République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université de Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj
Faculté des Mathématiques et d'Informatique
Département d'informatique

MEMOIRE

Présenté en vue de l'obtention du diplôme

Master en informatique

Spécialité : Ingénierie de l'informatique décisionnelle

THEME

Étude comparative des approches de détection de communautés

Présenté par :

- Bendifallah Aymen.
- Zouaoui Youcef Bahaa Edin .

Devant le jury composé de:

Président : DR. BENABID SONIA

Examinateur : DR. ZOUAOUI HAKIMA

Encadrante : Dr.Saifi Aabdelhamid.

2024/2025

Remerciements

Je tiens à exprimer ma plus profonde gratitude à mes parents, pour leur amour inconditionnel, leur patience, leur soutien indéfectible et leurs nombreux sacrifices. Leur confiance et leur présence constante ont été pour moi une source inestimable de force et de motivation.

Je tiens à remercier chaleureusement les membres du jury, Dr. Benabid Sonia et Dr. Zouaoui Hakima, pour avoir accepté d'évaluer notre travail.

Je remercie également mes frères et l'ensemble de ma famille pour leur soutien moral, leurs encouragements et leur présence bienveillante tout au long de mon parcours.

Je souhaite aussi adresser mes sincères remerciements à mes amis, pour leur amitié, leur écoute, leur aide, et tous les moments de partage qui ont rendu cette période plus agréable et moins stressante.

Enfin, je tiens à remercier chaleureusement mon encadrant, Monsieur Saifi Abdelhamid, pour sa disponibilité, ses conseils pertinents, son encadrement rigoureux et son accompagnement tout au long de ce travail. Son soutien a grandement contribué à la réalisation de ce mémoire.

À toutes et à tous, merci du fond du cœur.

AYMEN/YOUCEF

Abstract

The analysis of complex networks has vast and varied applications, ranging from social networks to biological and technological systems. Community detection is a key task in this analysis, as it allows the division of a network into subgroups of nodes that are densely connected to each other but weakly connected to the rest of the network. This study focuses on unsupervised algorithms for community detection, which are capable of identifying these structures without requiring labeled data. These methods rely on various approaches such as the optimization of modularity, spectral methods, and dynamic processes. The thesis provides a detailed analysis of unsupervised techniques, highlighting their advantages and challenges, particularly regarding scalability and robustness in large networks. The study also presents a comparative evaluation of these algorithms across several types of complex networks to assess their effectiveness and reliability in detecting communities. Finally, potential improvements to existing algorithms are discussed, with a focus on their application in modern, large-scale networks

Keywords : anomaly detection, network security, supervised learning, unsupervised learning, machine learning, Louvain algorithm, Walktrap algorithm, Infomap algorithm, Leiden algorithm, Clauset-Newman-Moore algorithm, Label Propagation, clustering, Python, datasets

Résumé

L’analyse de la structure des réseaux complexes a des applications vastes et variées, allant des réseaux sociaux aux systèmes biologiques et technologiques. La détection de communautés est une tâche clé dans cette analyse, permettant de diviser un réseau en sous-groupes de nœuds qui sont densément connectés entre eux, mais faiblement reliés avec le reste du réseau. Cette étude se concentre sur les algorithmes non supervisés de détection de communautés, qui sont capables de détecter ces structures sans avoir recours à des données étiquetées. Ces méthodes reposent sur différentes approches telles que l’optimisation de la modularité, les méthodes spectrales, et les processus dynamiques.Le mémoire présente une analyse détaillée des techniques non supervisées, met en lumière leurs avantages et leurs défis, notamment la scalabilité et la robustesse dans des réseaux de grande taille. L’étude propose également une évaluation comparative de ces algorithmes sur plusieurs types de réseaux complexes, afin de déterminer leur efficacité et leur capacité à détecter des communautés de manière fiable. Enfin, des pistes d’amélioration des algorithmes existants sont discutées, avec un accent particulier sur leur application dans des réseaux modernes à grande échelle.

Mots clés : Mots-clés : détection d’anomalies, sécurité réseau, apprentissage supervisé, apprentissage non supervisé, apprentissage automatique, algorithme Louvain, algorithme Walktrap, algorithme Infomap, algorithme Leiden, algorithme Clauset-Newman-Moore, Label Propagation, clustering, Python,

ملخص

تحليل الشبكات المعقدة له تطبيقات واسعة ومتنوعة، تتراوح من الشبكات الاجتماعية إلى الأنظمة البيولوجية والتكنولوجية.تُعد **اكتشاف المجتمعات** مهمة أساسية في هذا التحليل، حيث تسمح بتقسيم الشبكة إلى مجموعات فرعية من العقد التي تكون مرتبطة بشكل كثيف مع بعضها البعض ولكنها ضعيفة الاتصال مع بقية الشبكة. تركز هذه الدراسة على **الخوارزميات غير المشروطة** لاكتشاف المجتمعات، التي يمكنها تحديد هذه الهياكل دون الحاجة إلى بيانات موسومة. تعتمد هذه الطرق على مجموعة من الأساليب مثل تحسين **التجانس**، و**الطرق الطيفية**، و**العمليات الديناميكية**.يقدم هذا البحث تحليلاً مفصلاً للأساليب غير المشروطة، مع تسليط الضوء على مزاياها وتحدياتها، لا سيما فيما يتعلق بـ **قابلية التوسع** و **القدرة على التحمل** في الشبكات الكبيرة. كما تقدم الدراسة تقييماً مقارناً لهذه الخوارزميات عبر عدة أنواع من الشبكات المعقدة لتقييم فعاليتها وموثوقيتها في اكتشاف المجتمعات. أخيرًا، يتم مناقشة التحسينات المحتملة على الخوارزميات الحالية، مع التركيز على تطبيقاتها في الشبكات الحديثة واسعة النطاق.

الكلمات المفتاحية : أمن الشبكات، التعلم الخاضع للإشراف، التعلم غير الخاضع للإشراف، التعلم الآلي، خوارزمية لوفين، خوارزمية واكتراب، خوارزمية إنفوماب، خوارزمية لايدن، خوارزمية كلاوسيت-نيومان-مور، نشر التسميات، التجميع، بايثون، مجموعات البيانات

Table des matières

Liste des tableaux

Table des figures

Introduction générale

L'analyse des réseaux complexes est devenue un domaine central dans de nombreuses disciplines contemporaines, telles que les réseaux sociaux, les systèmes biologiques, les réseaux de communication, ou encore l'Internet des objets. Ces réseaux, constitués de nœuds interconnectés par des liens, permettent de modéliser des relations variées entre entités.
Comprendre leur structure sous-jacente est essentiel pour appréhender leur fonctionnement, leur dynamique et les comportements collectifs qui en émergent. Parmi les tâches fondamentales de cette analyse, la détection de communautés joue un rôle clé, en identifiant des groupes de nœuds fortement connectés entre eux mais faiblement liés au reste du réseau. Cette détection facilite la simplification et la compréhension des réseaux complexes, avec des applications variées allant de la recommandation de contenu à la découverte de médicaments.

Contexte

La détection de communautés dans les réseaux complexes est une approche essentielle pour comprendre la structure et le fonctionnement de divers systèmes interconnectés. Elle permet d'identifier des groupes de nœuds (ou entités) qui sont plus densément connectés entre eux qu'avec le reste du réseau, révélant ainsi des structures sous-jacentes cruciales pour l'analyse et l'optimisation de ces systèmes.

Problématique

Malgré l'intérêt croissant et l'importance capitale de la détection de communautés, cette tâche demeure particulièrement complexe. La variété des types de réseaux, leur grande échelle avec parfois des millions de nœuds, ainsi que la présence de données bruitées ou incomplètes rendent l'analyse difficile. Les méthodes classiques reposent souvent sur des approches supervisées nécessitant des données préalablement étiquetées ou une connaissance a priori des structures à détecter. Or, dans la plupart des cas réels, ces informations ne sont ni disponibles ni fiables, limitant ainsi l'applicabilité de ces techniques. De ce fait, les méthodes non supervisées ont émergé comme une solution prometteuse, s'appuyant uniquement sur la structure topologique des réseaux sans besoin de supervision extérieure. Toutefois, ces approches rencontrent plusieurs obstacles majeurs : leur capacité à gérer efficacement la scalabilité dans les grands réseaux, ainsi que la difficulté à évaluer objectivement la qualité des communautés détectées en l'absence de vérité terrain claire. Ces défis posent des questions cruciales quant à la robustesse, à la pertinence et à la fiabilité des résultats produits.

CONTRIBUTION

Ce travail s'appuie sur une démarche rigoureuse visant à évaluer des méthodes non supervisées de détection de communautés dans les réseaux complexes. La première étape consiste en une revue exhaustive de la littérature afin d'identifier les algorithmes les plus pertinents, tels que Louvain, Leiden, Infomap, Walktrap, Clauset-Newman-Moore et Label Propagation. Ces méthodes, couvrant des approches fondées sur l'optimisation de la modularité et NMI avec ari, les techniques spectrales et les dynamiques de propagation, sont ensuite implémentées ou adaptées pour les besoins de l'étude.

Les algorithmes sont testés sur divers jeux de données, incluant des réseaux réels et synthétiques, afin d'évaluer leur performance selon plusieurs critères : qualité des communautés détectées (mesurée notamment par la modularité), temps de calcul, consommation mémoire et robustesse face aux données bruitées. Cette évaluation comparative permet d'identifier les forces et faiblesses de chaque méthode, en fonction de la taille et de la topologie des réseaux étudiés.

Enfin, une analyse approfondie des résultats obtenus conduit à formuler des recommandations pratiques pour le choix des algorithmes en fonction des contraintes spécifiques des réseaux analysés. Des pistes d'optimisation sont également proposées pour améliorer la performance globale des méthodes étudiées, en tenant compte des enjeux de scalabilité et de robustesse.

Organisation du mémoire

Ce mémoire est structuré en trois chapitres :

- **Chapitre 1 :**Modélisation des réseaux complexes .
  Ce chapitre introduit les fondements théoriques des réseaux complexes, les types de graphes, ainsi que les principales métriques et outils utilisés pour analyser leur structure.
- **Chapitre 2 :**La Détection de communautés .
  Ce chapitre se concentre sur l'application d'algorithmes non supervisés, pour la détection de communautés. Il examine les avantages et limites de ces approches sur des cas concrets.
- **Chapitre 3 :** Conception et Implémentation de l'Approche Proposée .
  Le dernier chapitre présente l'implémentation des algorithmes sélectionnés, les jeux de données utilisés, les résultats obtenus ainsi qu'une évaluation comparative des performances. Des perspectives d'amélioration y seront également discutées.

Chapitre 1

Les Réseaux Complexes

1.1 Introduction

La notion de réseau varie selon les disciplines : pour un chasseur du XIX siècle, il s'agit d'un filet pour capturer du gibier, tandis qu'en physique, un réseau désigne un dispositif optique basé sur l'interférence d'ondes diffractées. En informatique, un réseau représente un ensemble de ressources matérielles et logicielles assurant la communication entre divers terminaux. Plus largement, en français, le terme désigne un ensemble de lignes entrecroisées, une idée issue du latin rete (filet), à l'origine de l'adjectif « réticulé ». Selon une définition générale tirée du dictionnaire, un réseau peut être vu comme une répartition de points liés par différentes interactions — comme dans un réseau urbain interconnectant des villes par des relations économiques ou politiques. Cette diversité d'interactions nous conduit à nous interroger sur les réseaux complexes : il s'agit d'une catégorie de systèmes complexes, définis comme des ensembles d'éléments en interaction, dont le comportement global ne peut se réduire à la somme de leurs parties [1]. Ainsi, un réseau complexe est un réseau d'entités interconnectées, dont les interactions engendrent des propriétés émergentes, impossibles à prévoir à partir des seules caractéristiques individuelles [2]. C'est sur cette base conceptuelle que s'appuie le présent travail.

1.2 Definition :

Les réseaux complexes sont présents dans de nombreux domaines aussi divers les uns que les autres : biologie, sociologie, psychologie, informatique, ... Ils recouvrent ainsi des réseaux aussi variés que le réseau Internet, les réseaux d'humains, ou encore les réseaux de protéines. De nombreuses études ont été réalisées à leur sujet. Ces réseaux peuvent être regroupés en quatre catégories [2] :

- **Les réseaux sociaux :**Un réseau social est un ensemble de personnes ou de groupes de personnes possédant des schémas de contacts ou d'interactions entre eux [3] (Fig. 1.1). C'est à partir de ce type de réseaux que la modélisation du monde réel a été introduite de façon empirique grâce à l'expérience de Milgram [56] qui sera décrite plus tard. La plupart des études concernant ces réseaux souffrent de problèmes d'imprécision, de subjectivité, et d'échantillons de petite taille [4]
- **Les réseaux d'informations :**Un réseau d'informations peut être rapporté à l'exemple classique d'un réseau de citations entre papiers scientifiques. La structure des informations étant stockée dans les nœuds, c'est pour cela que l'on utilise le terme réseau d'informations. Le World Wide Web avec ses pages web (contenant des informations) et ses hyperliens

est également un réseau d'informations (à ne pas confondre avec le réseau Internet qui est le réseau physique reliant les ordinateurs du monde entier entre eux).

- **Les réseaux technologiques :**Un réseau technologique est un réseau créé par l'homme principalement pour la distribution d'un service ou d'énergie. Les réseaux électriques, aériens, d'ordinateurs, en font partie.
- **Les réseaux biologiques**Un réseau biologique est un réseau d'éléments touchant au vivant. Un exemple de réseau biologique peut être un réseau d'interactions entre protéines.

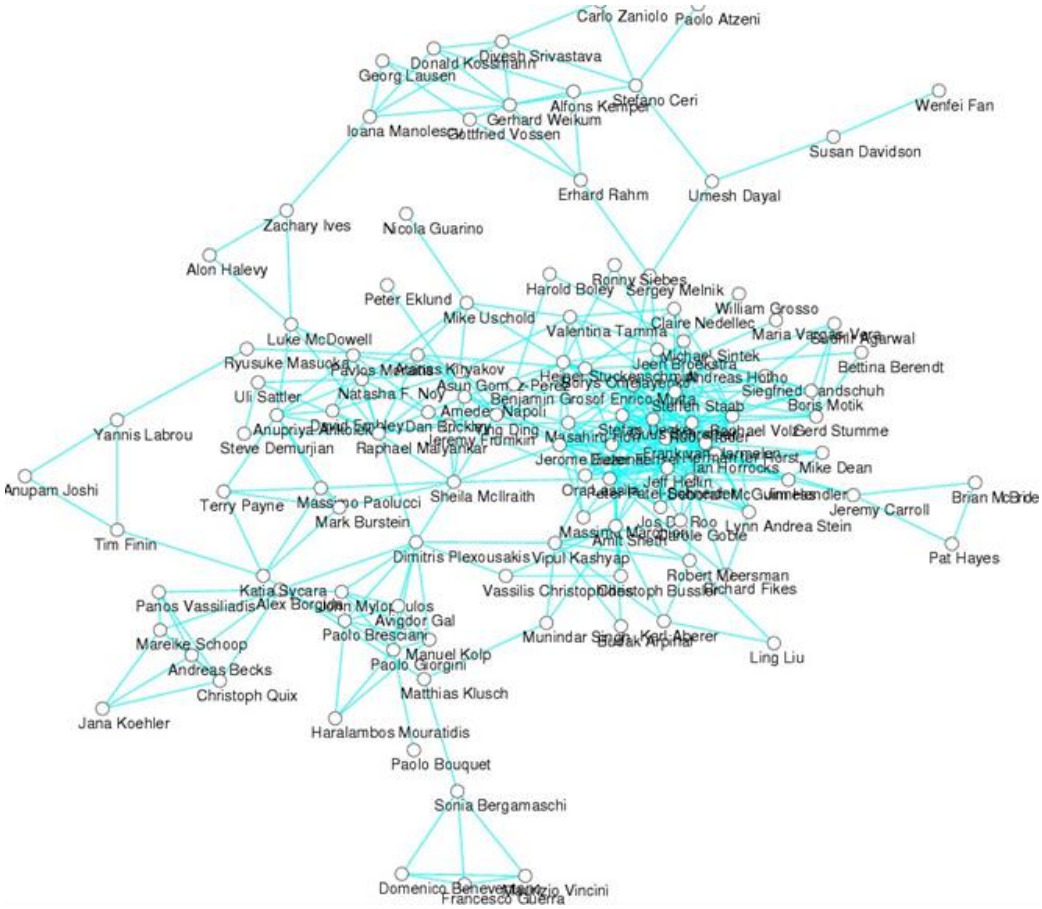

FIGURE 1.1 – Réseau de collaboration scientifique entre chercheurs[43]

Cette figure représente un réseau de collaboration scientifique. Chaque point (ou nœud) correspond à un chercheur, identifié par son nom, et chaque ligne (ou arête) relie deux chercheurs ayant coécrit au moins une publication ensemble. Le schéma forme un graphe complexe où l'on observe un grand groupe central très dense, indiquant que de nombreux chercheurs collaborent fréquemment entre eux, ainsi que quelques groupes plus petits ou des individus en périphérie, moins connectés au réseau principal. Ce type de visualisation permet de mettre en évidence les liens de collaboration, d'identifier les chercheurs les plus connectés (souvent au centre du graphe), et de mieux comprendre la structure des échanges scientifiques au sein d'une communauté ou d'un domaine de recherche.

