Université Echahid Cheick Larbi Tebessi – Tébessa
Faculté des Science Exactes et Sciences de la Nature et de la Vie
Département des Mathématiques et d'Informatique

Thèse

En vue de l'obtention du diplôme de

Doctorat LMD en

Informatique

Spécialité : **Réseaux des Systèmes Intelligents**

Visualisation interactive des données pour l'analyse et la recherche exploratoire dans un contexte big data

---

Présentée par : **Moustafa Sadek KAHIL**

Directeur de thèse : **Pr. Abdelkrim BOURAMOUL**

Co-directeur de thèse : **Pr. Makhlouf DERDOUR**

Soutenu le : 26 janvier 2023, devant le jury :

Pr. Mohamed AMROUNE	Université Echahid Cheick Larbi Tebessi – Tébessa	Président
Pr. Mohamed Lamine KHERFI	Université Kasdi Merbah – Ouargla	Examinateur
Pr. Mohamed Ridda LAAOUAR	Université Echahid Cheick Larbi Tebessi – Tébessa	Examinateur
Pr. Hakim BENDJENNA	Université Echahid Cheick Larbi Tebessi – Tébessa	Examinateur
Dr. Hichem TALBI	Université Constantine 2 – Abdelhamid Mehri	Examinateur
Pr. Abdelkrim BOURAMOUL	Université Constantine 2 – Abdelhamid Mehri	Rapporteur
Pr. Makhlouf DERDOUR	Université Lari Ben Mhidi – Oum El Bouaghi	Corapporteur

Année Universitaire : **2022 – 2023**

Remerciements

Ce travail n'aurait jamais été mené à terme sans le soutien de Dieu Tout Puissant qui m'a donné la volonté et la force de le réaliser.

J'exprime mes plus sincères remerciements à Pr. Abdelkrim BOURAMOUL et Pr. Makhlouf DERDOUR d'avoir assuré la direction de ce travail et de m'avoir guidé tout le long de ce parcours.

Je tiens également à remercier les membres du jury Pr. Mohamed Amroune, Pr. Mohamed Lamine KHERFI, Pr. Mohamed Ridda LAAOUAR, Pr. Hakim BENDJENNA et Dr. Hichem TALBI d'avoir accepté d'examiner cette thèse.

Enfin, j'adresse ma gratitude à tous ceux qui ont contribué de quelque façon que ce soit à l'aboutissement de ce travail.

Dédicaces

Grâce à Dieu j'ai accompli cette thèse après des années de travail parfois pénible mais toujours exaltant. Je dédie ce modeste effort à mes parents, mes frères et sœurs, qui m'ont soutenu tout au long de mon parcours, mes neveux Ahmed Yacine, Yasmine, Khaoula ainsi que mon beau-frère Boubaker. Je fais un clin d'œil particulier à papa qui m'a accompagné en revoyant la rédaction de la thèse chapitre après chapitre en veillant à l'usage du bon français.

Résumé

Big Data est un phénomène omniprésent dans la vie quotidienne. Il comporte les données qui sont générées tout le temps via différentes manières humaines, matérielles et logicielles. Ces données multidimensionnelles sont caractérisées par le volume dramatiquement élevé, la variété aussi bien en termes de source que de format, ainsi que la vélocité qui traduit la vitesse extraordinaire de leur génération et la nécessité de les stocker et les traiter dans un temps raisonnable. A cause de son implication croissante dans différents domaines et différentes applications de tout genre, Big Data ne cesse d'engendrer, jour après jour, de multiples défis qui nécessitent une considération pour être traités. La visualisation des données présente un axe important de ces défis. Elle consiste à présenter les données graphiquement afin de faciliter aux utilisateurs le processus d'exploration, de recherche et d'analyse. Cependant, malgré la multitude des techniques et des outils de visualisation, les dimensions de Big Data, connues sous le terme : 3Vs, rendent le processus de visualisation difficile à réaliser sans avoir recours à des méthodes modernes pour supporter ces dimensions.

Cette thèse vise à traiter le problème de visualisation des données en considérant deux volets, à savoir la visualisation des données et l'analytique visuelle. Le premier volet vise à améliorer la visualisation des données volumineuses et massives tout en considérant la satisfaction des contraintes de visualisation, notamment l'interactivité, la mise en échelle et la dynamique des données. Le second volet a pour objectif de considérer l'aspect de préférences de visualisation. Cet aspect, largement répandu au sein de la communauté scientifique, s'intéresse à l'amélioration de l'expérience utilisateur pour permettre à ce dernier d'explorer et/ou d'analyser graphiquement les données massives d'une manière consistante selon ses intérêts et orientations.

Dans le premier volet, deux approches ont été proposées dont une permet de préparer les grands data-sets à la visualisation interactive structurée et l'autre offre une méthode de visualiser les graphes à grande échelle d'une manière effective tout en assurant la mise en échelle et la personnalisation. Ces deux approches ont montré une consistance en termes d'effectivité, de rapidité et de faible complexité. Dans le second volet, une approche proposée consiste à assimiler le problème de visualisation exploratoire comme un problème de recommandation. Cela permet de considérer l'aspect de préférences des utilisateurs afin de leur recommander les données qui sont susceptibles de les intéresser et sous les formes graphiques qui leur sont adéquates. Cette approche, basée sur la régression, a montré, elle aussi, une performance remarquable face aux métriques d'évaluation, à savoir la perte, MSE, MAE et R-carré.

Mots clefs :

Big Data, mise en échelle, visualisation interactive, recherche exploratoire, analyse, analytique visuelle

Abstract

Big Data is a ubiquitous phenomenon in daily life. It comprises the multidimensional data which is generated all the time via different human, hardware, and software ways. This data is characterized by the dramatically high volume, the variety both in terms of source and format, and the velocity which reflects the extraordinary speed of data generation and the need to store and process it in a reasonable time. Because of its growing involvement in different fields and applications, Big Data continues to generate, day after day, multiple challenges that require to be addressed. Data visualization presents an important focus among these challenges. It consists in presenting data graphically in order to facilitate to users the process of exploration, research and analysis. However, despite the multitude of visualization techniques and tools, the dimensions of Big Data, known as 3Vs, make the visualization process difficult to achieve without resorting to modern methods in order to support these dimensions.

This thesis aims to address the problem of data visualization by considering two sides, namely data visualization and visual analytics. The first side aims to improve the visualization of large and massive data while considering the satisfaction of constraints related to it, in particular interactivity, scalability and data dynamics. The second side aims to consider the aspect of viewing preferences. This aspect, widespread within the scientific community, is concerned with improving the user experience in order to allow the latter to explore and/or graphically analyse massive data in a consistent manner according to their interests and orientations.

In the first side, two approaches have been proposed: the first one allows to prepare large data-sets for structured interactive visualization, and the second one offers a method of visualizing large-scale graphs in an effective way while ensuring scalability and customization of the graphical presentation. Both approaches showed effectivity in terms of rapidity and low complexity. In the second side, a proposed approach based on regression consists in assimilating the exploratory visualization problem as a recommendation problem. This makes it possible to consider the aspect of user preferences in order to recommend to them the data that are likely to interest them and in the appropriate visualizations. This approach also showed good performance in terms of loss, MSE, MAE, and R-square.

Keywords:

Big Data, Scaling, Interactive Visualization, Exploratory Research, Analysis, Visual Analytics

ملخص

البيانات الضخمة ظاهرة منتشرة في كل مكان في الحياة اليومية. وهي تشتمل على البيانات التي يتم إنشاؤها طوال الوقت عبر طرق بشرية وأجهزة وبرامج مختلفة. تتميز هذه البيانات متعددة الأبعاد بالحجم الكبير والتنوع من حيث المصدر والشكل، فضلاً عن سرعة توليدها الكبيرة والحاجة إلى تخزينها ومعالجتها في وقت معقول. بسبب استعمالات البيانات الضخمة المتزايدة في مختلف المجالات، تظهر تحديات متعددة تتطلب المعالجة. يمثل العرض البصري للبيانات جزءاً مهمًا لهذه التحديات. ويرتكز هذا الأخير على تقديم البيانات بشكل مرئي من أجل تسهيل عملية الاستكشاف والبحث والتحليل للمستخدمين. إلا أنه على الرغم من تعدد تقنيات وأدوات التصور، فإن خصائص البيانات الضخمة، تجعل عملية العرض البصري صعبة التحقيق في حال عدم اللجوء إلى الأساليب الحديثة.

تهدف هذه الأطروحة إلى معالجة مشكلة العرض البصري للبيانات من خلال اعتبار جانبين: عرض البيانات البصري والتحليلات المرئية. يهدف الجانب الأول إلى تحسين عرض البيانات الضخمة بطريقة مرئية مع مراعاة تلبية القيود المفروضة، ولا سيما التفاعل والتدرج وديناميكية البيانات. في حين يهدف الجانب الثاني إلى معالجة جانب تفضيلات العرض البصري بالنسبة للمستخدمين. وهذا الجانب الذي يشغل حيزا معتبرا في المجتمع العلمي يهتم بتحسين تجربة المستخدم لتمكينه من استكشاف وتحليل البيانات الضخمة بطريقة مرئية متسقة وفقًا لاهتماماتهم وتوجهاتهم.

في الجانب الأول تم اقتراح طريقتين، إحداهما تمكن من تهيئة البيانات الضخمة للعرض التفاعلي المنظم والأخرى تقدم طريقة للعرض المرئي للبيان، خصوصا الشبكات المعقدة، بطريقة فعالة مع ضمان التدرج وتخصيص العرض البياني. كلتا الطريقتين أظهرتا فعالية معتبرة من حيث السرعة وانخفاض التعقيد. بينما تهدف الطريقة المقدمة في الجانب الثاني إلى استيعاب مشكلة العرض المرئي الاستكشافي عن طريق استعمال أنظمة الاقتراح وبالاعتماد على نموذج انحدار. وهذا يمكن من استعمال تفضيلات المستخدمين من أجل اقتراح البيانات التي من المحتمل أن يهتموا بها، وعن طريق النماذج الرسومية المناسبة لهم. وقد أظهرت هذه الطريقة كذلك جدواها من خلال النتائج الجيدة التي تم تقييمها حسب معايير تقييم نماذج الانحدار.

الكلمات الدالة: البيانات الضخمة، التدرج، العرض المرئي التفاعلي، البحث الاستكشافي، التحليل، التحليلات المرئية

Table des Matières

Liste des figures

Liste des tables

Liste des algorithmes

Introduction générale

Mise en contexte

Big data désigne un phénomène relativement récent qui reflète l'explosion des volumes des données et qui est largement répandu dans différents domaines tels que la médecine, le commerce électronique, les réseaux sociaux, l'astronomie, etc. Ces données, souvent hétérogènes, connaissent une croissance exponentielle de volumes avec leur flux important. Cela remet encore une fois en cause les problèmes de stockage et de traitement de manière efficace, notamment face aux exigences du temps réel (M. Chen et al., 2014; Padgavankar & Gupta, 2014; Schintler & McNeely, 2022). En effet, la dépendance des données, qui sont considérées aujourd'hui comme « le nouveau pétrole », au sein de différentes industries est à l'origine d'une telle explosion. Leur exploitation est de plus en plus indispensable pour les entreprises et les compagnies afin de prendre des décisions correctes qui rassurent les processus subséquents, notamment le « Business Intelligence » (Lee, 2017).

