

Troisième partie

Système réalisé

Sommaire

Chapitre 5

Implémentation du système et résultats

Sommaire

5.1 Introduction

Ce chapitre explique les résultats de l'évaluation de tous les algorithmes et comment ils ont été obtenus. Diverses mesures d'évaluation ont été utilisées pour effectuer l'évaluation. En raison de la grande quantité de données, la complexité temporelle de certains algorithmes était énorme. Toutefois, pour résoudre ce problème, nous avons sous-échantillonné de manière aléatoire 1/10e des données d'apprentissage et adapté 20 modèles différents avec des hyper paramètres finement ajustés. Les paramètres tels que le sous-échantillonnage de 1/10e des données et l'entraînement de 20 modèles exactement ont été calculés sur la base d'expériences. Les valeurs ont été finalisées lorsque les performances du modèle ne changeait plus, même si la taille de l'échantillon ou le nombre de modèles augmentait.

5.2 K-means

La performance du clustering K-means n'est pas gravement affectée par le vaste volume de données. En raison de la nature de catégorisation du clustering K-means, contrairement à d'autres algorithmes dans les expériences de cette thèse, les données ne sont pas sous-échantillonnées en plusieurs morceaux. K-means peut traiter un grand nombre de données et les catégorise en groupes dans lesquels les points de données sont similaires les uns aux autres. Cette expérience vise à détecter et à distinguer les groupes de points de données qui contiennent un maximum de points de données malveillants et non malveillants. Toutes les données d'apprentissage sont introduites dans l'algorithme de clustering k-means avec des valeurs de variables k variées pour générer k modèles. La valeur de l'inertie est calculée pour chaque modèle k ; l'inertie est définie comme la somme des distances au carré (SSD) des échantillons par rapport à leur plus proche voisin. Cette métrique est utilisée pour sélectionner la meilleure valeur de k, produisant ainsi le meilleur modèle de clustering k-means. La figure est un graphique en forme de coude de la SSD en fonction du nombre de clusters. D'après la figure 5.1, la valeur de k = 5 semble être satisfaisante.

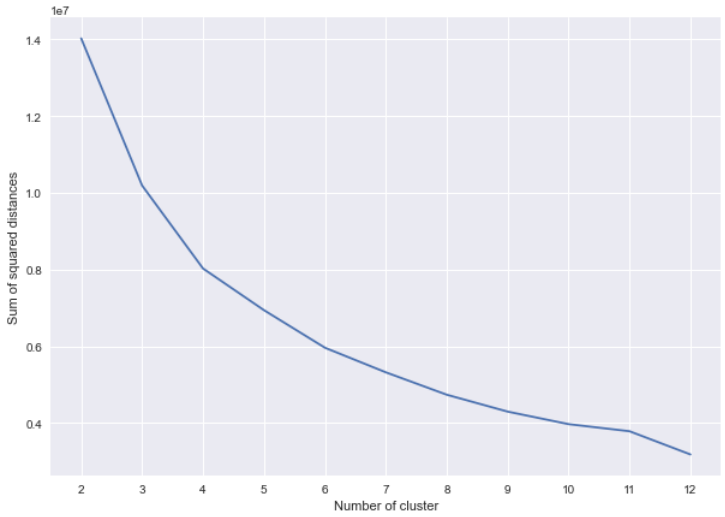

FIG. 5.1: Tracé entre le k-cluster et la somme des distances au carré des échantillons à leur centre de cluster le plus proche.

Le modèle avec la valeur k = 5 est évalué avec des données de formation et des données de test. Le tableau 5.1 décrit les statistiques des mesures d'évaluation en séparant les points de données malveillants des points de données non malveillants. points de données non malveillants. En se basant sur les données des tableaux 5.1 et 5.2, on peut déduire que le modèle a classé la majorité des points de données malveillants dans les clusters 2 et 4. Au contraire, les points de données non malveillants sont principalement désignés dans le cluster 1, le cluster 3 et le cluster 5.

Cluster	Nombre de données malicieuses	Pourcentage des données malicieuses	Nombre de données régulières	Pourcentage de données régulières
1	6	9.091%	2310150	28.178%
2	0	0.0%	1272386	15.52%
3	44	66.667%	1645813	20.075%
4	16	24.242%	961016	11.722%
5	0	0.0%	2009076	24.506%

TAB. 5.1: Distribution des cluster avec les données d'entraînement

Cluster	Nombre de données malicieuses	Pourcentage des données malicieuses	Nombre de données régulières	Pourcentage de données régulières
1	1	6.25%	576492	28.127%
2	0	0.0%	317811	15.506%
3	10	62.5%	411777	20.09%
4	4	25.0%	240452	11.732%
5	1	6.25%	503079	24.545%

TAB. 5.2: Distribution des cluster avec les données de test

La figures 5.2 est une illustration de l'espace tridimensionnel qui permetet de comprendre clairement comment les données malveillantes et non malveillantes

sont regroupées et distinguées les unes des autres. Dans l'espace tridimensionnel, chaque axe est représenté par une composante principale extraite des données réelles.

Afin de créer une matrice de confusion de prédiction conventionnelle, les clusters avec la majorité des points de données malveillantes sont considérés comme des clusters. Ces points de données malveillants (Cluster 2 et Cluster 4) ont été fusionnés en un seul. Inversement, les clusters avec la majorité des points de données non malveillants (Cluster 1, Cluster 3 et Cluster 5) ont également été fusionnés en un seul. Ces clusters fusionnés ont été étiquetés comme points de données malveillants et non malveillants pour calculer une matrice de confusion finale. En se basant sur les résultats, on peut observer que le modèle de clustering k-means a obtenu des résultats satisfaisants. Le modèle avec k = 5 a été en mesure de détecter 67% des données malveillantes et 64% des données non malveillantes dans les données d'entraînement. Il a également détecté avec succès 56% des données malveillantes dans les données de test ainsi que 64% des données non malveillantes tandis que les faux positifs n'étaient que de 44%, La figure 5.3 en est l'illustration.

