République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed El Bachir El Ibrahimi B.B.A.
Faculté de Mathématique et Informatique

Département d'Informatique

Mémoire

En vue de l'obtention du Diplôme de Master

Domaine : Mathématique et Informatique
Filière : Informatique
Spécialité : technologies de l'information et de la communication (TIC)

Thème

L'Apprentissage Automatique Pour La Prédiction De Lien Dans Les Réseaux Complexes

Représenté par :
BOUABDALLAH Maroua **DRIAI Ibtissem**

Devant le jury :
Président : Dr.CHARIKHI MOURAD
Examinateur : Dr.BELAZOUG MOUHOUB
Encadrant : Dr.SAIFI Abdelhamid

Année universitaire : 2023/2024

Remercîment

Je remercie Dieu de m'avoir accordé la santé, le bien-être, ainsi que la capacité et le courage d'accomplir ce travail.

J'adresse mes sincères remerciements à mon encadrant Dr**.Saifi Abdelhamid**, pour son soutien inestimable, ses conseils avisés et son assistance tout au long de ce travail, ainsi que pour le temps qu'il m'a consacré. Son aide a été déterminante dans le succès de ce projet.

Je tiens à remercier les membres du jury Dr.CHARIKHI MOURAD Dr.BELAZOUG MOUHOUB pour leur temps, leur expertise et leurs commentaires précieux et utiles sur ce projet. Merci beaucoup pour votre soutien et vos conseils.

Merci beaucoup à ma famille pour son soutien et sa patience à mon égard pendant cette période difficile. Ils m'ont apporté leur amour et leur soutien, et je n'oublie pas mes amis et mes proches qui m'ont toujours soutenu et encouragé.

Merci à toutes et à tous.

Dédicace

Je tiens c'est avec grande plaisir que Je dédie ce modeste travail

A tous ceux qui me connaissent, en particulier :

A l'être la plus cher de ma vie, mon père.

A celui qui m'a fait de moi une femme, la plus belle mère du monde ,ma mère

A mes chères frères et sœurs Et leurs enfants

Et ma deuxième famille Belabbass

A tous mes amis et collègues.

À mes professeurs et mentors, merci pour vos conseils précieux et votre encouragement constant. Que ce travail soit le reflet de votre inspiration et de votre dévouement.

Avec gratitude et respect

Maroua

Dédicace

Je tiens c'est avec grande plaisir que Je dédie ce modeste travail

A tous ceux qui me connaissent, en particulier :

A l'être la plus cher de ma vie, mon père.

A celui qui m'a fait de moi une femme , la plus belle mère du monde ,ma mère

A mes chères frères et sœurs Et leurs enfants

Et ma deuxième famille Zemmit

A tous mes amis et collègues.

À mes professeurs et mentors, merci pour vos conseils précieux et votre encouragement constant. Que ce travail soit le reflet de votre inspiration et de votre dévouement.

Avec gratitude et respect

Ibtissem

RESUME

Ce travail explore les concepts de la théorie des graphes pour modéliser et analyser les réseaux complexes en mettant l'accent sur l'utilisation du Machine Learning Les méthodes examinées incluent des mesures de similarité basées sur les voisins communs, des mesures basées sur la longueur des chemins, Nous avons également évalué l'efficacité de différents algorithmes de classification, tels que le Support Vector Machine (SVM), le K-Nearest Neighbors (KNN)…Nos résultats montrent que certaines combinaisons de ces méthodes et algorithmes permettent d'obtenir des prédictions précises des classes de liens dans les réseaux complexes, ouvrant ainsi de nouvelles perspectives pour leur analyse et leur application dans divers domaines

Mots-clés : prédiction de lien, Algorithmes de classification, Réseaux complexes

Abstract

This work explores graph theory concepts to model and analyze complex networks with an emphasis on the use of machine learning. Methods examined include similarity measures based on common neighbors, measures based on the length of paths, We also evaluated the effectiveness of different classification algorithms, such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN)…Our results show that certain combinations of these methods and algorithms make it possible to obtain accurate predictions of link classes in complex networks, thus opening new perspectives for their analysis and application in various fields

Keywords: link prediction, Classification algorithms, Complex networks

تلخيص

يستكشف هذا العمل مفاهيم نظرية الرسم البياني لنمذجة وتحليل الشبكات المعقدة مع التركيز على استخدام التعلم الآلي، وتشمل الأساليب التي تم فحصها مقاييس التشابه بناءً على الجيران المشتركين، والمقاييس المستندة إلى طول المسارات، وقمنا أيضًا بتقييم فعالية خوارزميات التصنيف المختلفة. مثل(Support Vector Machine (SVM ، و K-Nearest Neighbors (KNN)).تظهر نتائجنا أن مجموعات معينة من هذه الأساليب والخوارزميات تجعل من الممكن الحصول على تنبؤات دقيقة لفئات الارتباط في الشبكات المعقدة، وبالتالي فتح آفاق جديدة لتحليلها وتطبيقها في مختلف المجالات

الكلمات المفتاحية: التنبؤ بالارتباط، خوارزميات التصنيف، الشبكات المعقدة

SOMMAIRE

Liste des tableaux

Liste des Figures

INTRODUCTION GENERALE

INTRODUCTION GENERALE

Le travail de recherche porte principalement l'étude des réseaux complexes est un domaine de recherche scientifique émergent et actif largement inspiré par les résultats expérimentaux de réseaux du monde réel. Des chercheurs de divers domaines scientifiques ont utilisé des réseaux complexes pour modéliser des systèmes complexes, qui sont caractérisés par leurs structures topologiques complexes et souvent peu communes dans des réseaux plus simples, se retrouvent dans une multitude de domaines, notamment les réseaux sociaux, biologiques, technologiques et d'information. On prendre le réseau social comme un exemple le réseau social (un terme plus général est un réseau complexe) une approche standard pour modéliser la communication dans un groupe ou une communauté de personnes ,les relations entre les individus changent continuellement alors La compréhension et l'analyse de ces réseaux revêtent une importance cruciale on peut représenter comme un modèle graphique dans lequel un nœud correspond à une personne ou à un réseau social cela nous prendre à la théorie des graphes qui est fondamentale pour appréhender ces systèmes, car elle offre des outils et des concepts permettant de décrire et d'analyser leurs propriétés structurelles et comportementales , elle fournit un cadre conceptuel puissant pour modéliser et analyser ces réseaux, en les représentant sous forme de graphes où les nœuds représentent les entités du système étudié et les arêtes représentent les relations entre ces entités ,on peuvent identifier des schémas et des tendances dans la structure du réseau pour L'objectif principal est de prédire les liens potentiels entre les nœuds du réseau qui n'existent pas encore en basées sur leurs propriétés et les liens actuellement observés les motifs et les caractéristiques topologiques .

La prédiction des liens dans les réseaux complexes ayant des applications pratiques et théoriques dans de nombreux domaines, notamment la sociologie, la biologie, et l'ingénierie des réseaux. Dans les réseaux sociaux en ligne, par exemple, elle permet de recommander de nouveaux amis ou contacts, améliorant ainsi l'expérience utilisateur et la rétention sur la plateforme. Dans les réseaux biologiques, elle aide à identifier de nouvelles interactions protéiques ou génétiques, offrant des informations précieuses pour comprendre les processus biologiques fondamentaux et développer des traitements médicaux.

Motivation

La prédiction des liens constitue un défi majeur en raison de la complexité des réseaux et de leur dynamique volatile. Par exemple, les réseaux sociaux évoluent constamment avec de nouveaux utilisateurs et des interactions changeantes, ce qui rend difficile la prévision précise des futures connexions entre les nœuds du réseau. Premièrement, le processus d'obtention de

toutes les relations existantes entre toutes les paires de nœuds possibles n'est pas facile à réaliser et entraîne généralement la perte de nombreux liens dans le réseau. Deuxièmement, il existe des paires d'individus dans le réseau qui ne sont pas connectés au moment de l'acquisition des données mais qui sont très susceptibles de se connecter dans un avenir proche. La motivation pour résoudre ce problème réside dans la nécessité de développer des méthodes et des techniques efficaces pour prédire les liens dans réseaux complexes. Cela prendrait en charge des applications pratiques telles que la recommandation de contenu, la découverte de médicaments et l'optimisation du réseau de transport. En surmontant les défis liés à la complexité et à la dynamique des réseaux, nous pouvons améliorer notre compréhension des systèmes complexes et ouvrir de nouveaux horizons pour l'innovation et le progrès dans de nombreux domaines.

Contribution

L'importance de la prédiction de liens a inspiré de nombreux chercheurs de diverses disciplines scientifiques à développer de nouveaux algorithmes de prédiction de liens qui peuvent non seulement être appliqués au problème spécifique pour lequel ils ont été développés, mais peuvent également être généralisés aux réseaux obtenus dans d'autres domaines. Dans ce contexte, notre projet se concentre sur l'utilisation de l'apprentissage automatique pour la prédiction de liens dans les réseaux complexes.

Nous examinons l'efficacité de l'apprentissage automatique dans la résolution du problème de prédiction de liens grâce à sa capacité à apprendre à partir de données étiquetées. Nous avons choisi d'explorer plusieurs approches supervisées, dans le but de sélectionner les modèles (algorithmes) les plus efficaces pour ce travail basé sur les concepts de la théorie des graphes.

Nous avons utilisé une base de données de 5 réseaux complexes, couvrant une variété de domaines. Ces réseaux ont plusieurs variables telles que le nombre de nœuds, le nombre de liens, le degré moyen, le coefficient de regroupement, la longueur moyenne du chemin le plus court et d'autres propriétés structurelles. Après avoir préparé la base de données, nous avons effectué l'évaluation en utilisant plusieurs méthodes d'apprentissage et algorithmes de classification (KNN, SVM, AdaBoost, et Random Forest) et différentes mesures de similarité basées sur les voisins communs, des mesures basées sur la longueur des chemins chacune ayant des caractéristiques uniques qui les rendent adaptées à différentes configurations de réseaux et types de données. La performance des modèles et des algorithmes sera évaluée à l'aide d'un ensemble de critères, tels que (Précision, Rappel, F1-Score) .

Après avoir analysé les résultats, nous serons en mesure d'identifier les modèles les plus efficaces et les meilleures métriques de similarité pour contribuer à améliorer le processus de prédiction des liens dans les réseaux complexes.

Organisation de la thèse

Organiser le travail aide à structurer les idées et à atteindre les objectifs, Notre travail a donc été divisé en trois chapitres.

Le premier chapitre comprend tous les concepts de base des réseaux complexes, ainsi que leurs types et concepts liés à la théorie des graphes.

Le deuxième chapitre, nous nous sommes concentrés sur la prédiction de liens et les différentes méthodes utilisées pour analyser les réseaux complexes afin de mieux comprendre leur structure et leur comportement. Nous avons également abordé les algorithmes de classification tels que le KNN et SVM pour mieux comprendre le réseau et classer les différents éléments du réseau afin de comprendre le rôle de chaque élément au sein du réseau.

Nous conclurons notre rapport avec **un troisième chapitre** en fournissant des définitions des techniques utilisées pour tester et mettre en œuvre des expériences, et en les expliquant.

Enfin, nous terminons notre thèse par une **conclusion générale.**

Chapitre I

Introduction

Chapitre I : Introduction

I. 1 Introduction

Les réseaux complexes fait l'objet de plusieurs recherches qui ont été proposées pour découvrir la structure du réseau pour aident à comparer et choisir la stratégie la plus appropriée, d'autre part la théorie des graphes s'est développée substantiel de la connaissance pour devenus extrêmement utiles comme représentation d'une grande variété de réseaux dans différents secteurs dans la vie réelle.

I.2 Les réseaux complexes

Les réseaux complexes sont des systèmes composés de nombreux éléments interconnectés, dotés de propriétés émergentes qui ne peuvent être comprises uniquement par l'analyse de chaque élément séparément. Ils se distinguent par des structures non triviales, une forte interconnectivité et une dynamique évolutive.

Ces réseaux comprennent des nœuds (individus, groupes, organisations) et des connexions (liens, arêtes, relations) [1] qui peuvent représenter des interactions comme l'amitié, les accords économiques, les connexions Internet, les connexions neuronales ou les interactions protéiques.