En revanche, la visualisation des données est définie comme le processus de les présenter graphiquement à travers différentes techniques de présentation visuelle d'une manière efficiente, structurée et accessible à l'utilisateur afin qu'il comprenne les informations qu'elles véhiculent et pour lui simplifier les processus d'analyse et de recherche exploratoire (Ali et al., 2016; Andrienko et al., 2020; Godfrey et al., 2016). Classiquement, l'objectif de visualisation se limitait à mapper des objets graphiques à des données. Aujourd'hui, dans un ensemble de données volumineuses, le processus de visualisation consiste plutôt à en sélectionner les plus consistantes selon des critères précis. L'aspect d'interactivité est, lui aussi, très considéré de nos jours. Il consiste à introduire l'utilisateur dans la visualisation comme une partie intégrante à travers laquelle la présentation visuelle peut changer. Cela donne une dynamique aux systèmes de visualisation ; tout utilisateur peut interagir avec la visualisation via des fonctionnalités qui lui sont fournies. Ces dernières assurent la personnalisation de la présentation des données selon l'intérêt de chaque utilisateur (Dimara & Perin, 2020; Godfrey et al., 2016). Elles peuvent se manifester dans différentes formes telles que les paramètres de sélection, de filtrage et de recherche. Par conséquent, la visualisation interactive peut se voir comme une intersection de différentes disciplines, à savoir la présentation graphique, l'interaction homme/machine et la recherche d'information.

Problématique

Les données dans le phénomène Big Data se caractérisent par le volume important dû à la génération quotidienne des données qui dépassent les 200 exaoctets par jour[1]. Ces données, venues de différentes sources, sont de formats hétérogènes qui peuvent être structurés, semi-structurés ou non-structurés. Toutes ces caractéristiques ont engendré plusieurs enjeux qui touchent différents axes et qui affectent l'extraction de valeur à partir des données, d'où la question clef : les données sont omniprésentes, mais comment en extraire de la valeur ? La visualisation interactive des données, elle aussi, fait partie de ces enjeux. Au-delà des techniques et des outils de visualisation classiques et modernes, le défi est comment visualiser les données hétérogènes de tels volumes dans un temps raisonnable tout en satisfaisant les contraintes de visualisation, en considérant l'aspect multidimensionnel de la mise en échelle et en respectant toutes les dimensions de l'aspect utilisateur, notamment l'interactivité et les préférences (Nair et al., 2016; Ward et al., 2015). De plus, l'Analytique Visuelle (Visual Analytics) est récemment largement considérée (Andrienko et al., 2020; Zgraggen et al., 2017), peut-être plus que la visualisation. En effet, elle comprend la visualisation des données comme une tâche de son processus. Ce dernier est considéré comme tout un cycle qui a pour objectif de visualiser les informations sur la base d'une séquence d'étapes d'analyse qui finit par leur extraction et leur présentation graphique. La visualisation n'est alors qu'une tâche dans ce cycle. A cet effet, l'intérêt de visualisation des données dans le contexte

[1] https://www.statista.com/statistics/871513/worldwide-data-created/

Big Data s'est élargie pour couvrir, non seulement une tâche de mappage de données avec des formes graphiques, mais tout un processus d'extraction d'informations et de connaissances (Andrienko et al., 2020; M. Chen et al., 2014; Fiaz et al., 2016). Par conséquent, d'autres problèmes apparaissent et touchent différents aspects, notamment la visualisation des données multidimensionnelles, la visualisation des structures complexes de manière efficace, ainsi que la prise en considération des préférences de l'utilisateur. Les solutions à ces problèmes doivent satisfaire la dynamique des données qui est soumise aux contraintes de mise en échelle et de temps réel. En effet, il existe de nombreuses solutions qui servent à visualiser les data-sets multidimensionnels (Dash et al., 2008; Qin et al., 2018; Soylu et al., 2013), mais dont la complexité est élevée ou qu'elles ne fournissent pas de visualisation structurée. De même, d'autres solutions et des outils visent à analyser et à présenter graphiquement les structures complexes basées graphe tout en satisfaisant les contraintes de mise échelle et de temps réel (Agrawal et al., 2015; Ali et al., 2016; Raghav et al., 2016), sauf que les visualisations qu'elles fournissent peuvent être incompréhensibles par les utilisateurs à cause du nombre important de leurs composants. Finalement, plusieurs travaux offrent des mécanismes pour considérer l'aspect de préférences des utilisateurs afin de les assister pendant l'exploration visuelle (Golfarelli & Rizzi, 2019; Qin et al., 2018; Soylu et al., 2013). Sauf qu'ils nécessitent beaucoup d'expertises pour être généralisés, d'autant plus qu'ils exigent des ressources de calcul importantes afin de satisfaire les contraintes de mise en échelle et de temps réel.

Contributions

Sur la base des problématiques issues de la visualisation des données dans le contexte Big Data, cette thèse vise à proposer des solutions pour y remédier. A cette fin, trois contributions ont été proposées pour considérer trois axes qui concrétisent ces problématiques. La première contribution (Kahil et al., 2021a) considère les problèmes de visualisation des grands data-sets de manière structurée, interactive et de faible complexité. Différentes applications s'inscrivent dans ce contexte ; les data-sets, notamment les semi-structurés, ne nécessitent pas toujours d'être analysés à travers des modèles complexes en termes ressources et de temps de calcul. A cet égard, il serait préférable qu'ils soient traités via des techniques moins complexes. Pour cela, cette contribution vise à les prétraiter via une approche moins complexe basée sur l'heuristique gloutonne. Cette approche permet d'extraire les patterns et les organiser pour les visualiser de manière structurée, hiérarchique si possible.

La deuxième contribution (Kahil et al., 2019, 2021b) véhicule une nouvelle approche pour visualiser les données qui sont présentées dans une structure de graphe. En effet, les graphes à grande échelle occupent un grand espace dans le phénomène de Big Data ; ils existent dans les réseaux sociaux, les plateformes académiques, les réseaux d'énergie, etc. Ces graphes sont dotés de nombres si importants de nœuds et de relations qu'il est impossible de les visualiser sans avoir recours à une méthode qui organise leur présentation. Une manière d'y remédier est de visualiser séparément les communautés qui composent un graphe à grande échelle. Ainsi, le nombre de nœuds et de relations à visualiser dans un temps précis est réduit et leur visualisation est, par conséquent, possible. L'approche proposée assure la distribution du processus de détection des communautés afin de réduire le temps de calcul et offre un mécanisme d'interactivité à travers lequel l'utilisateur peut personnaliser la visualisation des communautés selon ses besoins et intérêts.

La troisième contribution (Kahil et al., In press) cible l'aspect de préférences de l'utilisateur en matière de techniques de visualisation et de patterns qui l'intéressent pendant le processus d'exploration visuelle. A cette fin, une approche proposée consiste à considérer le problème de visualisation pour la recherche exploratoire comme un problème de recommandation. Les systèmes de recommandation, très répandus dans différentes applications telles que l'e-commerce, ont pour objectif de fournir à chaque utilisateur

les éléments de données susceptibles de l'intéresser. Ce concept est adopté par l'approche proposée pour recommander à chaque utilisateur aussi bien les patterns de données que les formes de visualisation qui pourraient l'intéresser. Pour cela, cette approche propose trois solutions alternatives dont la sélection est faite selon le domaine d'application. Les trois alternatives comprennent une solution brute, une solution d'apprentissage automatique et une solution basée sur les réseaux de neurones profonds.

Structure de la thèse

La structure générale de cette thèse couvre deux grandes parties composées chacune de trois chapitres. La première partie est consacrée à la présentation des concepts théoriques liés à l'objet de notre thèse. Le premier chapitre introduit le concept de Big Data avec ses différentes dimensions, à savoir la signification des différentes Vs qui le caractérisent, la classification des données dans le contexte Big Data, les enjeux majeurs de cette technologie, l'aspect d'échelonnabilité qui représente un de ses piliers, sa relation avec les technologies contemporaines, ainsi que les systèmes et les plateformes de stockage et de traitement qui marquent cette technologie. Le deuxième chapitre présente de multiples aspects de la fouille des données qui contient l'ensemble de méthodes et techniques utilisées pour résoudre les différents problèmes de Big Data. Pour cela, ce chapitre introduit les différentes techniques utilisées dans la phase de pré-traitement des données, y compris le nettoyage et la préparation, pour présenter en fin de compte des techniques et des méthodes utilisées dans l'étape de fouille des données. Le troisième chapitre définit la visualisation comme une étape du processus d'extraction de connaissances à partir des données. Pour cela, il introduit son concept, y compris les conventions qui lui sont liées et les contraintes de visualisation qui représentent les mesures de validation de toute solution de visualisation. Il cite également différents outils et techniques de visualisation classiques et modernes et met en évidence des exemples de leur application. Après cela il projette le concept de visualisation sur le phénomène de Big Data pour déterminer les problèmes et les défis qui surgissent dans ce contexte.

La seconde partie, contenant elle-aussi trois chapitres, véhicule une présentation des contributions proposées pour traiter les problèmes de visualisation selon les axes définis. Le quatrième chapitre, présentant la première contribution, vise à préparer les data-sets volumineux pour la visualisation multidimensionnelle. Comme l'approche proposée est basée sur l'heuristique gloutonne, ce chapitre introduit le concept porté par cette heuristique, puis l'applique sur le problème de sélection des patterns à visualiser afin de réduire la complexité de cette tâche et de garantir une visualisation structurée, interactive et personnalisable. Il compare l'approche proposée avec des travaux qui s'inscrivent dans le même contexte selon des critères désignés pour montrer son applicabilité et sa performance. Le cinquième chapitre présente la deuxième contribution qui vise le traitement du problème de visualisation des graphes à grande échelle à travers la détection des communautés. A cet effet, il introduit ce concept avec ses différentes dimensions puis il décrit l'approche proposée pour distribuer le processus de détection des communautés afin de l'accélérer. Enfin, il applique une architecture proposée pour la visualisation multi-niveau des graphes et des différentes structures hiérarchiques. Le sixième chapitre décrit l'approche proposée pour considérer le problème de visualisation exploratoire des données comme un problème de recommandation. A cette fin, il véhicule une étude du concept de systèmes de recommandation et des méthodes employées pour résoudre leurs problèmes. Après quoi, en assimilant le problème de visualisation exploratoire au problème de recommandation, il présente l'approche proposée pour résoudre ce problème conformément à une architecture qui assure le déroulement fiable de ce processus. La conclusion générale résume les contributions proposées ainsi que les perspectives et les futurs travaux liés à chacune de ces contributions.

Partie I: Concepts théoriques de big data, fouille des données et visualisation interactive

Chapitre 1: Big Data – aperçu sur les concepts et les technologies

1 Introduction

La prise des décisions basée sur les données (data-driven decisions) a développé de nouveaux horizons de considérer les données. Auparavant, ces dernières étaient souvent d'importance limitée qui peut se résumer dans le stockage des informations sur le personnel d'une compagnie, l'analyse de quelques phénomènes de manière séparée, des études superficielles sur l'impact de quelques facteurs sur le déroulement d'un processus donné, etc. De nos jours, ces données représentent le pilier de toute une nouvelle technologie appelée Big Data et de toute une nouvelle science appelée science des données. Leur utilisation peut être associée à l'étude approfondie de tout un terrain d'intérêt, voire l'étude de l'intersection de plusieurs domaines éloignés. Business Intelligence consiste à sélectionner les décisions qui soient les plus rentables à un business à partir de l'analyse des données et l'extraction des connaissances à partir de celles-ci (Iqbal et al., 2016; Schintler & McNeely, 2022). Elle représente le chemin incontournable qu'empruntent les entreprises de nos jours afin de garder leur vivacité. Cela a donné un intérêt particulier aux données et aux méthodes de manipulation des données qui, à force de se multiplier de volume et de gagner de nouvelles caractéristiques, ont engendré de multiples défis nécessitant de nouvelles méthodes de stockage, de traitement et d'analyse. Tout cela a été l'acteur principal de l'apparition de Big Data.