1.3 Problématique modélisation d'un Réseaux complexe

Dans les réseaux réels, bien que les interactions locales — telles que la communication entre routeurs ou les réactions entre protéines — soient généralement bien comprises, les effets globaux résultant de l'ensemble de ces interactions demeurent encore globaux est cruciale, car

elle concerne des enjeux majeurs comme la propagation des virus (qu'ils soient biologiques ou informatiques), la stabilité des réseaux électriques, ou encore la résilience des grandes infrastructures. L'essor récent des capacités de traitement et de collecte massive de données a favorisé le développement de recherches sur ces réseaux. Il a notamment été observé que, malgré leurs différences apparentes, de nombreux réseaux partagent des propriétés macroscopiques similaires [7].mal appréhendés, notamment en raison de leur nature émergente [5]. Pourtant, la compréhension de ces phénomènes .

1.4 La théorie des graphes

Pour représenter les réseaux, la théorie des graphes paraît l'outil adéquat. C'est principalement cet outil qui a été utilisé dans les différentes études qui ont porté sur la modélisation de réseaux complexes. Dans cette partie, la théorie des graphes sera présentée, puis on étudiera les caractéristiques des propriétés structurelles d'un graphe pour arriver au problème de la dynamique, ainsi que les limites des études auxquelles on se confronte à l'heure actuelle. **Définition et concepts de base :**
La théorie des graphes trouve son origine en 1736, lorsque Euler démontra l'impossibilité de traverser une seule fois chacun des sept ponts de Königsberg tout en revenant à son point de départ. Cette problématique a marqué le début d'un champ mathématique aujourd'hui largement utilisé dans des disciplines variées, telles que la chimie (pour modéliser des structures moléculaires), la biologie (analyses génomiques), les sciences sociales (études de relations), ou encore l'industrie (notamment avec le problème du voyageur de commerce). Les graphes offrent un outil efficace pour représenter et résoudre des problèmes complexes, en modélisant les relations et interdépendances au sein de systèmes composés d'entités multiples — qu'il s'agisse de réseaux de communication, de transports, ou de diagrammes de projet. Outre leur utilité théorique, les graphes représentent également une structure de données fondamentale en informatique [8].

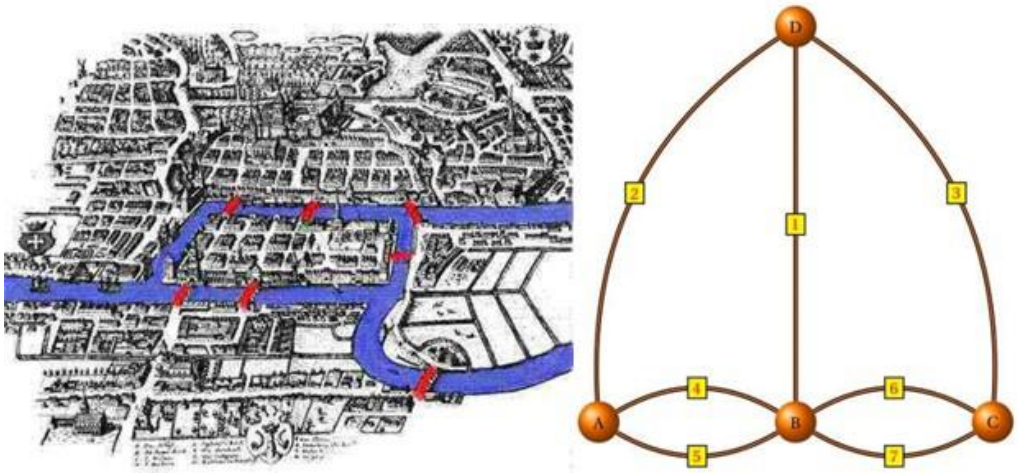

FIGURE 1.2 – Ponts de K¨onigsberg [44]

Cette figure illustre le célèbre problème des ponts de Königsberg. À gauche, on voit une carte ancienne de la ville de Königsberg (aujourd'hui Kaliningrad), traversée par une rivière et reliée par sept ponts (en rouge). Le défi historique consistait à trouver un chemin permettant de traverser chaque pont une seule fois sans jamais en retraverser un. À droite, la situation est modélisée sous forme de graphe : chaque zone de terre séparée par la rivière est représentée par un sommet (les cercles orange), et chaque pont par une arête reliant deux sommets. Cette

représentation graphique a permis à Euler de démontrer qu'il était impossible de réaliser ce parcours, posant ainsi les bases de la théorie des graphes et des chemins eulériens.

1.5 Concepts orientés

Dans beaucoup d'applications, les relations entre éléments d'un ensemble sont orientées, c'est-à-dire qu'un élément x peut être en relation avec un autre y sans que y soit nécessairement en relation avec x. On parle alors de graphe orienté (en anglais *directed graph* ou plus simplement digraph).

Définition 1 (Définition 1.1) *Un graphe $G = (X, U)$ est déterminé par :*

- *un ensemble $X = \{x_1, x_2, \ldots, x_n\}$ dont les éléments sont appelés sommets ou nœuds (ce dernier terme est plutôt utilisé dans le contexte des réseaux) ;*
- *un ensemble $U = \{u_1, u_2, \ldots, u_m\} \subseteq X \times X$ dont les éléments sont appelés arcs.*

Pour un arc $u = (x_i, x_j)$, x_i est l'extrémité initiale et x_j l'extrémité finale (ou bien origine et destination). L'arc u part de x_i et arrive à x_j. Un arc (x_i, x_i) est appelé une boucle.

Un p-graphe est un graphe dans lequel il n'existe jamais plus de p arcs de la forme (i, j) entre deux sommets quelconques. On appellera communément graphe un 1-graphe.

La densité d'un graphe est donnée par le quotient $\frac{m}{n^2}$, rapport du nombre effectif d'arcs sur le nombre maximal théorique.

1.6 Concepts non orientés

Lors de l'étude de certaines propriétés, il arrive que l'orientation des arcs ne joue aucun rôle. On s'intéresse simplement à l'existence d'arcs entre deux sommets (sans en préciser l'ordre). Un arc sans orientation est appelé une *arête*. L'ensemble U est alors constitué non pas de couples ordonnés, mais de *paires de sommets non ordonnées*.

Pour une arête (x_i, x_j), on dit qu'elle est *incidente* aux sommets x_i et x_j.

Dans le cas non orienté, au lieu de noter $G = (X, U)$ et $u = (x_i, x_j)$, on préfère souvent écrire $G = (X, E)$ et $e = [x_i, x_j]$.

Un *multigraphe* $G = (X, E)$ est un graphe pour lequel il peut exister plusieurs arêtes entre deux mêmes sommets.

Un graphe $G = (X, E)$ est dit *simple* :

1. s'il n'est pas un multigraphe ;
2. s'il n'existe pas de boucles.

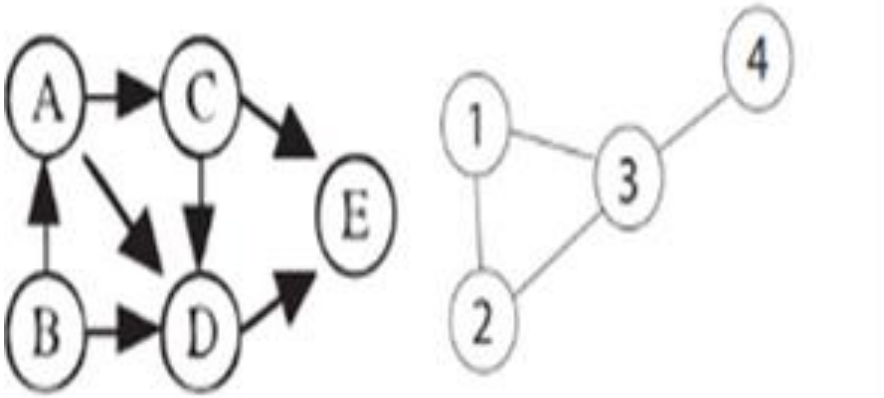

FIGURE 1.3 – Graphe orienté et graphe non orienté [45]

Cette figure présente deux types de graphes : à gauche, un graphe orienté où les sommets (A, B, C, D, E) sont reliés par des flèches indiquant des relations directionnelles, c'est-à-dire que chaque arête possède un sens précis (par exemple, de A vers C ou de B vers D), ce qui permet de représenter des situations où la direction des liens est importante. À droite, on trouve un graphe non orienté composé de sommets numérotés (1, 2, 3, 4) reliés par des arêtes simples, sans flèche ; ici, les connexions sont bidirectionnelles, ce qui signifie que le lien entre deux sommets n'a pas de sens particulier et qu'on peut circuler librement dans les deux directions. Cette illustration met donc en évidence la différence fondamentale entre un graphe orienté, où la direction des liens joue un rôle, et un graphe non orienté, où les liens sont symétriques.

1.6.1 Principales définitions

La nature des données constitue un aspect crucial dans la détection d'anomalies. Les données peuvent être structurées, semi-structurées ou non structurées, et leur qualité ainsi que leur quantité peuvent influencer la précision des modèles de détection d'anomalies. Des caractéristiques des données telles que la dimensionnalité et la distribution jouent également un rôle important [4].

Adjacence

- Deux sommets sont **adjacents** (ou *voisins*) s'ils sont joints par un arc.
- Deux arcs sont **adjacents** s'ils ont au moins une extrémité commune.

Degrés

- Le *demi-degré extérieur* de x_i, noté $d^+(x_i)$, est le nombre d'arcs ayant x_i comme extrémité initiale : $d^+(x_i) = \omega^+(x_i)$.

- Le *demi-degré intérieur* de x_i, noté $d^-(x_i)$, est le nombre d'arcs ayant x_i comme extrémité finale : $d^-(x_i) = \omega^-(x_i)$.
- Le *degré* de x_i est : $d(x_i) = d^+(x_i) + d^-(x_i)$. Dans un graphe non orienté, le degré d'un sommet est le nombre d'arêtes qui lui sont incidentes.

Graphe complémentaire Soit $G = (X, U)$, on définit le *graphe complémentaire* $\overline{G} = (X, \overline{U})$ tel que $(x_i, x_j) \in \overline{U} \iff (x_i, x_j) \notin U$.

Graphe partiel Soit $G = (X, U)$ et $U_p \subseteq U$, le graphe $G_p = (X, U_p)$ est un *graphe partiel* de G. Cela peut donner lieu à des *sommets isolés.*

Sous-graphe Soit $G = (X, U)$ et $X_s \subset X$. Le *sous-graphe* $G_s = (X_s, V)$ est défini par $V = \{(x, y) \in U \mid x, y \in X_s\}$. Pour tout $x_i \in X_s$, $\Gamma_s(x_i) = \Gamma(x_i) \cap X_s$.

Sous-graphe partiel Un *sous-graphe partiel* combine les deux définitions précédentes.

Exemple (Réseau routier)

- Graphe partiel : que les autoroutes.
- Sous-graphe : que la région Midi-Pyrénées.
- Sous-graphe partiel : que les autoroutes de Midi-Pyrénées.

Types de graphes

- **Graphe réflexif** : $\forall x_i \in X, (x_i, x_i) \in U$.
- **Graphe irréflexif** : $\forall x_i \in X, (x_i, x_i) \notin U$.
- **Graphe symétrique** : $\forall x_i, x_j \in X, (x_i, x_j) \in U \Rightarrow (x_j, x_i) \in U$.
- **Graphe asymétrique** : $\forall x_i, x_j \in X, (x_i, x_j) \in U \Rightarrow (x_j, x_i) \notin U$. $\Rightarrow$ G est aussi irréflexif.
- **Graphe antisymétrique** : $\forall x_i, x_j \in X$, si $(x_i, x_j) \in U$ et $(x_j, x_i) \in U$, alors $x_i = x_j$.
- **Graphe transitif** : si $(x_i, x_j) \in U$ et $(x_j, x_k) \in U$, alors $(x_i, x_k) \in U$.
- **Graphe complet** : $\forall x_i = x_j \in X, (x_i, x_j) \in U$.

Clique Une **clique** est un ensemble de sommets formant un *sous-graphe complet*. Soit $C \subset X$ une clique de G non orienté : pour tout $x_i, x_j \in C$, $(x_i, x_j) \in U$.

Tournoi Un *graphe complet et antisymétrique* est appelé un **tournoi**, car il représente le résultat d'un tournoi où chaque joueur affronte tous les autres une fois.

1.6.2 Autres graphes

Il existe d'autres classes de graphes particuliers, comme les graphes sans circuit, les graphes bipartis ou les graphes planaires. Nous n'entrerons pas dans leur description ici, les notions présentées ci-dessus étant suffisantes à la compréhension de ce chapitre.

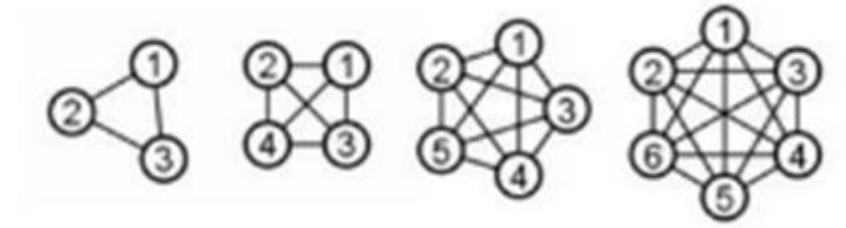

FIGURE 1.4 – Graphes complets avec 3, 4, 5 et 6 sommets [46]

présente des graphes complets avec 3, 4, 5 et 6 sommets. Un graphe complet, est un type de graphe dans lequel chaque sommet est relié à tous les autres sommets par une arête. Ainsi, dans chaque exemple de l'image, chaque point (ou sommet) est connecté à tous les autres points du même graphe. Plus le nombre de sommets augmente, plus le nombre d'arêtes croît rapidement, car chaque nouveau sommet se relie à tous les précédents. Ces graphes illustrent la structure maximale de connexions possibles pour un ensemble donné de sommets.

1.7 Modélisation des réseaux complexes par la théorie des graphes

Dans les différentes études concernant les réseaux complexes et la théorie des graphes, d'autres propriétés du graphe sont étudiées, car les définitions précédentes, même si elles donnent une idée du type de graphe auquel nous pouvons avoir à faire, ne sont pas suffisantes pour caractériser un réseau.

1.8 Modélisation existantes : Graphes Aléatoires Invariance d'Échelle Petits Mondes

1.8.1 Lexpérience de Milgram :

Le psychologue social Stanley Milgram a approfondi le concept des six degrés de séparation, initialement formulé en 1929 par l'écrivain hongrois Frigyes Karinthy dans sa nouvelle Chaînes. Cette théorie propose que toute personne dans le monde peut être reliée à une autre par une chaîne de connaissances ne dépassant pas six intermédiaires [11]. Dans sa première expérience, Milgram a demandé à des volontaires d'Omaha (Nebraska) d'envoyer une lettre à un agent de change à Boston (Massachusetts) en la transmettant uniquement à des connaissances jugées proches de la cible. Sur cinquante participants, seulement trois lettres ont atteint leur destination. Bien qu'une ait réussi en quatre jours, seulement 6 % des lettres ont abouti [11]. D'autres expériences menées par Milgram ont connu un taux de réussite encore plus faible, au point qu'elles n'ont pas été publiées. Des recherches ultérieures ont montré que des facteurs socioculturels — comme la couleur ou l'origine des personnes ciblées — influençaient fortement les résultats. Par exemple, les chaînes menant à une personne noire aboutissaient dans 13 % des cas, contre 33 % pour une personne blanche, même si la couleur n'était pas connue des participants [11]. Malgré les limites, ces expériences ont permis des avancées. Après plusieurs ajustements, Milgram a augmenté le taux de réussite à 35 %, et des chercheurs ultérieurs ont atteint jusqu'à 97 %. Il a aussi observé un effet d'entonnoir, révélant que quelques individus hautement connectés jouaient un rôle central dans les chaînes de transmission. Même dans l'expérience initiale, deux chaînes sur trois passaient par les mêmes personnes. C'est ainsi qu'a émergé la fameuse moyenne des « six degrés de séparation » [11].

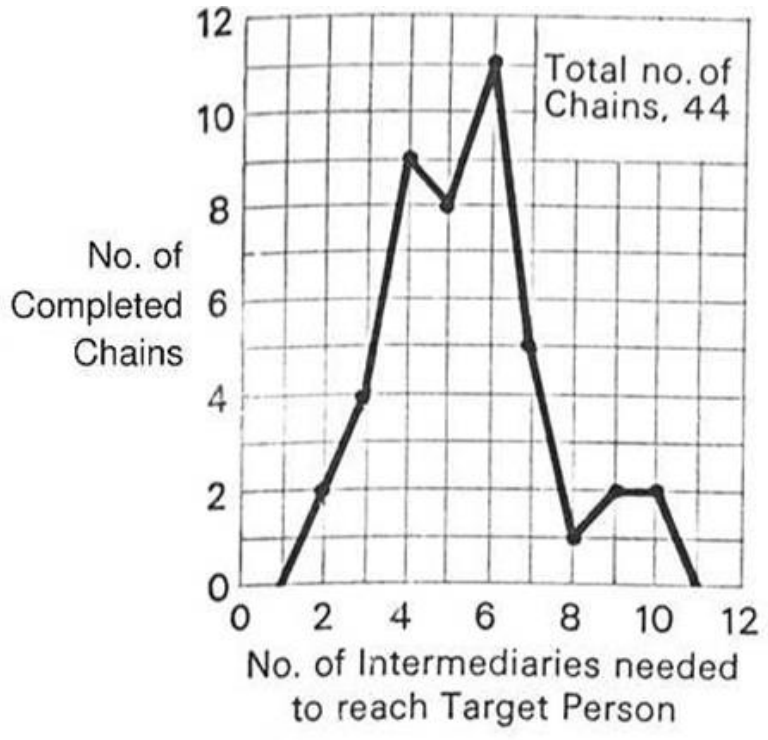

FIGURE 1.5 – Les six degrés de séparation de Milgram [47]

Ce graphique montre le nombre de chaînes complètes en fonction du nombre d'intermédiaires nécessaires pour atteindre une personne cible dans une expérience de réseau social. On constate

que la plupart des chaînes réussies passent par 4 à 7 intermédiaires, avec un maximum à 6, ce qui signifie qu'en moyenne, il faut environ six personnes pour relier deux inconnus. Au total, 44 chaînes ont permis d'atteindre la cible, illustrant ainsi le principe des « six degrés de séparation » : dans un réseau social, la distance entre deux personnes est généralement très courte.