Les graphes formés par ces réseaux sont souvent très complexes, avec les réseaux sociaux comme application la plus célèbre [2]. Cependant, leur étude a également suscité beaucoup de recherches dans divers domaines, notamment l'économie, les télécommunications, la biologie, l'intelligence artificielle, la bio-informatique, l'anthropologie, les sciences de l'information, la psychologie sociale et la sociolinguistique. [3]

Figure I-01 : Visualisation de réseaux complexes

I. 3Théorie des graphes

La théorie des graphes est une des pierres angulaires et la discipline mathématique et informatique qui étudie les graphes, lesquels sont des modèles abstraits de dessins de réseaux reliant des objets. Ces modèles sont constitués par la donnée de sommets (aussi appelés nœuds ou points, en référence aux polyèdres), et d'arêtes (aussi appelées liens ou lignes) entre ces sommets[6] ; ces arêtes sont parfois non symétriques (les graphes sont alors dits orientés) et sont alors appelées des flèches ou des arcs. Donc devenue une technique clé pour modéliser, analyser, simuler et comprendre ces topologies de réseaux complexes, aussi bien de manière statique que dynamique

I. 4 Définition et concepts de base des graphes

Comme l'indique le titre, nous nous intéressons aux théories des graphes et ses notions fondamentales relatives Nous allons présenter dans cette section brièvement.

I. 4.1 Définition d'un graphe

Le graphe est composé d'un ensemble de nœuds (aussi appelés sommets) reliés par paires par des arêtes. Les nœuds du graphe représentent des entités individuelles, tandis que les arêtes représentent les liens entre elles.[7]

Un graphe est défini un couple de nœuds G = (V, E) tel que :

- V (vertex en anglais) est un ensemble fini de sommets$V = \{v1, v2,\ldots,\}$

- E (Edge en anglais) est un ensemble d'arêtes liant certains couples de nœuds $E=\{e1,e2,\ldots,em\}$.

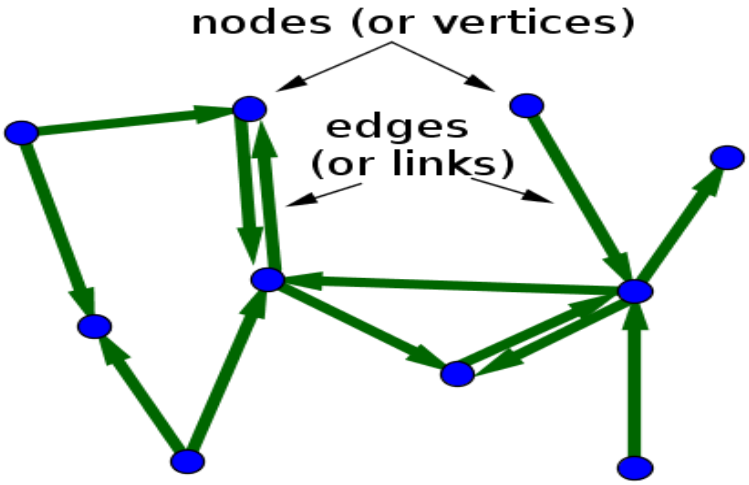


Figure I-02 : représentions général de graphe

- Le graphe peut être : **Graphe orienté** : Les arêtes ont une direction ou bien

Graphe non orienté : Les arêtes n'ont pas de direction

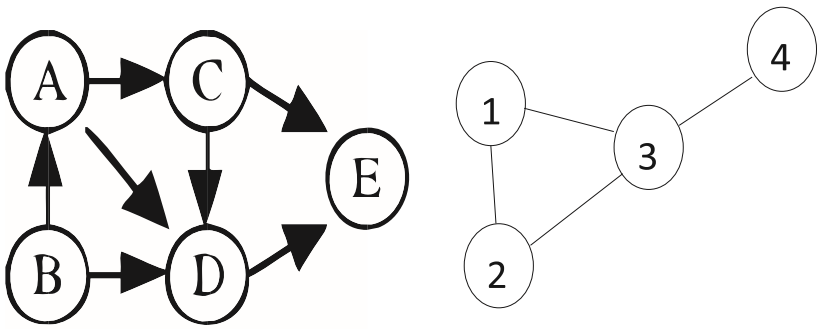


Figure I-03 : Graphe orienté et non-orient

I.4.2 Les types de graphes

il y a des différents types de graphes :

- Graphe Simple : est un graphe non orienté sans boucles (arêtes connectant un sommet à lui-même) et sans arêtes multiples entre deux sommets.

- Graphe Complet : est un graphe simple dans lequel chaque paire de sommets est connectée par une arête. Un graphe complet avec n sommets est noté Kn

- Graphe Biparti : est un graphe dont les sommets peuvent être divisés en deux ensembles disjoints U et V tels que chaque arête connecte un sommet de U à un sommet de V

- Graphe Pondéré : est un graphe dans lequel chaque arête est associée à un poids ou un coût. Les poids peuvent représenter des distances, des coûts, des capacités …[33]

I.4.3 Terminologie :

-**Nœud (ou sommet):** Un nœud représente un élément individuel dans un graphe. Il est noté par un symbole ou une étiquette

Sommets adjacents Deux sommets vi, vj ∈ V sont adjacents si l'arête {vi, vj} appartient à E. On dit aussi qu'ils sont voisins. L'ensemble des voisins de v se note ν(v)

-Lien (ou arête) : Une arête est une connexion entre deux nœuds dans un graphe.

Arêtes adjacentes : Deux arêtes sont adjacentes si elles ont au moins une extrémité en commun.

-L'ordre d'un graphe est le nombre de ses sommets.

-Une boucle est un arc ou une arête reliant un sommet à lui-même.

-Longueur d'un graphe : c'est le nombre d'arêtes d'un chemin.

-Le diamètre d'un réseau est la longueur du plus long chemin géodésique entre deux nœuds.

-**Un sous-graphe** est un graphe contenu dans un autre graphe, soit $G' \subset G$

-**Un cycle** est un circuit qui ne passe par chaque sommet qu'une seule fois, à l'exception du sommet de départ/arrivée.

I.4.4 Degré d'un graphe

Le degré d'un nœud **i** dans un graphe est le nombre de liens qu'il possède pour se connecter à d'autres nœuds. On désigne le degré d'un nœud **i** par **K**i. Les autres nœuds connectés à **i** sont appelés les voisins de ii ou le voisinage du nœud **i**. Pour un graphe non orienté comportant **n** nœuds [8]

I.4.5 Chemin d'un graphe

Un chemin dans un réseau est une suite de nœuds connectés par des liens, et sa longueur est le nombre de liens traversés. Les chemins peuvent repasser par les mêmes nœuds ou liens, sauf pour les chemins auto-évitants qui ne repassent jamais par le même nœud.

Le chemin plus court, est le chemin le plus direct entre deux nœuds, sans boucles, mais il peut y avoir plusieurs chemins possibles ou aucun.[9]

I.4.6 Densité d'un Graphe

La densité D d'un graphe simple est une mesure de combien d'arêtes (liens effectifs) sont présentes par rapport au nombre maximum possible d'arêtes.

Formule

$$D = \frac{2m}{n(n-1)}$$

- **m : Nombre d'arêtes présentes**
- **n : Nombre de nœuds**

I.4.7 La distance d'un graphe

La distance entre deux nœuds dans un réseau est le plus court chemin que quelqu'un doit emprunter pour passer d'un nœud à l'autre. La longueur moyenne du trajet est la moyenne de ces distances entre toutes les paires de nœuds, ce qui nous donne une idée de la facilité de transmission des informations sur de longues distances.[10]

Pour déterminer le nombre de chemins de longueur donnée entre deux nœuds, nous utilisons la matrice d'adjacence du graphe

I.4.8 Voisinage des nœuds dans les graphes

Le voisinage d'un nœud est l'ensemble des nœuds directement connectés à ce nœud par une arête. Donc, quand il existe un lien entre deux nœuds on les appelle des voisins ou adjacents.

Formule :

- **Voisinage de V :** $N(v) = \{ u \mid (v,u) \in E \}$

- **Degré de V** : $\deg |N(v)|$

N(v) : ensemble des nœuds

E : l'ensemble des arêtes du graphe

Exemple : Si le nœud A est connecté aux nœuds B et C, alors le voisinage de A est {B, C} et son degré est 2**.**

I.4.9 Connexité

Un graphe G = (X, U) est connexe si ∀i, j ∈ X, il existe une chaine entre i et j. On appelle composante connexe le sous-ensemble de sommets tels qu'il existe une chaine

entre deux sommets quelconques (Fig. 04). Un graphe est connexe s'il comporte une composante connexe maximale et une seule. Chaque composante connexe est un graphe connexe

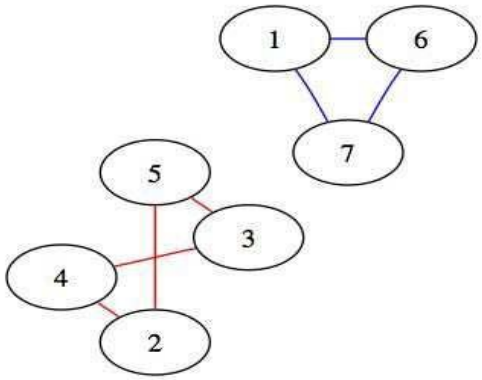


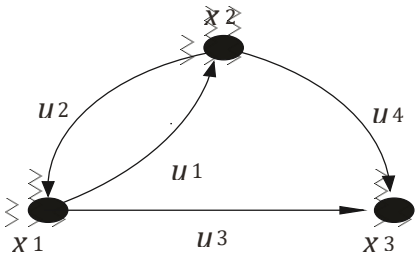
Figure I-04 : Exemple d'un Graphe connexe

I.5 Représentations d'un graphe

Il existe plusieurs façons de représenter un graphe, En particulier, elles ne sont pas ´équivalentes du point de vue de l'efficacité des algorithmes.[6] On distingue principalement la représentation par : matrice d'adjacence, liste d'adjacence

I.5.1 Matrice d'adjacence

Considérons un graphe. La matrice d'adjacence (Tab 01) fait correspondre les sommets origine des arcs (placés en ligne dans la matrice) aux sommets destination (placés en colonne). Dans le formalisme matrice booléenne, l'existence d'un arc (xi, xj) se traduit par la présence d'un 1 `à l'intersection de la ligne xi et de la colonne xj ; l'absence d'arc par la présence d'un 0 (dans un formalisme dit matrice aux arcs les éléments représentent le nom de l'arc)



	x_1	x_2	x_3	destination
x_1	0	1	1	
x_2	1	0	1	
x_3	0	0	0	
origine				

Tableau I-01 : Matrice d'adjacence d'un graphe

I.5.2 Listes d'adjacence

Dans la représentation par listes d'adjacence, on utilise un tableau TT de nn listes, une pour chaque sommet du graphe. Chaque liste d'adjacence T[i]T[i] contient les nœuds voisins du sommet ii. Cette méthode est plus efficace en termes de mémoire pour les graphes creux, car elle ne stocke que les voisins directs de chaque sommet. Cependant, l'accès aux prédécesseurs d'un sommet est plus compliqué avec cette méthode.[10]

Remarque : la représentation par matrice d'adjacence est plus adaptée pour les graphes denses, tandis que la représentation par listes d'adjacence est plus efficace pour les graphes creux.

I.6 l'application de théorie des graphes sur les réseaux complexes

La théorie des graphes a donné naissance à une multitude de théorèmes et d'algorithmes. Chacun d'entre eux a pour objectif de résoudre un problème pratique. Et pour cause, les graphes se retrouvent au sein de nombreuses applications quotidiennes comme aide à la décision, stratégie, optimisation (plus court chemin, GPS, coût minimal), réseaux de transports : chemins de fer, électricité, gaz, transport de l'énergie, Internet (réseau de l'information), ports et aéroports, ordonnancement des tâches, etc...