L'utilisation de Big Data ne cesse de s'étendre jour après jour dans l'industrie et dans les différents domaines. Ceci a certainement affecté le marché de Big Data dont le revenu a atteint 15 milliards $ en 2019, et il est prévu qu'il continue de croitre d'environ 30% chaque année pour dépasser les 68 milliards $ dans l'année 2025[1]. La réussite dans un tel domaine n'est pas acquise sans cause ; le volume des données a connu une croissance exponentielle qui a dépassé 64.2 zettaoctets dans l'année 2020[2], et 79 zettaoctets en 2021[3], avec une moyenne qui dépasse 216 exaoctets de données créées et/ou répliquées par jour. A évoquer l'exemple de Covid 19, il est constaté que les données ont été et continuent d'être nécessaires pour différents objectifs (Alsunaidi et al., 2021; Schintler & McNeely, 2022). On peut en citer les majeurs, à savoir les statistiques selon les catégories sociales pour déterminer les plus exposées et les plus vulnérables au virus, les conditions aériennes des endroits où ce dernier se propage le plus, le nombre des cas positifs en temps réel, etc. Toutes ces nouvelles caractéristiques des données qui marquent le phénomène de Big Data ont engendré différents enjeux, issus des exigences qui continuent d'apparaitre avec son émergence sur différentes applications. Par ailleurs, son intersection avec les technologies contemporaines représente, elle-aussi, un facteur important de ces enjeux. Parmi ces technologies on peut citer les ontologies, le cloud computing, les data centres, l'internet des objets, le Blockchain et l'apprentissage automatique. Les ontologies permettent de trouver des modèles schématiques pour les données de multiples applications d'entreprise en en extrayant les pertinentes et les intégrant dans un graphe sémantique (Schintler & McNeely, 2022). Cela permet d'alléger le processus de recherche de l'information depuis les différentes sources. (Eine et al., 2017; Konys, 2017). Les data centres, avec leurs différentes architectures telles que SAN, NAS et DAS (Padgavankar & Gupta, 2014; Patgiri, 2019; Schintler & McNeely, 2022), sont devenus de plus en plus indispensables pour les applications Big Data ; ils assurent le stockage, la sauvegarde et la récupération des données volumineuses en temps réel. IoT est caractérisé par des milliers de périphériques qui s'interagissent et génèrent des volumes importants de données. Cela crée des défis qui sont directement liés à Big Data à savoir la collecte, le stockage et l'analyse des données volumineuses et hétérogènes dans des délais raisonnables (Arulkumar et al., 2019; Patgiri, 2019; Schintler & McNeely, 2022). La sécurité pendant la collecte et le traitement des données elle-aussi représente un défi à ne pas négliger. L'emploi de

[1] https://www.statista.com/statistics/947745/worldwide-total-data-market-revenue/
[2] https://www.statista.com/statistics/871513/worldwide-data-created/
[3] https://www.analyticsinsight.net/top-10-big-data-statistics-you-must-know-in-2021/

Blockchain peut y remédier (Deepa et al., 2022). Les solutions modernes des différents problèmes d'analyse et d'analytique dans Big Data reposent principalement sur les modèles d'apprentissage automatique (Machine Learning), y compris l'apprentissage profond (Deep Learning) (Q. Zhang et al., 2018).

Le reste de ce chapitre est organisé comme suit : La deuxième section présente les notions générales liées à Big Data qui couvrent sa définition, ses caractéristiques, ses enjeux et opportunités, sa classification selon différents critères, la propriété de mise en échelle et les notions d'analyse et d'analytique dans ce phénomène. La troisième section introduit le concept de NoSQL qui représente les bases de données modernes utilisées pour le stockage des données Big Data. Elle énumère et illustre également les types de ces bases de données et leurs caractéristiques. La quatrième section décrit les politiques de traitement à adopter dans un contexte Big Data. La cinquième section discute les plateformes de Big Data les plus répandues qui composent l'écosystème de Hadoop, ce dernier étant la concrétisation représentative du cycle de Big Data. La sixième section met en évidence le concept d'analytique dans le contexte Big Data et sa différence avec l'analyse classique. La septième section conclut ce chapitre.

2 Big Data : concepts, classification et technologies

Big Data est le terme qui reflète l'explosion des données qui a marqué les années récentes dans le monde entier. Ceci est le résultat de dépendance indispensable des données dans les différents secteurs industriels, économiques, scientifiques, médicaux, commerciaux, etc. Ce concept fait référence aux données volumineuses de nature complexe qui ne peuvent pas être traitées via les techniques et systèmes traditionnels de Business intelligence et d'analyse (Oussous et al., 2018; Patgiri, 2019). Il repose alors sur de nouvelles visions, valeurs économiques et découvertes traduites par des techniques et systèmes avancés qui ont été développés pour supporter les nouvelles caractéristiques des données et considérer les défis qui leur sont liés. Big Data est généralement défini par les 3Vs qui le caractérisent et qui sont décrits ci-dessous.

2.1 Dimensions de Big Data

La notion des 3Vs fait référence aux grandes caractéristiques qui ont marqué données dans sur lesquelles est connu le phénomène de Big Data depuis son apparition, à savoir : le volume, la variété et la vélocité. Aujourd'hui, avec la révolution qu'ont connu les solutions de stockage et de traitement, ces trois challenges ne représentent plus les seuls piliers de Big Data. Il est même devenu un consensus au milieu scientifique que l'existence de juste deux de ces challenges est suffisant pour qualifier qu'une situation est de Big Data (Gorodov & Gubarev, 2013). Par ailleurs, d'autres aspects, traduits par d'autres Vs, ont apparu avec l'émergence de Big Data dans de nombreuses applications. Et ce phénomène a vécu une transition de 3Vs à 7Vs, 10Vs et même 13Vs. La présente section est essentiellement consacrée à la description des 3Vs, mais présente aussi les autres aspects des 10Vs de manière superficielle.

2.1.1 Volume

Le volume des données a inscrit une croissance impressionnante qui a atteint 79 zettaoctets en 2021. Ce chiffre devrait continuer d'augmenter pour atteindre 181 zettaoctets dans l'année 2025. Elles sont issues de différentes sources telles que les technologies contemporaines, les entreprises, les interactions humaines, etc. (N. Khan et al., 2018). La propriété du volume est devenue un enjeu très considéré qui a engendré des problèmes qui touchent aussi l'axe du stockage que du traitement (N. Khan et al., 2018; McAfee et al., 2012). L'agrégation des données est devenue nécessaire et indispensable pour remédier à ces défis.

2.1.2 Variété

La variété des données est la propriété qui reflète la nature hétérogène de celles-ci en termes de leurs sources et leurs formats.

Source des données

L'interconnexion des différentes technologies a engendré une multitude des sources des données qui caractérisent Big Data. Les données peuvent provenir des capteurs et différents périphériques, du web, des stations météorologiques, etc. (Kahil et al., 2020; Oussous et al., 2018). Plus que jamais avec l'Internet des Objets (IoT : Internet of Things), les capteurs sont omniprésents. Ils reçoivent les signaux de toute nature (température, mouvement, sons, …) et les transforment en données pour que ces dernières soient exploitées pour différentes fins. Le web représente une source importante, si ce n'est la plus importante, des données qui marquent Big Data. Les données sont récupérées partout des sites web, des forums de discussion, des plateformes d'e-commerce, des réseaux sociaux, etc. (Fan et al., 2014; Oussous et al., 2018). Une bonne partie des données web est récupérée depuis les différents types des journaux (logs) (Marx, 2013; Sagiroglu & Sinanc, 2013; Schintler & McNeely, 2022). Ces données peuvent être exploitées pour de divers objectifs : traçabilité, extraction des informations pour recommander des services, …

Formats

Les données peuvent venir sous différents formats à savoir structurées, semi-structurées, non-structurées et multi-structurées (Kahil et al., 2019; Qi, 2020; Sagiroglu & Sinanc, 2013; Tariq RS, 2015). Tous ces formats peuvent arriver à la fois comme entrée au même système et doivent être traitées ensemble.

Structurées

Les données structurées sont celles qui sont arrangées selon un schéma déterminé et fixé qui est représenté par un modèle tabulaire composé de colonnes et de lignes. Les bases des données relationnelles sont l'exemple de cette classe des données. Ces qui caractérise les données structurées est qu'elles sont faciles à manipuler comparées aux autres classes des données. Parmi les opérations utilisées pour cette manipulation sont la projection, la sélection et la jointure. Les bases des données relationnelles, bien que classiques, présentent une solution de stockage pour de multiples applications.

Semi-structurées

Les données semi-structurées sont celles qui à la fois ne suivent pas un schéma fixé, mais qui ne sont pas brutes ou complètement non-structurées non plus. Elles contiennent généralement des éléments structurés tels que les métadonnées et les balises organisationnelles qui permettent de simplifier l'analyse et mettre en échelle les données. Parmi les formats des données semi-structurées sont CSV, TSV, JSON, XML, HTML, Parquet et les graphes.

Non structurées

Les données non-structurées ne suivent aucun modèle ou schéma et qui ne peut, par conséquent, pas être stockées dans les bases de données relationnelles. Parmi les données non-structurées les plus connues sont les données textuelles et les données multimédia telles que les vidéos, les images, l'audio, les données scientifiques, les données météorologiques, etc.

Données Multi-structurées

Les données multi-structurées sont celles qui peuvent contenir simultanément tous les formats des données précédemment cités, c-à-d. structurées, semi-structurées et non-structurées. Ce type de données, caractérisant le plus l'ère de Big Data, est le produit direct de l'intersection des technologies qui a engendré une ubiquité marquante. Parmi les exemples de ce type de données sont les pages web

dynamiques et les réseaux sociaux qui peuvent contenir à la fois des données multimédia et textuelles (non-structurées), des tables (structurées) et des data-sets et codes html, xml, … (semi-structurées).

2.1.3 Vélocité

La vélocité est un des piliers de Big Data qui continue de représenter un défi très considéré aujourd'hui. Elle est relative à l'aspect du temps qui reflète la rapidité d'acquisition, de stockage et de traitement des données conformément aux spécificités de ces dernières qui peuvent être générées continument en temps réel et aux exigences de l'application (N. Khan et al., 2018; Tariq RS, 2015). Ces spécificités reflètent essentiellement la durée de vie des données dont la valeur peut se limiter voire se perdre avec le temps. La vélocité a donc un impact direct sur la rapidité du processus de prise de décision

De nos jours, le nombre de challenges de Big Data à considérer continue d'augmenter avec l'émergence de multiples défis à cause de la propagation des technologies interconnectées. Par conséquent, les 3Vs ne sont plus les seuls à prendre en considération ; d'autres Vs ont apparu. Dans la communauté scientifique, on parle des 5Vs, 7Vs, 10Vs, 13Vs, … Dans ce qui suit, les autres challenges qui font partie des 10Vs, montrés dans **Figure 1-1**, sont brièvement décrits.