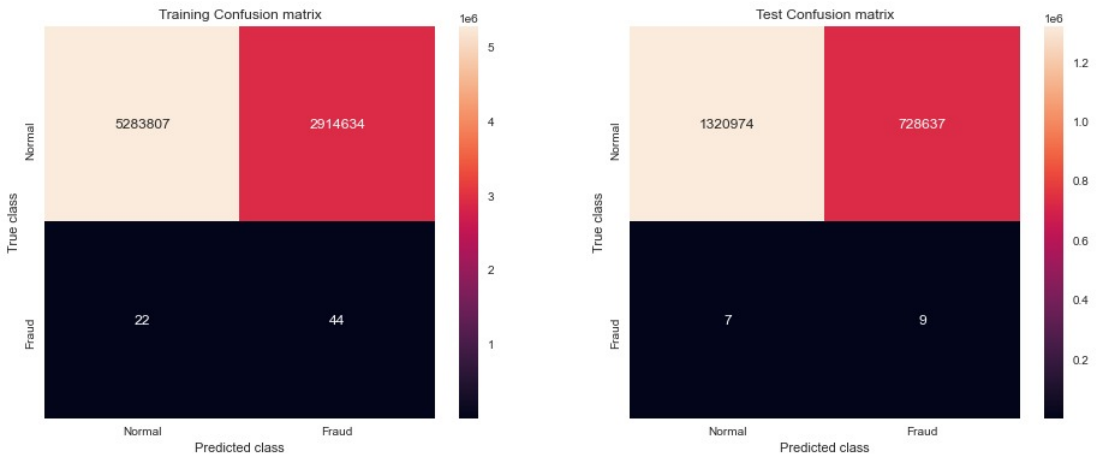

FIG. 5.3: Matrice de confusion des nombres de formation et de validation pour k=5

5.3 Isolation Forest

Pour gérer l'extrême déséquilibre des classes, la technique de suréchantillonnage des minorités synthétiques (SMOTE) est utilisée pour générer davantage d'échantillons d'entraînement synthétiques malveillants afin que l'algorithme puisse détecter plus efficacement un modèle d'anomalies dans les données. Le rapport pour les points de données synthétiques nouvellement générés est également estimé par une

l'optimisation des hyperparamètres. Tous les hyperparamètres sont optimisés en minimisant une fonction de perte. Pour ce cas de recherche, la fonction de perte choisie est le complément du score F1. Les paramètres optimisés estimés pour la forêt d'isolation sont 40 N-estimateurs et une fraction de contamination/sur-échantillonnage de 0,057. Tous les modèles formés sont évalués à l'aide d'un seul ensemble de test, qui contient des données non synthétiques non vues par tous les modèles. Le tableau 5.3 décrit les paramètres d'évaluation pour les données d'apprentissage et le tableau 5.4 décrit les paramètres d'évaluation pour les données de test.

Iteration	Accuracy	Precision	Recall	F1	ROC
13	0.931334	0.370682	0.391774	0.380937	0.724573
4	0.925377	0.315913	0.329332	0.322483	0.721707
17	0.931344	0.370739	0.391731	0.380946	0.716042
16	0.933363	0.388501	0.410648	0.399268	0.715528
10	0.931744	0.374291	0.395605	0.384653	0.714033
9	0.929847	0.357441	0.377244	0.367076	0.711494
20	0.931748	0.374238	0.395284	0.384473	0.710784
15	0.929943	0.358038	0.377223	0.367380	0.710547
12	0.931709	0.373642	0.393871	0.383490	0.703796
7	0.930783	0.365867	0.386724	0.376007	0.703310
5	0.930560	0.363916	0.384670	0.374005	0.701172
18	0.928444	0.345352	0.365047	0.354926	0.701047
11	0.932133	0.377710	0.399243	0.388178	0.699587
1	0.930795	0.365930	0.386681	0.376020	0.699114
19	0.931675	0.373694	0.395006	0.384054	0.695487
8	0.932296	0.378302	0.397124	0.387485	0.691118
2	0.929723	0.356521	0.376731	0.366347	0.686369
6	0.931431	0.370690	0.389228	0.379733	0.680089
3	0.927448	0.336414	0.355160	0.345533	0.676338
14	0.929987	0.358061	0.376281	0.366945	0.647852

Tab. 5.3: Métriques d'évaluation de Isolation forest avec les données d'entraînement

La sélection d'un modèle sur la base de ces métriques d'évaluation est un défi ; cependant, pour cette thèse, la métrique F1-score est choisie comme référence car nous sommes intéressés par la combinaison optimisée de précision et de rappel d'un modèle. En analysant les illustrations visuelles et en se basant sur la métrique score-f1 la plus élevée et sur une courbe ROC et d'autres métriques d'évaluation assez raisonnables, l'itération 7 est favorable. La figure 5.4 représente sa courbe (ROC).

La matrice de confusion donne une vue d'ensemble de l'évaluation des classificateurs. Les mesures de confusion de la figure représentent l'évaluation de la classification sous forme de pourcentage. Sur la base de la figure 5.5, on peut déterminer que 50% des transactions malveillantes ont été correctement détectées

Iteration	Accuracy	Precision	Recall	F1	ROC
10	0.961959	0.000103	0.500000	0.000205	0.837556
13	0.961705	0.000102	0.500000	0.000204	0.835131
7	0.961918	0.000102	0.500000	0.000205	0.832576
20	0.962248	0.000103	0.500000	0.000207	0.832478
15	0.961392	0.000101	0.500000	0.000202	0.829213
5	0.961548	0.000102	0.500000	0.000203	0.820845
2	0.960923	0.000100	0.500000	0.000200	0.819646
16	0.962528	0.000104	0.500000	0.000208	0.816566
17	0.961509	0.000101	0.500000	0.000203	0.814341
3	0.959909	0.000097	0.500000	0.000195	0.813449
8	0.963009	0.000106	0.500000	0.000211	0.813292
12	0.962393	0.000104	0.500000	0.000208	0.811826
14	0.961525	0.000101	0.500000	0.000203	0.810969
19	0.962106	0.000103	0.500000	0.000206	0.809928
4	0.958805	0.000083	0.437500	0.000166	0.808554
11	0.962402	0.000104	0.500000	0.000208	0.807388
18	0.960789	0.000100	0.500000	0.000199	0.806789
9	0.961001	0.000088	0.437500	0.000175	0.806708
1	0.961842	0.000102	0.500000	0.000205	0.798591
6	0.962163	0.000103	0.500000	0.000206	0.796915

TAB. 5.4: Métriques d'évaluation de Isolation forest avec les données de test

et, tout aussi important, 96% des transactions non malveillantes ont également été détectées, des transactions non malveillantes ont également été correctement classées comme des transactions normales.