1.8.2 Modèle aléatoire d'Erdős-Rényi

La définition d'un des tout premiers modèles théoriques de réseau a été établie en 1959 par Paul Erdős et Alfréd Rényi. Ce modèle minimaliste consiste en un ensemble de n nœuds reliés par des arêtes qui sont placées de manière aléatoire uniforme entre des paires de nœuds.

Erdős et Rényi ont proposé plusieurs versions de ce modèle. La plus couramment étudiée est celle dénommée $G_{n,p}$, où chaque arête entre deux nœuds est présente avec une probabilité p, et absente avec une probabilité $1 - p$, indépendamment des autres arêtes.

Souvent, les propriétés d'un graphe $G_{n,p}$ ne sont pas exprimées en fonction de p, mais plutôt en fonction du **degré moyen** z des nœuds. Le nombre moyen d'arêtes dans un graphe $G_{n,p}$ est donné par :

$$\mathrm{E}[m] = \frac{n(n-1)}{2} \cdot p$$

Chaque arête ayant deux extrémités, le nombre moyen de connexions (ou demi-arcs) est donc :

$$n \cdot z = n(n-1)p$$

Ainsi, on peut déduire que le degré moyen z est :

$$z = \frac{n(n-1)p}{n} = (n-1)p \simeq np \quad \text{(pour } n \text{ grand)}$$

L'approximation np est d'autant plus précise que n est grand. Ainsi, une fois que l'on connaît n, chaque propriété exprimable en fonction de p peut également l'être en fonction de z. Le modèle d'Erdős–Rényi possède plusieurs propriétés intéressantes. Par exemple, une caractéristique remarquable démontrée dans les articles originaux de l'époque est qu'une phase de transition apparaît en fonction de z, ce qui provoque la formation d'une composante géante.

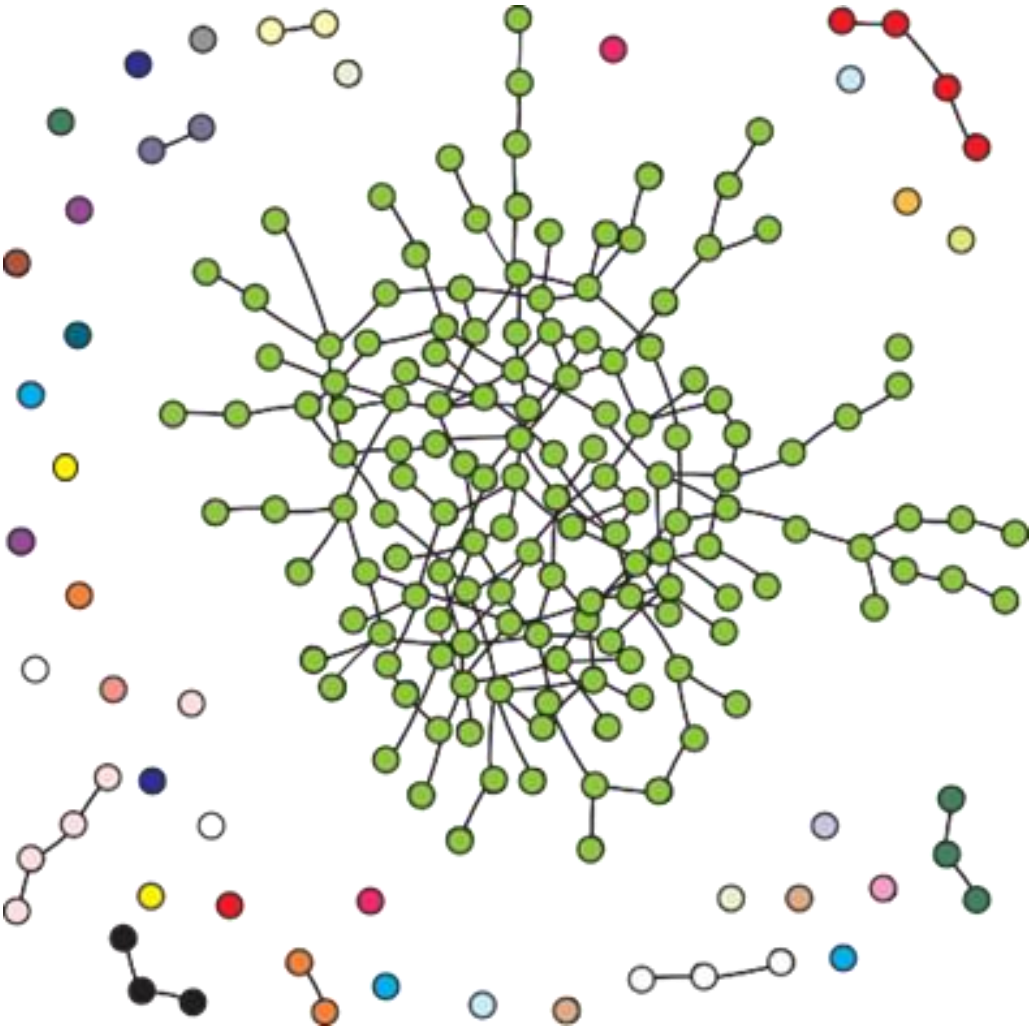

FIGURE 1.6 – Graphe aléatoire avec sa composante géante au centre [48]

Une composante géante (nœuds verts) :
Un groupe très dense de nœuds interconnectés entre eux, formant le cœur du graphe. Tous les nœuds de cette composante peuvent atteindre les autres, directement ou indirectement. C'est la plus grande composante du graphe.

Composantes rouges (foncé et clair) :
Petits groupes de nœuds interconnectés, mais complètement séparés de la composante géante. Ils forment des structures locales indépendantes.

Composantes bleu clair et bleu foncé :
Petits sous-graphes dans lesquels les nœuds sont connectés entre eux, mais pas au reste du graphe. Ils représentent de petits îlots d'interactions.

Composantes violettes :
Groupes de nœuds isolés du cœur du graphe, connectés uniquement entre eux, probablement des entités spécifiques non intégrées au système global.

Composantes orange :
Sous-ensembles plus petits encore, avec parfois seulement deux ou trois nœuds, représentant des connexions faibles ou isolées.

Composantes noires :
Mini-groupes souvent très restreints, voire des paires de nœuds connectés entre eux uniquement.

Composantes jaunes :

Petits groupes également isolés, ayant des connexions internes mais aucune vers l'extérieur.

Composantes roses :
Nœuds faiblement connectés, uniquement à l'intérieur d'un petit groupe, non reliés à la structure principale.

Composantes blanches / gris très clair :
Composantes de taille très réduite, souvent invisibles à grande échelle, mais qui montrent l'existence de relations locales.

Composantes grises :
Nœuds dispersés dans de petits clusters, ou seuls, qui n'ont aucune relation avec les autres groupes.

représente un graphe aléatoire, où chaque point (ou nœud) est relié à d'autres par des arêtes tracées de façon aléatoire. Au centre, on observe un grand ensemble de nœuds verts interconnectés : il s'agit de la composante géante, c'est-à-dire le plus grand groupe de nœuds qui sont tous reliés, directement ou indirectement, les uns aux autres. Autour de cette composante centrale, on distingue plusieurs petits groupes de nœuds de couleurs différentes, chacun formant une composante connexe distincte, mais beaucoup plus petite. Certains nœuds sont même isolés ou ne forment que de très petits ensembles. Ce schéma illustre ainsi comment, dans un graphe aléatoire, une structure dominante émerge : la composante géante, qui regroupe une grande partie des nœuds, tandis que le reste du graphe est constitué de petits groupes ou d'éléments isolés.

Une composante est un sous-ensemble de noeuds du graphe qui s'avere ^etre un sous- graphe connexe (c'est-a-dire que tous les noeuds sont connect es entre eux par au moins un chemin). Pour des petites valeurs de z, quand il y a peu de noeuds dans le graphe, il n'est pas surprenant de constater que la plupart des noeuds ne sont pas connectes entre eux, et que les composantes sont petites, avec une taille moyenne qui reste constante au fur et a mesure que le graphe augmente en taille. Cependant, il y a une valeur critique de z a partir de laquelle la plus grosse des composantes du graphe contient une fraction finie S du nombre total de noeuds, sa taille nS augmentant lin´eairement avec la taille du graphe entier. Cette composante est la composante g´eante. En g´en´eral, en plus de celle-ci, il y a d'autres composantes, mais elles restent petites, ayant une taille moyenne qui reste constante au fur et a mesure que le graphe s'agrandit. La phase de transition 'a laquelle la composante g´eante se forme se declenche La

formation d'une composante géante dans un graphe aléatoire fait penser au comportement de nombreux réseaux réels. Par exemple, un réseau social où les gens sont connectés est très dense et possède probablement une composante géante. Pourtant, les graphes aléatoires diffèrent des réseaux réels sur de nombreux points. Deux différences notables ont été remarquées par Strogatz [7] et par Albert et Barabási [10]. Premièrement, les réseaux réels montrent de forts coefficients de regroupement (clustering) [11], alors que ce n'est pas le cas pour le modèle d'Erdős et Rényi. En effet, dans ce modèle, les probabilités qu'une paire de nœuds soit connectée par une arête sont indépendantes, il n'y a donc pas de différence au niveau probabilité de connexion entre deux nœuds ayant un voisinage commun et deux nœuds n'en ayant pas. Cela signifie que le coefficient de regroupement pour un graphe aléatoire est tout simplement égal à C=pC = pC=p ou, de manière équivalente, C=znC =znC=nz.

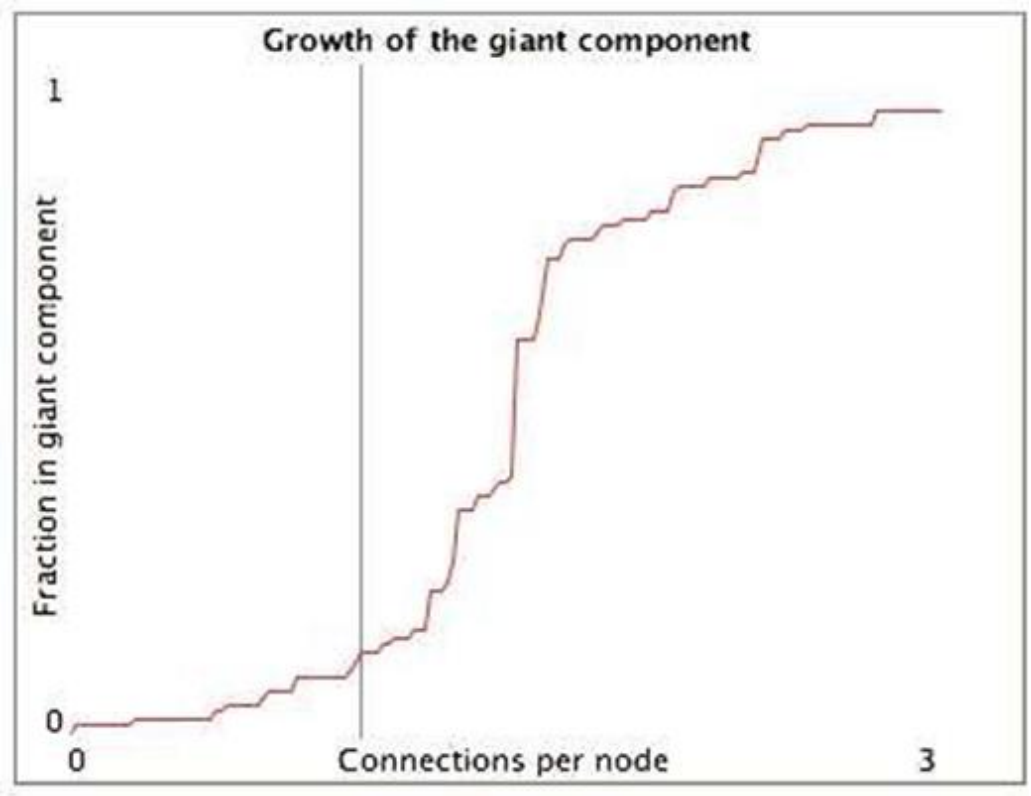

FIGURE 1.7 – se de transition de formation de la composante géante dans un graphe [49]

Le schéma montre qu'en augmentant le nombre moyen de connexions par nœud dans un graphe aléatoire, il existe un seuil critique où une grande composante connectée apparaît soudainement, reliant la majorité des nœuds. Comme on peut le constater, la correspondance du coefficient de regroupement entre réseaux réels et graphes aléatoires n'est pas bonne. Les graphes aléatoires ne reproduisent donc pas cette propriété importante des réseaux réels. La seconde chose qui diffère entre les réseaux réels et les graphes aléatoires est la distribution de leur degré [10]. La probabilité pkp qu'un nœud dans un graphe aléatoire possède un degré égal à k est donnée par la distribution binomiale suivante :

$$p^k = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

Quand $n \gg kz$, on obtient :

$$p_k = \frac{z^k e^{-z}}{k!}$$

1.8.2.1 Les graphes aléatoires généralisés

Il existe une classe de graphes proches des graphes aléatoires d'Erdős et Rényi, à la seule différence que la distribution des degrés des nœuds ne suit pas forcément une loi de Poisson. On peut, en théorie, lui faire suivre n'importe quelle loi de distribution. Abordée en premier lieu par Bender et Canfield en 1978 [47], la généralisation de ces graphes a été formalisée en 1995 par Molloy et Reed [46]. La méthode de construction de base est de se tenir à une séquence de degrés spécifique, c'est-à-dire à un ensemble ki de degrés des nœuds i = 1, ..., n. En fait, cet ensemble sera choisi de manière à ce que la fraction de nœuds ayant comme degré k tende vers la distribution de degrés désirée pk au fur et à mesure que n augmente. Une fois que chaque nœud possède sa séquence de degrés, la méthode pour générer le graphe est la suivante : on donne à chaque nœud i un nombre ki de "points de connexion" (un point de connexion correspond à l'extrémité d'une arête), et on choisit des paires de ces points de connexion de manière aléatoire uniforme pour joindre les arêtes. Quand tous les points de connexion ont été utilisés, le graphe résultant est un membre choisi de façon aléatoire parmi l'ensemble des graphes possédant la séquence du degré désiré. Plus récemment, une classe générale de graphes encore plus large intitulée « modèles de graphes aléatoires non homogènes » a été formalisée par Söderberg en 2002 [12]. Ces graphes sont construits en imposant un type de structure aux nœuds. Les modèles de cette classe ne sont pas restreints au seul choix de la distribution du degré, cependant, une infinité de modèles existe pour chaque type de distribution. Ces modèles peuvent, dans certains cas, correspondre à certains types de réseaux dynamiques, même si leur défaut majeur est le côté artificiel de leur génération

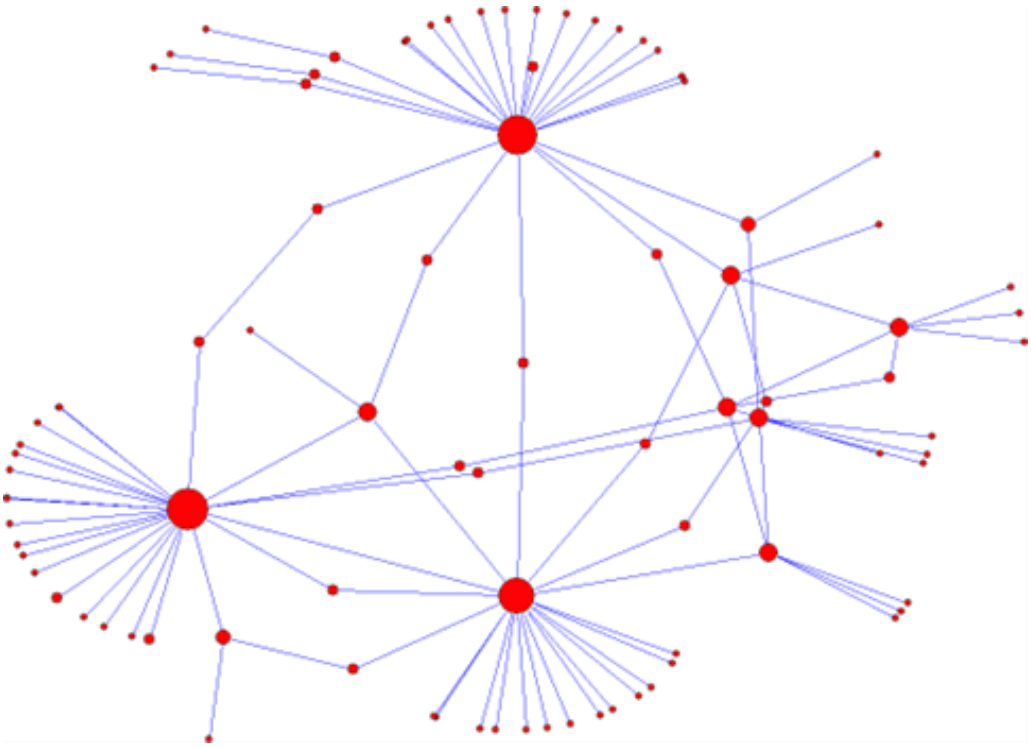

FIGURE 1.8 – Graphe aléatoire généralisé de Molloy Reed suivant une loi de puissance [50]

Nœuds très gros (hubs rouges)

- **Quelques nœuds** ont un très grand nombre de connexions.
- Ce sont des **nœuds dominants** qui jouent le rôle de **centres de liaison** du réseau.
- Dans les réseaux réels, ces hubs pourraient représenter :

- Des utilisateurs influents (réseaux sociaux),
- Des serveurs ou routeurs majeurs (réseaux informatiques),
- Des gènes ou protéines très interconnectés (réseaux biologiques).