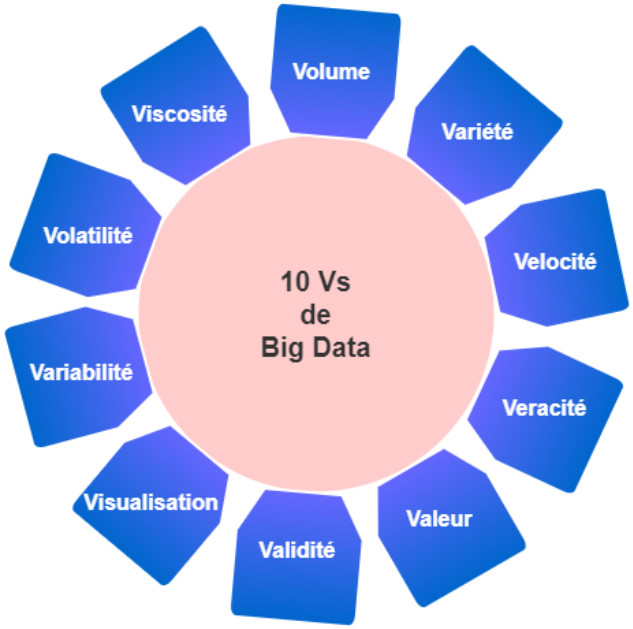

Figure 1-1 : 10Vs de Big Data

2.1.4 Véracité

La véracité des données collectées se concentre sur la qualité et l'exactitude, ainsi qu'elle définit le niveau de confiance de celles-ci vis-à-vis la prise de décision sur leur base. Selon le facteur de véracité, les données peuvent être bonnes, mauvaises ou indéfinies en fonction de la manière de leur collecte. Cela peut présenter l'inconsistance, l'incomplétude, l'ambigüité, l'approximation, etc. (N. Khan et al., 2018; Lee, 2017; Tariq RS, 2015).

2.1.5 Validité

Bien que la validité soit confondue avec la véracité, ces deux concepts sont différents. Il se peut que l'une de ces deux fasse l'objet d'un défis sans que l'autre ne soit concernée. La validité des données implique que ces dernières sont passables du statut exploratoires aux statut actionnables pour assurer une prise de données correcte et efficace (N. Khan et al., 2018; Lee, 2017). Elle est proportionnelle à l'application ; les données peuvent être valides pour une application et invalides pour une autre.

2.1.6 Visualisation

La visualisation des données dans le contexte Big Data présente un défi important à explorer. Cette importance est liée à l'impact qu'elle peut véhiculer pour les analystes dans le processus d'analyse et d'exploration des données. D'abord, avec les données de quantités importantes, il est impossible de les visualiser toutes à la fois. Ensuite, la multitude des techniques de visualisation rendent difficile d'en choisir les adéquates pour un objectif fixé. Enfin, des contraintes liées à la visualisation doivent être satisfaites afin d'organiser la présentation des données aux utilisateurs à temps tout en assurant l'interactivité avec ceux-ci (Andrienko et al., 2020; Gorodov & Gubarev, 2013; Kahil et al., 2020; Schintler & McNeely, 2022).

2.1.7 Variabilité

La variabilité reflète l'inconsistance de flux des données durant le processus de collecte. Elle peut être consultée selon le temps (N. Khan et al., 2018; Schintler & McNeely, 2022). Cet axe a attiré plus d'attention avec l'accroissement d'utilisation des multimédia qui sont devenus aujourd'hui une partie intégrante de la vie quotidienne pour toutes les catégories sociales.

2.1.8 Volatilité

Comme le temps réel est devenu une caractéristique de Big Data, la volatilité est relative à la durée de vie des données en termes de stockage et validité de ces dernières (N. Khan et al., 2018; Roy et al., 2020; Schintler & McNeely, 2022). Selon le domaine d'application, il est essentiel de définir le point de temps après lequel les données ne sont plus pertinentes à un processus d'analyse. Pour cela, des règles sont à déterminer pour considérer la disponibilité des données conformément au flux important de ces dernières et aux exigences du temps de l'application en question.

2.1.9 Viscosité

Cette propriété est relative au degré de corrélation et d'interdépendance entre les différentes structures de Big Data. Ces dernières sont le résultat de la collecte des données depuis de multiples sources qui sont souvent hétérogènes. L'importance de la viscosité surgit du fait que les petits changements au niveau de ces corrélations peuvent causer d'importants effets sur le comportement des systèmes d'analyse (N. Khan et al., 2018; Roy et al., 2020; Schintler & McNeely, 2022).

2.1.10 Valeur

La valeur, traduisant l'objectif primordial de tout processus de d'analyse des données, fait référence à l'utilité des données analysées et leur impact sur la décision prise lors de ce processus (Hashem et al., 2015; N. Khan et al., 2018; Roy et al., 2020; Schintler & McNeely, 2022). Elle repose sur l'exactitude de l'analyse des données et est mesurée en comparant les résultats de cette dernière avec d'autres processus d'analyse.

2.2 Classification de Big Data

Les données dans le contexte Big Data sont classifiées selon différents critères conformément aux spécificités de ces données et aux applications. Parmi ces derniers on trouve (Hashem et al., 2015; Kahil et al., 2020) (1) la source des données, (2) le format des données, (3) le type de stockage utilisé dans l'application en question, (4) le type du traitement adopté et (5) l'objectif d'analyse assigné.

Comme est déjà mentionné, Big Data est caractérisé par multiples sources de données issues de l'intersection des différentes technologies. Les pages web, IoT, les machines, les capteurs, les transactions et les sites d'e-commerce sont des exemples des sources données Big Data. Le format des données reflète la propriété de variété des données qui représente la 2ème V de Big Data. Les données sont alors classifiées en structurées, non-structurées, semi-structurées ou peuvent contenir ces trois types

à la fois. NoSQL est le type des bases de données fréquemment utilisé pour le stockage des données Big Data. Quatre classes de bases de données NoSQL peuvent être distinguées : (1) les BDD orientées document, (2) orientées colonne, (3) basées graphe et (4) basées clef-valeur (Bhogal & Choksi, 2015; Davoudian et al., 2018; Kahil et al., 2020; Storl et al., 2015). Selon le type du traitement, toute solution peut adopter le traitement batch ou le traitement temps-réel (Bajaber et al., 2016; Fegaras, 2016; Lee, 2017; Schintler & McNeely, 2022). Le premier type est celui qui utilise généralement les données qui sont stockées sur disque, tandis que le second utilise les données stockées sur mémoire. Contrairement au premier, le second type est utilisé dans les applications où les résultats doivent être obtenus rapidement. L'analyse des données est un processus multidimensionnel dont les tâches varient selon l'objectif assigné. Elle peut cibler le pré-traitement des données, la transformation, l'extraction des connaissances, la présentation des résultats, etc. **Figure 1-2** résume classification des applications Big Data selon les critères mentionnés.

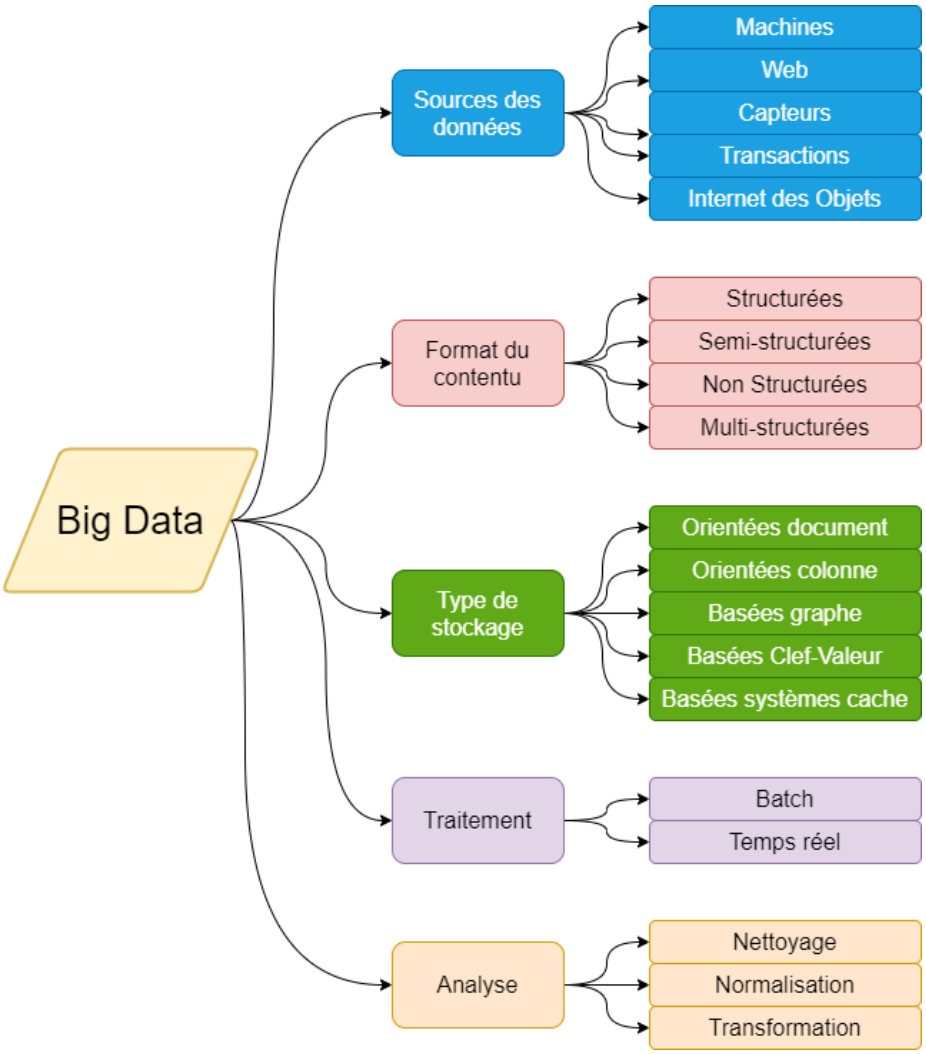

Figure 1-2 : Classification de big data selon différents critères (insprée de (Hashem et al., 2015))

2.3 Enjeux de Big Data

Avec l'émergence de Big Data dans les différents domaines, des multiples enjeux ne cessent d'apparaitre jour après jour. Ces derniers concernent les différentes phases qui composent le cycle de Big Data notamment (Jagadish et al., 2014; Kahil et al., 2020; Qi, 2020; Tariq RS, 2015) : l'acquisition des données, le nettoyage, l'intégration et l'agrégation de ces données, l'analyse et la modélisation et l'interprétation des résultats. Les trois premières étapes constituent la phase de gestion dans Big Data,

tandis que les deux dernières composent le phase analytique (Tariq RS, 2015). **Figure 1-3** résume les enjeux majeurs issus de chacune de ces étapes.

Figure 1-3 : Enjeux Principaux de Big Data (inspirée de (Tariq RS, 2015))

2.3.1 Acquisition des données

L'acquisition des données implique leur récupération depuis différentes sources disponibles à une application précise. A cause de l'hétérogénéité des données récupérées, les tâches d'analytique deviennent lourdes et difficiles à accomplir sans passer par un processus de traitement efficace (Lee, 2017; Schintler & McNeely, 2022). Ce processus comprend essentiellement le filtrage des données acquises selon différents critères telles que les formats et les sources, la réduction de dimensions de ces données afin de réduire la complexité des processus subséquents en termes de temps et de mémoire, la génération des métadonnées et la considération de la fidélité des données (Schintler & McNeely, 2022; Tariq RS, 2015).

2.3.2 Nettoyage des données

Le nettoyage des données est une étape importante qui affecte directement la performance du processus d'analytique des données (Kahil et al., 2019; Lee, 2017; Schintler & McNeely, 2022). Il a pour mission de transformer les données en formats analytiques tout en tenant en compte leur fiabilité (Tariq RS, 2015). Les données non fiables sont traitées de différentes méthodes qui sont discutées dans le chapitre **2**.

2.3.3 Intégration et agrégation des données

L'intégration des données englobe le choix des SGBD Big Data adéquats conformément aux données hétérogènes qui viennent depuis différentes sources afin d'optimiser le processus d'agrégation (Lee, 2017; Schintler & McNeely, 2022). Pour cela, les SGBD doivent assurer la portabilité pour qu'ils opèrent sur différentes plateformes. Ils doivent également être échelonnables afin de pouvoir supporter les données volumineuses ajoutées ou mises à jour.