5.4 Histogram based outlier score

Le score d'aberration basé sur l'histogramme (HBOS) n'est pas aussi sensible au volume massif de données. Le temps de calcul de HBOS peut être raisonnable avec une grande quantité de données d'entraînement. Cependant, pour que la comparaison des résultats avec d'autres algorithmes soit équitable, les données d'apprentissage sont sous-échantillonnées de manière aléatoire en 20 morceaux, et chaque morceau contient 1/10e des données avec des points de données malveillants substantiels. La technique de suréchantillonnage de minorités synthétiques (SMOTE) est utilisée pour équilibrer partiellement le déséquilibre des classes afin que l'algorithme puisse rechercher un modèle reconnaissable. Tous les hyper-paramètres, y compris le ratio de sur-échantillonnage, sont estimés par le processus de réglage des hyperparamètres. Les tableaux 5.5 et 5.6 illustrent les statisqtique de chaque itteration avec les données d'entrainement et de test.

L'illustration visuelles ci-dessous est une plongée plus profonde dans le modèle sélectionné, qui est de l'itération 13. La caractéristique d'exploitation du récep-

Iteration	Accuracy	Precision	Recall	F1	Macro-F1	ROC
6	0.742560	0.382035	0.471760	0.422183	0.628284	0.599742
14	0.741051	0.380322	0.474949	0.422401	0.627759	0.595092
11	0.741033	0.380249	0.474709	0.422261	0.627687	0.592896
10	0.741860	0.380838	0.471162	0.421212	0.627549	0.587042
15	0.741630	0.380451	0.471001	0.420911	0.627316	0.593749
2	0.740640	0.379514	0.474008	0.421530	0.627189	0.601697
4	0.740666	0.379378	0.473097	0.421086	0.626996	0.593190
16	0.740225	0.378610	0.472598	0.420415	0.626506	0.596930
19	0.740149	0.378500	0.472627	0.420359	0.626448	0.600446
3	0.740242	0.378539	0.472103	0.420175	0.626403	0.592166
8	0.740228	0.378484	0.471936	0.420075	0.626351	0.597607
17	0.740025	0.378282	0.472475	0.420164	0.626308	0.592665
18	0.739915	0.378000	0.471892	0.419760	0.626077	0.599659
9	0.739697	0.377623	0.471652	0.419432	0.625836	0.594727
12	0.739896	0.377772	0.470868	0.419214	0.625819	0.598939
13	0.740423	0.378195	0.468933	0.418704	0.625804	0.588402
5	0.739422	0.377051	0.470863	0.418767	0.625417	0.590200
20	0.739351	0.376796	0.470119	0.418316	0.625181	0.593388
1	0.739564	0.376945	0.469232	0.418056	0.625150	0.592564
7	0.738907	0.376036	0.469668	0.417669	0.624700	0.588883

Tab. 5.5: Métriques d'évaluation de Histogram based outlier score (HBOS) avec les données d'entrainement

Iteration	Accuracy	Precision	Recall	F1	Macro-F1	ROC
6	0.811016	0.000026	0.625000	0.000052	0.628284	0.791550
14	0.807372	0.000025	0.625000	0.000051	0.627759	0.748616
11	0.807059	0.000025	0.625000	0.000051	0.627687	0.770593
10	0.809834	0.000026	0.625000	0.000051	0.627549	0.774287
15	0.809840	0.000026	0.625000	0.000051	0.627316	0.748306
2	0.807051	0.000025	0.625000	0.000051	0.627189	0.791561
4	0.807451	0.000025	0.625000	0.000051	0.626996	0.789694
16	0.806858	0.000025	0.625000	0.000051	0.626506	0.774686
19	0.807048	0.000025	0.625000	0.000051	0.626448	0.791532
3	0.806692	0.000025	0.625000	0.000051	0.626403	0.775532
8	0.807075	0.000025	0.625000	0.000051	0.626351	0.774370
17	0.806233	0.000025	0.625000	0.000050	0.626308	0.788920
18	0.807062	0.000025	0.625000	0.000051	0.626077	0.791563
9	0.807332	0.000025	0.625000	0.000051	0.625836	0.774856
12	0.807333	0.000025	0.625000	0.000051	0.625819	0.791589
13	0.808792	0.000025	0.625000	0.000051	0.625804	0.792737
5	0.806302	0.000025	0.625000	0.000050	0.625417	0.791519
20	0.807168	0.000025	0.625000	0.000051	0.625181	0.774792
1	0.807096	0.000025	0.625000	0.000051	0.625150	0.774699
7	0.805768	0.000025	0.625000	0.000050	0.624700	0.791502

Tab. 5.6: Métriques d'évaluation de Histogram based outlier score (HBOS) avec les données de test

teur (ROC) de validation a donné de bien meilleurs résultats que la caractéristique d'exploitation du récepteur (ROC) d'apprentissage pour le score de valeurs aberrantes basé sur l'histogramme (HBOS). Trouver le seuil optimal pour le modèle est une tâche difficile et nécessite des recherches et des expérimentations approfondies. Néanmoins, la bibliothèque utilisée par cette thèse pour détecter les anomalies possède un algorithme intégré, qui sélectionne de manière optimale le seuil avec les meilleurs résultats possibles. La figure 5.6 montre un comportement intéressant à proximité de valeurs de seuil élevé, le modèle commence à s'orienter vers une sous-adaptation, puis vers une sur-adaptation. Ce phénomène est dû à l'augmentation de la valeur du seuil par rapport à l'évaluation du modèle.