Nœuds moyens ou petits (autour des hubs)

- Ils sont **majoritairement connectés à un seul hub** ou à quelques voisins.
- Ce sont des **nœuds périphériques**, souvent dépendants des hubs pour accéder au reste du réseau.
- Ce type de structure reflète une **topologie hiérarchique**.

$$P(k) \sim k^{-\gamma}$$

- $P(k)$: probabilité qu'un nœud ait un degré k
- k : degré du nœud (nombre de liens)

- γ (gamma) : exposant de la loi de puissance, généralement compris entre 2 et 3 pour les réseaux réels

Cette loi implique qu'il y a beaucoup de nœuds avec peu de connexions et quelques nœuds (appelés "hubs") avec un très grand nombre de connexions. C'est une caractéristique typique des réseaux complexes (réseaux sociaux, Internet, etc.), où la structure n'est pas homogène.

1.9 Conclusion

Tout au long de ce chapitre, un bon nombre de modèles de réseaux complexes, basés sur la théorie des graphes, a été présenté, de manière plutôt chronologique. Au fil des recherches, de nombreuses caractéristiques et propriétés ont été étudiées, pour au final, but ultime de la modélisation et de la simulation, correspondre le plus au monde réel. La question de la correspondance réseaux réels - réseaux modélisés reste cependant sans réponse satisfaisante. La plupart des modèles présentés possèdent, il est vrai, une ou plusieurs caractéristiques appartenant aux réseaux réels. Cependant, même si le dernier modèle de Lebhar paraît pertinent, en particulier pour les réseaux informatiques, il est évident qu'une grande part d'incertitude existe quant à une modélisation efficace et une compréhension globale des réseaux complexes. Si les études autour de la structure sont les plus nombreuses, alors peut-être une meilleure compréhension de la fonction permettrait d'éclaircir le problème posé [19], ou tout simplement en allant au-delà d'une modélisation utilisant de simples graphes, comme les hypergraphes ou les matroïdes par exemple, que la théorie présentée dans le prochain chapitre englobera. Car aussitôt que le comportement fonctionnel des réseaux complexes sera compris, la modélisation de ces réseaux n'en deviendra que plus évidente et compréhensible. Dans la suite du document, nous allons montrer qu'il est possible d'obtenir des modélisations plus fines se rapprochant davantage des cas du réel.

Application dans les systèmes biologiques et les systèmes de santé : La détection de communautés dans les réseaux biologiques a une signification importante, tels que les réseaux de protéines, les réseaux alimentaires, les réseaux métaboliques, etc. Dans les réseaux de protéines, elle a été appliquée de manière à détecter les complexes protéiques. Dans les systèmes de soins de santé, la détection de communautés peut aider à analyser la croissance rampante des cellules dans un tissu pulmonaire (cancer des poumons).

Chapitre 2

La Détection de communautés

2.1 Introduction

Beaucoup de systèmes complexes dans divers domaines comme la biologie, l'informatique, la linguistique, le commerce, etc., peuvent être représentés de manière abstraite par des réseaux. Une des caractéristiques communes que l'on retrouve dans de nombreux réseaux concerne l'existence de zones plus densément connectées que d'autres. Ces zones sont habituellement appelées communautés et correspondent intuitivement à des groupes de nœuds plus fortement connectés entre eux qu'avec les autres nœuds du réseau, ce qui signifie que la détection de communautés a pour rôle de classer les membres du réseau en groupes. Pour faire la détection de communautés dans un réseau donné, nous avons besoin d'une représentation facile et pratique, comme la représentation par graphe. Ce chapitre tente de rapporter l'essentiel concernant la notion de communauté. Mais avant cela, il convient d'introduire au préalable des cas de modélisation par des graphes, ainsi que certaines notions et définitions utiles de la théorie des graphes.

2.2 Lapprentissage automatique

nous présenterons l'apprentissge automatique, ses principaux types ainsi qu'une présentation détaillée de l'algorithme qu'on va utiliser dans notre approche "l'algorithme K-Means". "Récemment, les réseaux sociaux font partie de notre vie quotidienne, comme par exemple Facebook, Twitter, Instagram, etc. Un réseau social est un ensemble d'individus reliés par différents types de liens : amitié, fraternité, profession, etc. L'interaction entre individus se fait par l'envoi de messages, le partage de photos, etc. Pour comprendre ces interactions et la structure de ces réseaux, une analyse intéressante a été réalisée : il s'agit de la détection des communautés. L'application des algorithmes d'apprentissage automatique dans les tâches de détection de communautés dans les réseaux a attiré une grande attention ces dernières années. Dans ce chapitre, nous nous intéressons à l'apprentissage automatique, à sa définition, à ses différents types, ainsi qu'à certaines de leurs variantes. Nous concluons ce chapitre par une explication détaillée de l'algorithme utilisé dans l'approche proposée, qui est l'algorithme K-means."

2.2.1 Méthodes d'Apprentissage Automatique

5.1 Méthodes d'Apprentissage Automatique L'apprentissage automatique, ou Machine Learning (ML), est une discipline scientifique qui consiste à développer des algorithmes capables d'ap-

prendre à résoudre une tâche spécifique sans être programmés explicitement . L'apprentissage automatique est une branche de l'intelligence artificielle qui permet à un logiciel d'apprendre et de reconnaître des modèles complexes dans les données, de la même manière qu'un être humain. Plus la quantité de données collectées est grande, plus la machine améliore ses compétences. Les méthodes d'apprentissage automatique peuvent être classées en plusieurs catégories : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement

Apprentissage supervisé "L'apprentissage supervisé (SL) est un algorithme d'apprentissage automatique permettant d'acquérir des informations sur la relation entrée-sortie d'un système, en se basant sur un ensemble donné d'échantillons d'apprentissage entrée-sortie étiquetées. Autrement dit, c'est un système qui reçoit à la fois les données en entrée et les données attendues en sortie. Les données en entrée et en sortie sont étiquetées pour établir une base d'apprentissage qui servira pour le traitement ultérieur des données [23, 55]. L'objectif de l'apprentissage supervisé est de construire un modèle capable d'apprendre la correspondance entre l'entrée et la sortie, et de prédire la sortie du système en fonction de nouvelles entrées [59]. L'apprentissage supervisé est généralement utilisé dans les domaines de la classification et de la régression [59]."

Apprentissage non supervisé L'apprentissage non supervisé (UL) est un type d'apprentissage automatique qui consiste à ne disposer que de données d'entrée et pas de variables de sortie correspondantes. Dans ce type d'apprentissage, les algorithmes sont laissés à leurs propres mécanismes pour découvrir et présenter la structure intéressante des données ; il n'y a pas de réponse correcte ni d'enseignant [4]. Les réponses que l'on cherche à prédire ne sont pas disponibles dans les jeux de données. Ici, l'algorithme utilise un jeu de données non étiquetées. On demande alors à la machine de créer ses propres réponses. Elle propose ainsi des réponses à partir d'analyses et de groupement de données. L'apprentissage non supervisé comprend deux catégories d'algorithmes : Algorithmes de Regroupement (Clustering en anglais) et les algorithmes de Réduction de la dimentiaLITE [9].

Apprentissage par renforcement "L'apprentissage par renforcement (Reinforcement Learning ou RL) est généralement utilisé pour enseigner à une machine comment exécuter une séquence d'étapes. Il diffère de l'apprentissage supervisé et non supervisé. Dans ce cadre, les scientifiques programment un algorithme pour accomplir une tâche en lui fournissant des indices positifs ou négatifs au fur et à mesure de son apprentissage. Le programmeur définit les règles des récompenses, mais laisse à l'algorithme la liberté de décider des actions à entreprendre pour maximiser la récompense, et ainsi accomplir la tâche de manière optimale [5, 49]."

2.2.2 Lapprentissage automatique et la détection de communautés

Aujourd'hui, un grand nombre d'algorithmes d'apprentissage automatique, et en particulier les algorithmes non supervisés, ont été utilisés pour la détection de communautés. Li et al. (2016) [58] ont proposé un algorithme amélioré de détection de communautés basé sur l'Analyse en Composantes Principales (PCA) . Les résultats des simulations montrent que l'algorithme proposé peut détecter les communautés avec plus de précision dans les réseaux complexes. Gujral et al. (2019) [36] ont proposé un algorithme d'apprentissage automatique efficace pour la détection de communautés, baptisé HACD (Hierarchical Agglomerative Community Detection). Cet algorithme combine l'information locale d'un graphe avec la propagation de l'appartenance, et ils ont démontré l'efficacité de cette approche dans la détection des communautés. Huan Li,

Ruisheng Zhang, Zhili Zhao et Xin Liu (2021) [57] ont présenté et évalué une nouvelle perspective pour la détection de communautés, basée sur un modèle d'apprentissage automatique. Ils ont développé un algorithme amélioré de propagation des étiquettes, nommé LPA-MNI, en combinant la fonction de modularité et l'importance des nœuds avec la LPA d'origine. Ils ont prouvé que cet algorithme offrait une meilleure précision, une modularité plus élevée et un nombre de communautés plus raisonnable par rapport aux autres algorithmes comparés. Aftab et al. (2021) [2] ont proposé un autre algorithme d'apprentissage automatique pour la détection de communautés dans les réseaux sociaux. Ils ont conçu un cadre de regroupement basé sur la communauté pour identifier les utilisateurs ayant des intérêts similaires. À cette fin, ils ont proposé un cadre hybride combinant les algorithmes MiniBatch K-means et DBSCAN, appelé Hybrid DBSCAN. Ce cadre s'adapte bien à la taille de l'ensemble de données.

2.3 Communauté

La notion de communautés dans les graphes n'a pas de définition formelle. Cependant, l'existence de zones plus densément connectées que d'autres est le résultat d'une présence de structures de graphes dont les nœuds se sont regroupés en communautés du fait de leur ressemblance ou de leurs intérêts communs [32]. Cette ressemblance ou ce partage peut avoir des interprétations différentes selon la nature et le type du réseau d'interaction considéré (réseaux sociaux, réseaux biologiques, etc.). Nous allons donner ici deux définitions des communautés, l'une sémantique et l'autre structurelle [61, 1].

- **Définition sémantique**

  Une communauté est un ensemble de nœuds qui partagent les mêmes centres d'intérêt ou ayant le même profil.

- **Définition structurelle**

  Une communauté est un ensemble de nœuds fortement liés entre eux et faiblement liés avec les autres nœuds du graphe.

2.3.1 Structure communautaire

$$\mathsf{C} = \{C_1, C_2, C_3, \ldots, C_k\}$$

où C désigne la structure communautaire (ou couverture), et chaque C_i représente une communauté, i.e. un sous-ensemble de nœuds fortement connectés entre eux. La **taille** de la structure communautaire, notée $|\mathsf{C}|$, indique le nombre total de communautés détectées[17].

$$C_1 = \{1, 6, 9, 10\}, \quad C_2 = \{2, 4, 5\}, \quad C_3 = \{3, 7, 8, 11, 12\}$$

et la structure communautaire correspondante est :

$$\mathsf{C} = \{C_1, C_2, C_3\}$$

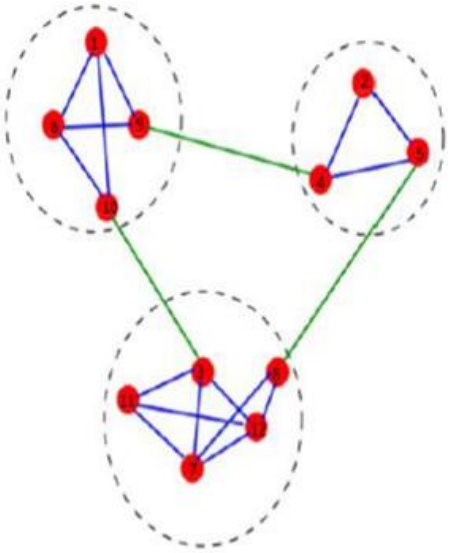

FIGURE 2.1 – Structure de communauté dans le réseaux [51]

La structure de communauté où certains nœuds sont chevauchants est connue sous le nom de structure de communauté chevauchante (voir figure 1.3). La structure de communauté où tous les nœuds sont non chevauchants est connue sous le nom de structure de communautés disjointes [1]. La détection des structures communautaires peut être définie par une classification des nœuds du réseau plus densément connectés que d'autres, afin de construire des classes connexes d'utilisateurs ayant les mêmes caractéristiques au regard d'une mesure de similarité se référant à des intérêts communs [18, 1].

2.3.2 Objectifs de la détection des communautés

Parmi les objectifs de cette notion de communautés, on peut citer ce qui suit [27] :

- La détection de communautés a pour objectif de révéler de nouvelles relations et d'extraire de nouvelles propriétés dans un réseau représenté sous forme de graphe, dans le but de comprendre la structure d'un tel réseau, de révéler des informations, etc.

- Identification des acteurs centraux d'un système : dans le domaine de la sécurité informatique, par exemple, il est important de savoir quels sont les nœuds (leaders) les plus importants qui doivent être attaqués (ou mieux sécurisés si l'on est du côté administrateur) pour déstabiliser (ou stabiliser) un système ou un réseau.

- Fournir un résumé de la structure du réseau.

- Avoir de nombreuses propriétés sur le réseau, comme l'importance d'un acteur donné (son influence, sa popularité, etc.) par rapport aux autres.

- Étudier la similarité entre les individus d'une même communauté et donc le degré d'interaction, puis mesurer ensuite la force de la relation entre eux.

- Extraction des différents profils, centres d'intérêt, sujets d'actualité dont la population parle.

- Connaître la tendance politique d'une population, connaître les goûts et les opinions des gens sur les produits proposés sur le marché.

- La connaissance d'un individu peut induire la connaissance des autres en relation avec lui ou avec le reste du groupe, le cas échéant.

- Mise en place d'une stratégie de marketing (systèmes de recommandation) : à partir de la connaissance de l'ensemble des communautés, nous pouvons connaître les différents profils (ou les intérêts communs regroupant les membres), et ensuite faire une diffusion (publicité, recommandation personnalisée, etc.) d'informations précises à un ensemble bien connu d'utilisateurs.

- Solutions pour minimiser / maximiser la diffusion.

- Extraire les connaissances du réseau et mettre en évidence les principales propriétés du réseau.

2.3.3 Applications de la détection des communautés

Plusieurs applications de la détection des communautés existent. On peut citer quelques-unes :

o Application dans les systèmes biologiques et les systèmes de santé La détection de communautés dans les réseaux biologiques a une signification importante, tels que les réseaux de protéines, les réseaux alimentaires, les réseaux métaboliques, etc. Dans les réseaux de protéines, elle a été appliquée de manière à détecter les complexes protéiques. Dans les systèmes de soins de santé, la détection de communautés peut aider à analyser la croissance rampante des cellules dans un tissu pulmonaire (cancer des poumons).

o Application dans la détection de fraudes et d'anomalies Par exemple, dans les transactions bancaires, les techniques de l'exploration de données (data mining en anglais), telles que la détection de communautés, analysent le réseau de transactions en classifiant les transactions en communautés. Si une nouvelle transaction n'appartient à aucune communauté, elle est considérée comme une anomalie ou une fraude.

o Applications scientifiques et académiques Des algorithmes de détection de communautés sont utiles dans le domaine de la recherche scientifique. Leur utilité réside dans la capacité de classer les auteurs, leurs publications, les années et les lieux de publication, etc. Ces algorithmes

peuvent prédire de nouvelles relations entre auteurs (collaboration scientifique) et peuvent proposer de nouveaux papiers aux auteurs suivant leur profil. Ce système peut être réduit à un cas simple, comme une bibliothèque où nous pouvons proposer des livres aux étudiants suivant leur spécialité, analyser la similarité entre livres et entre étudiants, etc.

2.3.4 Algorithmes de détection de communautés

Il existe plusieurs algorithmes de détection de communautés. Dans cette section, nous présentons les algorithmes utilisés pour comparer notre travail.

Girvan et Newman C'est l'algorithme hiérarchique divisif le plus connu, abrégé sou- vent par GN pour désigner les auteurs Girvan et Newman, qui ont introduit une mesure de centralité appelée centralité d'intermédiarité des liens (Edge-Betweenness Centrality) pour partitionner un graphe. Cette mesure de centralité est définie comme le nombre de plus courts chemins entre deux nœuds qui passent par une arête. Cet algorithme est particulièrement in- tuitif. La première étape consiste à calculer cette edge-betweenness pour toutes les arêtes du graphe, puis à retirer celle ayant la plus haute betweenness. Ce processus est itéré jusqu'à ce que la dernière arête soit retirée. Dans une seconde phase, à partir du graphe sans arêtes, les arêtes sont réintroduites dans l'ordre inverse, ce qui fournit une hiérarchie très fine du réseau, car en ajoutant une arête reliant deux communautés, on ajoute un niveau hiérarchique, les deux communautés étant regroupées dans une super-communauté L'idée de cet algorithme est la suivante : si un lien se trouve fréquemment sur les plus courts chemins entre les nœuds du graphe, alors il ne se trouve pas au sein d'une communauté donnée, mais il relie des portions distantes du graphe (des communautés distinctes)

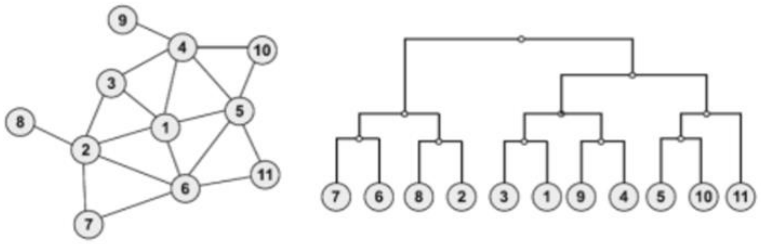

FIGURE 2.2 – Exemple d'un dendrogramme hiérarchique pour Newman [52]

Le schéma de l'image représente un graphe et son dendrogramme hiérarchique, illustrant la méthode de détection de communautés de Newman. Initialement, chaque nœud du graphe forme sa propre communauté, puis, à chaque étape, les groupes les plus proches ou similaires sont regroupés deux à deux, ce qui se traduit par la construction progressive du dendrogramme. Ce processus, appelé méthode agglomérative, permet de visualiser la structure hiérarchique des communautés et de sélectionner la partition optimale selon la modularité, c'est-à-dire la configuration qui maximise la densité des liens à l'intérieur des groupes par rapport aux liens entre eux. En résumé, le schéma montre comment on passe d'un réseau d'individus à une organisation hiérarchique des communautés, facilitant l'analyse et l'interprétation des structures sous-jacentes.