Voici les plus applications courantes de la théorie des graphes :

I.6.1 Les réseaux sociaux :

Ce terme englobe tous les sites web, applications mobiles et plateformes qui facilitent la création de liens sociaux en ligne. Ces systèmes fournissent aux utilisateurs des outils et des interfaces favorisant les interactions. Aujourd'hui, il existe une multitude de médias sociaux. Certains se concentrent sur des thématiques spécifiques, tandis que d'autres sont limités à des zones géographiques ou des communautés particulières, comme les réseaux sociaux(Facebook , Instagram...) d'écoles ou d'entreprises

Figure I-05 : Exemple de réseaux sociaux

I.6.2 Les réseaux d'information :

Un réseau informatique est constitué d'appareils informatiques interconnectés capables d'échanger des données et de partager des ressources entre eux. Ces appareils utilisent des protocoles de communication, un ensemble de règles, pour transmettre des informations via des technologies physiques ou sans fil. Fig07

I.6.3 Les réseaux technologiques :

On regroupe généralement sous la bannière des réseaux technologiques les réseaux de distribution d'information ou de distribution électrique, qui sont couramment étudiés. Un réseau similaire fréquemment étudié est le réseau formé par les routeurs qui permettent la distribution de données à l'échelle de l'internet. Ces réseaux sont généralement obtenus à l'aide d'observations des routes empruntées par les données pour se rendre d'un ordinateur à l'autre. La catégorie des réseaux technologiques ne comporte toutefois pas seulement des réseaux informatiques. Certains chercheurs ont étudié les propriétés des réseaux de distribution électrique D'autres ont étudié, par exemple, les réseaux routiers ou les réseaux de trafic aérien

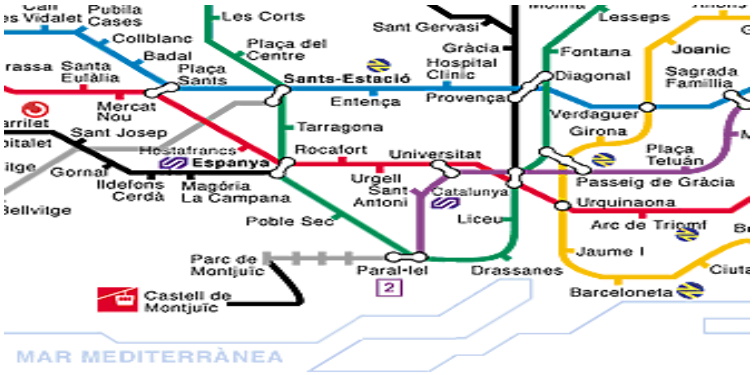


Figure I-06 : Représentation de réseau de Metro

I.6.4 Les réseaux biologiques

Un grand nombre de systèmes biologiques peuvent être représentés sous forme de réseaux. Par exemple, les réactions qui résultent de l'interaction entre différentes protéines peuvent être modélisées sous la forme d'arcs reliant des nœuds représentant des protéines un autre type d'étude intéressant concerne la modélisation des réseaux de neurones. Ces réseaux sont particulièrement difficiles à reconstituer, mais leur étude offre un aperçu précieux de la structure sous-jacente au fonctionnement du système nerveux.

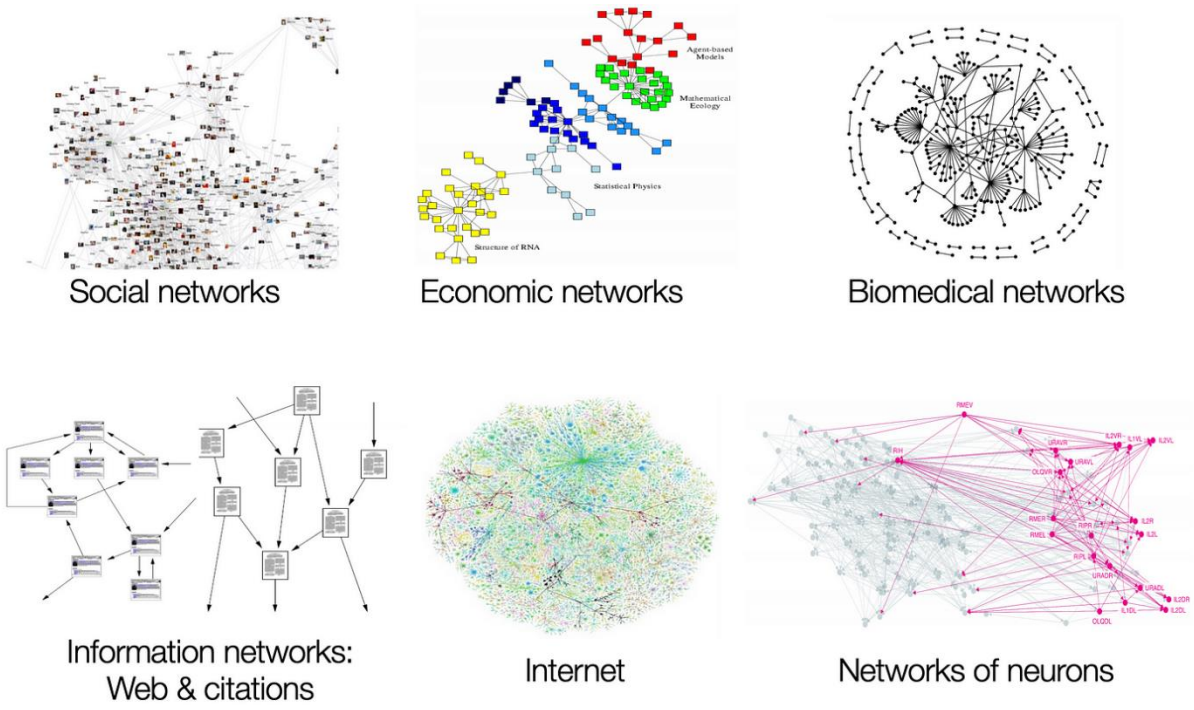


Figure I-07 : Des exemples de réseaux complexes

Conclusion

Dans ce premier chapitre, nous avons présenté les notions principales relatives à la théorie des graphes et démontré son rôle crucial dans la modélisation des réseaux complexes. Nous avons exploré les concepts fondamentaux tels que les nœuds, les arêtes, et les types de graphes, et discuté de leur pertinence dans divers contextes, En détaillant les différentes catégories de graphes, nous avons établi une base solide pour comprendre comment ces structures peuvent être utilisées pour représenter et analyser des systèmes complexes. Cette introduction fournit un cadre théorique essentiel pour les chapitres suivants, où nous approfondirons les méthodes et les algorithmes spécifiques employés pour la prédiction des liens et l'analyse des réseaux

Chapitre II

Prédiction de liens dans Réseaux complexes

Chapitre II : Prédiction de liens dans Réseaux complexes

II.1 Introduction

La prédiction de liens est importante dans de nombreux domaines de la vie, anticiper les relations potentielles entre différentes entités, prédire les comportements futurs et fournir des informations importantes pour la prise de décision. Dans ce chapitre, nous aborderons les différentes méthodes d'apprentissage utilisées pour prédire les connexions, en plus de celles-ci. Algorithmes de classification appliqués.

II.2 la prédiction de liens

La prédiction de liens consiste à inférer de nouveaux liens entre entités sur la base des liens existants. La plupart des approches existantes s'appuient sur l'apprentissage de vecteurs de traits latents pour l'encodage des entités et des relations. En général cependant, les traits latents ne sont pas facilement interprétables. Les approches à base de règles sont interprétables mais un ensemble de règles différent doit être appris pour chaque relation. Nous proposons une nouvelle approche qui n'a pas besoin de phase d'apprentissage et qui peut fournir des explications intelligibles pour chaque inférence. Elle repose sur le calcul de Concepts de plus proches. [11] Voici les étapes clés du processus de prédiction de liens avec l'apprentissage automatique [38] [39] [40] :

Collecte des Données : Acquisition des données du réseau, comprenant les entités et les relations entre elles. Ces données peuvent provenir de divers domaines tels que les réseaux sociaux, les réseaux biologiques, les réseaux de transport

Ingénierie des Caractéristiques : Extraction de caractéristiques pertinentes à partir des données du réseau. Cela peut inclure des mesures de similarité entre les entités, des attributs des nœuds, des motifs de sous-graphes

Construction du Dataset : Formation d'un ensemble de données comprenant des exemples positifs (liens existants) et des exemples négatifs (paires de nœuds non connectées).

Choix du Modèle : Sélection d'un modèle d'apprentissage automatique approprié pour la tâche de prédiction de liens. Cela peut inclure des méthodes traditionnelles telles que la régression logistique, ou des approches plus complexes comme les réseaux de neurones ou les méthodes d'ensemble comme les Forêts Aléatoires.

Entraînement du Modèle : Entraînement du modèle sur l'ensemble de données en utilisant les exemples positifs et négatifs. Le modèle apprend à partir des caractéristiques des données à prédire de nouvelles connexions dans le réseau

Validation du Modèle : Évaluation des performances du modèle sur un ensemble de validation ou à l'aide de techniques de validation croisée pour s'assurer de sa capacité à généraliser aux données non vues.

Prédiction de Liens : Utilisation du modèle entraîné pour prédire de nouvelles connexions dans le réseau. Le modèle attribue une probabilité à chaque paire de nœuds d'être liée, en fonction des caractéristiques extraites.

Évaluation des Performances : Évaluation des performances de la prédiction de liens à l'aide de métriques telles que la précision, le rappel, le F1-score, l'AUC-ROC, etc.

En combinant des techniques d'apprentissage automatique avec une ingénierie de caractéristiques appropriée, la prédiction de liens permet de découvrir de nouvelles relations potentielles dans les réseaux complexes, offrant ainsi des informations précieuses pour diverses applications telles que la recommandation de produits, l'identification de communautés, la prédiction de collaborations scientifiques.

Apprentissage automatique

L'apprentissage automatique, (Machine Learning) est une branche de l'intelligence artificielle qui se concentre sur le développement de techniques permettant aux ordinateurs d'apprendre à partir de données et de faire des prédictions ou des décisions sans être explicitement programmés pour chaque tâche. Il repose sur la création d'algorithmes capables d'identifier des modèles et des relations dans les données, puis d'utiliser ces informations pour effectuer des tâches variées telles que la classification, la régression, la détection d'anomalies. [34]

l existe plusieurs types d'apprentissage automatique, parmi lesquels :

-Apprentissage Supervisé

-Apprentissage Non Supervisé

-Apprentissage par Renforcement

Dans ce rapport nous allons utiliser l'apprentissage supervisé

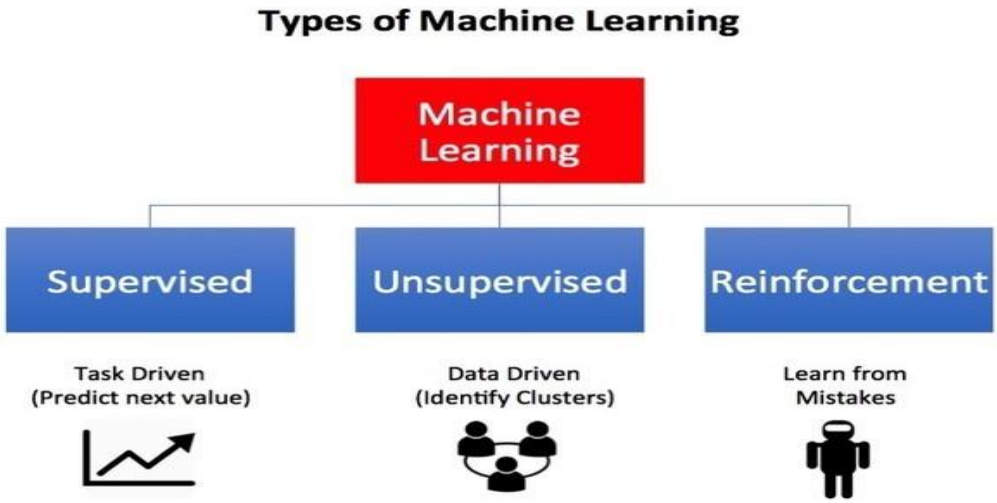


Figure II-08: schéma représente les types de machine Learning

L'apprentissage supervisé

L'apprentissage supervisé est une méthode d'apprentissage automatique où un modèle est entraîné à partir d'un ensemble de données étiquetées, c'est-à-dire des données d'entraînement comprenant des entrées et des sorties correctes. Ce processus permet au modèle de faire des prédictions ou des décisions précises lorsqu'il rencontre de nouvelles données. L'algorithme évalue sa précision en utilisant une fonction de perte et ajuste ses paramètres jusqu'à ce que l'erreur soit minimisée.[35]

L'apprentissage supervisé peut être appliqué à deux types de problèmes principaux :

Classification : Attribue des données de test à des catégories spécifiques en reconnaissant des entités. Les algorithmes courants incluent les classificateurs linéaires, les machines à vecteurs de support (SVM), les arbres de décision, les k plus proches voisins (k-NN), et les forêts d'arbres décisionnels (Random Forest).

Régression : Comprend la relation entre les variables dépendantes et indépendantes pour établir des projections. Les algorithmes populaires incluent la régression linéaire, la régression logistique, et la régression polynomiale.

II.3 Vus sur les Méthode d'apprentissage appliquées à la prédiction des liens

Les méthodes de prédiction de liens se répartissent en grandes catégories basées sur une variété de métriques et de techniques pour évaluer la similarité et la connectivité entre les entités. Nous présenterons ici une classification des méthodes de prédiction de liens en fonction de leurs approches et objectifs spécifiques.