2.3.4 Analyse et modélisation des données

L'efficacité de l'analyse et d'analytique des données Big Data repose sur la considération de multiples défis à savoir la manière de normaliser les outils d'analyse avec les BDD choisies dans l'étape d'intégration et d'agrégation des données, la stratégie à suivre avec les données dynamiques et/ou bruyantes et le choix des modèles adéquats aux données et à l'objectif d'analytique assigné (Fan et al., 2014; Schintler & McNeely, 2022).

2.3.5 Interprétation des résultats

L'interprétation des données est faite conformément aux objectifs fixés avant le commencement du processus d'analytique. A cette étape, l'enjeux principal est de tirer les conclusions significatives à partir des hypothèses (Andrienko et al., 2020; Fan et al., 2014).

2.4 Echelonnabilité

L'échelonnabilité, appelée parfois l'adaptabilité et l'extensibilité, présente la traduction la plus proche du mot anglais « scalability ». Cette propriété reflète la mise en échelle, un aspect primordial qui caractérise Big Data (Fan et al., 2014; Kahil et al., 2021a; Schintler & McNeely, 2022; D. Singh & Reddy, 2015). Cet aspect fait référence à la capacité d'un système de s'adapter aux situations où les volumes données croissent de façon rapide, (Jagadish et al., 2014; Schintler & McNeely, 2022; Tariq RS, 2015). Par conséquent, l'échelonnabilité doit être satisfaite en améliorant les performances des systèmes via des stratégies précises pour qu'ils puissent continuer de fonctionner correctement tout en satisfaisant les contraintes existantes telles que la contrainte du temps réel, … Pour ce faire, nous distinguons deux stratégies de mise en échelle (scaling) : horizontale (Scale-Out) et verticale (Scale-Up) (D. Singh & Reddy, 2015). L'objectif dans les deux cas est de maximiser les performances d'un système pour pouvoir supporter le traitement des données dont le volume augmente rapidement.

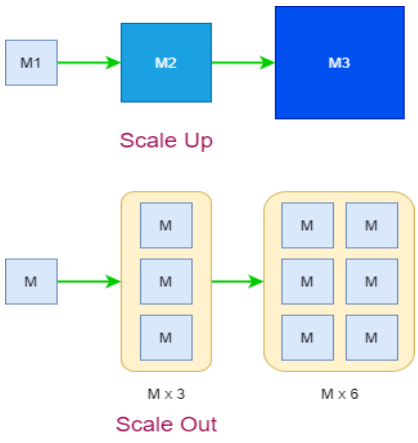

Figure 1-4 : Stratégies de Mise en Echelle (Scaling)

La politique suivie dans les deux stratégies est expliquée dans ci-dessous conformément à ce qui est illustré dans **Figure 1-4**. Scale-Up consiste à améliorer les machines qui déjà existent en étendant leurs unités de stockage (mémoires) et de traitement (processeurs), tandis que Scale-Out consiste à ajouter des machines au lieu d'améliorer celles existantes. Chacune de ces deux techniques présente des avantages et des inconvénients. Le choix de la meilleure stratégie revient alors aux spécificités et aux besoins de l'application en question. **Table 1-1** (D. Singh & Reddy, 2015) montre les avantages et les inconvénients de chacune des deux stratégies.

Table 1-1 : Avantages et inconvénients des deux stratégies de mise en échelle

	Scale-up	**Scale-out**
Avantages	Les performances peuvent être prises simplement par la majorité des outils logiciels La reconfiguration matérielle est facile à réaliser dans les machines	Facilité des étapes d'augmentation des performances Moins coûteux Simplicité de mise en échelle des systèmes selon les besoins
Inconvénients	La mise en échelle est limitée aux spécificités des machines Plus coûteux Les performances supplémentaires peuvent ne pas être complètement utilisées au commencement du traitement Le système doit être puissant pour supporter les charges du traitement	Les outils logiciels doivent supporter le traitement parallèle et la distribution des données La majorité des outils logiciels ne supportent pas ce type de mise en échelle

3 NoSQL

Les bases des données relationnelles sont les types des systèmes classiques qui sont dédiés à stocker les données structurées. Ces dernières suivent un schéma défini à travers lequel elles sont organisées dans des tables. Bien que plusieurs systèmes de stockage aient été proposés pour supporter les données massives, notamment les entrepôts de données (Cuzzocrea et al., 2013; Giceva & Sadoghi, 2018) et les bases de données de traitement parallèle massif (MPP : Massively Parallel Processing) (Schintler & McNeely, 2022), ils ne supportent pas les formats hétérogènes des données. En plus, ils sont essentiellement dédiés à manipuler les données via SQL.

Le terme NoSQL (Not Only SQL) désigne le nouveau type des systèmes de gestion de bases de données non-relationnelles. Ces bases de données, souvent distribuées, sont dédiées au stockage des données non-structurées. Elles ne nécessitent pas de schéma fixe pour stocker et gérer les données (Davoudian et al., 2018; Jing Han et al., 2011; Kahil et al., 2021a). Elles sont caractérisées par la grande échelonnabilité au volume des données hétérogènes et homogènes qui excède les pétaoctets et leur capacité d'opérer sur différentes plateformes. Comme ces bases de données ne sont pas fondées sur la jointure à cause de la complexité des données volumineuses, l'analyse de ces dernières est souvent difficile et nécessite des techniques avancées pour être réalisée. Techniquement, les bases de données NoSQL stockent les informations dans des documents semi-structurés tels que XML, JSON, BSON (JSON binaire) (Atzeni et al., 2020; Davoudian et al., 2018). Le type de documents JSON offre une représentation flexible des données non-structurées afin de simplifier le processus du stockage et de gestion. Les données structurées peuvent toutefois être manipulées à travers ce format. Par conséquent, le stockage et la gestion via les bases de données NoSQL peuvent combiner à la fois les données structurées et non-structurées et peuvent même gérer les premières via les requêtes SQL.

Le choix d'une base de données NoSQL se fait selon des critères relatifs aux besoins et spécificités des domaines d'application. Ces bases de données peuvent être catégorisées en quatre types à savoir orientées colonne, orientées document, basées clef-valeur et basées graphe (Patgiri, 2019; J. Wang et al., 2020). Cette catégorisation n'est pas exclusive ; il existe des BDD NoSQL qui peuvent être considérées en plus qu'une catégorie. Par exemple : Riak appartient à la fois aux BDD basées clef-valeur et à celles basées orientées document. Ces catégories sont décrites ci-dessous (Atzeni et al., 2020; Bhogal & Choksi, 2015; Cattell, 2011; Davoudian et al., 2018; Jing Han et al., 2011; Schintler & McNeely, 2022). De plus, les base de données basées système cache (Cache System-based databases)

peuvent être considérées comme une autre catégorie de NoSQL. Redis et Memcache peuvent s'inscrire sous type.

3.1 BDD orientées colonne

Les BDD orientées colonne consistent à traiter chaque colonne séparément des autres. Toutes les colonnes sont stockées de manière contigüe. Ce type de BDD est largement utilisé pour la gestion des entrepôts des données et business intelligence, mais aussi pour la gestion des relations entre les clients et tant d'autres applications. Elles sont très performantes pour les requêtes d'agrégation en raison de la disponibilité des données dans les colonnes. Parmi les BDD orientées colonnes populaires on trouve HBase, Cassandra, et Hypertable.

3.2 BDD basées clef-valeur

Dans les bases de données basées clef-valeur, les données sont stockées selon le principe des tables de hachage. Chaque pièce de données est stockée comme une paire clef-valeur où chaque clef est unique, tandis que la valeur est dotée d'un type qui peut être différent d'une pièce à une autre. Parmi ces types on trouve les chaines de caractères (textes), JSON, etc. Les BDD basées clef-valeur peuvent être utilisées pour stocker les données sans schémas telles que les tables associatives, les dictionnaires et les collections. Redis, Dynamo et Riak sont représentatives des BDD basées clef-valeur.

3.3 BDD orientées document

Comme les BDD basées clef-valeur, les orientées document stockent les données dans des paires clef-valeur, sauf que les valeurs sont stockées dans des documents formatés en XML ou JSON. Ces bases de données sont capables de comprendre ces valeurs et d'effectuer des requêtes sur elles. Elles sont utilisées dans les applications d'E-commerce, les plateformes de blogging, l'analytique en temps réel, etc. Les BDD orientées document représente un type de BDD NoSQL très utilisé. Parmi ces BDD il y a MongoDB, CouchDB, Riak, Amazon SimpleDB et Lotus Notes.

3.4 BDD basées graphe

Le principe de ce type de bases de données est de considérer les données qui peuvent avoir des représentations basées sur les graphes. Ce type de représentation est le résultat de la connexité des données qui traduit les relations directes et/ou indirectes entre ces données et qui forme éventuellement un graphe (réseau) de données. Concrètement, chaque enregistrement est représenté par un nœud et est relié avec les autres enregistrements par des relations différentes. Il peut alors y avoir de multiples relations à l'intérieur de la même base de données. Les bases de données basées sur les graphes peuvent être trouvées dans différentes situations telles que les bio-informatiques, les réseaux sociaux, les systèmes de recommandations des films, … En plus de Neo4j, de nombreuses BDD basées graphes ont été développées telles que Amazone Neptune, DataStax, Infinite Graph, OrientDB, FlockDB, ArangoDB, GRAKN, etc.

4 Traitement dans Big Data

Le traitement des données englobe toutes les étapes du cycle de Big Data à partir de la collecte des données jusqu'à l'interprétation et la livraison des connaissances aux API. Avec le flux des données important, les tâches du traitement présentent toujours une préoccupation à ne pas prendre à la légère. Leur accomplissement doit suivre des stratégies effectives conformément aux exigences de l'application en question. Concrètement, le traitement des données se déroule principalement selon deux stratégies alternatives. Batch et Temps réel (Kahil et al., 2020; Lee, 2017; Oussous et al., 2018; J. Wang et al., 2020).

4.1 Traitement batch

Le traitement batch est le type de traitement qui vise les données statiques. Ces dernières sont généralement volumineuses dans le contexte Big Data et sont stockées d'une manière distribuée. Ce type de traitement est fondé sur le principe que, une fois le traitement commence, il n'y a plus davantage de données externes à considérer. Les tâches sont alors accomplies dans un ordre séquentiel et sans arrêt. Ce type de traitement a plusieurs applications telles que la gestion financière dans les compagnies, l'analyse des transactions soumises par les entreprises au cours de périodes précises, etc.

4.2 Traitement en temps réel

Le traitement en temps réel (Stream Processing) est appliqué sur les données dynamiques. Le flux des données qui font l'objet du traitement en temps réel n'est pas soumis à des limitations de temps, ces dernières sont traitées dès qu'elles entrent aux systèmes. Ceci dit, les systèmes de traitements ne nécessitent pas de stocker des volumes importants de données pour effectuer les différentes tâches. Parmi les applications illustratives du traitement en temps réel on trouve la détection de fraude, l'analyse des sentiments dans les réseaux sociaux, la supervision des logs et l'analyse des comportement des clients.

Le traitement batch et le traitement en temps réel ont chacun ses propres caractéristiques et ses applications. **Table 1-2** résume les grandes différences entre eux (Kolajo et al., 2019).