L'algorithme de Louvain : C'est une simple méthode hiérarchique ascendante de détection de communautés proposée en 2008 par Blondel et al. de l'Université de Louvain [8]. Il part de l'hypothèse que chaque nœud est une communauté, puis il regroupe chaque paire de nœuds adjacents dans une même communauté en maximisant la modularité. Il comprend deux phases. Premièrement, il cherche les « petites » communautés en optimisant la modularité de manière locale (séparément, au niveau de chaque communauté). Deuxièmement, il regroupe les nœuds d'une même communauté et construit un nouveau réseau dont les nœuds sont les communautés, avec pour poids la somme des poids des arêtes entre les deux communautés. Ces deux phases produisent un nouveau niveau hiérarchique de découpage en communautés. L'algorithme s'arrête lorsqu'aucune fusion de la première phase n'améliore plus la modularité [15, 27]. Cet algorithme est très efficace en termes de temps d'exécution ; il permet d'analyser des réseaux typiques de deux millions de nœuds en deux minutes [8]. De plus, la qualité des communautés détectées est très bonne [22].

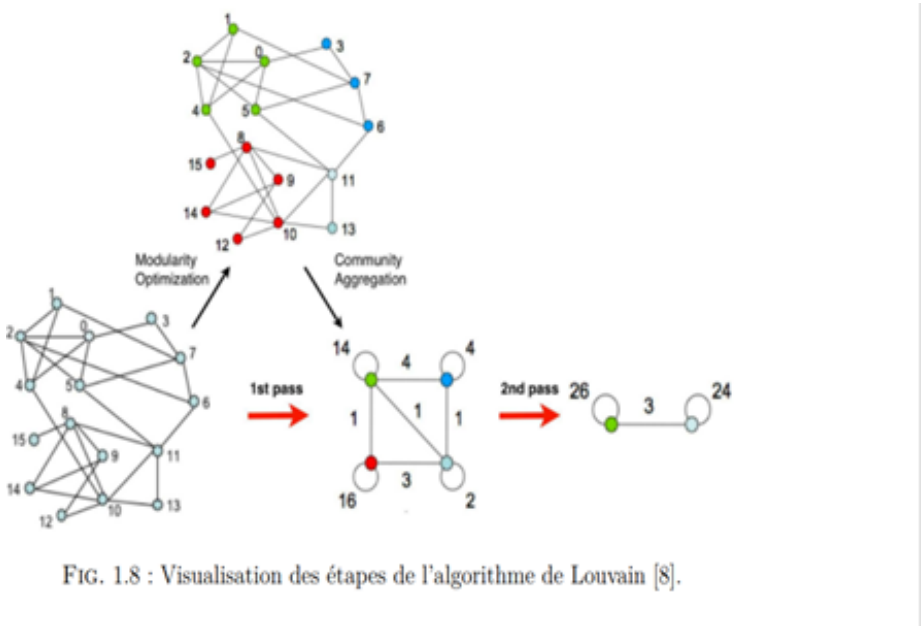

FIG. 1.8 : Visualisation des étapes de l'algorithme de Louvain [8].

FIGURE 2.3 – Visualisation des étapes de l'algorithme de Louvain [53]

Le schéma illustre le fonctionnement de l'algorithme de Louvain, utilisé pour détecter des communautés dans un réseau. L'algorithme commence par regrouper chaque nœud dans sa propre communauté, puis il déplace progressivement les nœuds entre communautés dans le but d'optimiser la modularité, c'est-à-dire de maximiser la densité des liens à l'intérieur des groupes par rapport aux liens entre eux. Après cette première phase, les communautés identifiées sont agrégées en super-nœuds, formant un nouveau graphe simplifié sur lequel l'algorithme recommence le processus. Cette alternance entre optimisation locale et agrégation se répète jusqu'à ce qu'aucune amélioration ne soit plus possible, permettant ainsi de révéler efficacement la structure hiérarchique des communautés dans le réseau.

L'algorithme Infomap : L'algorithme Infomap a été créé par Rosvall et Bergstrom (2011) [96]. Cet algorithme fait appel à une équation carte généralisée (generalized map equation). Emprunté de la théorie de l'information, ce concept est utilisé afin d'obtenir une hiérarchie de partitions à niveau variable. Les paramètres de cette équation récursive sont obtenus par marches aléatoires. Cet algorithme est similaire à la méthode Louvain, c'est-à-dire qu'en première phase, une liste de sommets triée aléatoirement est parcourue et chaque opération vise à trouver le voisin avec lequel le sommet sélectionné constitue une communauté minimisant

la valeur de l'équation de carte jusqu'à ce qu'un minimum soit atteint. L'algorithme procède alors de façon récursive à une nouvelle phase essentiellement identique, mais à un niveau supérieur, soit avec les communautés et sous-communautés. Pour chaque phase, l'ordre aléatoire des sommets ou modules est réévalué après chaque parcours. Cet algorithme possède l'avantage d'arriver à des partitions hiérarchisées sans avoir à choisir le nombre de communautés ou sous-communautés ni le nombre de niveaux de l'hiérarchie. De plus, Infomap prouve son efficacité sur de grands réseaux par sa rapidité

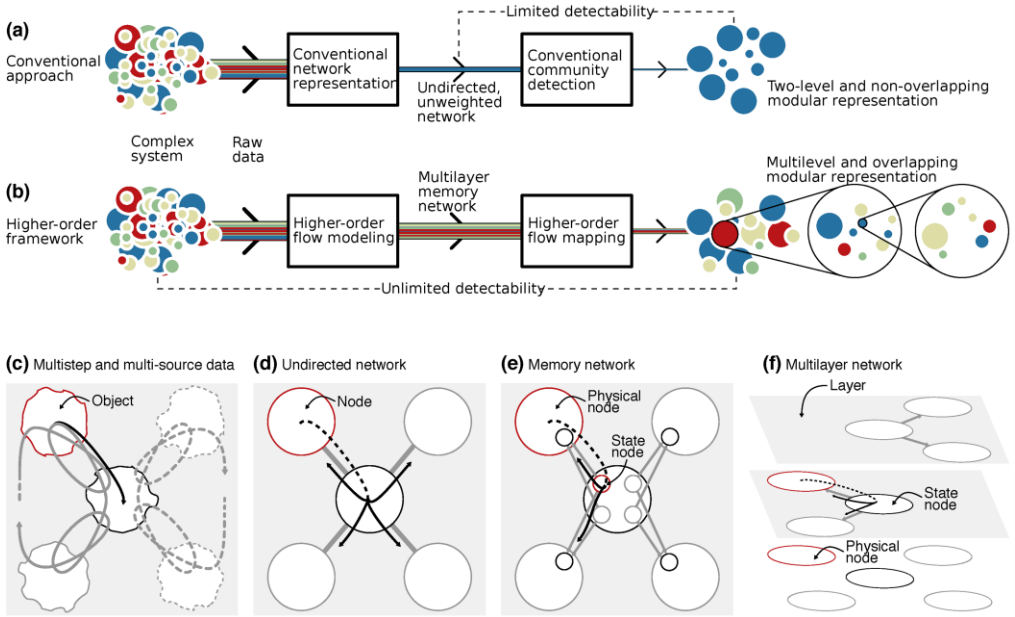

FIGURE 2.4 – illustre la différence entre l'approche conventionnelle de détection de communautés [54]

Cette figure compare deux approches de détection de communautés dans les réseaux complexes. L'approche classique (a) simplifie les données en un réseau non orienté et non pondéré, ce qui limite la détection des communautés à une structure simple, non chevauchante et à deux niveaux. En revanche, l'approche avancée (b) utilise des réseaux à mémoire et multicouches, qui tiennent compte des dépendances temporelles ou contextuelles dans les données. Cela permet de détecter des communautés plus fines, potentiellement chevauchantes et hiérarchiques. Les parties (c) à (f) illustrent cette richesse : (c) montre les données complexes avec dépendances multi-étapes, (d) un réseau classique, (e) un réseau à mémoire avec des nœuds d'état représentant le contexte, et (f) un réseau multicouche où les couches représentent différentes dimensions ou sources. Cette approche offre une représentation plus fidèle et plus puissante des structures cachées dans les systèmes complexes. Il s'agit de la représentation de la trajectoire suivie par un marcheur aléatoire dans un réseau. Cette représentation doit capturer les différents nœuds visités et les transitions entre eux, de manière à refléter l'évolution du parcours dans le graphe ou réseau. Un codage binaire de l'ensemble des parcours avec le moins de caractères possibles : L'objectif est de compresser les informations relatives aux parcours en utilisant un codage binaire. Ce codage doit être conçu pour réduire au maximum la longueur de la séquence tout en préservant l'intégrité des données, permettant ainsi de représenter chaque parcours de manière efficace Un codage à deux niveaux (sur les groupes puis sur les nœuds) permettant à un même code d'être utilisé pour des nœuds différents :

Il s'agit d'un codage hiérarchique dans lequel les informations sont d'abord organisées et codées au niveau des groupes de nœuds (communautés ou sous-communautés), puis au niveau des nœuds individuels. Ce codage permet de réutiliser un même code pour des nœuds qui, bien

qu'étant distincts, appartiennent à des groupes similaires, optimisant ainsi la compression.

Une visualisation des groupes sous forme de graphe quotient : Un graphe quotient est un graphe dans lequel les nœuds représentent des classes d'équivalence (dans ce cas, des groupes ou communautés). Chaque groupe est représenté par un seul nœud, et les arêtes entre ces nœuds représentent les connexions entre les groupes dans le graphe d'origine. Cette visualisation permet de simplifier la structure globale du réseau tout en conservant l'information essentielle sur les relations entre les communautés.

L'algorithme Walktrap une méthode de détection de communautés dans des graphes basée sur les parcours aléatoires. L'idée principale de cet algorithme est de considérer les communautés comme des groupes de nœuds qui sont fortement connectés entre eux par des parcours aléatoires. En d'autres termes, deux nœuds seront considérés comme faisant partie de la même communauté si un marcheur aléatoire les traverse fréquemment ensemble. Le processus commence par attribuer à chaque nœud une communauté individuelle. Ensuite, des marches aléatoires sont effectuées sur le graphe pour déterminer la similarité entre les nœuds. La similarité entre les nœuds est calculée en fonction de la fréquence des chemins traversant les mêmes nœuds, ce qui permet de mesurer à quel point les nœuds sont liés entre eux par des marches aléatoires. À chaque itération, l'algorithme fusionne les communautés les plus similaires, c'est-à-dire celles dont les nœuds partagent des parcours aléatoires similaires. Ce processus est répété jusqu'à ce qu'une structure stable de communautés soit atteinte. L'algorithme repose sur une approche hiérarchique. Initialement, chaque nœud est traité comme une communauté distincte, mais au fur et à mesure que les communautés fusionnent, l'algorithme construit une hiérarchie de communautés imbriquées. La mesure de la similarité utilisée est généralement basée sur les parcours aléatoires de petite taille, et la fusion des communautés se fait en fonction de cette mesure. Une des forces de Walktrap est qu'il ne nécessite pas de paramétrer le nombre de communautés à l'avance. L'algorithme ajuste la taille des communautés en fonction de la structure du graphe, ce qui permet de mieux adapter la détection à la réalité du réseau étudié. L'algorithme est souvent utilisé pour analyser des graphes complexes, comme les réseaux sociaux, les réseaux biologiques, ou les réseaux de citation scientifique, où l'identification de groupes cohérents de nœuds peut apporter des informations utiles sur la structure sous-jacente du réseau. Cependant, une des limitations de Walktrap réside dans sa complexité computationnelle. Pour de très grands graphes, le calcul des parcours aléatoires et des similarités entre nœuds peut devenir coûteux. Des améliorations, comme l'utilisation de méthodes d'échantillonnage ou des optimisations des marches aléatoires, peuvent être envisagées pour améliorer l'efficacité de l'algorithme sur de grands réseaux. En résumé, Walktrap est un algorithme robuste pour la détection de communautés basé sur les parcours aléatoires, permettant de révéler des structures hiérarchiques dans les graphes sans nécessiter un nombre préétabli de communautés.

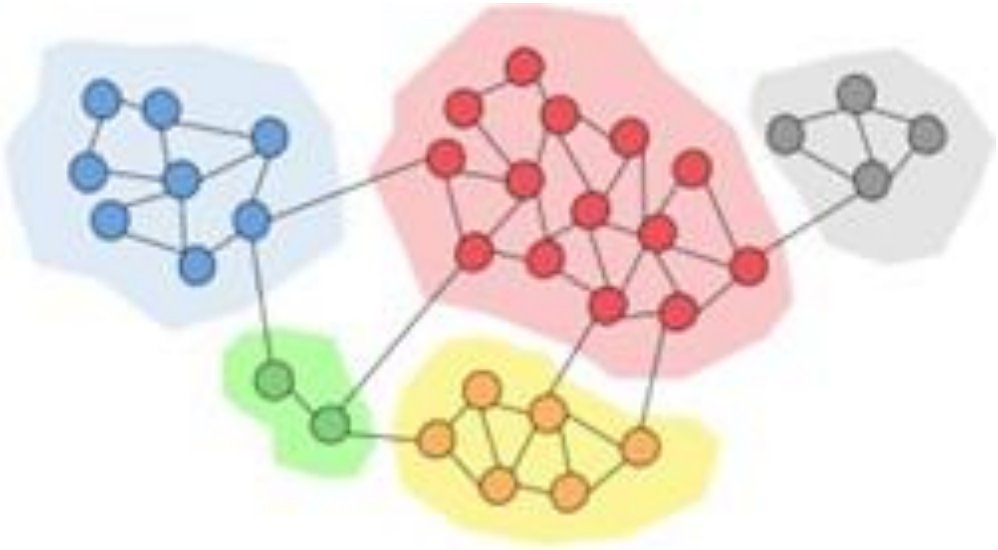

FIGURE 2.5 – exemple de partitionnement d'un réseau en 5 communautés la majorité des connexions sont intra-communautaires [55]

Communauté bleue : composée de nœuds fortement connectés entre eux, formant un sous-ensemble dense du graphe.
Communauté verte : plus petite, elle relie la bleue et l'orange, jouant un rôle de **pont** dans le réseau.
Communauté rouge : la plus grande, elle est composée de nombreux nœuds connectés de manière interne.
Communauté jaune/orange : également dense, elle présente une structure interne cohérente avec peu de connexions vers l'extérieur.
Communauté grise : isolée sur le côté, elle montre un groupe de nœuds faiblement connecté au reste du graphe.

e schéma illustre le résultat de l'algorithme Walktrap appliqué à un graphe pour détecter des communautés. Cet algorithme repose sur le principe que des marches aléatoires courtes ont tendance à rester confinées à l'intérieur des communautés, c'est-à-dire des groupes de nœuds densément connectés entre eux. À partir de ces marches, Walktrap calcule une distance entre les nœuds ou groupes de nœuds, puis utilise une approche hiérarchique agglomérative pour fusionner progressivement les communautés les plus proches. Le résultat final est une partition du graphe en plusieurs communautés, comme le montrent les zones colorées du schéma (bleu, vert, rouge, jaune et gris), où chaque couleur correspond à un groupe de nœuds étroitement liés. Ce type de structure révèle la capacité de Walktrap à regrouper efficacement les nœuds selon leur proximité topologique.

L'algorithme de Leiden L'algorithme de Leiden est une méthode non supervisée de détection de communautés dans les graphes, conçue comme une amélioration de l'algorithme de Louvain. Il vise à optimiser une fonction de qualité (comme la modularité ou le modèle de Potts) tout en garantissant que les communautés détectées soient fortement connexes, ce que ne garantit pas Louvain. L'algorithme procède en trois étapes : une phase de raffinement local, où chaque nœud est déplacé vers la communauté qui maximise le gain de qualité ; une phase de stabilisation, qui réorganise les nœuds mal placés pour renforcer la cohésion interne des communautés ; et une phase d'agrégation, où un graphe réduit est construit à partir des communautés trouvées, sur lequel le processus est répété. Plus stable, plus précis et garantissant la connexité des communautés, Leiden surpasse Louvain, notamment en évitant les partitions arbitraires et en assurant une convergence vers de meilleurs optima. Il reste efficace sur de grands graphes tout en produisant des résultats plus fiables. Ce travail a été formalisé par Traag, Waltman et van Eck en 2019 dans Scientific Reports [Traag et al., 2019].

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c, c)$$

L'algorithme de Leiden peut également être utilisé avec d'autres fonctions comme la modularité généralisée ou la CPM (Constant Potts Model).