On a proposé ce graphe pour appliquer les méthodes

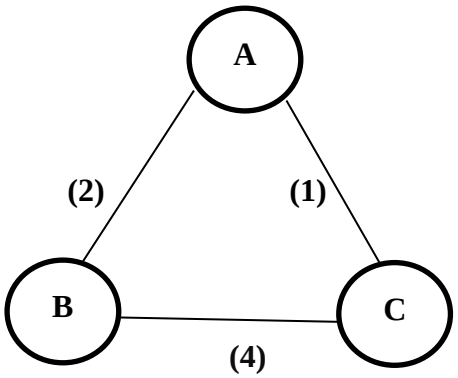


Figure II-09 : Exemple de graphe d'ordre 3

II.3.1 Mesures de similarité basées sur les voisins communs

- **Voisins communs (Hasan et Zaki, 2011)**

Est une technique locale simple pour estimer la similarité entre deux nœuds dans un réseau donné. Le degré de similarité entre deux nœuds est représenté par le nombre de voisins communs entre eux. Si deux nœuds ont une relation étroite, il est probable qu'ils partagent un grand nombre de voisins communs, ce qui augmente la probabilité d'avoir un lien entre eux. Cette hypothèse a été confirmée par plusieurs études qui ont observé une corrélation entre le nombre de voisins communs entre des paires de nœuds et la probabilité de liaison entre eux. La fonction utilisée pour déterminer le degré de similarité entre les nœuds est :

$$\textbf{Score } s(x,y)=|\Gamma(x)\cap\Gamma(y)| \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

$\boldsymbol{\Gamma(x)}$ Représente l'ensemble des voisins du sommet x

$\boldsymbol{\Gamma(y)}$ Représente l'ensemble des voisins du sommet y

Exemple : D'après le graphe précédent, Figure09

$\Gamma(A)=\{B,C\}$

$\Gamma(B)=\{A,C\}$

$\Gamma(C)=\{A,B\}$

Voisins communs de A et B : $\Gamma(A)\cap\Gamma(B)=\{C\}$ 1 voisin commun

Voisins communs de A et C : $\Gamma(A)\cap\Gamma(C)=\{B\}$ 1 voisin commun

Voisins communs de B et C : $\Gamma(B)\cap\Gamma(C)=\{A\}$ 1 voisin commun

- **Le coefficient de Jaccard (Hasan et Zaki, 2011)**

Le coefficient de Jaccard c'est une mesure statistique utilisée pour comparer la similarité des ensembles d'échantillons. Elle est habituellement notée J (x, y) où x et y représentent deux nœuds différents d'un réseau.

En prévision de liens, tous les voisins d'un nœud sont traités comme un ensemble et la prédiction est faite par calcul et le classement de la similarité de l'ensemble de chaque paire de nœud voisin.

L'expression mathématique de cette méthode est la suivante :

$$\mathbf{s(x,y)} = \frac{|\,\mathbf{\Gamma(x)\cap\Gamma(y)}\,|}{|\mathbf{\Gamma(x)}\cup\mathbf{\Gamma}\,\mathbf{(y)}|} \text{.....(2)}$$

- **Adamic/Adar (Adamic et Adar, 2003)**

Cette mesure de similarité, initialement proposée par Lada Adamic et Eytan Adar, était destinée à mesurer la similarité entre deux entités en fonction de leurs caractéristiques partagées [Adamic and Adar 2003]. Cependant, le poids de chaque caractéristique est pénalisé logarithmiquement par sa fréquence d'apparition. Si nous considérons les voisins comme des caractéristiques, cela peut être écrit comme suit :

$$\mathbf{AA(x,y)} = \sum_{\boldsymbol{z}\,\boldsymbol{\epsilon}\,\mathbf{\Gamma(x)\cap\Gamma(y)}} \frac{\mathbf{1}}{\mathbf{\log}\ |\mathbf{\Gamma}\,\mathbf{(z)}\,|} \text{.....(3)}$$

$\mathbf{\Gamma(x)}$ représente l'ensemble des voisins du sommet x

$\mathbf{\Gamma(y)}$ représente l'ensemble des voisins du sommet y

$\mathbf{\Gamma(x)\cap\Gamma(y)}$ est l'ensemble des voisins communs aux sommets **x** et **y**

$|\mathbf{\Gamma}\,\mathbf{(z)}\,|$ est le nombre de voisins du sommet z

Exemple : D'après le graphe précédent, Figure 09

Γ (A)={B,C}

Γ (B)={A,C}

Γ (C)={A,B}

Adamic /Adar de (A ,B)

Les voisins communs de A et B sont : Γ(A)∩Γ(B)={C}

Le nombre de voisins de *C* est : |Γ(*C*)|=2

Donc : AA(A,B)= $\frac{1}{\log 2}$ ≈1.4427

II.3.2 Mesures basées sur la longueur des chemins

- **Chemin le plus court (Hasan et Zaki, 2011)**

  Cette caractéristique traditionnelle correspond à le plus petit nombre d'arêtes formant un chemin entre une paire de sommets ou bien trouver le plus court chemin entre deux sommets d'un graphe (orienté ou non orienté)

Chemin le plus court = | chemin la plus petite | .....(4)

Exemple : La distance entre A et B est 1 (liaison directe).

II.3.3 Mesures basées sur la connectivité globale

- **Somme des voisins (Hasan et al., 2006)**

  Total A partir des voisins de chaque sommet d'une paire,

  Une valeur spécifique est calculée en fonction des valeurs de ses voisins.

$$\mathbf{S(v)} = \sum_{u \in N(v)} W(u,v) \quad \textbf{.....(5)}$$

V représente le nœuds (sommets) qui nous calculerons le total des voisins .

u est un nœud voisin spécifique de v.

$W(u,v)$ est le poids de l'arête , entre v et u .

N (v) est l'ensemble des nœuds voisins de v qui sont connectés directement à v.

Exemple :

Pour le nœud A :

Somme des voisins(A) = w(A,B) + w(A,C) = 2+1= 3

Pour le nœud B :

Somme des voisins(B)=w(B,A)+w(B,C) = 2+4 = 6

- **Somme des éléments intermédiaires**

Prise en compte de la structure d'un graphe biparti, la somme des éléments intermédiaires fait référence au total nombre d'éléments connectés aux deux sommets Qui forment une paire .

Dans un graphe bipartite G = (U,V,E) où U et V sont les deux ensembles disjoints de sommets et E est l'ensemble des arêtes, pour une paire de sommets (u,v) avec u ∈ U et v ∈ V la somme des éléments intermédiaires SI(u,v) est définie comme :

$$\mathbf{SI\ (u,v) =| N(u) \cap N(v)| .....(6)}$$

Où N(u) désigne l'ensemble des voisins du sommet u et N(v) désigne l'ensemble des voisins du sommet v. Autrement dit, cela représente le nombre de sommets dans l'ensemble U qui sont connectés à v, ainsi que le nombre de sommets dans l'ensemble V qui sont connectés à u.

Exemple :

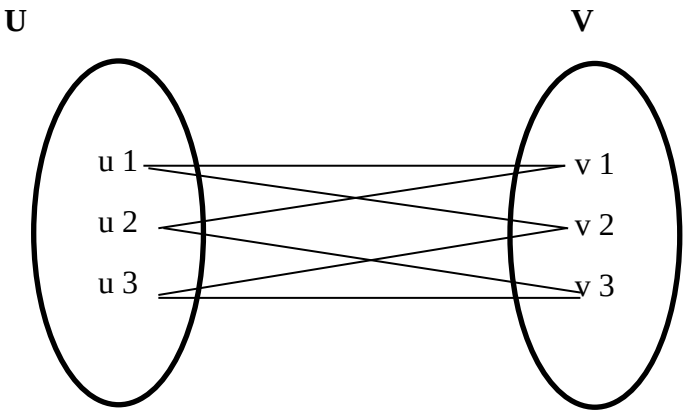


Figure II-10 : Exemple de graphe biparti

Pour la paire de sommets (u1, v2):

Les voisins de u1 (dans V) sont {v1,v2}

Les voisins de v2 (dans U) sont {u1,u2}

N(u1) ∩ N(v2) est vide donc **SI(u1,v2)=0**

Pour la paire de sommets (u1, v1):

Les voisins de u1 (dans V) sont {v1,v2}

Les voisins de v2 (dans U) sont {u1,u2}

N(u1) ∩ N(v1) est le sommet u1 donc **SI(u1,v1)=1**

- **Attachement préférentiel (Barabasi et al., 2002)**

  Est basée sur le principe que deux nœuds qui ont beaucoup de relations ont tendance à être liés :

  AP (x,y) = || Γ(x) || * || Γ(y) || .....(7)

|| **Γ(x)** || est le degré du sommet x

Exemple : D'après le graphe précédent, Figure09

||Γ(*A*)||= 2

||Γ(*B*)||= 2

||Γ(*C*)||= 2

AP(A,B)=||Γ(A)||×||Γ(B)||=2×2=4

AP(A,C)=||Γ(A)||×||Γ(C)||=2×2=4

AP(B,C)=||Γ(B)||×||Γ(C)||=2×2=4

- **Indice de Leicht-Holme-Newman (Leicht et al., 2006)**

  Cet indice est défini comme le rapport de chemins réels de longueur deux entre deux nœuds et une valeur proportionnelle à la longueur attendue nombre de trajets de longueur deux entre eux . Ses propres auteurs proclamer que cet indice est une mesure d'équivalence structurelle plus sensible que d'autres comme l'indice Salton ou l'indice Jaccard. La fonction de similarité définie par cet indice peut être calculé comme :

  $$\boldsymbol{LHN(x,y)} = \frac{|\,\boldsymbol{\Gamma(x)\cap\Gamma(y)}\,|}{|\,\boldsymbol{\Gamma(x)}\,|\,|\boldsymbol{\Gamma}\,(\boldsymbol{y})|} \textbf{.....(8)}$$

. Exemple : D'après le graphe précédent, Figure 09

||Γ(A)||=||Γ(B)||= ||Γ(C)||=2

Γ(*A*)∩Γ(*B*)={*C*}

LHN(A,B) = $\frac{1}{2\times2}$ = 2.25

II.3.4 Mesures de centralité et de cohésion

- **Coefficient de clustering (Hasan and Zaki, 2011)**

Ce coefficient exprime la probabilité que deux nœuds soient connectés, en considérant qu'ils partagent un voisin en commun.

$$\mathbf{CC} = \frac{2*\text{nombre de lien entre les voisins de i}}{\text{nombre de voisins de i}*(\text{nombre de voisins de i}-1)} \text{ .....(9)}$$

Exemple :

Pour le nœud A :

Nombre de liens entre les voisins de 1 : il y a un lien entre les nœuds 2 et 3

Nombre de voisins de 1 : 2(nœuds 2 et 3).

Donc : $\mathbf{CCA} = \frac{\mathbf{2*1}}{\mathbf{2*(2-1)}} = \mathbf{1}$

- **Clustering moyen des nœuds (Saramäki et al., 2007)**

Regroupement local moyen de tous les sommets d'un graphe

$$\mathbf{Cm} = \frac{1}{\mathrm{N}} \sum_{i=1}^{N} C_i \text{ .....(10)}$$

N est le nombre total de nœuds dans le graphe

C_i est le coefficient de clustering du nœud i

Exemple

Grâce à l'exemple précédent on a déjà : **CCA =CCB=CCB=1**

Et **N= 3** donc : **Cm =** $\frac{1}{3}$**(1+1+1)=** $\frac{1}{3}$**(3) =1**

- **Centralité de proximité (Freeman, 1978)**

La centralité de proximité indique la proximité d'un nœud par rapport à tous les autres nœuds du réseau. Il est calculé comme la moyenne de la longueur du chemin le plus court entre le nœud et tous les autres nœuds du réseau , Elle peut être exprimée comme suit **[18]** :

$$\boldsymbol{C(u)} = \frac{\mathbf{1}}{\sum_y \boldsymbol{d}(\mathbf{u},\mathbf{y})} \text{ .....(11)}$$

C(u) est la centralité de proximité du sommet

d(u,y) est la distance entre le sommet u et le sommet y

Exemple :

Pour le sommet A

d(A,u)= d (A,B)+ d (A,C)=1+1=2

donc : $\boldsymbol{C(A) = \frac{1}{2}}$

- **Degré moyen des voisins (Barrat et al., 2004)**

Le degré moyen des voisins, c'est-à-dire la moyenne des degrés des paires de sommets, est utilisée pour déterminer la connectivité moyenne des voisins d'un nœud,

$$\langle k \rangle = \frac{1}{|N(v)|} \sum_{u \in N(v)} \deg(u) \quad \text{.....(12)}$$

Deg(u) est le degré du voisin u de v.

$|N(v)|$ est le nombre des voisins du nœud v.

Exemple : D'après le graphe précédent, Figure 09.

Deg (A) =2 (connecté a B et C)

Deg (B) =2 (connecté a A et C)

Deg (C) =2 (connecté a A et B)

$\langle k \rangle = \frac{1}{3}$ (deg (A)+deg(B)+deg(c)) = $\frac{1}{3}$ (2+2+2)= = $\frac{6}{3}$ =2

II.4 Algorithmes de classification

L'algorithme de classification fait partie des méthodes d'apprentissage supervisé. C'est-à-dire que les prédictions sont réalisées à partir de données historiques.