La matrice de confusion pour le score de valeurs aberrantes basé sur l'histogramme (HBOS) illustré dans la figure 5.7 révèle que, bien que 62% des transactions malveillantes aient été détectées avec succès et que 81% des transactions non malveillantes aient également été détectées correctement dans l'ensemble de test, le taux de faux positifs a été assez important, soit 38% pour cet algorithme. Pour l'évaluation de la formation et du test, le modèle semble fonctionner avec des faux positifs et des faux négatifs modérés. Bien que la signification des mesures de performance soit entièrement basée sur le cas d'utilisation, à savoir, ce mémoire, dans un cas d'utilisation idéal, vise à maximiser la détection des vrais positifs et des vrais négatifs tout en minimisant les faux positifs.

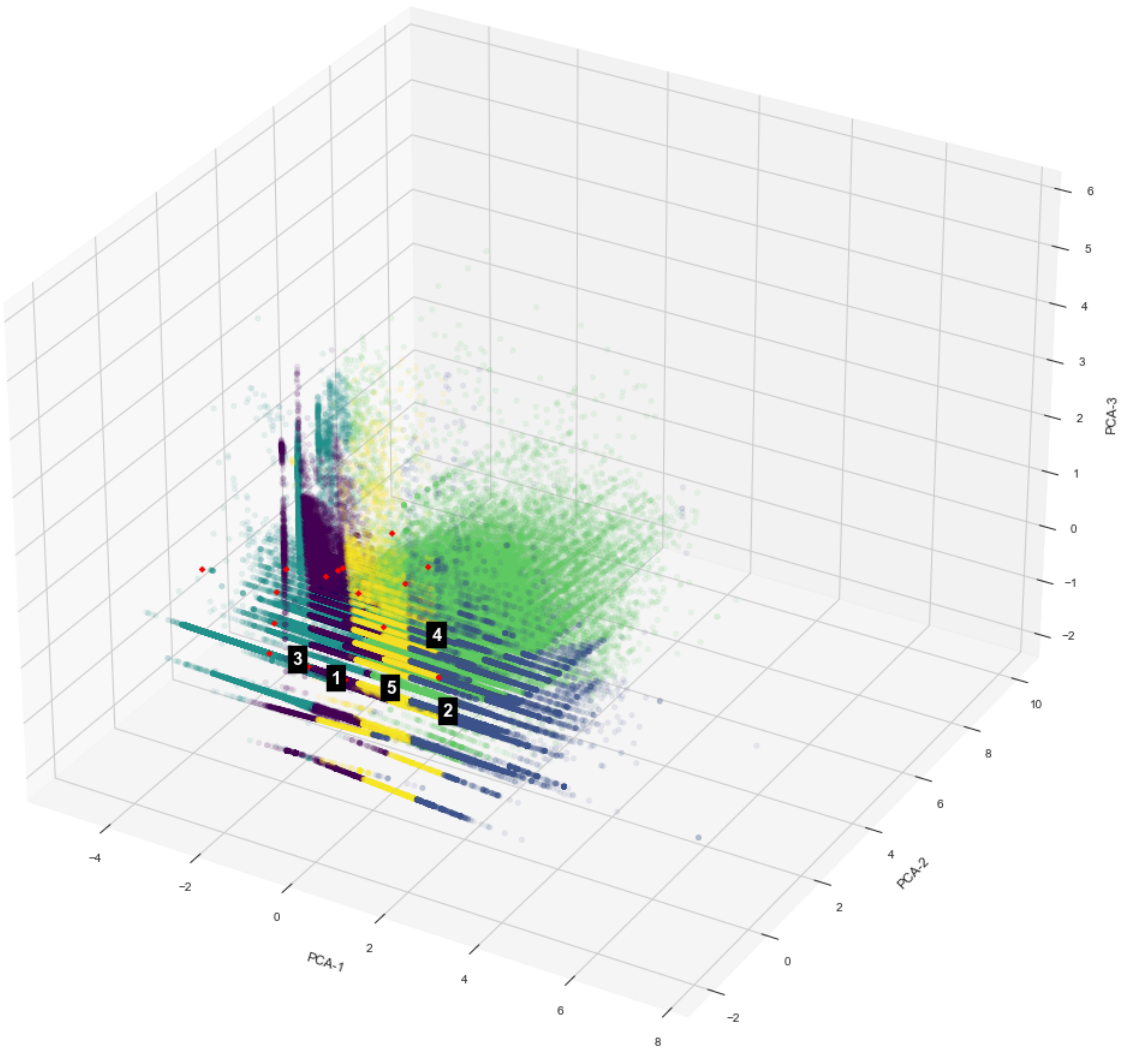

FIG. 5.2: Visualisation 3D de la distribution des points de données avec k=5

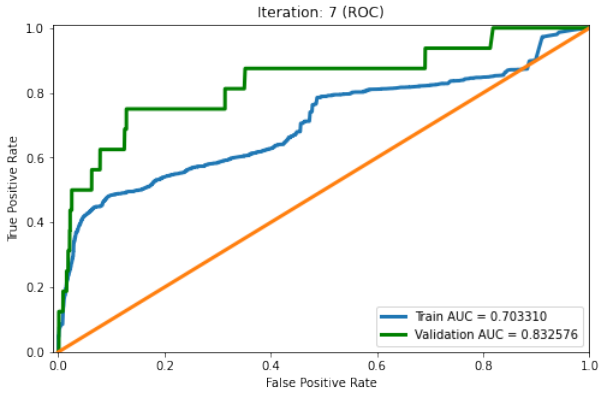

FIG. 5.4: Zone d'apprentissage et de validation sous la courbe (AUC) de la courbe caractéristique de fonctionnement du récepteur (ROC) pour l'itération 7