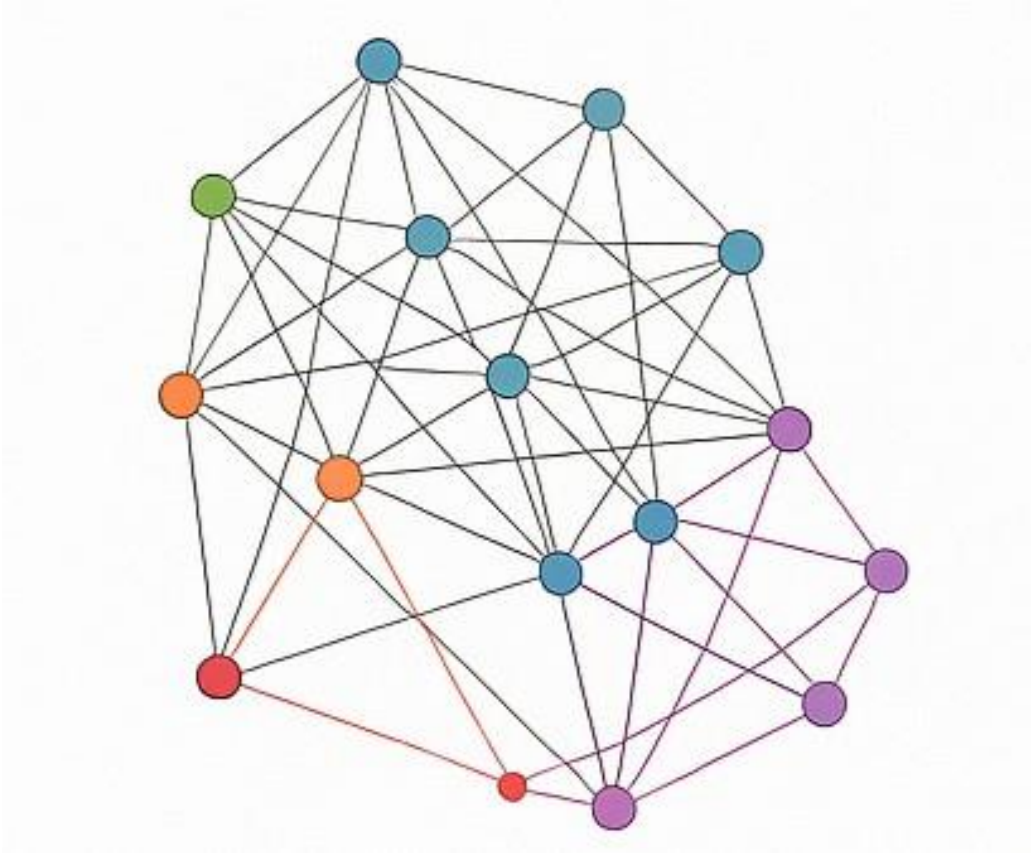

FIGURE 2.6 – FONCTION DALGORITHME LEIDEN [56]

Bleu clair : Tous les nœuds bleus appartiennent à une même communauté centrale, très connectée.

Violet : Les nœuds violets forment une autre communauté, plus localisée sur la droite du graphe.

Orange : Les nœuds oranges constituent une petite communauté distincte à gauche.

Rouge : Le nœud rouge (en bas à gauche) est isolé dans une petite communauté.

Vert : Le nœud vert (en haut à gauche) est aussi dans une communauté distincte.

illustre une détection de communautés dans un graphe, où chaque nœud est coloré selon la communauté à laquelle il appartient. Contrairement à une segmentation par zones, ici les communautés sont représentées par des couleurs distinctes appliquées aux nœuds eux-mêmes, mettant en évidence les regroupements topologiques. Ce type de visualisation est typique des algorithmes tels que Louvain ou Leiden, qui fonctionnent en optimisant une mesure de modularité. L'objectif est de maximiser la densité des connexions à l'intérieur des communautés tout en minimisant celles entre les communautés. Le résultat obtenu montre plusieurs groupes de nœuds fortement interconnectés entre eux, avec des liaisons plus faibles vers les autres groupes, révélant la structure communautaire sous-jacente du graphe.

L'algorithme de Label Propagation est une méthode de détection de communautés non supervisée, simple et efficace, qui repose sur une idée intuitive : les nœuds d'un graphe ont tendance à adopter l'étiquette (ou label) de la majorité de leurs voisins. Voici une présentation structurée de l'algorithme :

- Chaque nœud du graphe commence avec une étiquette unique, généralement son propre

identifiant.

- À chaque itération, chaque nœud met à jour son étiquette en adoptant l'étiquette la plus fréquente parmi ses voisins.

- Le processus continue jusqu'à stabilisation, c'est-à-dire lorsque les étiquettes ne changent plus ou oscillent entre quelques valeurs.

- Chaque nœud v se voit attribuer une étiquette unique Lv=v

- Pour chaque nœud v (dans un ordre aléatoire ou fixé), on met à jour son étiquette :

$$L_v = \arg\max_l |\{u \in \mathsf{N}(v) : L_u = l\}|$$

où N(v)est l'ensemble des voisins de vvv, et on choisit l'étiquette la plus fréquente parmi eux. En cas d'égalité, on choisit aléatoirement parmi les étiquettes les plus fréquentes Lorsque les étiquettes ne changent plus (convergence), ou qu'un nombre maximal d'itérations est atteint.

2.4 Mesures d'évaluation de la qualité des structures communautaires

Actuellement, il existe de nombreuses mesures pour évaluer l'efficacité des algorithmes de détection de communautés. Parmi ces métriques, on a :

2.4.0.1 La Modularité (Q)

"C'est l'une des métriques fréquemment utilisées pour mesurer la qualité de la détection communautaire des réseaux. Elle a été proposée par Girvan et Newman en 2004 [76]. Les réseaux à forte modularité présentent des connexions denses entre les nœuds au sein des modules, mais des connexions éparses entre les nœuds de différents modules. La modularité (Q) est définie par [57, 115] :" Les corrections sont mineures, mais elles améliorent la structure de la phrase. Si vous avez d'autres passages à corriger, je suis à votre disposition !

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j)$$

Aij : Représente la matrice d'adjacence du réseau.

ci : Représente la communauté à laquelle le nœud i est assigné.

c : Représente le nombre total de communautés.

ki : Représente le degré du nœud i.

kj : Représente le degré du nœud j.

m : Représente le nombre total d'arêtes dans le réseau. Ainsi

$$\delta(c_i, c_j) = \begin{cases} 1, & \text{Si le nœud } i \text{ et le nœud } j \text{ sont dans la même communauté,} \\ 0, & \text{sinon.} \end{cases}$$

La valeur de Q est comprise entre 1 et +1. Plus cette valeur est proche de 1, plus la force de la structure communautaire dans le réseau est élevée, et donc la qualité de la détection des communautés est meilleure

2.4.0.2 l'information mutuelle normalisée (NMI)

"C'est une mesure de similarité qui évalue la similarité entre deux partitions, A et B, où A représente la partition réelle du réseau et B la partition détectée par les algorithmes expérimentaux de détection de communautés. Elle est fondée sur la théorie de l'information [26]. Pour deux partitions A et B d'un réseau, la valeur de l'NMI est calculée à l'aide de l'équation

$$NMI(A, B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} N_{ij}\log\left(\frac{N_{ij}N}{N_i N_j}\right)}{\sum_{i=1}^{C_A} N_i \log\left(\frac{N_i}{N}\right) + \sum_{j=1}^{C_B} N_j \log\left(\frac{N_j}{N}\right)}$$

A : représente la partition réelle du réseau.

B : représente la partition découverte par les algorithmes de détection de communautés.

CA : représente le nombre de communautés dans la partition A.

CB : désigne le nombre de communautés dans la partition B.

N : représente le nombre total de nœuds dans le réseau.

Nij : représente le nombre de nœuds communs entre la communauté i de la partition A et la communauté j de la partition B.

Ni : représente le nombre de nœuds dans la communauté réelle i (c'est la somme de la ligne i de la matrice Nij).

Nj : représente le nombre de nœuds dans la communauté calculée j (c'est la somme de la colonne j). La valeur de NMI peut varier entre 0 et 1. Plus la valeur de NMI est proche de 1, plus les deux partitions sont similaires. En d'autres termes, lorsque deux partitions A et B sont complètement différentes, NMI(A, B) = 0. Si NMI prend sa valeur maximale, qui est égale à 1, cela signifie que la partition A correspond exactement à la partition B."

2.4.0.3 Indice de Rand ajusté (ARI) :

"Dans la mesure où l'indice de Rand varie de façon très importante entre deux partitions tirées au hasard, ce que Vinh appelle la « constant baseline property », une version « ajustée » a été créée, dont l'espérance est nulle lorsque les partitions sont sélectionnées au hasard [41]. L'indice de Rand ajusté (ARI, pour Adjusted Rand Index) corrige cet effet en normalisant l'indice de Rand (RI). C'est une adaptation de l'indice de Rand conçue pour être insensible au nombre de classes [20, 102]. L'ARI est défini par l'équation

$$ARI\,(P_1, P_2) = \frac{RI(P_1, P_2) - E(RI(P_1, P_2))}{\max(RI(P_1, P_2)) - E(RI(P_1, P_2))}$$

"Où E(RI(P1,P2)) est l'espérance de la valeur de l'indice de Rand, autrement dit, l'indice obtenu en partitionnant les données au hasard. Si les partitions sont identiques, l'ARI vaut 1. L'ARI est une mesure corrigée pour la chance : l'espérance de l'ARI de deux partitions tirées aléatoirement vaut 0 [20]. Nous allons comparer notre travail avec d'autres travaux à l'aide de la mesure d'évaluation Q (la Modularité)."

2.5 Conclusion

Ce chapitre a exploré les concepts clés liés à la détection de communautés dans les réseaux, en mettant l'accent sur l'importance de comprendre les structures internes des graphes. Il a détaillé différentes approches et algorithmes pour identifier les communautés, tels que l'algorithme de Girvan et Newman, Louvain, et Infomap. De plus, il a introduit les mesures d'évaluation de la qualité des partitions communautaires, telles que la modularité, l'indice de Rand ajusté et la NMI, qui permettent de mesurer l'efficacité des algorithmes de détection. En outre, l'usage de l'apprentissage automatique dans ce domaine a été abordé, notamment à travers des méthodes comme k-means et DBSCAN, pour améliorer la détection des communautés dans les réseaux complexes.

Chapitre 3

Conception et Implémentation de l'Approche Proposée

3.1 Introduction

Les réseaux complexes sont des structures fondamentales utilisées pour modéliser des systèmes comportant de nombreux éléments interconnectés. Ces réseaux sont présents dans divers domaines, tels que les réseaux sociaux, où les individus interagissent, et les réseaux biologiques, représentant des interactions entre gènes, protéines ou autres entités biologiques. Dans de tels réseaux, l'identification des communautés — groupes de nœuds fortement interconnectés — joue un rôle clé pour comprendre la structure sous-jacente et les dynamiques du système étudié. Afin de détecter ces communautés, plusieurs algorithmes ont été développés, chacun avec des approches distinctes. Parmi ceux-ci, les algorithmes Louvain, Infomap et Walktrap se distinguent par leur efficacité et leur application dans des contextes variés. Ces méthodes sont évaluées à travers des critères de performance tels que la modularité, l'ARI, le NMI, permettant ainsi une comparaison approfondie de leurs résultats. L'analyse des performances de ces algorithmes offre un éclairage précieux sur leur adaptabilité et leur efficacité pour différents types de réseaux complexes

3.2 Environnement de développement

Dans cette section, nous présenterons les environnements matériels et logiciels que nous avons utilisés pour développer et exécuter nos modèles et approches proposés.

3.2.1 L'environnements logiciel

est le standard de facto pour la communication et la publication de documents scientifiques. Il est disponible en tant que logiciel libre.

Paython

Python est un langage de programmation interprété de haut niveau, multipara digme, couvrant la programmation fonctionnelle, procédurale et orientée objet, développé par Guido van

Rossum. Il peut être utilisé dans de nombreux contextes, avec des bibliothèques spécialisées adaptées à tout type d'utilisation. Cependant, il est surtout utilisé comme langage de script pour automatiser des tâches simples mais fastidieuses. Il est également utilisé avec succès dans des milliers d'applications commerciales réelles dans le monde entier, y compris de nombreux système de grande envergure et critiques

Tensorflow TPU

TensorFlow TPU (Tensor Processing Unit) est une accélération matérielle spéci- fique développée par Google pour l'exécution de modèles d'apprentissage automa- tique et d'intelligence artificielle. Les TPU sont conçus pour offrir des performances optimales lors de l'entraînement et de l'inférence de modèles de deep learning, en particulier pour les tâches intensives en calcul. Ils sont largement utilisés dans les environnements cloud pour accélérer les charges de travail de machine learning à grande échelle. TensorFlow est compatible avec les TPU, ce qui permet aux utilisa- teurs de tirer parti de cette accélération matérielle pour leurs projets d'apprentissage automatique.

Keras

Keras est une bibliothèque open-source écrite en Python qui peut s'exécuter sur TensorFlow et permet d'interagir avec des réseaux de neurones profonds et des algorithmes d'apprentissage automatique. Elle a été initialement écrite par François Chollet. Elle a été développée pour rendre la mise en œuvre des modèles d'appren- tissage profond aussi rapide et facile que possible.

Sickit-learn

Scikit-Learn est un package d'apprentissage automatique pour le langage de pro- grammation Python qui comprend des méthodes telles que les forêts aléatoires, la régression logistique, les machines à vecteurs de support, et d'autres.

Notebooks jupyter

Les notebooks Jupyter sont un outil web open-source pour développer et partager des documents qui incluent du code en direct, des équations, des visualisations et du texte dans plusieurs formats. Les codes de cette thèse sont disponibles sous forme de notebooks Jupyter.

Google Colab

Google Colab est un environnement de recherche créé par Google pour des raisons de recherche et d'éducation, axé sur la distribution de la recherche en apprentissage automatique. C'est un environnement de notebook Jupyter basé sur le cloud qui ne nécessite aucune installation. Cela signifie que vous pouvez entraîner vos mo- dèles sur un GPU K80 gratuitement tant que vous avez un compte Google. Google Colaboratory Cloud offre un TPU gratuit de 16 Go RAM .

Kaggle

Kaggle est une communauté en ligne de scientifiques des données et d'apprentis en machine learning, appartenant à Google LLC. Il permet aux utilisateurs de trouver et de publier des ensembles de données, d'explorer et de créer des modèles dans un environnement de science des données basé sur le web qui est un notebook ou un script. Les Kernels Kaggle offrent un GPU Cloud gratuit NVidia K80 GPUs et 16 Go de RAM gratuits, la session est limitée à 6 heures.

MAtplotlib

Matplotlib est une bibliothèque Python utilisée pour tracer et visualiser des don- nées sous forme de graphiques. Elle peut être combinée avec les bibliothèques de calcul scientifique NumPy et SciPy. Elle peut exporter des formats matriciels (PNG, JPEG) et des formats vectoriels (PDF, SVG).

3.2.2 L'environnement Matériel

Composant Matériel Configuration

Ordinateur Dell Inc. XPS 13 9305 Processeur 11th Gen Intel(R) Core(TM) i5-1135G7 cœur(s), 8 processeur(s) logique(s) @ 2.40GHz, 2419 MHz, 4 Mémoire (RAM) 16,00 Go Système d'exploitation Windows 64 bits

3.3

Dans le cadre de notre projet, nous avons suivi un processus méthodique pour analyser les données. Tout d'abord, nous avons commencé par la collecte de données, une étape cruciale pour garantir la qualité et la pertinence des informations. Ensuite, ces données ont subi un prétraitement rigoureux pour les préparer à l'analyse. Nous avons ensuite appliqué trois méthodes distinctes : Louvain, Infomap, et Walktrap,et liedn avec newamn chacune offrant une perspective unique sur les données. Enfin, les résultats obtenus ont été soumis à une phase d'entraînement et validation pour évaluer leur efficacité et leur précision.

3.4 collection des données

Les bases de données sont cruciales pour organiser et analyser des informations diverses. Elles comprennent des données relationnelles et non relationnelles, ainsi que des réseaux complexes tels que les réseaux biologiques et sociaux. Ces systèmes permettent d'évaluer la robustesse des algorithmes sur des ensembles de données variés, en tenant compte de la taille, du degré moyen, des coefficients de clustering et des indices d'hétérogénéité

3.4.1 Ensembel de donnée sur communautés

L'ensemble de données utilisé dans cette étude comprend plusieurs réseaux de communautés bien connus, issus de domaines variés tels que la biologie, les réseaux sociaux, la politique, les communications institutionnelles et les infrastructures techniques. Parmi eux, on retrouve par exemple le réseau Dolphins, représentant les interactions sociales entre dauphins, Polbooks, illustrant les co-achats de livres à orientation politique, ou encore Football, un réseau de rencontres sportives entre équipes universitaires américaines. À cela s'ajoutent des réseaux plus complexes comme EU-Core, retraçant les échanges d'e-mails au sein d'une institution européenne, ou AS (Autonomous Systems), représentant les connexions entre systèmes autonomes sur Internet. Chacun de ces réseaux est modélisé sous forme de graphe, les nœuds représentant les entités (individus, livres, équipes, etc.) et les arêtes, leurs interactions. La diversité topologique de ces graphes permet de tester la robustesse et la généralisabilité des algorithmes de détection de communautés dans des contextes très hétérogènes.

3.4.2 Prétraitement

Dans le cadre de cette étude, j'ai moi-même réalisé un prétraitement sur l'ensemble des jeux de données utilisés. Cette étape a permis de normaliser les graphes en convertissant les identifiants de nœuds en entiers afin d'assurer la compatibilité avec les algorithmes implémentés. Lorsque cela était nécessaire, je me suis également assuré de ne conserver que le composant connexe principal de chaque réseau, afin de garantir la cohérence des résultats. Par ailleurs, j'ai extrait les étiquettes de vérité terrain (ground-truth), lorsqu'elles étaient disponibles, pour permettre une évaluation fiable des performances des méthodes de détection de communautés.