À l'inverse de l'apprentissage non supervisé où il n'y a pas de classes prédéfinies. Il faut donc constituer les catégories en fonction des attributs communs, pour ensuite réaliser la prédiction. **[13]**

II.4.1 L'algorithme SVM (Support Vector Machine)

est un algorithme d'apprentissage supervisé utiliser pour la classification et la régression pour trouver les limites de décision qui séparent les différentes classes dans l'espace des fonctionnalités, tout en maintenant la distance maximale entre les points les plus proches de chaque classe. Les données sont séparées en différentes classes en utilisant l'idée de marge maximale, où. la marge représente la distance entre une ligne de séparation des données et le point le plus proche de l'un des ensembles.

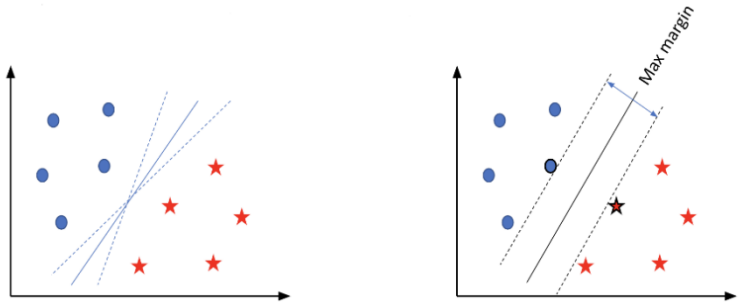


Figure II-11: L'algorithme SVM

II.4.2 L'algorithme KNN

Un algorithme d'apprentissage supervisé fonctionne sur le principe selon lequel chaque point de données situé à proximité les uns des autres appartient à la même classe. Autrement dit, les choses proches les unes des autres se ressemblent.

En bref, dans l'algorithme du K-voisin le plus proche lorsque K = 3, la distance entre le point cible et trois points voisins est mesurée. Si les deux points les plus proches appartiennent à un certain groupe et que le troisième point appartient à un groupe différent, le point cible sera classé en fonction de la majorité, c'est-à-dire qu'il sera classé comme faisant partie du groupe auquel appartiennent les deux points les plus proches.

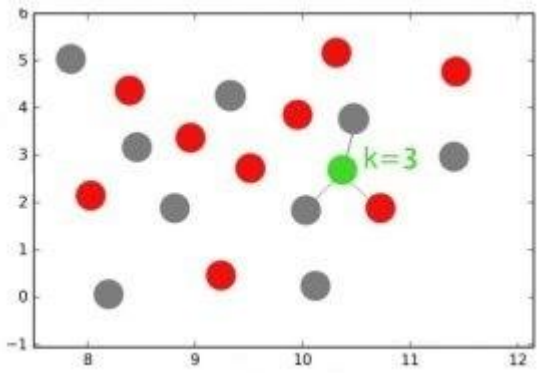


Figure II-12 : L'algorithme KNN

II.4.3 L'algorithme Naïve Bayes (NB)

L'algorithme Naive Bayes est un modèle de classification basé sur le théorème de Bayes et l'hypothèse « naïve » et prédit en fonction de la probabilité qu'un élément soit présent. Il est efficace dans la classification en raison de sa simplicité et de sa rapidité. Et on utiliser apré la Préparation des données puis calculant les probabilités conditionnelles qui sont calculées à l'aide de l'équation suivante :

$$\mathbf{P(A|B)} = \frac{\mathbf{P(B|A)\,P(A)}}{\mathbf{P(B)}}$$

sachant que :

P(A|B) est la probabilité conditionnelle de l'événement A étant donné l'événementB

P(A) Probabilité A

II.4.4 L'algorithme CART

L'algorithme CART (Classification and Regression Trees) est une méthode d'apprentissage automatique utilisée à la fois pour la classification et la régression. Elle s'appuie sur un arbre de décision, où les données sont divisées en petits sous-groupes afin que chaque sous-groupe soit homogène. L'arborescence puis choisit la valeur seuil qui fournit la meilleure division des données, et ainsi de suite. Le processus est répété pour chaque sous-groupe jusqu'à ce qu'un état d'arrêt soit atteint.

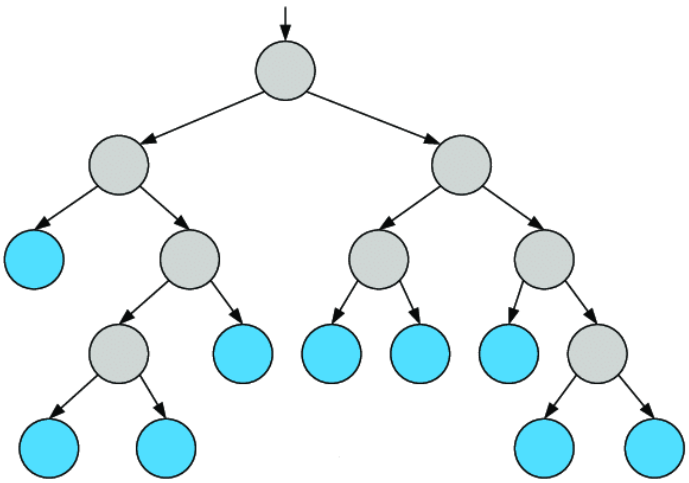


Figure II-13 : L'algorithme CART

II.4.5 AdaBoost (Adaptive Boosting)

AdaBoost ou Adaptive Boosting, est une méthode d'ensemble utilisée en apprentissage supervisé qui combine plusieurs classificateurs faibles pour former un classificateur fort. La méthode fonctionne en pondérant les erreurs des classificateurs faibles et en ajustant ces poids à chaque itération pour se concentrer sur les erreurs les plus difficiles à corriger.[36]

- **Principe** :
    1. Initialisation des poids de toutes les observations de manière égale.
    2. Entraînement d'un classificateur faible sur les données pondérées.
    3. Calcul de l'erreur du classificateur et ajustement des poids : augmenter les poids des erreurs et diminuer les poids des bonnes prédictions.
    4. Répétition du processus pour un nombre d'itérations donné ou jusqu'à ce que l'erreur soit minimisée.

II.4.6 Foret Aléatoires (Random Forest)

La méthode des Forêts Aléatoires est un algorithme d'ensemble qui utilise un grand nombre d'arbres de décision pour effectuer des tâches de classification et de régression. Chaque arbre est formé sur un sous-échantillon aléatoire des données, et les décisions finales sont prises par un vote majoritaire ou une moyenne des prédictions des arbres individuels.[37]

- **Principe** :
    1. Génération de plusieurs sous-échantillons aléatoires des données d'entraînement avec remplacement (Bootstrap sampling).
    2. Construction d'un arbre de décision pour chaque sous-échantillon, en utilisant une sélection aléatoire de caractéristiques à chaque nœud.
    3. Agrégation des prédictions de tous les arbres pour obtenir la prédiction finale (vote majoritaire pour la classification, moyenne pour la régression).

II.5 Les mesures d'évaluation

Nous avons deux méthodologies peuvent être utilisées pour performance la technique de prédiction de lien

La première approche ressemble aux méthodes de classification traditionnelles, dans lesquelles le classificateur prédit la classe (existence ou absence de lien) pour chaque paire de nœuds, et les performances sont mesurées en comparant ces prédictions avec les classes réelles.

La deuxième approche capitalise sur les scores ou probabilités générés par les techniques de prédiction de liens, classant toutes les paires de nœuds en fonction de ces scores. Les t premières

paires sont considérées comme des instances positives (indiquant un lien prédit), où t est égal au nombre de liens dans l'ensemble de test. Les paires restantes sont traitées comme des instances négatives, ce qui suggère l'absence de lien entre elles. Cette approche évalue la capacité de la technique à classer les paires vraies positives parmi toutes les paires.[17]

		Classe réelle	
		Positif (lien)	Négatif (pas de lien)
Classe prévue	Positif (lien)	True Positive (TP)	False positive (FP)
	Négatif (pas de lien)	False negative (FN)	True negative(TN)

Tableau II-02 : matrice de confusion pour une prédiction de lien

TP : Nombre de liens existe dans Gpredicted et existe dans Gtest

TN : Nombre de liens n'existe pas dans Gpredicted et existe dans Gtest

FP : Nombre de liens existe dans Gpredicted et n'existe pas dans Gtest

FN : Nombre de liens n'existe pas dans Gpredicted et n'existe pas dans Gtest

Après l'implémentation des mesures citées au-dessus sur les Data-Frame. On calcule les valeurs positives et négatives des vrais et faux liens. En fin, on va calculer les critères suivants

Précision : est une mesure de la performance d'un modèle de classification. Elle indique la proportion de prédictions positives correctes par rapport à l'ensemble des prédictions positives effectuées par le modèle. En d'autres termes, la précision mesure la qualité des prédictions positives, c'est-à-dire le pourcentage des éléments correctement identifiés parmi ceux que le modèle a étiquetés comme positifs. Et elle est définie comme suit :

$$\text{précision} = \frac{\text{TP}}{(\text{TP}+\text{FP})}$$

Une précision élevée signifie que le modèle fait peu d'erreurs lorsqu'il prédit des éléments comme positifs.

Rappel : est une mesure de la performance d'un modèle de classification, qui évalue la capacité du modèle à identifier tous les éléments pertinents dans un ensemble de données. Le rappel indique la proportion de vrais positifs parmi tous les éléments qui sont réellement positifs et il est défini comme suit :

$$\text{Rappel} = \frac{\text{TP}}{(\text{TP}+\text{FN})}$$

Accuracy : est une mesure de la performance globale d'un modèle de classification. Elle indique la proportion de prédictions correctes (à la fois positives et négatives) par rapport à l'ensemble des prédictions effectuées. En d'autres termes, l'exactitude mesure le pourcentage total de classifications correctes effectuées par le modèle et se calcule comme suit :

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

Une grande précision signifie que le modèle commet globalement peu d'erreurs dans ses prédictions.

F_Mesure : une mesure qui combine la précision et le rappel. Elle est définie comme suit :

$$\text{F}_{\text{Mesure}} = \frac{2*(\text{Précision}*\text{Rappel})}{(Précision+Rappel)}$$

Cette mesure est particulièrement utile pour évaluer les modèles dans des contextes où il existe un compromis entre précision et rappel, et où les classes peuvent être déséquilibrées. Elle varie de 0 à 1, où 1 indique une performance parfaite.

II.6 Conclusion

Ce chapitre nous a permis d'explorer différentes méthodes d'apprentissage appliquées à la prédiction de liens. Dans divers domaines de recherche et d'application, En examinant des algorithmes spécifiques, nous avons pu comprendre comment ces techniques peuvent être utilisées pour modéliser et prédire les relations entre entités.

Chapitre III

Tests et expérimentations

Chapitre III : Tests et expérimentations

III.1 Introduction

Ce chapitre est consacré aux expériences menées dans notre étude. Nous expliquons les outils utilisés dans le développement du projet, tels que le choix du langage de programmation, de l'environnement de programmation et des appareils utilisés. La prédiction de réaction appliquée dans ce travail est présentée et les résultats expérimentaux obtenus sont illustrés. Nous continuons ensuite à discuter des résultats obtenus.