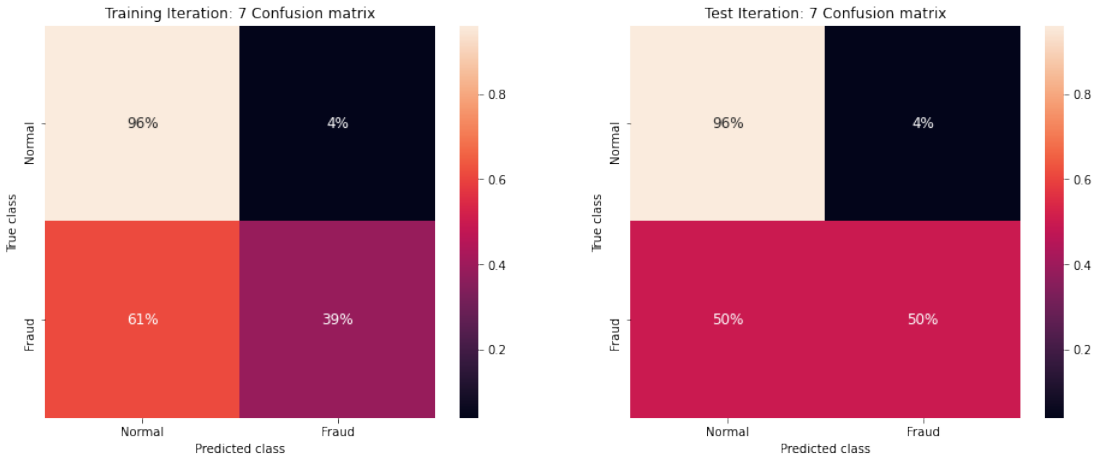

FIG. 5.5: Matrice de confusion des nombres de formation et de validation pour l'itération 7

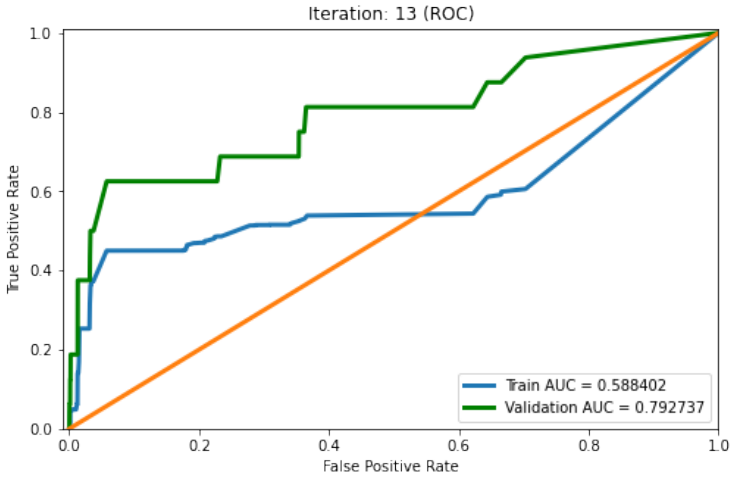

FIG. 5.6: Zone d'apprentissage et de validation sous la courbe (AUC) de la courbe caractéristique de fonctionnement du récepteur (ROC) pour l'itération 13

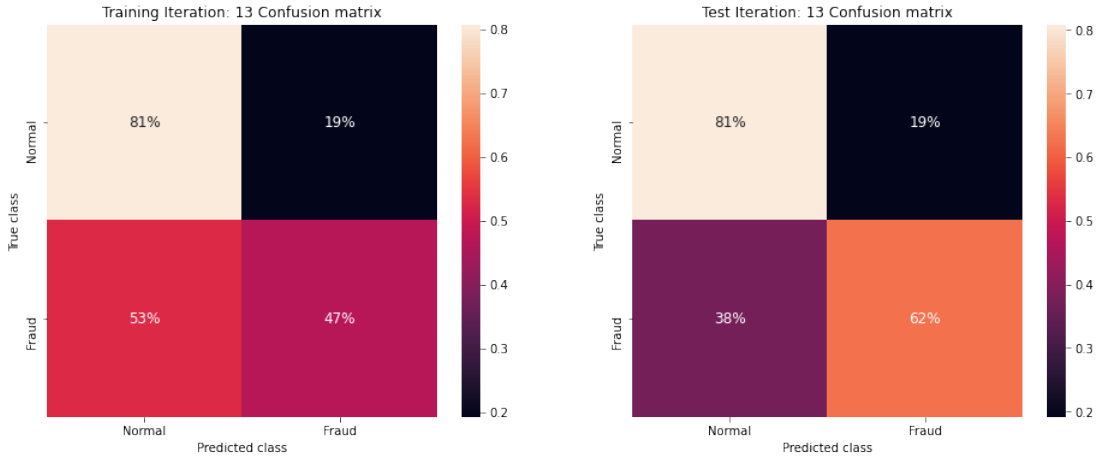

FIG. 5.7: Matrice de confusion des nombres de formation et de validation pour l'itération 13

Chapitre 6

Discussion des résultats :

Sommaire

6.1 Introduction

Ce chapitre décrit la comparaison entre les résultats de toutes les expériences réalisées pour ce mémoire. Au total, trois algorithmes sont utilisés pour détecter les transactions malveillantes et non malveillantes dans l'ensemble de données bitcoin utilisé pour cette recherche. Le tableau 7.1 décrit les statistiques d'évaluation calculées pour chaque algorithme.

Toutes les mesures d'évaluation ont leur signification en fonction du type de solution requise. Ce mémoire de recherche vise à maximiser la détection des transactions malveillantes et non malveillantes tout en minimisant les cas de faux positifs. Certains algorithmes sont plus performants que d'autres, tandis que le classificateur d'ensemble prend en compte le vote de tous les algorithmes pour prendre une décision.