Table 1-2 : Comparaison entre le traitement batch et le traitement en temps réel

Traitement batch	Traitement en temps réel
Les données sont collectées au fil du temps	Le flux des données est continu
L'intégralité données sont traitées après avoir été entièrement collectées	Les données sont traitées en parties
Dédié à traiter des données volumineuses qui ne sont pas contraintes au temps.	Dédié à traiter les données qui sont contraintes au temps
Lent	Rapide

5 L'écosystème Hadoop

Hadoop, lancé en 2006, a précédé l'apparition du concept actuel de Big Data et a été son principal outil de stockage et de traitement distribués des données. A l'origine, il était essentiellement composé d'un système de stockage appelé HDFS, de la base de données Hbase et du mécanisme de traitement MapReduce. L'évolution de Big Data et son émergence dans différentes discipline a attiré l'attention de la communauté scientifique et des grandes compagnies de l'industrie logicielle telles que Microsoft, Apache et IBM. Cet intérêt a été à l'origine de développement d'une multitude de solutions optimisées pour résoudre les problèmes et considérer les enjeux des différentes phases du cycle de Big Data. La majorité de ces solutions, supportées même par les grands fournisseurs de Cloud, sont basées sur Hadoop. Ce dernier s'est alors transformé d'une plateforme qui traite des tâches limitées en tout un écosystème avec une grande portée sur les solutions modernes destinées à remédier aux problèmes de Big Data qui ne cessent d'apparaitre. Il existe différentes architectures qui sont proposées par les grandes compagnies dans le marché de Big Data telles que Cloudera, IBM, Databricks, Hortonworks, Infochimps et Google Dataproc (Erraissi, 2017; LTIM et al., 2017). En termes d'outils, les différences entre ces architectures ne sont pas majeures car la majorité des outils qu'ils emploient s'inscrivent dans le projet de Hadoop et sont licenciés par Apache. Ces architectures, illustrées dans (Baaziz & Quoniam, 2014; R. Singh & Kaur, 2016; Uzunkaya et al., 2015; Zhu et al., 2021), diffèrent alors essentiellement au niveau d'abstraction qui traduit le processus adopté pour la résolution des problèmes Big Data. **Figure 1-5**, inspirée de ces différentes architectures, montre l'exemple d'un écosystème de Hadoop

représenté par des couches avec des exemples d'outils implémentés sur Hadoop qui peuvent être utilisés au niveau de chacune.

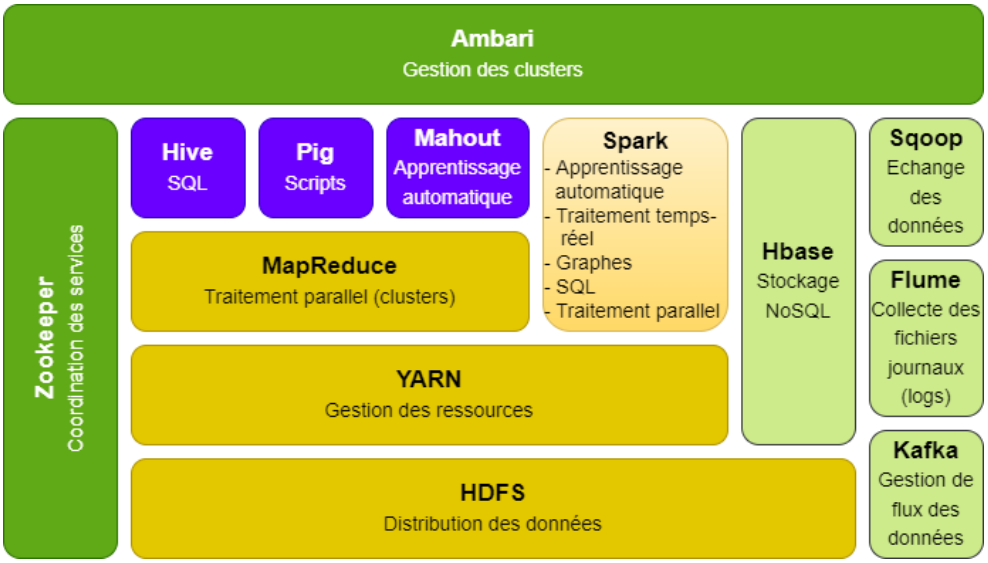

Figure 1-5 : Ecosystème de Hadoop (inspirée de (Baaziz & Quoniam, 2014; R. Singh & Kaur, 2016; Uzunkaya et al., 2015; Zhu et al., 2021))

En général, on peut séparer les outils qui font parties de l'écosystème de Hadoop selon l'objectif en trois catégories : le stockage de données, la gestion de l'écosystème et le traitement des données. La gestion de l'écosystème comprend les ressources, la coordination entre les outils et la gestion de flux et d'échange des données à l'intérieur de l'écosystème. Selon cette catégorisation, les systèmes faisant parties de cet écosystème sont brièvement décrits ci-dessous.

5.1 Stockage des données

Il y a de multiples mécanismes qui sont utilisés pour stocker les données volumineuses dans cet écosystème. Ces mécanismes peuvent être des systèmes de fichiers distribués, des bases de données au sein de Hadoop ou d'autres supports de stockage externes.

Le système de fichiers distribué Hadoop (HDFS : Hadoop Distributed File System) est un système dédié au stockage distribué des données dans l'écosystème Hadoop. Il peut supporter des centaines de nœuds dans le cluster pour offrir un stockage de données rapide et fiable sur les disques, que ces données soient structurées ou non-structurées (LTIM et al., 2017; Oussous et al., 2018). Il consiste à présenter les données dans des blocks afin de rendre plus facile la tâche de leur distribution et, par conséquent, la distribution du traitement. Il est caractérisé par la distributivité et la réplicabilité des données, d'où sa fiabilité (Schintler & McNeely, 2022). La réplication des données est basée sur le fait que ces dernières sont stockées dans chaque block et qu'elles ont des copies sur d'autres nœuds. Ainsi, en cas de pannes, HDFS continue de fonctionner normalement. Il a essentiellement été conçu pour supporter le traitement batch des opérations de haute latence. HDFS est basé sur l'architecture maitre-esclave (Patgiri, 2019; Uzunkaya et al., 2015). Comme le montre **Figure 1-6**, il est composé d'un nœud de noms et des nœuds de données (Patgiri, 2019; Uzunkaya et al., 2015; Vohra, 2016). Le premier, qui joue le rôle de superviseur, gère toutes les interactions entre les nœuds des données. A tout instant donné, le nœud des noms est unique. Cela assure l'organisation d'accès des nœuds des données pour le stockage, la lecture et l'écriture des données. Les nœuds de données se chargent du stockage des données dans les blocs.

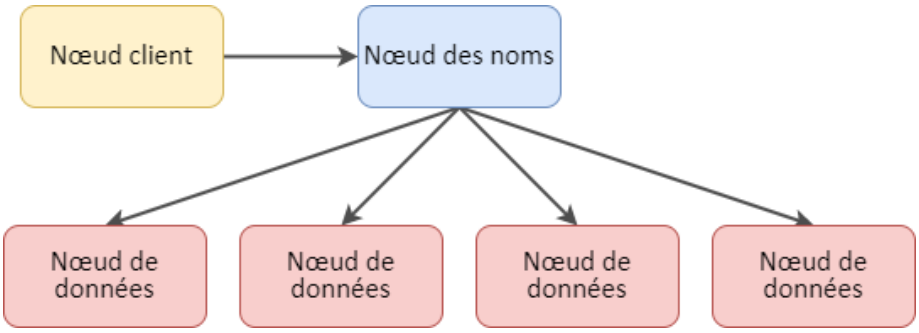

Figure 1-6 : Architecture du système de fichiers HDFS (inspirée de (Uzunkaya et al., 2015))

La fiabilité de HDFS est assurée par différentes méthodes de prévention telle que la sauvegarde régulière des métadonnées dans plusieurs nœuds, la coordination entre les nœuds via des outils spécialisés tels que Zookeeper (Vohra, 2016), etc.

Bien que HDFS soit le système de stockage le plus connu dans le contexte Big Data, tant d'autres systèmes existent et présentent des avantages qui remédient à différents problèmes tels que la récupération des données dans les cas de catastrophes, l'optimisation de secours (Hot Standby), etc. (Patgiri, 2019). Parmi ces systèmes on trouve GFS, QFS, HopsFS, GlusterFs, QFS, Ceph FS (Patgiri, 2019).

Il existe également différentes bases de données qui assurent le stockage dans l'écosystème Hadoop telles que la base de données NoSQL Hbase (Uzunkaya et al., 2015, 2015; Zhu et al., 2021) et l'entrepôt de données échelonnable Hive (X. Liu et al., 2014; Oussous et al., 2018; R. Singh & Kaur, 2016).

5.2 Gestion

La gestion dans l'écosystème de Hadoop est un concept multidimensionnel qui peut être lié aux ressources de toute sorte (mémoire, processeurs, …), aux nœuds (clients et trackers), à la communication au sein de l'écosystème et au flux de données selon leurs types. Parmi les frameworks basés Hadoop qui s'inscrivent dans l'axe de gestion on peut citer : YARN, FLUME, KAFKA, Zookeepr et Ambari. YARN veut dire « Encore un autre négociateur de ressources » (Yet Another Resource Negociator) a été originairement conçu pour gérer les ressources dans l'écosystème de Hadoop, mais il est devenu tout un système distribué de traitement sur lequel différents frameworks tels que Spark, Hive et Storm peuvent être utilisés dans le même cluster Hadoop (Erraissi, 2017; X. Liu et al., 2014; Vohra, 2016). Flume est un framework flexible et échelonnable qui utilise le flux continu des données pour la collecte, l'agrégation et la transmission des volumes importants des données des journaux (logs) depuis les serveurs afin de les stocker dans HDFS ou d'autres support externes tels que Hbase (X. Liu et al., 2014; Vohra, 2016). Kafka est un système basé sur la messagerie qui est utilisé pour l'intégration des données provenant de différentes sources en temps réel dans les clusters Hadoop. Il sert à assurer la disponibilité de ces données pour le l'accès et traitement en temps réel par multiples systèmes qui opèrent de manière concurrentielle (X. Liu et al., 2014; Vohra, 2016). Ces systèmes peuvent être des connecteurs de bases de données, des nœuds de traitement, des serveurs, des applications consommatrices, etc. Zookeeper est un framework fiable qui sert à coordonner les processus distribués en fournissant des services opérationnels au sein des clusters de Hadoop. Ces derniers sont essentiellement constitués d'un service de configuration distribuée, un service de synchronisation et un registre de nommage des systèmes distribués basés Hadoop (Y. Li et al., 2018; Vohra, 2016). Ils ont alors pour mission de coordonner et de gérer les machines d'un cluster Hadoop (Vohra, 2016; Wadkar & Siddalingaiah, 2014). Ambari est un framework de gestion, de provisionnement, de supervision et de sécurisation des clusters Hadoop développé par Hortonworks. Il offre une plateforme sécurisée et consistante pour le contrôle

opérationnel. Cette plateforme permet d'automatiser les différentes opérations dans les clusters Hadoop (Wadkar & Siddalingaiah, 2014). Ambari offre une visibilité complète sur l'état des clusters, ce qui facilite la capture et la résolution des problèmes qui peuvent avoir lieu dans son sein.

5.3 Traitement

Le traitement des données dans l'écosystème de Hadoop est maintenu par de nombreux outils. Ces derniers offrent des fonctionnalités et des performances différentes selon les exigences des problèmes en question et le type du traitement adéquat, à savoir Batch et temps réel. Ci-dessus sont décrits des frameworks qui sont largement utilisés pour différentes tâches du traitement.