III.2 Environnement Expérimental

Les expérimentations ont été effectuées sur un PC (MAS-PC) avec un processeur Intel ® Pentium ® CPU B960 @2.20GHz

4.00G de RAM fonctionnant sous le système d'exploitation Windows 64 bits

III.3 Technologies utilisées

Pour réaliser l'implémentation nous avons utilisé environnements de développement suivant :

III.3.1 Python

Le langage Python est un langage de programmation open source multiplateformes et orienté objet. Grâce à des bibliothèques spécialisées, Python s'utilise pour de nombreuses situations comme le développement logiciel, l'analyse de données, ou la gestion d'infrastructures. Il n'est donc pas, comme le langage HTML par exemple, uniquement dédié à la programmation web. Langage de programmation interprété, Python permet l'exécution du code sur n'importe quel ordinateur. Utilisable aussi bien par des programmeurs débutants qu'experts, Python permet de créer des programmes de manière simple et rapide. **[15]**

Figure III-**14** : logo de langage Python

III.3.2 le navigateur Anaconda

Anaconda Navigator est une interface utilisateur graphique (GUI) de bureau inclus dans Anaconda Distribution qui vous permet de lancer des applications et de gérer des packages, des environnements et des canaux conda sans utiliser de commandes d'interface de ligne de commande (CLI). Navigator peut rechercher des packages sur Anaconda.org ou dans un référentiel Anaconda local. Il est disponible pour Windows, MacOs et Linux. **[16]**

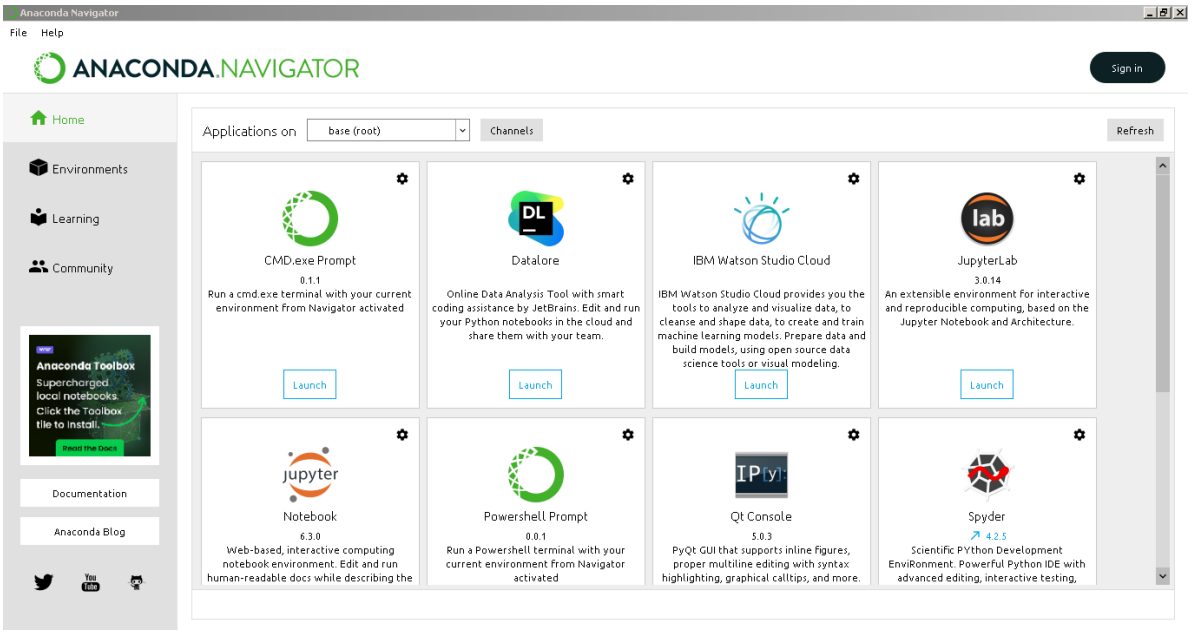


Figure III-15 : Navigateur Anaconda

III.3.3 Spyder

Spider est un puissant environnement de développement interactif pour le langage Python, doté de fonctionnalités d'édition avancées et de tests interactifs. Il offre un ensemble d'outils puissants pour le développement, le débogage et l'analyse de données en plus d'un environnement informatique numérique. Spyder prend en charge l'utilisation de bibliothèques Python populaires telles que NumPy, SciPy et Matplotlib, permettant une manipulation efficace des données et des graphiques. Il s'agit d'un environnement de développement intégré (IDE) gratuit inclus Anaconda

FigureIII-16 : spyder logo

.

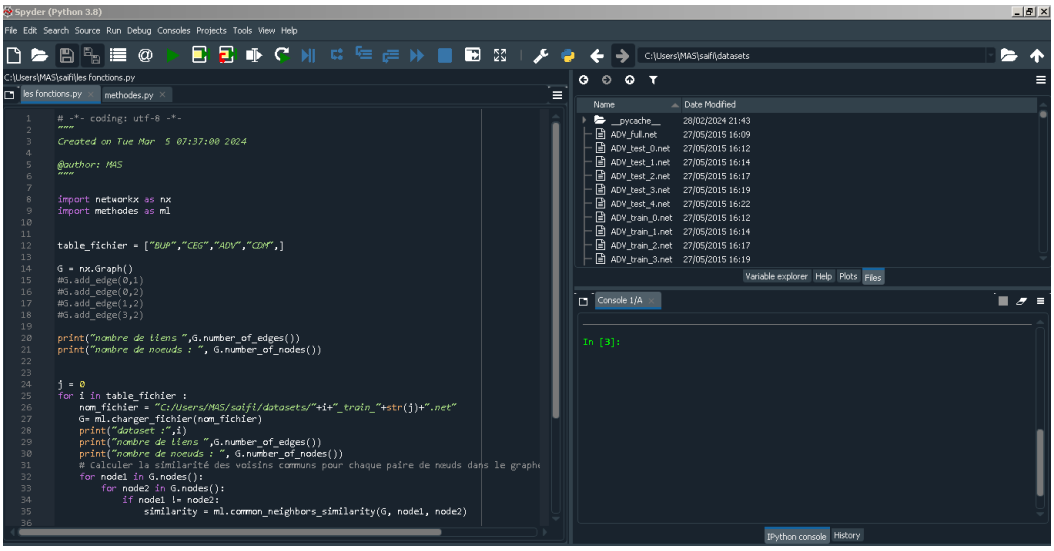


Figure III-17 : Interface Spyder

III.4 Bibliothèques utilisées

- **NetworkX**

NetworkX est une bibliothèque Python pour traiter et créer des réseaux complexes. C'est un programme gratuit qui vous permet de créer des graphiques et d'ajouter des nœuds et des arêtes.

- **NumPy**

NumPy est une bibliothèque pour le langage de programmation Python, permettant de créer et de manipuler de grands tableaux Matrices multidimensionnelles et fonctions mathématiques.

Figure III-18: NumPy logo

- **Matplotlib**

Matplotlib est une bibliothèque de visualisation de données en Python. Elle permet de créer des graphiques statiques, animés et interactifs de haute qualité. Matplotlib est particulièrement utile pour la génération de graphiques en deux dimensions, tels que des histogrammes, des graphiques en barres, des graphiques linéaires, des diagrammes de dispersion[31]

Figure III-19: Matplotlib Logo

- **Pandas**

Pandas est une bibliothèque Python utilisée pour le traitement et l'analyse des données, qui permet de traiter et de stocker des matrices numériques. Pandas est largement utilisé dans le domaine de la science des données.

Figure III-20: Pandas logo

- **Sklearn**

Scikit-learn est une bibliothèque de machine Learning en Python qui fournit des outils simples et efficaces pour l'analyse des données et l'apprentissage automatique. Elle comprend une large gamme d'algorithmes de machine Learning pour la classification, la régression, le clustering, la réduction de dimension, et le prétraitement des données. Scikit-learn est bien intégrée avec d'autres bibliothèques scientifiques de Python comme NumPy et SciPy [32]..

Figure III-21: Sklearn logo

III.5 Description du Dataset

Dans notre travail, nous avons utilisé une base de données contenant 22 réseaux provenant de diverses sources et domaines d'application. On a choisi (8) réseaux soigneusement sélectionnés pour représenter une large gamme de propriétés, incluant des tailles variées, des degrés moyens, des coefficients de regroupement et des indices d'hétérogénéité. Le tableau ci-dessous résume les propriétés structurelles des réseaux utilisés dans nos expériences.

Nom	Nombre Des nœuds	Nombre Des liens	(k)	C	ASPL	D	H	R
EML	1133	5451	9.62	0.22	3.61	8	1.9421	0.0782
HMT	2426	16630	13.71	0.54	3.15	10	3.1011	0.0474
NSC	1461	2742	3.75	0.69	2.59	17	1.8486	0.4616
SMG	1024	4916	9.6	0.31	2.98	6	3.9475	-0.1925
UAL	332	2126	12.81	0.63	2.74	6	3.4639	-0.2079
CEG	297	2148	14.46	0.29	2.46	5	1.8008	-0.1632
PUB	105	441	8.4	0.49	3.08	7	1.4207	-0.1279
INF	410	2765	13.49	0.46	3.63	9	1.3876	0.2258

Tableau III-03 : Les valeurs correspondantes aux différents éléments utilisés dans notre Échantillon de données

HMT : est un réseau social.

UAL : est un réseau de trafic aéroportuaire.

EML : est un réseau d'individus qui partagent des e-mails.

PUB : est un réseau de blogs politiques.

INF : est un réseau de contacts en face à face dans une exposition.

CEG : est un réseau de biologique.

SMG, NSC : sont des réseaux de coauteurs pour différents domaines d'études.

Les données sont accessibles à l'adresse suivante: https://vlado.fmf.uni-lj.si/pub/networks/data/

- **Nombre de nœuds** : Le nombre total de nœuds dans le réseau.
- **Nombre de liens** : Le nombre total de liens (arêtes) dans le réseau.
- **k** : Degré moyen des nœuds.
- **C** : Coefficient de clustering moyen.
- **ASPL** : Longueur moyenne du plus court chemin.
- **D** : Diamètre du réseau.
- **H** : Hauteur de l'arborescence couvrante.
- **r** : Coefficient d'assortativité

Ces ensembles de données diversifiés nous permettent d'évaluer les algorithmes de prédiction des liens dans différents contextes et types de réseaux. Chacun de ces réseaux présente des caractéristiques uniques qui sont essentielles pour comprendre les dynamiques et les interactions complexes au sein des réseaux. La compréhension approfondie de ces propriétés structurelles nous aide à adapter et à optimiser les modèles de prédiction de liens pour obtenir des résultats plus précis et pertinents.

III.6 Les processus de prédiction des liens

Dans cette section, nous détaillons les étapes suivies pour prédire les liens dans les réseaux complexes en utilisant l'apprentissage automatique supervisé. Le processus se décompose en plusieurs étapes principales, qui sont essentielles pour évaluer et optimiser les performances de notre modèle :

1. Traitement de Données

Nous avons divisé notre dataset en deux classes :

Classe 1 : Liens existants :

- Les liens existants dans le graphe initial sont considérés comme des exemples positifs.
- Chaque lien est représenté comme une paire de nœuds connectés.

Classe 0 : Liens non existants :

- Des paires de nœuds non connectées sont générées aléatoirement pour représenter des exemples négatifs.
- Cette classe aide à équilibrer le dataset en ajoutant des exemples de non-liens

2. Fusion et Calcul de la Similarité

Fusion des Classes :

Les paires de nœuds des deux classes (liens existants et non existants) sont combinées en un seul dataset. Chaque paire de nœuds est associée à une étiquette binaire indiquant la présence (1) ou l'absence (0) de lien

Calcul des Mesures de Similarité :

Ces mesures sont calculées pour chaque paire de nœuds dans les classes de liens existants et non existants, et sont ensuite utilisées comme caractéristiques pour l'entraînement et l'évaluation du modèle de prédiction de liens. (,Coefficient de Jaccard, Common Neighbors...)

3. Normalisation et Division en Ensembles d'Entraînement et de Test

Nous avons préparé nos données pour l'entraînement du modèle en les normalisant Les mesures de similarité pour s'assurer qu'elles ont une distribution comparable, ce qui aide à optimiser la performance du modèle d'apprentissage automatique. La technique courante de normalisation qu'on a utilisé c'est :

Mise à l'échelle (Min-Max Scaling) : Transformation des valeurs pour qu'elles se situent entre 0 et 1.

Division du Dataset :

Après la normalisation de dataset on a divisé en deux ensembles d'entraînement et de test.

Ensemble d'Entraînement (80%) : Utilisé pour entraîner le modèle.

Ensemble de Test (20%) : Utilisé pour évaluer les performances du modèle.

4. Entraînement et Évaluation du Modèle

Nous avons Sélectionné et entraîner un modèle de machine Learning, puis évaluer sa performance.

Premièrement nous utilisons Divers algorithmes de machine Learning sont appliqués sur notre dataset. Les modèles couramment utilisés incluent : Régression Logistique, Arbres de Décision, Forêts Aléatoires, Machines à Vecteurs de Support (SVM)

Et pour évaluer Les performances du modèle en utilisant des métriques telles que : Accuracy , F1-score, Précision , Recall

On a déjà verra la définition des méthodes dans 2-ème chapitre

En suivant ces étapes méthodiques, nous nous assurons que notre modèle de prédiction de liens est à la fois robuste et précis, capable de généraliser efficacement aux données non vues.