6.2 Discussion

Ici une comparaison de tous les algorithmes est faite et le tableau 6.2 décrit les statistiques d'évaluation calculées.

Classifer	Accuracy	Balanced-Accuracy	Macro-Precision	Macro-Recall	Macro-F1	Macro-ROC	Precision	Recall	F1	ROC
IForest	0.957096	0.853549	0.500067	0.853549	0.489175	0.877862	0.000136	0.7500	0.000273	0.877862
HBOS	0.864890	0.807446	0.500020	0.807446	0.463819	0.850437	0.000043	0.7500	0.000087	0.850437
Kmeans	0.642473	0.477488	0.499999	0.477488	0.391168	NA	0.000007	0.3125	0.000014	NA
Ensemble	0.899701	0.824851	0.500028	0.824851	0.473660	0.862428	0.000058	0.7500	0.000117	0.862428

TAB. 6.1: Statistiques de tous les algorithmes

Sur la base des données du tableau 6.2 et de l'illustration comparative de la figure 6.1 de l'aire sous la courbe du ROC, ainsi que l'efficacité et de la comparaison des courbes de précision et de rappel des figures 6.2 et 6.3, il est difficile de déterminer le meilleur choix.

Cependant, il existe un compromis entre la robustesse et les performances optimales. En termes de robustesse, la méthode d'ensemble semble être un meilleur choix car elle prend en compte un vote décisif de six classificateurs avant de prendre sa décision, ce vote peut avoir des poids variables ajustés pour chaque algorithme en fonction du cas d'utilisation. Néanmoins, pour une performance solo optimale, la concurrence est rude entre Isolation Forest, HBOS et K-means. La figure 6.4 est un histogramme représentant la précision de chaque modèle sur l'ensemble de test. Il n'est pas judicieux de se baser uniquement sur la précision pour sélectionner un modèle dans ce cas d'utilisation particulier. S'appuyer uniquement sur la précision peut être trompeur.

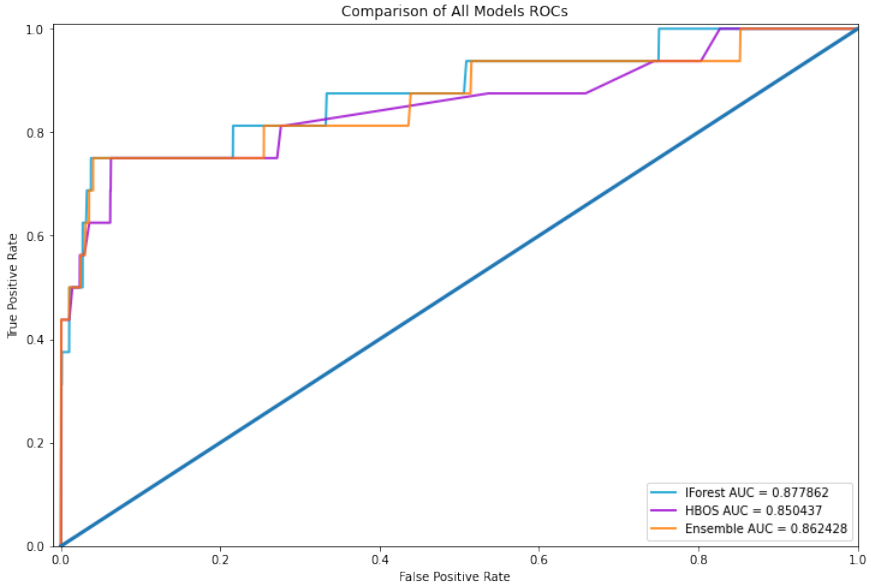

FIG. 6.1: comparaison des courbes ROc pour tout les algorithmes

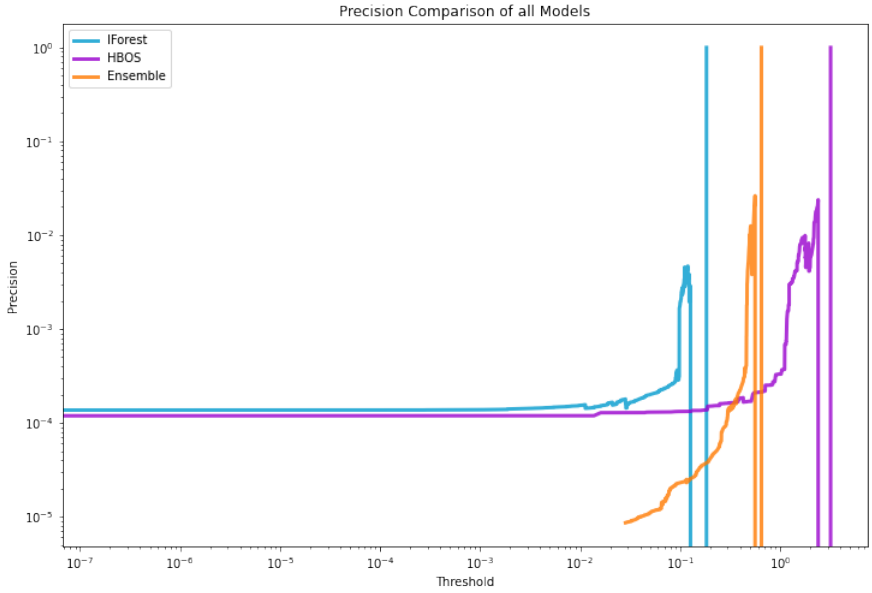

FIG. 6.2: Comparaison des précisions de tous les algorithmes

Les métriques de macro-évaluation sont des indicateurs utiles dans les scénarios où les performances globales du système doivent être prises en considération. L'histogramme de la figure 6.5 représente une comparaison du rappel macro pour tous les algorithmes, l'histogramme de la figure 6.6 illustre la comparaison des