3.4.3 datasets

Un réseau social classique représentant les relations d'amitié entre 34 membres d'un club de karaté. Il est souvent utilisé comme référence pour tester les algorithmes de détection de communautés, notamment parce qu'il possède une division connue résultant d'un conflit interne.

Karate (Zacharys Karate Club) : Un réseau social classique représentant les relations d'amitié entre 34 membres d'un club de karaté.Il est souvent utilisé comme référence pour tester les algorithmes de détection de communautés, notamment parce qu'il possède une division connue résultant d'un conflit

Dolphins : Ce réseau représente les interactions sociales observées entre 62 dauphins vivant au large de la Nouvelle-Zélande. Les liens indiquent des associations fréquentes entre individus.

Polbooks (Political Books) : Réseau de co-achats de 105 livres politiques sur Amazon lors de l'élection présidentielle américaine de 2004. Les livres sont classés selon leurs orientations idéologiques (libéral, conservateur, neutre).

Football (College Football) Réseau des rencontres entre équipes universitaires de football américain (115 équipes). Les communautés reflètent généralement les conférences sportives

auxquelles les équipes appartiennent.

EU-Core : Réseau d'échanges d'e-mails entre employés d'une institution de recherche européenne. Chaque nœud représente un employé, et une arête symbolise un échange d'e-mail. Ce réseau est plus dense et complexe que les précédents.

AS (Autonomous Systems) : Représente l'infrastructure d'Internet via les connexions entre systèmes autonomes (AS). Chaque nœud est un AS, et les arêtes indiquent les liens de communication entre eux. Ce jeu de données est le plus vaste et complexe de l'étude.

3.4.4 Approches

3.4.4.1 Louvain

méthode Louvain est une approche pour la détection de communautés de manière efficace. Voici les hyperparamètres : Ces hyperparamètres permettent de contrôler le comportement

Paramètre	**Valeur par défaut**	**Description**
max_iterations	10	Nombre maximal d'itérations de l'optimisation de la modularité par niveau.
Optimizer	ADAM	pour l'optimisation du modèle Keras
Epochs	3	pour l'entraînement du modèle Keras
Batch Size	8	pour le traitement par lots dans l'entraînement Keras

TABLE 3.1 – Paramètres du louvaine

de l'algorithme Louvain, influençant la taille des communautés détectées, la précision de la convergence et la performance de l'exécution.

3.4.4.2 INFOMAP

méthode INFOMAP est une approche pour la détection de communautés de manière efficace. Voici les hyperparamètres :

Paramètre	Valeur par défaut	Description
max_iterations	10	Spécifie le nombre maximal d'itérations que l'algorithme doit effectuer.
Optimizer	ADAM	Définit l'algorithme d'optimisation utilisé pour ajuster les poids et les paramètres du modèle.
tolerance	1e-4	Définit la tolérance à l'erreur ou à la variation dans les résultats.

TABLE 3.2 – Paramètres du infomap

Ces hyperparamètres permettent de contrôler le comportement de l'algorithme infomap influençant la taille des communautés détectées, la précision de la convergence et la performance de l'exécution .

3.4.4.3 WALKTRAP

méthode WALKTRAP est une approche pour la détection de communautés de manière efficace. Voici les hyperparamètres :

Paramètre	Valeur par défaut	Description
Fonction d'activation	relu	Fonction d'activation utilisée dans les couches du modèle Keras.
Optimizer	ADAM	Optimiseur utilisé pour l'entraînement du modèle Keras.
Batch size	8	Taille du batch pour l'entraînement dans Keras.
Epochs	3	Nombre d'époques pour entraîner le modèle Keras.

TABLE 3.3 – Paramètres du walktrap

Ces hyperparamètres permettent de contrôler le comportement de l'algorithme walktrap, influençant la taille des communautés détectées, la précision de la convergence et la performance de l'exécution.

3.4.4.4 Leiden

méthode Leiden est une approche pour la détection de communautés de manière efficace. Voici les hyperparamètres :

Paramètre	Valeur par défaut	Description
resolution_parameter	1.0	Granularité des communautés.
n_iterations	2	Nombre d'itérations d'optimisation.
seed	None	Graine aléatoire.
partition_type	Modularity	Type de partition (ex : ModularityVertexPartition).

TABLE 3.4 – Paramètres de LIEDEN

Ces hyperparamètres permettent de contrôler le comportement de l'algorithme leiden influençant la taille des communautés détectées, la précision de la convergence et la performance de l'exécution.

3.4.4.5 Clauset-Newman-Moore et Label Propagation

Clauset-Newman-Moore et Label Propagation n'ont pas d'hyperparamètres car ce sont des algorithmes heuristiques simples conçus pour fonctionner sans réglages externes. Clauset-Newman-Moore est une méthode gloutonne qui fusionne automatiquement les communautés pour maximiser la modularité, sans contrôle sur la granularité ou le nombre de communautés. Label Propagation fonctionne par itérations où chaque nœud adopte le label majoritaire de ses voisins, dépendant uniquement de l'ordre de mise à jour des nœuds, sans paramètre réglable. Cette simplicité rend ces algorithmes rapides et faciles à utiliser, mais limite la possibilité de personnaliser ou d'ajuster finement les résultats.

3.5 Entraînement et validation

Après avoir appliqué plusieurs algorithmes de détection de communautés en apprentissage non supervisé sur l'ensemble des graphes, vous avez utilisé plusieurs métriques pour évaluer la qualité des partitions : NMI, ARI et modularité. Ces indices permettent de mesurer la similarité entre la partition détectée et la partition de référence (NMI, ARI), ainsi que la cohésion interne des communautés dans le graphe (modularité). Chaque algorithme attribue un label de communauté à chaque nœud du graphe, ce qui permet de comparer directement les classifications obtenues aux vérités terrains stockées dans les attributs des nœuds. Contrairement aux méthodes supervisées, la détection de communautés ne nécessite pas de division en ensembles d'entraînement et de validation, car elle s'appuie directement sur la structure complète du graphe et sur l'ensemble des nœuds. L'évaluation se fait donc sur la totalité des nœuds, ce qui permet d'estimer la robustesse et la pertinence des différentes méthodes comparées.

3.6 Résultats et discussion

Dans ce travail, nous avons appliqué six algorithmes de détection de communautés sur plusieurs jeux de données afin d'évaluer leurs performances. Les algorithmes Louvain, Leiden et Infomap se sont distingués par des scores élevés en modularité, NMI et ARI, indiquant une bonne qualité de partition et une forte correspondance avec les communautés de référence. Les méthodes

Clauset-Newman-Moore et Label Propagation ont également produit des résultats intéressants, avec une détection variable du nombre de communautés selon les graphes. L'étude des nœuds individuellement a permis de mieux comprendre comment certains nœuds, notamment ceux situés aux frontières des communautés, influencent la qualité globale des partitions. Ces résultats montrent que le choix de l'algorithme dépend du compromis recherché entre précision, qualité des communautés et efficacité de calcul, et soulignent l'importance d'une évaluation à la fois globale et locale pour une analyse complète.

3.7 comparaison et segmentation résultats

3.7.1 Modularité

TABLE 3.5 – comparaison résultats Modularité

Méthode	Karate	Dolphins	Polbooks	Football	EU-Core	AS	Total
Louvain	0.4151	0.5196	0.5270	0.6042	0.4333	0.6303	0.5216
Infomap	0.4020	0.4941	0.5262	0.5934	0.4247	0.5728	0.5022
Walktrap	0.3532	0.4888	0.5070	0.6029	0.3719	0.5579	0.4803
Leiden	0.4198	0.5241	0.5269	0.6046	0.4301	0.6449	0.5251
Clauset-Newman-Moore	0.3807	0.4955	0.5020	0.5497	0.3665	0.5899	0.4807
Label Propagation	0.2797	0.1121	0.4974	0.4811	0.5831	0.0888	0.3525

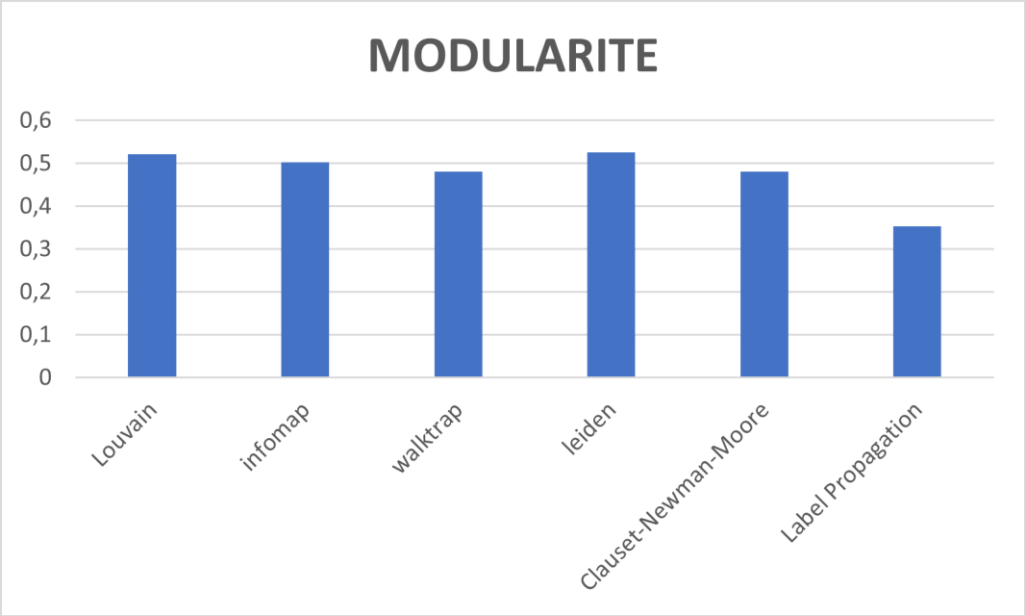

FIGURE 3.1 – Segmentation modularité

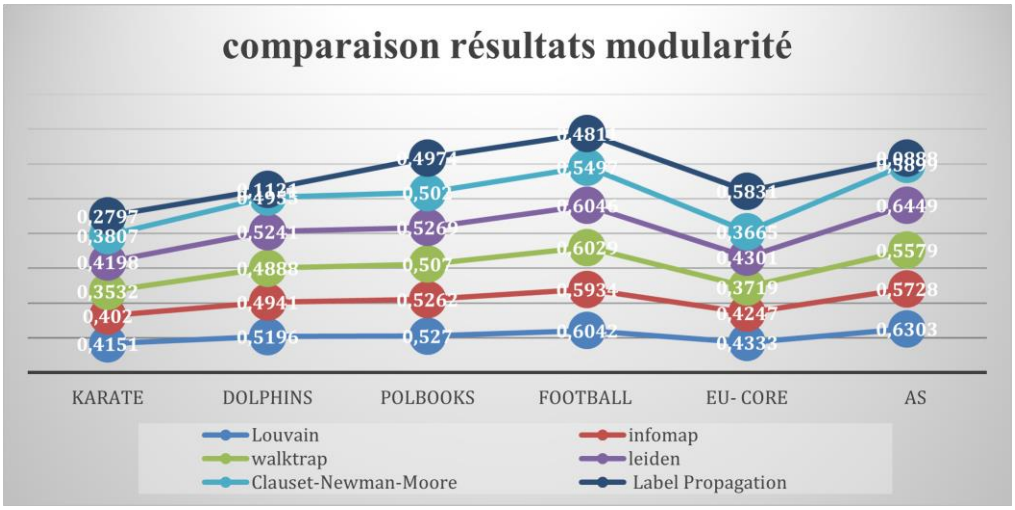

FIGURE 3.2 – comparaison de résultats Modularité

l'algorithme Leiden obtient les meilleurs résultats globaux, suivi de près par Louvain, qui reste une solution robuste et performante sur la majorité des graphes, notamment les plus grands comme as et football. Infomap se situe juste en dessous, offrant des partitions de qualité correcte mais légèrement moins bonnes en modularité. Walktrap et Clauset-Newman-Moore affichent des performances plus modestes, avec des scores plus faibles sur plusieurs réseaux. Enfin, Label Propagation présente des résultats très faibles et instables, ce qui limite son intérêt dans ce contexte. Ainsi, Leiden et Louvain s'imposent comme les algorithmes les plus adaptés pour maximiser la modularité des partitions.

3.7.2 ARI

TABLE 3.6 – comparaison résultats ARI

Méthode	**Karate**	**Dolphins**	**Polbooks**	**Football**	**EU-Core**	**AS**	**Total**
Louvain	0.5998	0.2708	0.6605	0.7041	0.3416	0.2822	0.4765
Infomap	0.7022	0.4254	0.6649	0.7685	0.2836	0.2951	0.5233
Walktrap	0.3331	0.4167	0.6534	0.8154	0.1975	0.2621	0.4464
Leiden	0.5414	0.3329	0.6752	0.8069	0.3656	0.2002	0.4870
Clauset-Newman-Moore	0.6803	0.4509	0.6379	0.4741	0.1695	0.1842	0.4328
Label Propagation	0.0879	0.0879	0.2638	0.5942	0.7510	0.0917	0.3128

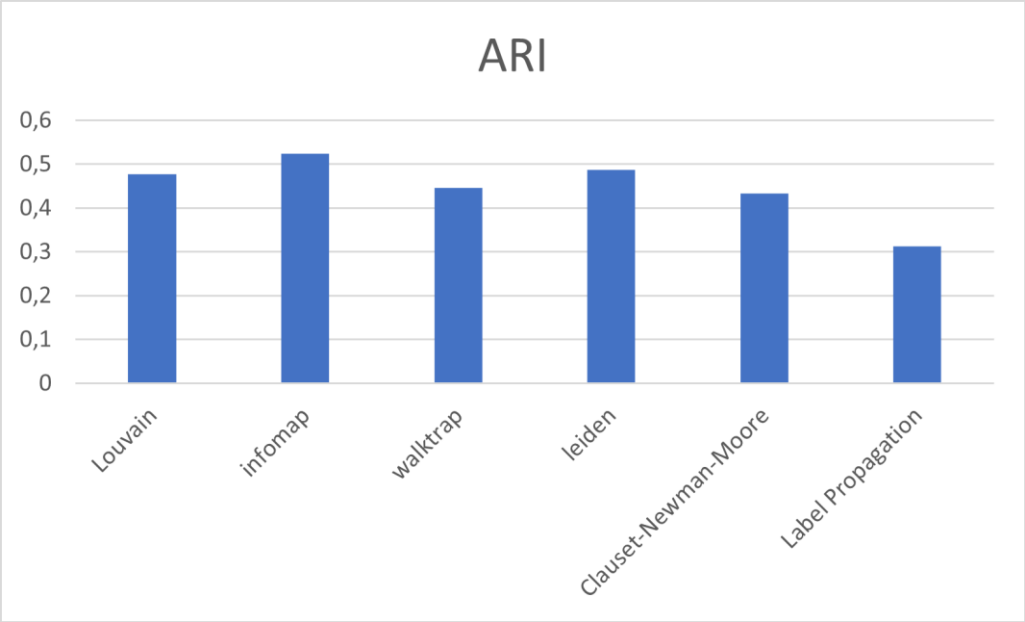

FIGURE 3.3 – Segmentation avec ARI

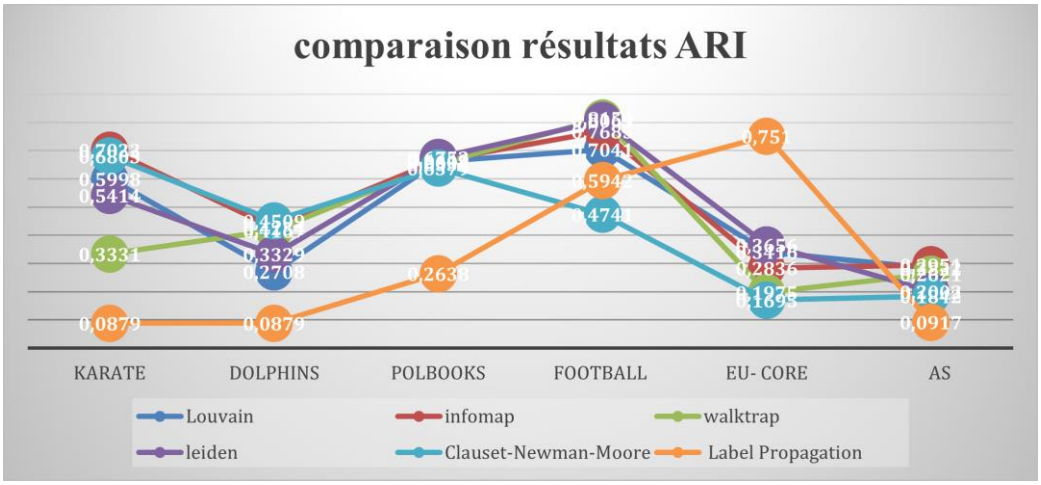

FIGURE 3.4 – comparaison de résultats ARI

Selon la métrique ARI, l'algorithme Infomap obtient les meilleurs résultats globaux, suivi par Leiden et Louvain, qui affichent également de bonnes performances. Infomap se distingue notamment sur les graphes football, polbooks et dolphins. En revanche, Walktrap et Clauset-Newman-Moore présentent des scores plus modestes, tandis que Label Propagation reste instable, avec des résultats très faibles sur la majorité des graphes, malgré un pic de performance sur eu-core. Globalement, Infomap s'impose comme le plus efficace selon cette métrique.