Nous avons repris tous les processus dans le schéma suivant :

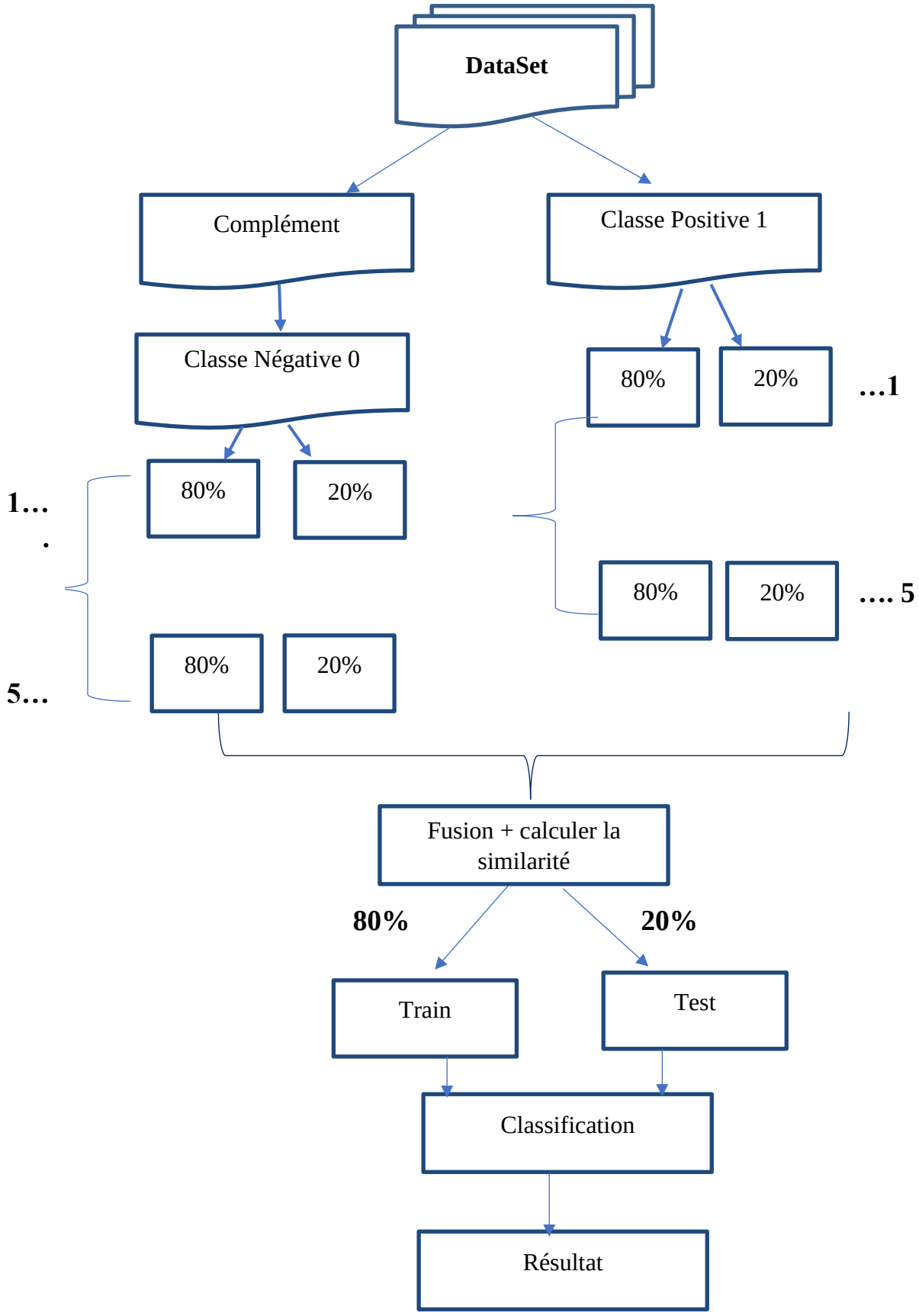


Figure III-22: schéma représente le Processus de prédiction des liens

III.7 Résultats et expérimentation

Dans cette section, nous présentons les résultats expérimentaux obtenus à partir de l'application de différents algorithmes d'apprentissage automatique pour la prédiction de liens dans des réseaux complexes. Nous avons évalué les performances de quatre modèles : RandomForest, AdaBoost, SVM, et KNN, en utilisant plusieurs métriques telles que l'accuracy, la précision, le rappel et le F1-score. En outre, nous avons comparé le temps d'exécution de chaque modèle pour chaque ensemble de données. Les jeux de données utilisés proviennent de divers domaines, allant des réseaux sociaux aux réseaux biologiques, ce qui nous permet d'évaluer la robustesse et la généralisation des modèles.

III.7.1 Résultat de Accuracy :

	Total	AdaBoost	KNeighbors	RandomForest	SVM
BUP	0,88094504	0,88094504	0,880945044	0,880945044	0,880945044
CEG	0,94809616	0,94809616	0,948096159	0,948096159	0,948096159
INF	0,95063291	0,95063291	0,950632911	0,950632911	0,950632911
SMG	0,93033119	0,93490668	0,934906683	0,934906683	0,921180196
UAL	0,96080192	0,97295307	0,972953066	0,972953066	0,93649962

Tableau III-04:Résultats de Accuracy

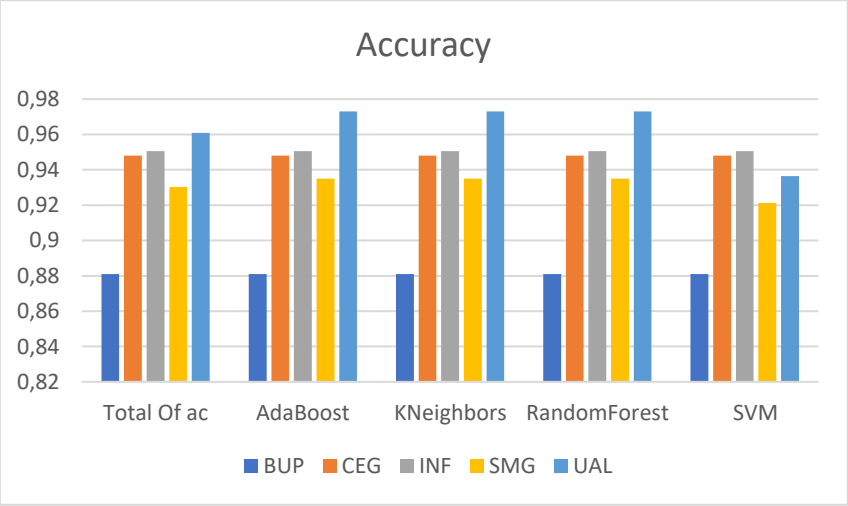


Figure III-23 : Diagramme du Résultats de Accuracy

Commentaires:

- Tous les modèles montrent des Accuracy élevées sur tous les ensembles de données, indiquant une bonne capacité à prédire les liens corrects.
- Les variations minimes entre les méthodes sur chaque ensemble de données suggèrent une performance comparable dans la plupart des cas.
- SVM montre une légère baisse sur SMG par rapport aux autres ensembles, indiquant une spécificité potentielle des caractéristiques de ces données.

III.7.2 Résultat de Précision :

	Total	AdaBoost	KNeighbors	RandomForest	SVM
BUP	0,808072653	0,80807265	0,808072653	0,808072653	0,808072653
CEG	0,906034947	0,90603495	0,906034947	0,906034947	0,906034947
INF	0,910258995	0,910259	0,910258995	0,910258995	0,910258995
SMG	0,878509585	0,88483572	0,884835721	0,884835721	0,865857313
UAL	0,928277148	0,94871661	0,948716605	0,948716605	0,887398236

Tableau III-05 : Résultats de Précision

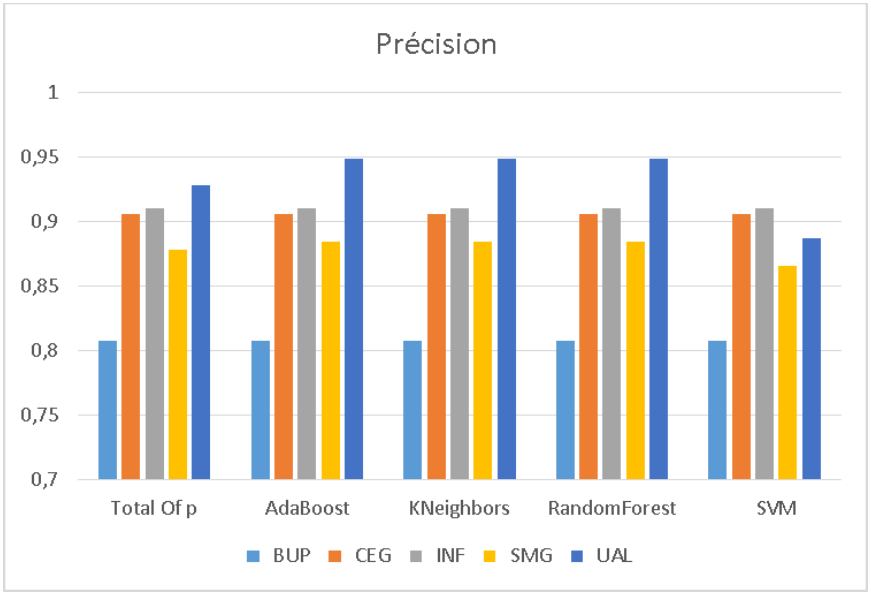


Figure III-24 : Diagramme du Résultats de Précision

Commentaires:

- Tous les modèles présentent une précision très élevée (> 0,80) sur tous les ensembles de données, indiquant une capacité à prédire avec précision les liens corrects.
- AdaBoost, KNeighbors et RandomForest montrent des performances similaires, tandis que SVM a légèrement plus basse précision sur SMG.
- La précision reste cohérente sur tous les ensembles, ce qui souligne la robustesse des modèles.

III.7.3 Résultat de F1-score :

	Total	AdaBoost	KNeighbors	RandomForest	SVM
BUP	0,893736318	0,893736318	0,893736318	0,893736318	0,893736318
CEG	0,950683268	0,950683268	0,950683268	0,950683268	0,950683268
INF	0,952990038	0,952990038	0,952990038	0,952990038	0,952990038
SMG	0,935122623	0,938892518	0,938892518	0,938892518	0,927582834
UAL	0,962554723	0,973674295	0,973674295	0,973674295	0,940315579

Tableau III-07 :Résultats de F1-score

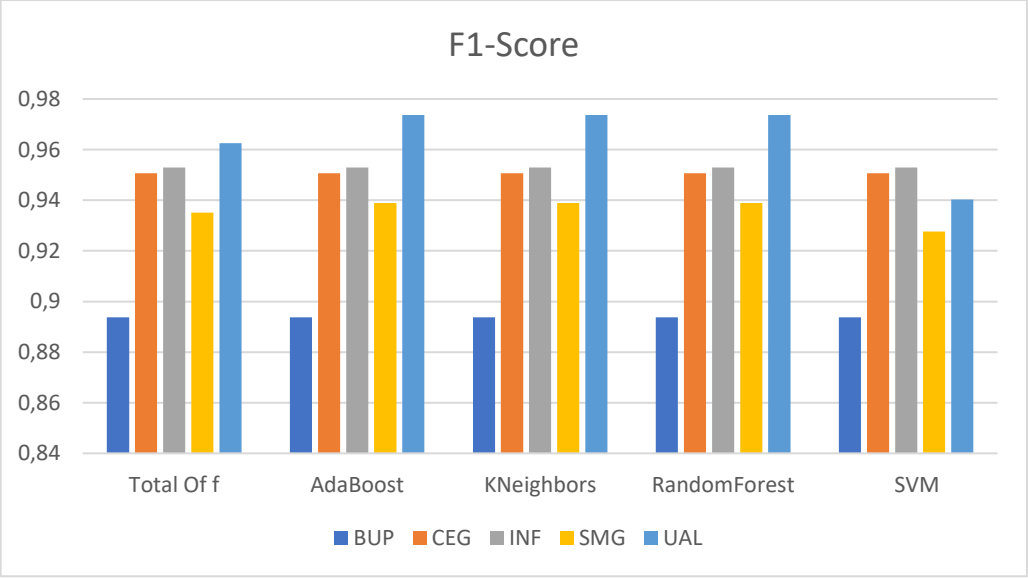


Figure III-25: Diagramme du Résultats de F1-score

Commentaires:

- Les scores F1 sont élevés (> 0,89) pour tous les modèles sur tous les ensembles de données, indiquant un bon équilibre entre précision et rappel.
- AdaBoost, KNeighbors et RandomForest montrent des performances similaires, tandis que SVM a des scores légèrement inférieurs sur SMG et UAL.
- La cohérence des scores F1 sur tous les ensembles montre la fiabilité des modèles dans la prédiction des liens corrects.