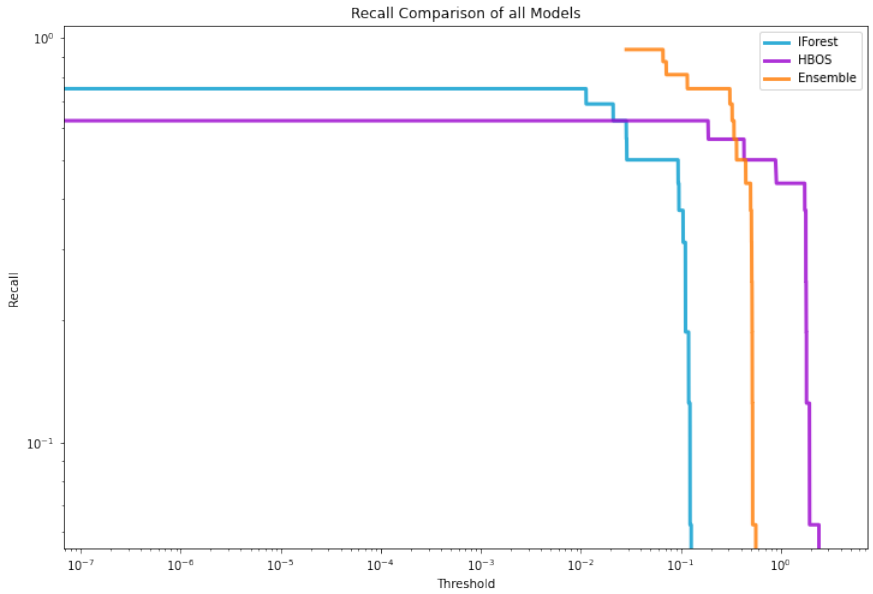

FIG. 6.3: Comparaison des rappels de tous les algorithmes

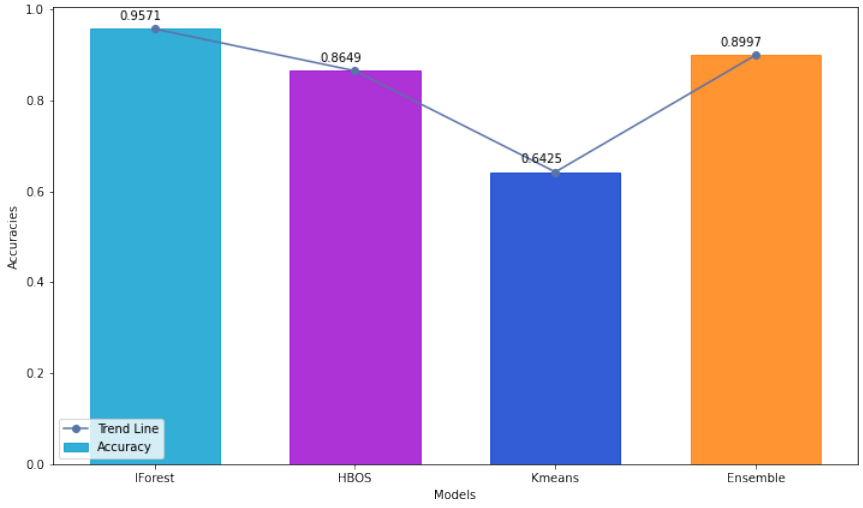

FIG. 6.4: Accuracy de tous les modèles

scores macro-f1 pour tous les algorithmes. D'après les illustrations visuelles de ces mesures d'évaluation macro, la méthode d'ensemble est plus stable et plus robuste. Cependant, la forêt d'isolement est légèrement plus performant. Le choix d'un algorithme parmi ceux-ci est difficile et peut nécessiter des informations plus approfondies sur les scénarios d'utilisation.

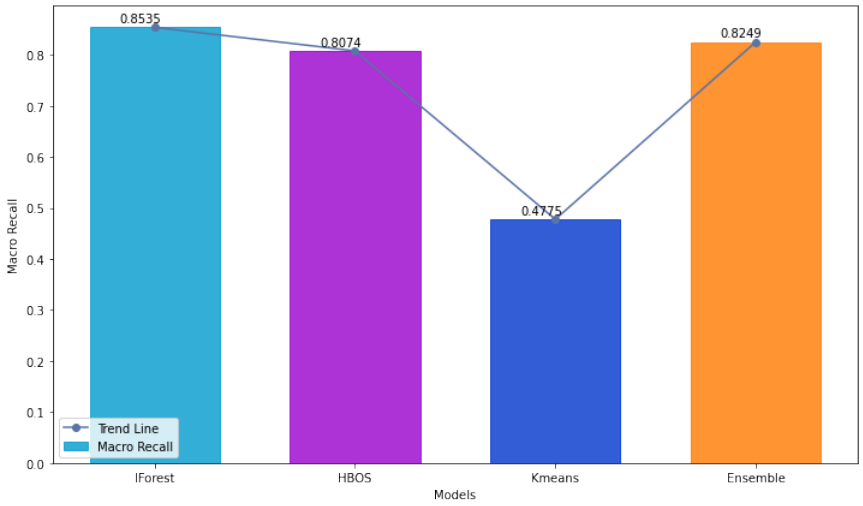

FIG. 6.5: Histogramme comparant la métrique de rappel macro pour tous les algorithmes.