TABLE 3.7 – comparaison résultats NMI

Méthode	Karate	Dolphins	Polbooks	Football	EU-Core	AS	Total
Louvain	0.7071	0.4743	0.5559	0.8506	0.5959	0.4915	0.6126
Infomap	0.6995	0.5674	0.5537	0.8905	0.6212	0.4356	0.6280
Walktrap	0.5042	0.5373	0.5427	0.8874	0.5804	0.5117	0.5939
Leiden	0.6873	0.5124	0.5737	0.8903	0.5967	0.4859	0.6244
Clauset-Newman-Moore	0.6925	0.5727	0.5308	0.6977	0.4753	0.4430	0.5686
Label Propagation	0.2075	0.2075	0.4790	0.5341	0.8697	0.1797	0.4129

3.7.3 Nmi

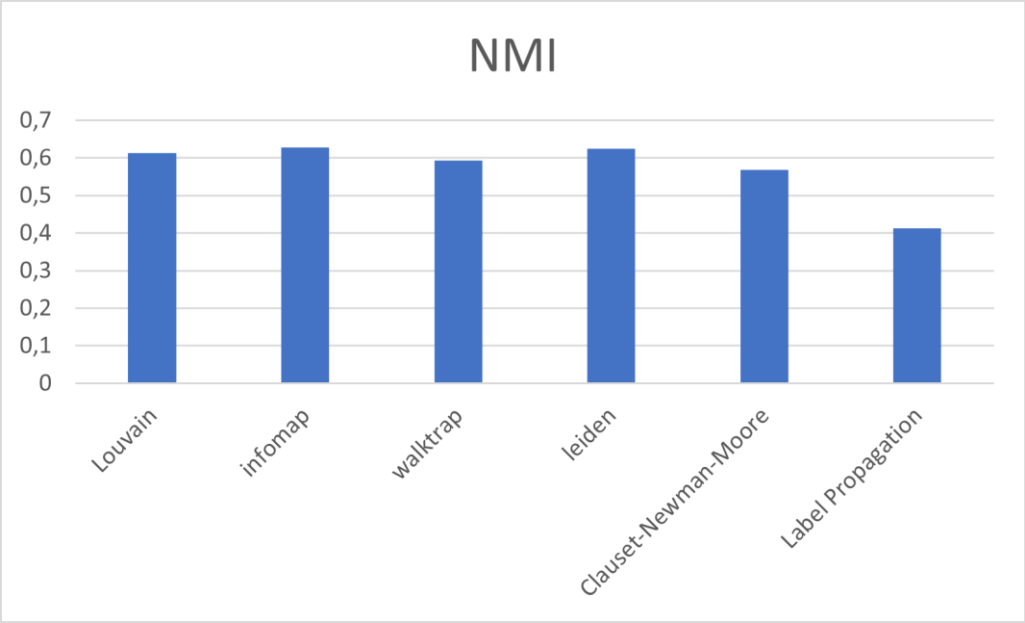

FIGURE 3.5 – Segmentation avec NMI

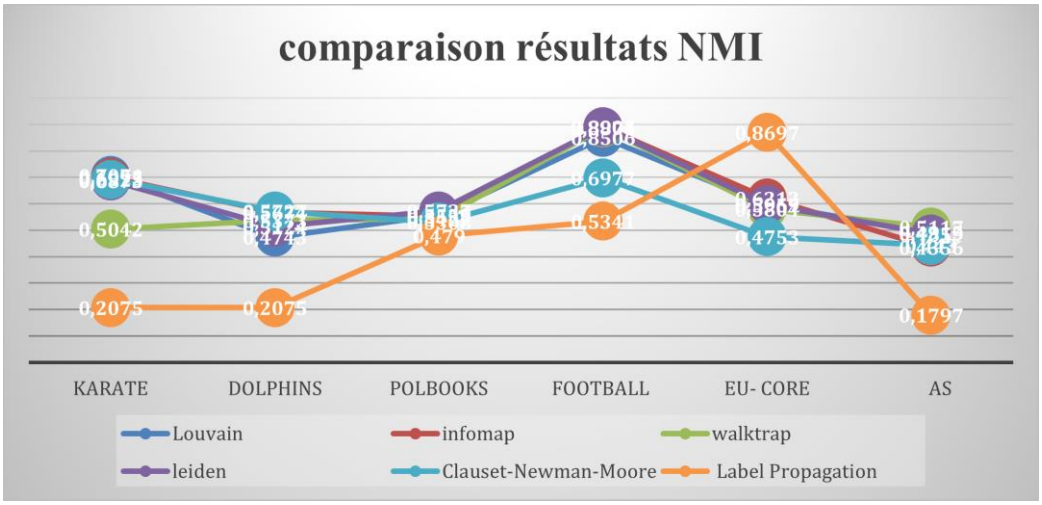

FIGURE 3.6 – comparaison de résultats NMI

Selon la métrique NMI, l'algorithme Infomap se positionne en tête avec la meilleure perfor-

mance globale (0.6280), suivi de près par Leiden (0.6244) et Louvain (0.6126), tous trois offrant d'excellents résultats, notamment sur les graphes football et eu-core. Walktrap obtient également un score satisfaisant, tandis que Clauset-Newman-Moore reste légèrement en retrait. En revanche, Label Propagation affiche une performance très instable, avec un score global nettement inférieur (0.4129) malgré un pic élevé sur eu-core. Ainsi, Infomap, Leiden et Louvain apparaissent comme les algorithmes les plus fiables selon cette métrique.

3.8 CONCLUSION

Ce chapitre a présenté une analyse approfondie de plusieurs algorithmes de détection de communautés appliqués à divers réseaux complexes. En utilisant des ensembles de données variés (réseaux sociaux, biologiques, techniques, etc.), nous avons évalué six méthodes — Louvain, Leiden, Infomap, Walktrap, Clauset-Newman-Moore et Label Propagation — à l'aide de métriques telles que la modularité, l'ARI et le NMI. Les résultats montrent que Leiden et Louvain offrent les meilleures performances globales en termes de modularité, tandis que Infomap se distingue selon l'ARI et le NMI. Walktrap et Clauset-Newman-Moore affichent des résultats intermédiaires, et Label Propagation se révèle instable et peu performant. L'étude souligne l'importance de choisir un algorithme adapté aux caractéristiques du réseau et au critère d'évaluation privilégié. Ce travail met en lumière la complémentarité des approches, la diversité des graphes étudiés et la nécessité d'une évaluation fine, à la fois globale et locale, pour une détection de communautés efficace.

Conclusion générale

Ce mémoire a porté sur la détection de communautés dans les réseaux complexes, un enjeu fondamental pour comprendre la structure et les dynamiques des systèmes interconnectés. Nous avons mené une étude comparative approfondie de six algorithmes non supervisés majeurs : Louvain, Infomap, Walktrap, Leiden, Clauset-Newman-Moore et Label Propagation.

Louvain se distingue par sa rapidité d'exécution et son efficacité à maximiser la modularité, ce qui en fait un choix privilégié pour l'analyse de grands réseaux. Cependant, il peut parfois produire des communautés peu cohérentes. Leiden, une amélioration récente de Louvain, corrige ces faiblesses en garantissant des communautés bien connectées et en conservant une excellente scalabilité.

Infomap repose sur la théorie de l'information pour détecter des structures communautaires fines, offrant ainsi une grande précision dans la détection des groupes cohérents, notamment dans des réseaux où la modularité n'est pas optimale. Walktrap utilise des marches aléatoires pour capturer la proximité des nœuds et s'avère particulièrement efficace pour produire des partitions stables et proches des références, comme l'indiquent les scores élevés en ARI (Adjusted Rand Index) et NMI (Normalized Mutual Information).

L'algorithme Clauset-Newman-Moore exploite une approche hiérarchique adaptée aux très grands graphes. Bien qu'il soit très efficace en termes de complexité, sa précision est parfois moindre, ce qui le rend plus adapté à des analyses exploratoires à grande échelle. Enfin, Label Propagation se caractérise par sa rapidité exceptionnelle et son absence de paramètre, mais il peut souffrir d'instabilités et de résultats variables selon la topologie du réseau.

Malgré leurs performances, ces méthodes font face à des défis majeurs. La scalabilité demeure un enjeu critique avec l'augmentation exponentielle des données. La simplification nécessaire des graphes peut entraîner une perte d'information importante, affectant la qualité des partitions. De plus, l'évaluation de la qualité des communautés est compliquée par l'absence fréquente de vérités terrain dans les données réelles.

Ces limites ouvrent de nombreuses pistes de recherche prometteuses. L'adoption d'algorithmes récents comme Leiden, l'hybridation d'approches traditionnelles avec des techniques d'apprentissage automatique (clustering avancé, réseaux de neurones sur graphes) peuvent améliorer la robustesse et la précision. L'intégration de métriques multi-critères, combinant modularité, stabilité et cohérence topologique, pourrait permettre une évaluation plus complète. Le développement d'outils interactifs pour la visualisation et l'analyse dynamique des communautés est également essentiel. Enfin, l'adaptation de ces méthodes à des réseaux évolutifs, la gestion de données hétérogènes (attributs, temporalité) et la création de benchmarks standardisés constituent des objectifs clés pour rendre ces techniques opérationnelles dans des domaines sensibles tels que la cybersécurité, la médecine personnalisée ou la gestion d'infrastructures critiques.

Bibliographie

[1] Sofiane Ben Amor. Percolation, prétopologie et multialéatoires, contributions à la modélisation des systèmes complexes : exemple du contrôle aérien. Thèse de doctorat, École Pratique des Hautes Études (EPHE), Paris, 2008.

[2] M. E. J. Newman. The Structure and Function of Complex Networks. *SIAM Review*, 45(2) :167–256, 2003.

[3] Alain Degenne and Michel Forsé. Les réseaux sociaux. Une analyse structurale en sociologie. Armand Colin, Paris, 1994.

[4] Stanley Milgram. The Small World Problem. *Psychology Today*, 1(1) :61–67, 1967.

[5] Mark Newman, Albert-László Barabási, and Duncan J. Watts. The Structure and Dynamics of Networks. Princeton University Press, 2006.

[6] Annick Lesne and Michel Laguës. Chapitre 3 : L'universalité comme conséquence de l'invariance d'échelle. In *Invariance d'échelle - Des changements d'états à la turbulence*, Belin, pages 58–105, Paris, 2003.

[7] Steven H. Strogatz. Exploring Complex Networks. *Nature*, 410(6825) :268–276, 2001.

[8] Pierre Lopez. Graphes. 2005.

[9] Colin Cooper and Alan Frieze. The Cover Time of the Giant Component of a Random Graph. 2006.

[10] Albert-László Barabási and Réka Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439) :509–512, 1999.

[11] Duncan J. Watts and Steven H. Strogatz. Collective Dynamics of 'Small-World' Networks. *Nature*, 393(6684) :440–442, 1998.

[12] Bo Söderberg. A General Formalism for Inhomogeneous Random Graphs. *Physical Review E*, 66 :066121, 2002.

[13] Geoffrey Canright, Andreas Deutsch, Mark Jelasity, and Frederick Ducatelle. Structures and Functions of Dynamic Networks. PhD thesis, 2004.

[14] Santo Fortunato. Community Detection in Graphs. *Physics Reports*, 486(3-5) :75–174, 2010.

[15] Nassira Lograda. La détection de communautés dans les réseaux sociaux. Thèse de doctorat, 2019.

[16] Nedioui Med Abdelhamid. Fouille et apprentissage automatique dans les réseaux sociaux dynamiques. Mém. de mast., 2015.

[17] Mustapha Merazka. Détection de communautés dans les réseaux sociaux. Thèse de doctorat, 2014.

[18] Rachid Djerbi. Détection de communautés dans les réseaux sociaux. Thèse de doctorat, 2021.

[19] Bolin Chen et al. Identifying Protein Complexes and Functional Modules—from Static PPI Networks to Dynamic PPI Networks. *Briefings in Bioinformatics*, 15(2) :177–194, 2014.

[20] Nadia Chouchani. Une approche de détection des communautés d'intérêt dans les réseaux sociaux : application à la génération d'IHM personnalisées. Thèse de doctorat, 2018.

[21] Muhammad Aqib Javed et al. Community Detection in Networks : A Multidisciplinary Review. *Journal of Network and Computer Applications*, 108 :87–111, 2018.

[22] Mark E. J. Newman and Michelle Girvan. Finding and Evaluating Community Structure in Networks. *Physical Review E*, 69(2) :026113, 2004.

[23] Olivier Gach. Algorithmes mémétiques de détection de communautés dans les réseaux complexes : techniques palliatives de la limite de résolution. Thèse de doctorat, 2013.

[24] Vincent D. Blondel et al. Fast Unfolding of Communities in Large Networks. *Journal of Statistical Mechanics : Theory and Experiment*, 2008(10) :P10008, 2008.

[25] Maël Canu. Détection de communautés orientée sommet pour des réseaux mobiles opportunistes sociaux. Thèse de doctorat, 2017.

[26] Michel Crampes and Michel Plantié. Partition et recouvrement de communautés dans les graphes bipartis, unipartis et orientés. In *24èmes Journées francophones d'Ingénierie des Connaissances*, 2013.

[27] Martin Rosvall and Carl T. Bergstrom. Maps of Random Walks on Complex Networks Reveal Community Structure. *Proceedings of the National Academy of Sciences*, 105(4) :1118–1123, 2008.

[28] Huan Li et al. LPA-MNI : An Improved Label Propagation Algorithm Based on Modularity and Node Importance for Community Detection. *Entropy*, 23(5) :497, 2021.

[29] Yan Yuan et al. An Influence Maximisation Algorithm Based on Community Detection. *International Journal of Computational Science and Engineering*, 22(1) :1–14, 2020.

[30] Leon Danon et al. Comparing Community Structure Identification. *Journal of Statistical Mechanics : Theory and Experiment*, 2005(9) :P09008, 2005.

[31] Lawrence Hubert and Phipps Arabie. Comparing Partitions. *Journal of Classification*, 2(1) :193–218, 1985.

[32] David Combe. Détection de communautés dans les réseaux d'information utilisant liens et attributs. Thèse de doctorat, 2013.

[33] Jinfang Sheng et al. Research on Community Detection in Complex Networks Based on Internode Attraction. *Entropy*, 22(12) :1383, 2020.

[34] Jean-Francis Roy. Apprentissage automatique avec garanties de généralisation à l'aide de méthodes d'ensemble maximisant le désaccord. Thèse de doctorat, 2018.

[35] Alassane Samba. *Science des données au service des réseaux d'opérateur : proposition de cas d'utilisation, d'outils et de moyens de déploiement*. Thèse de doctorat, octobre 2018.

[36] Pádraig Cunningham, Matthieu Cord, et Sarah Jane Delany. *Supervised learning*. In : Machine Learning.

[37] Erik G. Learned-Miller. *Introduction to supervised learning*. Department of Computer Science, University of Massachusetts.

[38] Bing Liu. *Supervised learning*. In : Web Data Mining. Springer, 2011, p. 63–132.

[39] Mohamed Alloghani et al. *A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science*. Janvier 2020, p. 3–21. ISBN : 978-3-030-22474-5. DOI : 10.1007/978-3-030-22475-2_1.

[40] Giuseppe Bonaccorso. *Machine Learning Algorithms*. Packt Publishing Ltd, 2017.

[41] Alex M. Andrew. *Reinforcement Learning : An Introduction* par Richard S. Sutton et Andrew G. Barto, Adaptive Computation and Machine.

[42] Leslie Pack Kaelbling, Michael L. Littman et Andrew W. Moore. *Reinforcement Learning : A Survey*. Journal of Artificial Intelligence Research, vol. 4 (1996), p. 237–285.

[43] L. Li et al. *Community detection algorithm based on local expansion k-means*. Neural Network World, vol. 26, no. 6 (2016), p. 589.

[44] Ekta Gujral et al. *HACD : Hierarchical Agglomerative Community Detection in Social Networks*. In : 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP). IEEE, 2019, p. 1–6.

[45] Huma Aftab et al. *Hybrid DBSCAN based Community Detection for Edge Caching in Social Media Applications*. Juillet 2021. DOI : 10.1109/IWCMC51323.2021.9498609.

[46] Michael Molloy et Bruce Reed. *A critical point for random graphs with a given degree sequence*. Random Structures and Algorithms, vol. 6, p. 161–180, 1995.

[47] (s.d.). *Graphe de réseau montrant les connexions entre diverses personnes* [Image]. Pngsucai. https://www.pngsucai.com/png/907304.html

[48] A. Naumowicz. *A Note on the Seven Bridges of Königsberg Problem*. Formalized Mathematics, vol. 22, no. 2 (2014), p. 177–178. https://doi.org/10.2478/forma-2014-0018

[49] (s.d.). *Graphe de réseau montrant les connexions entre diverses personnes* [Image]. Pngsucai. https://www.pngsucai.com/png/907304.html

[50] M. Mokhtari et A. Ghasemi. *From left to right : top a 1, 2, 3, 4-plex ; Bottom a 4-plex and a 3-plex with*. Journal of Structural Engineering, vol. 144, no. 7 (2018), 04018123. https://doi.org/10.1061/(ASCE)ST.1943-541X.0002075

[51] J. Smith. vol. 22, no. 2 (2014), p. 177–178

[52] Stanley Milgram. *The Small World Problem*. Psychology Today, vol. 1, no. 1 (1967), p. 61–67. http://snap.stanford.edu/class/cs224w-readings/milgram67smallworld.pdf

[53] D. S. Bassett. *Network Analysis I II*. Présentation, été 2011. Brain Mapping.

[54] Martin Rosvall et Carl T. Bergstrom. *Higher-order network science*. Nature Physics, vol. 18 (2022), p. 394–398. https://doi.org/10.1038/s41567-021-01432-1

[55] A. Ould Mohamed Moctar et I. Sarr. *Détection de communautés statiques et dynamiques*. Revue, 2016.

[56] Vincent A. Traag, Ludo Waltman et Nees Jan van Eck. *From Louvain to Leiden : guaranteeing well-connected communities*. Scientific Reports, vol. 9, no. 1 (2019), 5233.