III.7.4 Résultat de la comparaison entre les méthodes selon les mesures de performances :

	AdaBoost	Kneighbors	RandomForest	SVM
Précision moyenne	0,899583	0,875384	0,8654243	0,855726
F1-Score moyenne	0,969568	0,943215	0,9120548	0,905441
Accuracy moyenne	0,782548	0,753624	0,7547842	0,714752

Tableau III-06: Résultats de moyenne de mesures de performances de chaque méthode

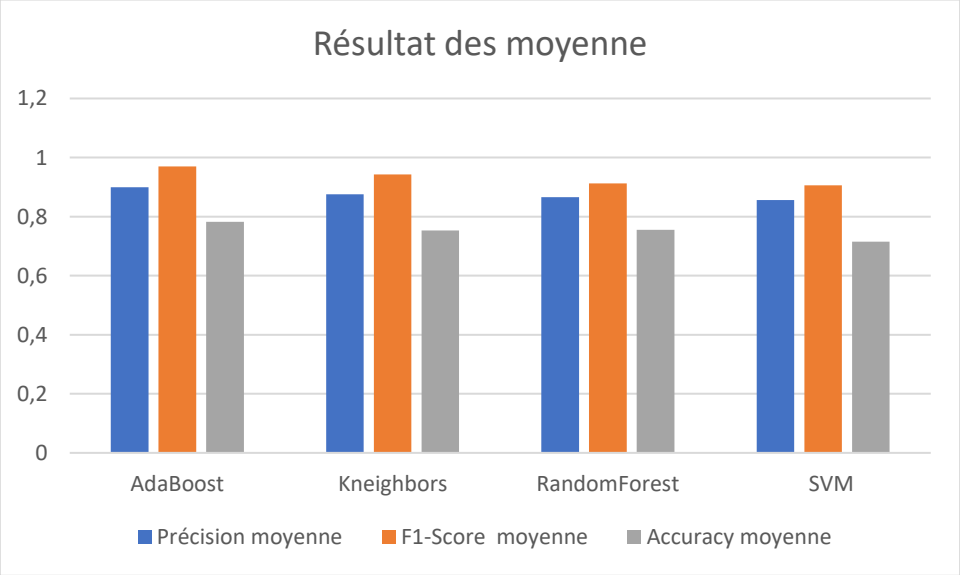


Figure III-26 : Diagramme Résultats de moyenne de mesures de performances de chaque méthode

Accuracy:

- AdaBoost maintient également la meilleure Accuracy moyenne, bien que toutes les méthodes montrent des scores assez élevés.
- KNeighbors et RandomForest ont des Accuracy moyennes proches, légèrement inférieures à celle de AdaBoost.
- SVM montre la plus basse Accuracy moyenne parmi les quatre méthodes, suggérant des résultats plus variables sur différents ensembles de données.

Précision:

- AdaBoost présente la meilleure précision moyenne parmi toutes les méthodes, indiquant sa capacité à minimiser les faux positifs lors de la prédiction des liens.
- KNeighbors et RandomForest suivent avec des valeurs proches mais légèrement inférieures à AdaBoost.
- SVM affiche la précision moyenne la plus basse parmi les quatre méthodes évaluées.

F1-score:

- AdaBoost obtient le meilleur F1-Score moyen, mettant en avant sa capacité à maintenir un équilibre élevé entre la précision et le rappel.
- KNeighbors suit avec un F1-Score moyen robuste, bien que légèrement inférieur à celui de AdaBoost.
- RandomForest et SVM montrent des performances moyennes plus basses en termes de F1-Score, avec SVM ayant la valeur la plus basse parmi les méthodes comparées.

III.8 Résumé des Résultats :

III.8.1 la meilleure mesure de performance

La meilleure mesure de performance parmi Accuracy, précision, et F1-score est le F1-score. Cette mesure est privilégiée car elle offre un équilibre entre la précision et le rappel, essentiels pour évaluer la capacité d'un modèle à prédire avec précision tout en identifiant correctement toutes les instances positives

III.8.2 Meilleur Modèle :

Parmi les modèles évalués pour la prédiction des liens, AdaBoost se démarque comme le choix optimal. Il affiche un F1-score exceptionnel, illustrant sa capacité à maintenir un équilibre optimal entre la précision et le rappel. Cette performance est soutenue par une précision solide et une capacité accrue à prédire avec précision les liens. KNeighbors présente également des performances robustes, bien qu'un peu inférieures à AdaBoost, tandis que RandomForest et SVM montrent des résultats plus variables. En résumé, AdaBoost est recommandé pour sa

capacité à fournir une prédiction précise et fiable des liens, ce qui en fait le choix idéal dans ce contexte analytique.

III.9 Conclusion

En conclusion, les résultats obtenus dans ce chapitre démontrent l'efficacité des algorithmes d'apprentissage automatique pour la prédiction de liens dans des réseaux complexes. Les modèles évalués, à savoir RandomForest, AdaBoost, SVM et KNN, ont tous montré des performances remarquables avec des scores élevés d'accuracy, de précision, et de F1-score sur différents ensembles de données provenant de divers domaines.

En outre, la comparaison des temps d'exécution entre les modèles a souligné l'importance de choisir le bon algorithme en fonction de la taille et de la complexité des données.

Ces résultats fournissent des bases solides pour la poursuite de la recherche dans ce domaine et peuvent avoir des implications pratiques dans de nombreux secteurs

Conclusion générale

Conclusion générale

Dans notre thèse intitulée L'Apprentissage Automatique Pour La Prédiction De Lien Dans Les Réseaux Complexes, nous avons exploré en profondeur le domaine des réseaux complexes et de la prédiction de liens en utilisant des techniques d'apprentissage automatique.

Nous avons examiné les fondements théoriques des réseaux complexes telle que le nombre de nœuds, le nombre de liens, le degré moyen, le coefficient de regroupement, la longueur moyenne du chemin le plus court et d'autres propriétés structurelles, mettant en évidence leur ubiquité dans de nombreux domaines et leur rôle crucial dans la modélisation des interactions entre entités. Cette partie introductive a posé les bases nécessaires à la compréhension des défis et des opportunités associés à la prédiction de liens.

Ensuite, nous avons démontré des méthodes efficaces pour prédire les connexions entre les entités dans divers types de réseaux en utilisant des techniques d'apprentissage automatique, en nous concentrant sur les méthodes supervisées. Nous avons étudié en détail des algorithmes tels que SVM, KNN, AdaBoost et Random Forest, dans laque KNN vise à classer de nouvelles données en fonction de la similarité avec des données existantes, SVM cherche à maximiser la marge entre différentes classes pour une séparation optimale, AdaBoost combine plusieurs classificateurs faibles pour renforcer la performance globale, et Random Forest utilise l'agrégation de multiples arbres de décision pour réduire la variance et améliorer la robustesse des prédictions. Ainsi que les mesures d'évaluation pour quantifier la performance de ces modèles.

Nous avons décrit en détail notre environnement de travail, notre méthodologie de traitement des données et notre approche pour entraîner et évaluer nos modèles. Les résultats de nos expériences ont confirmé la pertinence et l'efficacité de nos approches. Notamment, nous avons constaté que les méthodes SVM et AdaBoost se sont avérées être les plus performantes pour la prédiction de liens dans nos réseaux complexes. Elles ont surpassé les autres modèles en termes de précision, Accuracy et F1-score, démontrant une capacité exceptionnelle à identifier correctement les liens existants et à prévoir les nouvelles connexions potentielles.

En conclusion, cette thèse a été une exploration holistique du domaine de la prédiction de liens dans les réseaux complexes, allant des concepts théoriques aux applications pratiques. En

combinant une compréhension approfondie des réseaux avec des techniques d'apprentissage automatique avancées montre la supériorité des méthodes SVM et AdaBoost, ce travail ouvre de nouvelles perspectives pour la modélisation et l'analyse des interactions dans une grande variété de domaines., représentent une avancée significative dans notre compréhension des réseaux complexes et offrent des pistes prometteuses pour des recherches futures.

Bien que la plupart des objectifs fixés dans cette étude aient été atteints, il reste encore des perspectives et des améliorations potentielles qui pourraient être réalisées à l'avenir telles que:

- **Utiliser une base de données plus grande:** Intégrer un plus grand nombre de réseaux, y compris des réseaux plus complexes, pour évaluer la robustesse et la scalabilité des modèles.
- **Explorer d'autres méthodes d'apprentissage:** Examiner des techniques comme les réseaux de neurones profonds, les méthodes d'ensemble telles que le Gradient Boosting Machines (GBM) et les réseaux de neurones convolutifs pour capturer des relations non linéaires et des structures complexes dans les réseaux.
- **Utiliser d'autres mesures de performance:** Adopter des métriques telles que l'AUC (Area Under the Curve) et le ROC (Receiver Operating Characteristic) pour une évaluation plus complète de la capacité des modèles à prédire les liens dans les réseaux complexes.

Ces pistes de recherche futures contribueront à approfondir notre compréhension et à améliorer la précision des prédictions dans les réseaux complexes, renforçant ainsi l'applicabilité de l'apprentissage automatique dans ce domaine.

Bibliographie

Bibliographie

- Adamic, L. A., Adar, E., 2003. Friends and neighbors on the web. Social Networks, 25 (3), 211-230.
- Hasan, M. A., Zaki, M. J., 2011. A survey on Link Prediction in Social Networks. Social Network Data Analytics. Springer. 243-275.
- Hasan,M.A.,Chaoji,V.,Salem,S.,Zaki,M.,2006.LinkPrediction using Supervised Learning In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security, Counterterrorism and Security, SIAM Datan Mining Conference
- Leicht, E. A., Holme, P., Newman, M. E. J., 2006, Vertex similarity in networks. Physical Review E, 73 (2).
- Barabasi,A.L.,Jeong,H.,Neda,Z.,Ravasz,E.,2002.Evolution of the social network of scientific collaboration. Physica A: Statistical Mechanics and its Applications, 311 (3), 590-614.
- Saramäki,J.;Kivelä,M.;Onnela,J.;Kaski,K.&Kertesz, J.,2007.Generalizations of the clustering coefficient to weighted complex networks. Physical Review, 75 (2).
- Freeman, L. C., 1978. Centrality in social networks conceptual clarification. Social Networks, 1 (3).
- Barrat, A., Barthelemy, M., Pastor-Satorras, R. and Vespignani, A., 2004. The architecture of complex weighted networks. Proceedings of the National Academy of Sciences, 101. 3747-3752.
- Cbwar: Classification de binaires windows via apprentissage par renforcement
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

[2] L. A. N. Amaral, A. Scala, M. Barthelemy et H. E. Stanley, « Classes of small-world networks », Proceedings of the National Academy of Sciences, vol. 97, no 21, 10 octobre 2000, p. 11149–11152 (ISSN 0027-8424 et 1091-6490, PMID 11005838, PMCID PMC17168, DOI 10.1073/pnas.200327197

[3] Complex Networks,Juan Carlos Burguillo , University of Vigo

[4] HEC MONTREAL, Algorithmes parallèles en détection de communautés dans les réseaux complexes ,parPhilippe Gagnon

[7] An Introduction to Graph Theory and Complex Networks ,Maarten van Steen, January 2010

[8]Introduction to Graph Theory" by Douglas B. West, Pearson, ISBN: 978-0130144003

[9]"Graph Theory" by Reinhard Diestel, Springer-Verlag, ISBN: 978-3642173839

[10] "Graphs and Digraphs" by Gary Chartrand, Linda Lesniak, and Ping Zhang, CRC Press, ISBN: 978-1439826255)

[17] Liben-Nowell, D., Kleinberg, J. (2007). "The link-prediction problem for social networks". Journal of the American Society for Information Science and Technology

[20] Pedro G. Lind, Marta C. González, and Hans J. Herrmann. 2005 Cycles and clustering in bipartite networks. Physical Review E (72)

[31] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90-95.

[32] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830

[33] Diestel, R. (2005). Graph Theory (4th ed.). Springer.

[34] Mitchell, T. M. (1997). Machine Learning. McGraw-Hill.

[35] Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press

[35] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

[36] Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences

[37] reiman, L. (2001). Random forests. Machine Learning,

[38] Alpaydin, E. (2020). Introduction to Machine Learning. MIT Press.

[39] Liben-Nowell, D., & Kleinberg, J. (2007). The link prediction problem for social networks. Journal of the American society for information science and technology,

[40] Schölkopf, B., & Smola, A. J. (2002). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press.

Webgraphie

[1]https://aws.amazon.com/fr/whatis/computernetworking/#informatique%,technologies%20physiques fil

[5] https://datascientest.com/theorie-des-graphes-tout-savoir

[6] https://martajv9.wixsite.com/theoriedegraphes/repr%C3%A9sentations-1

[11] Hal open science(https://inria.hal.science/hal-02281775/document#:~:text=La%20pr%C3%A9diction%20de%20liens%20consiste,des%20entit%C3%A9s%20et%20des%20relations.)

[13] DataScientest (https://datascientest.com/algorithme-de-classification-definition-et-principaux-modeles

[15] futura-sciences (https://www.futura-sciences.com/tech/definitions/informatique-python-19349/)

[16] https://docs.anaconda.com/free/navigator/

[18] sciencedirect (https://www.sciencedirect.com/topics/computer-science/closeness-centrality)

[30] Esri https://pro.arcgis.com/fr/pro-app/latest/tool-reference/image-analyst/how-compute-accuracy-for-object-detection-works.htm#:~:text=Courbe%20pr%C3%A9cision%2Drappel%20%2D%20Il%20s,mod%C3%A8le%20de%20d%C3%A9tection%20d'objets