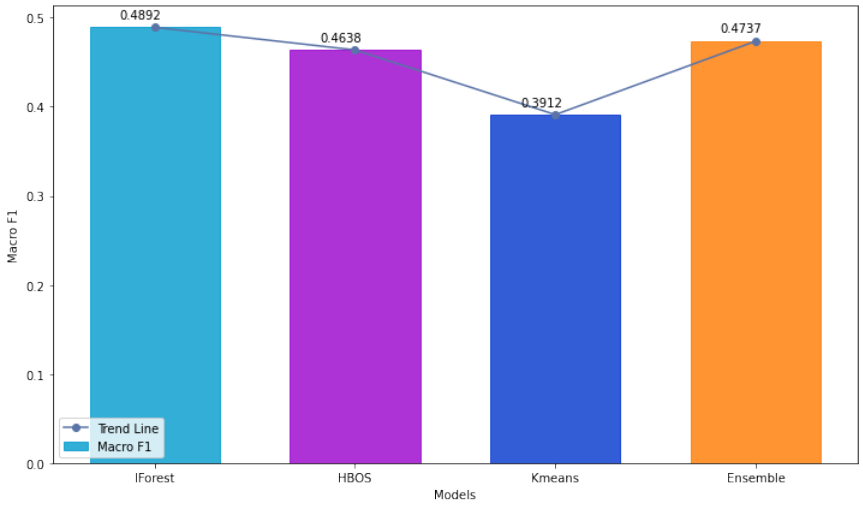

FIG. 6.6: Histogramme comparant la métrique macro f1 pour tous les algorithmes.

6.3 Conclusion

Dans ce chapitre j'ai présenté les résultats de chaque algorithme, une comparaison entre eux, après toutes ces expérimentations c'est la forêt d'isolement qui s'en sort le mieux en termes de résultat et de robustesse.

Conclusion

L'évaluation de plusieurs algorithmes non supervisés pour détecter les comportements anormaux dans une blockchain basée sur les sorties de transactions non dépensées (UTXO) montre que les activités malveillantes peuvent être identifiées avec succès. Cela va dans le sens d'autres recherches sur la détection des anomalies dans les blockchains. Si les paramètres d'évaluation des recherches précédentes avaient été divulgués, la comparaison aurait été encore plus évidente. Bien qu'elle puisse être indirectement comparée à la détection de la fraude par carte de crédit, la dynamique des données est entièrement différente et induit donc une variété de problèmes.

Il est possible que de meilleurs résultats auraient pu être obtenus avec des données plus malveillantes. La petite taille des points de données transactionnelles malveillantes dans l'ensemble de données signifie qu'il y a peu de modèles anormaux à reconnaître. Une tentative a été faite pour surmonter ce problème en générant synthétiquement des points de données malveillants tout en veillant à ce qu'ils ne soient pas exagérés. Cependant, cela affecte la qualité de la détection des anomalies, ce qui entraîne une baisse des performances des modèles.

La sélection des caractéristiques extraites de l'ensemble des données de transaction de la blockchain pour déterminer le comportement anormal semble un facteur essentiel car certaines variables peuvent détenir une perspective unique. Par exemple, lorsque la détéctions d'anomalies est traité comme un problème de série temporelle, il peut y avoir d'autres solutions. Cependant, ce mémoire se concentre sur la recherche de modèles anormaux en utilisant des techniques d'apprentissage non supervisé, et avoir un élément de série temporelle dans le problème augmenterait la complexité de la portée.

Néanmoins, les chercheurs ont expérimenté diverses techniques non supervisées pour détecter les transactions anormales dans une blockchain et ont découvert des moyens relativement efficaces pour y parvenir. Le manque de données pertinentes et les limitations de la puissance de calcul, ainsi que les capacités de traitement des données des algorithmes, induisent des défis complexes.

La détection des anomalies dans les blockchains peut s'améliorer à l'avenir si certains éléments progressent, comme la disponibilité de données pertinentes et riches. L'ingénierie des caractéristiques et l'approche du problème sous un autre angle, comme l'analyse des séries temporelles ou des réseaux, peuvent débloquer de nouvelles solutions potentielles. En outre, l'informatique distribuée peut être utilisé pour améliorer les capacités de calcul des algorithmes.

Bibliographie

[1] Varun Chandola, Arindam Banerjee et Vipin Kumar. « Anomaly detection : A survey ». In : *ACM computing surveys (CSUR)* 41.3 (2009), p. 1-58.

[2] Markus Goldstein et Andreas Dengel. « Histogram-based outlier score (hbos) : A fast unsupervised anomaly detection algorithm ». In : *KI-2012: Poster and Demo Track* (2012), p. 59-63.

[3] Zengyou He, Xiaofei Xu et Shengchun Deng. « Discovering cluster-based local outliers ». In : *Pattern Recognition Letters* 24.9-10 (2003), p. 1641-1650.

[4] Butian Huang et al. « Behavior pattern clustering in blockchain networks ». In : *Multimedia Tools and Applications* 76.19 (2017), p. 20099-20110.

[5] Fei Tony Liu, Kai Ming Ting et Zhi-Hua Zhou. « Isolation forest ». In : *2008 eighth ieee international conference on data mining*. IEEE. 2008, p. 413-422.

[6] James MacQueen et al. « Some methods for classification and analysis of multivariate observations ». In : *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. T. 1. 14. Oakland, CA, USA. 1967, p. 281-297.

[7] P Monamo, V Marivate et B Twala. *Unsupervised learning for robust Bitcoin fraud detection. In 2016 Information Security for South Africa (ISSA). 129–134*. 2016.

[8] Satoshi Nakamoto. *Bitcoin : A peer-to-peer electronic cash system*. Rapp. tech. Manubot, 2019.

[9] Thai Pham et Steven Lee. « Anomaly detection in bitcoin network using unsupervised learning methods ». In : *arXiv preprint arXiv :1611.03941* (2016).

[10] Mei-Ling Shyu et al. *A novel anomaly detection scheme based on principal component classifier*. Rapp. tech. MIAMI UNIV CORAL GABLES FL DEPT OF ELECTRICAL et COMPUTER ENGINEERING, 2003.

[11] Michail Vlachos, George Kollios et Dimitrios Gunopulos. « Discovering similar multidimensional trajectories ». In : *Proceedings 18th international conference on data engineering*. IEEE. 2002, p. 673-684.

[12] Zengsheng ZHONG et al. « SilkViser : A Visual Explorer of Blockchain-based Cryptocurrency Transaction Data ». In : *2020 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE. 2020, p. 95-106.

Table des figures

Liste des tableaux

Table des matières