5.3.1 MapReduce

Basé sur le paradigme Map Reduce, ce framework représente le noyau du traitement dans Hadoop. Développé à base de Java, il a pour mission d'assurer le traitement distribué dans les clusters de Hadoop (X. Liu et al., 2014; Schintler & McNeely, 2022; D. Singh & Reddy, 2015). Comme les données qu'il traite sont essentiellement stockées dans HDFS, MapReduce offre une mise en échelle aux volumes importants qui peuvent atteindre les pétaoctets. De plus, il offre plusieurs avantages tels que la flexibilité en termes d'accessibilité facile aux multiples sources et types de données, la rapidité du traitement grâce à son opérabilité parallèle et la simplicité de développement des solutions qui peut être réalisé via Python, Java et C++ (D. Singh & Reddy, 2015). Le modèle de MapReduce, dédié principalement au traitement Batch, est composé de deux tâches : Map et Reduce. Map se charge de convertir les données en data-set de sorte que chaque élément est représenté par un tuple (clef-valeur). Reduce, succédant Map, prend sa sortie comme entrée et combine les tuples en un autre ensemble de tuples qui soit plus petit (D. Singh & Reddy, 2015; R. Singh & Kaur, 2016). Dans les modèles antérieurs de Hadoop, l'architecture de MapReduce, comme le montre **Figure 1-7**, était composée de deux types d'opérateur, à savoir un traceur de jobs (job tracker) et des traceurs de tâches (task trackers) (Bajaber et al., 2016; Wadkar & Siddalingaiah, 2014). Le premier, exécuté sur le nœud des noms dans HDFS, a pour mission de (1) recevoir des requêtes de MapReduce de la part des nœuds clients, (2) déterminer l'emplacement des données au niveau des nœuds de données, (3) trouver les nœuds qui lui sont les meilleurs à exécuter la tâche désignée et (4) gérer les trackers de tâches et renvoyer le résultat de la tâche accomplie au client. Le second type, exécuté sur les nœuds de données, effectue les opérations de Map et de Reduce par l'administration et la communication avec le traceur de jobs.

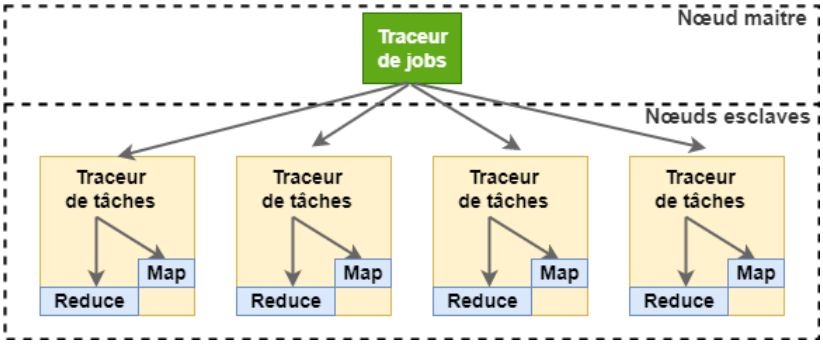

Figure 1-7: Architecture classique de hadoop mapreduce (inspirée de (Uzunkaya et al., 2015))

Dans les versions récentes de Hadoop, cette architecture a été remplacé par YARN qui contient l'application master à la place du tracker de jobs et les gestionnaires de nœuds à la place des trackers de tâches (Wadkar & Siddalingaiah, 2014). Ils sont gérés par le gestionnaire des ressources. L'application master supervise alors les tâches effectuées par les gestionnaires de nœuds.

Le traitement des erreurs dans MapReduce peut être fait via différentes méthodes conformément à la nature de l'erreur. Différents cas qui peuvent être distingués sont expliqués ci-dessous.

1- Lorsque des erreurs se produisent aux dans les tâches supervisées par l'application master, cette dernière peut procéder de deux manières : (a) réexécuter la tâche ou (b) relancer la tâche sur un autre nœud.
2- Si l'application master tombe en panne, elle est redémarrée par un négociateur de ressources tel que YARN (Oussous et al., 2018).
3- Si un nœud tombe en panne, le gestionnaire de ressources essaie de le redémarrer (Vohra, 2016).
4- Si le gestionnaire des ressources tombe en panne, une application maître alternative peut le remplacer selon le concept de haute disponibilité via des outils dédiés tels que Zookeper (Vohra, 2016).

Pig et Mahoot représentent d'autres frameworks utilisés pour le traitement dans les clusters Hadoop. Pig est une plateforme dédiée à réduire l'effort de se concentrer trop sur MapReduce lors de l'implémentation des solutions d'analytique des grands data-sets. (R. Singh & Kaur, 2016; Uzunkaya et al., 2015). Mahout est un framework qui est basé sur MapReduce et qui est dédié à traiter les problèmes d'apprentissage automatique. Basé sur Java, il contient différents algorithmes implémentés de classification tels que les Naïve Bayes, de clustering tels que K-means et de recommandation tels que le filtrage collaboratif (Aziz et al., 2018).

5.3.2 Apache Spark

Apache Spark est un framework de traitement des données volumineuses qui peut être utilisé via différents langages de programmation à savoir Scala, Python, Java et R. Largement utilisé par les grandes compagnies de nos jours et plus efficient que MapReduce, il peut opérer seul ou de manière distribuée au sein de l'écosystème de Hadoop sur HDFS, YARN, Mesos, etc. Il supporte le traitement des données structurées via SQL, les données en streaming, le traitement des graphes et l'apprentissage automatique (Aziz et al., 2018; Bajaber et al., 2016; X. Liu et al., 2014). Une application Spark a deux composants essentiels : un driver et des exécuteurs. Le driver, s'exécutant sur le nœud maitre, a pour mission de convertir le programme codé en tâches qui sont distribuées sur les exécuteurs. Ces derniers exécutent les tâches qui leur sont assignées sur les nœuds travailleurs (workers). Puisque les applications Spark s'exécutent en mémoire comme montré dans **Figure 1-8** (X. Liu et al., 2014), ce dernier est caractérisé par une performance qui peut être cent fois plus rapide que celle de MapReduce dans des situations où des résultats intermédiaires sont à stocker (Hazarika et al., 2017; Kahil et al., 2020). Un autre avantage de Spark est qu'il est doté d'une API facile aux développeurs qui leur permet d'implémenter des solutions d'une manière simple.

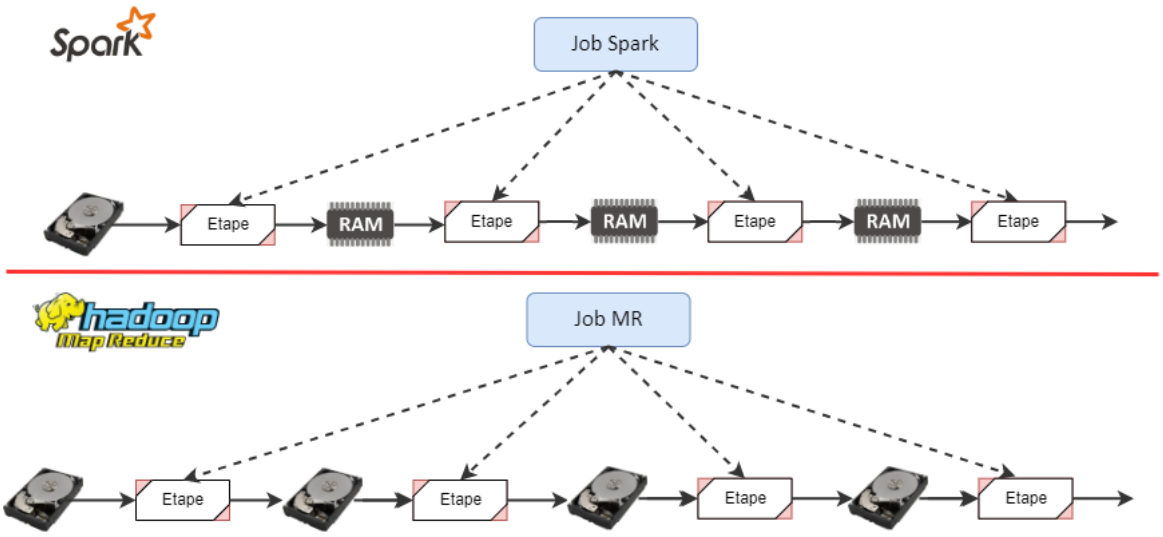

Figure 1-8 : Processus d'exécution des jobs dans Spark et Hadoop MapReduce (X. Liu et al., 2014)

RDD (Resilient Distributed Dataset) est la structure de base de Spark. C'est une abstraction qui représente une collection d'objets immutables qui peut être distribuée dans le cluster et traitée parallèlement tout en assurant la mise en échelle. Cette structure peut être créée à partir de différentes manières telles que les bases de données SQL ou NoSQL, les fichiers textes, HDFS, les buckets, etc. Elle supporte les opérations de jointure, de filtrage, d'échantillonnage et d'agrégation, en plus des fonctions de Map et de Reduce (Aziz et al., 2018; Bajaber et al., 2016; Wadkar & Siddalingaiah, 2014). A partir de RDD, deux autres structures ont été développées : les dataframes et les data-sets. Leur objectif est de représenter les données de manière structurée, c-à-d. sous forme de tables avec des lignes et des colonnes, afin de pouvoir les interroger efficacement et de faciliter les différentes tâches d'analyse subséquentes.

Comme le montre **Figure 1-9**, Spark a essentiellement cinq librairies qui sont dédiées à différentes tâches, à savoir Spark Core, Spark SQL, MLLib, Spark Streaming, GraphX.

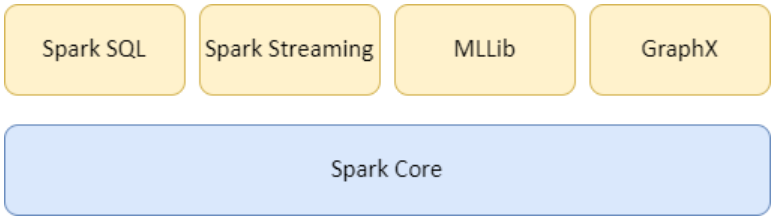

Figure 1-9 : Librairies de Spark (Aziz et al., 2018)

1- Spark Core représente le noyau de Spark. C'est à sa base que les autres librairies fonctionnent. Un « Core » divise une application Spark en tâches qu'il distribue sur multiples exécuteurs échelonnables aux exigences de l'application. On distingue deux types d'opérations dans le framework Spark : Transformations et actions. Une transformation a pour objectif d'appliquer une opération sur le RDD, la dataframe ou le data-set pour en extraire un résultat qui soit de la même nature que ces structures. Parmi les transformations on trouve les fonctions de Map, FlatMap, Filtrage, Reduce, Agrégation, etc. Une action, quant à elle, est une opération dont l'objectif est d'effectuer des instructions qui n'affectent pas les structures, mais plutôt pour d'autres fins telles que d'affichage du contenu d'un RDD ou d'une dataframe, la personnalisation d'affichage dans la console, etc.

2- Spark SQL est un package qui est destinée au traitement des données structurées et qui est largement utilisé de nos jours par les développeurs pour la création des applications Spark. Outre SQL, il peut aussi traiter les données non structurées et semi-structurées en leur attribuant des schémas. Il supporte le paradigme *Map Reduce*. L'unité utilisée dans Spark SQL est la *DataFrame* (Aziz et al., 2018). Bien qu'elle soit structurée du fait qu'elle est composée de lignes et colonnes, elle n'est physiquement qu'un RDD, et toutes les propriétés de ce dernier, notamment l'immutabilité et la distributivité, s'appliquent sur elle. Spark SQL peut lire et écrire les données dans différents formats tels que CSV, JSON, HDFS, ORC, Hive, Parquet, les différentes bases de données SQL et NoSQL, etc.
3- MLLib est un framework qui permet de faciliter la création des pipelines de l'apprentissage automatique distribué via Python ou R (Bonaccorso & Safari, 2018). De multiples algorithmes de régression, de classification, de clustering, de recommandation, de réduction de dimensionnalité et de fouille des données textuelles y sont implémentés et peuvent s'exécuter de manière distribuée pour accélérer le processus d'apprentissage.
4- Spark Streaming est une librairie qui assure le traitement en temps réel en Spark sans avoir recours à d'autres frameworks tels que Storm et Flink (Bajaber et al., 2016; Schintler & McNeely, 2022). Le principe de Spark Streaming est de découper le flux de données qui viennent continument en une série de micro-batches afin de les traiter via l'API Spark. Ainsi, le traitement en temps réel devient similaire au Batch lors de la phase du développement. Structured Streaming, lancé récemment, permet de créer des dataframes et des data-sets à partir des données en temps réel, d'assurer l'agrégation et d'exécuter les opérations relatives, y compris les requêtes SQL en temps réel.
5- GraphX contient un ensemble d'algorithmes de traitement des structures graphe tels que les algorithmes de parcours des graphes en profondeur et en largeur, la détection des communautés dans les réseaux, etc. (Bajaber et al., 2016). Les algorithmes implémentés utilisent principalement les RDD pour modéliser les données via le package GraphFrames.

6 Analytique dans Big Data

Big Data, d'un point de vue analytique, suit un cycle abstrait qui est très similaire au cycle d'extraction des connaissances à partir des données (KDD : Knowledge Discovery in Databases) (Adhikari & Adhikari, 2015) et à ses alternatives SEMMA (Sample, Explore, Modify, Model, Assess) (Echantillonner, Explorer, Modifier, Modéliser, Evaluer) (Azevedo & Santos, 2008; Shafique & Qaiser, 2014) et CRISP-DM (Azevedo & Santos, 2008; Shafique & Qaiser, 2014). Bien que de multiples grandes compagnies telles que IBM ont conçu leurs propres cycles de vie de Big Data (El Arass & Souissi, 2018), leurs concepts restent similaires. Il enveloppe des étapes primordiales selon lesquelles est fondée toute application qui s'inscrit dans un contexte Big Data. Comme le montre **Figure 1-10**, ces étapes sont : la collecte des données et l'enregistrement des données, le filtrage et l'enregistrement des données, la modélisation, l'analyse et l'interprétation de ces données et la présentation des résultats.

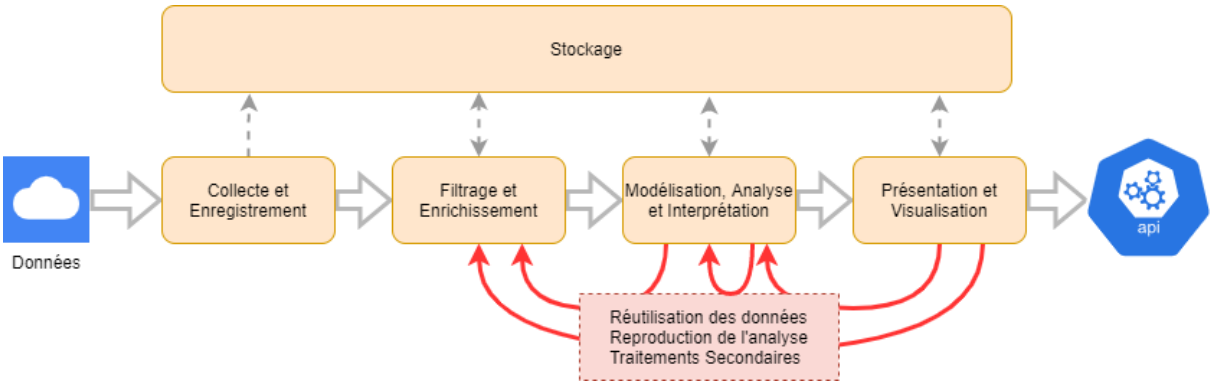

Figure 1-10 : Cycle de Big Data (Demchenko et al., 2014; El Arass & Souissi, 2018)

Des études proposent d'ajouter une étape préservation des données dans le cycle de Big Data (Pouchard, 2016; Sinaeepourfard et al., 2016) à travers laquelle sont définies les données à garder et à stocker pour l'utilisation postérieure telle que la dissémination ou de futurs traitements. La décision de garder les données est prise suivant des critères qui sont généralement relatifs à leur qualité et leur impact sur le résultat du processus d'analyse.

La prise de décision à partir des données dans Big Data, tout comme les données classiques, repose sur l'analyse de ces données. Classiquement, toute analyse se limitait dans des tâches ayant pour objectif d'étudier des aspects élémentaires d'un phénomène défini par des données. Dans l'ère de Big Data, ces tâches se sont élargies à cause de la nécessité de tirer des conclusions à partir de l'étude à la fois de multiples domaines chevauchés. Par conséquent, des méthodes plus avancées que l'analyse classiques sont devenues indispensables pour tirer des informations et des connaissances à partir des données. Elles s'inscrivent dans l'analytique. Cette dernière est un concept qui englobe toutes les activités et les processus qui liés aux données (Andrienko et al., 2020; Iqbal et al., 2016; D. Singh & Reddy, 2015). Son objectif est de faciliter aux analystes et ingénieurs l'accès et la compréhension des données et les connaissances qu'elles portent, la prise de décisions et la prédiction des évènements et des actions. L'analytique peut alors être considéré comme une généralisation de l'analyse des données classique, la dernière étant utilisée pour des tâches spécifiques. **Figure 1-11** montre la relation entre les deux concepts.

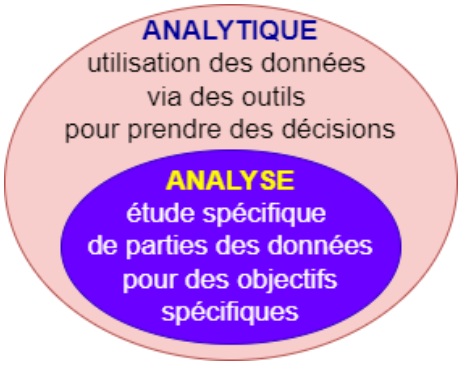

Figure 1-11 : Relation entre analyse et analytique

Il existe de multiples outils modernes dédiés à l'analytique des données. Parmi lesquels on trouve MS Excel, MS Power BI, Tableau, Google Analytics, Python (Pandas, Numpy, …), R analytics, …

7 Conclusion

Ce chapitre a introduit la technologie de Big Data et a couvert les concepts, les méthodes et les outils qui lui sont liés. En effet, Big Data est le résultat de l'explosion des données qui représentent une matière essentielle à travers laquelle les différentes industries modernes font le processus d'extraction des connaissances et de prise de décision afin de maximiser leur profit et d'éviter les fausses prédictions. Ce processus d'extraction suit de nos jours des cycles déterminés tels que KDD, CRISP-DM et SEMMA. Les dimensions de Big Data, connues sous le terme des Vs, traduisent ses caractéristiques et enjeux à considérer. Cette considération repose sur la proposition de solutions sous forme de méthodes, d'outils et de techniques qui traitent les différents problèmes tels que le stockage distribué et la réplication des données, le traitement de ces données en temps réel, la sécurisation des différents processus du traitement, la réalisation des solutions d'analytique et la mise en échelle des différents outils et méthodes proposées. L'analytique des données est l'objet sur lequel tout processus de prise de décision se déroule. En effet, elle est à l'origine de la création de nouveaux modèles, notamment d'apprentissage automatique qui ont pour objectif de tirer des conclusions à partir des données volumineuses et hétérogènes qui caractérisent les différents domaines d'applications modernes. Ils sont implémentés et supportés par les différentes plateformes de Big Data actuelles.

Chapitre 2: Fouille des données – concepts, techniques et outils

1 Introduction

La fouille des données représente l'étape la plus importante dans tout cycle d'extraction d'informations et de connaissances à partir des données. C'est à cette étape que la modélisation et l'exécution des solutions d'analyse et d'analytique des données se font. En effet, ces solutions sont basées sur de nombreux modèles dont ceux qui s'inscrivent dans l'apprentissage automatique reçoivent actuellement le plus grand intérêt. La raison est qu'ils ont montré des résultats prometteurs dans différentes applications d'analytique des données. Par ailleurs, afin d'obtenir des résultats efficients dans la phase de fouille des données, elle est généralement précédée par un processus de nettoyage de données qui couvre différentes dimensions. Le présent chapitre aborde ces différents aspects. Il commence par décrire différentes classes de techniques utilisées pour le nettoyage de données dans la deuxième section. La troisième section aborde les différents modèles et méthodes d'apprentissage automatique qui représentent aujourd'hui la solution incontournable aux problèmes d'analyse et d'analytique des données. La quatrième section aborde des techniques d'apprentissage automatique avancées qui sont très répandues récemment, notamment les modèles d'apprentissage profond. La cinquième section conclut ce chapitre.

2 Nettoyage et préparation des données

Le nettoyage et la préparation des données représentent un aspect d'important intérêt à considérer lors de tout processus d'analyse dans les cas réels. Précédant les autres étapes d'analyse et d'analytique, ils prennent souvent plus d'effort que celles-ci (De Jonge, Edwin & Van Der Loo, Mar, 2013). Cet intérêt particulier revient au fait que la qualité des données affecte directement les performances des modèles développés en se basant sur ces données ; si ces dernières contiennent des valeurs manquantes ou fausses par exemple, le modèle construit sera de faible performance et risquerait l'insuffisance ou le surapprentissage.

Techniquement, la phase de nettoyage et de préparation enveloppe de multiples méthodes qui visent différents axes. Parmi ces méthodes les plus importantes il existe (Berti-Equille, 2019; Krishnan et al., 2016): la détection des valeurs aberrantes, l'imputation des valeurs manquantes, l'optimisation des données et la réduction de dimensionnalité. Ces méthodes, étant non-séquentielles, peuvent être résumées dans les points suivants.

2.1 Détection d'anomalies (outlier detection)

La détection d'anomalies est le processus d'identifier et de traiter les données aberrantes dans les data-sets, d'où son autre appellation : la détection des valeurs aberrantes (Subasi, 2020). Les valeurs aberrantes dans data-set sont celles qui possèdent des propriétés contradictoires avec les propriétés de la majorité des valeurs existantes, ce qui y cause un comportement non-fréquent (van der Aalst, 2016). D'un point de vue statistique par exemple, ces propriétés concernent de paramètres statistiques qui décrivent les data-sets tels que la variance, la covariance et la médiane. Les valeurs aberrantes peuvent être constatées graphiquement comme des points qui sont loin de la majorité des points d'un data-set, en dehors de toutes les classes, les clusters ou les zones denses du data-set, etc. La définition des contradictions qui caractérisent ces valeurs est proportionnelle à différents critères tels que les types des données en question, la structure de celles-ci et le domaine d'application.

Les techniques les plus utilisées pour la détection d'anomalies sont généralement non-supervisées (Subasi, 2020; Z.-H. Zhou, 2021). Parmi lesquelles il y a le clustering, le calcul de distance (Z.-H. Zhou, 2021) et l'écart interquartile, aussi appelé l'étendue interquartile (IQR : Interquartile Range) (Berti-Equille, 2019). Le clustering est décrit dans la section ci-dessous. Tandis que le calcul de distance, il peut simplement être défini par une fonction $dist(x_i, x_j)$ qui mesure la distance entre deux points x_i et

x_j qui est (1) positive, c-à-d $dist(a,b) \geq 0$, (2) symétrique, c-à-d $dist(a,b) = dist(b,a)$ et (3) sous-additive, c-à-d $dist(a,b) \leq dist(a,c) + dist(c,b)$ (Z.-H. Zhou, 2021). Il existe plusieurs mesures pour calculer les distances ; la distance de Minkowski en est représentative et largement utilisée. Elle est définie par la formule suivante :

$$dist(a,b) = \left(\sum_{i=1}^{n} |a_i - b_i|^p \right)^{\frac{1}{p}}$$

Avec $P > 0$. En effet, si $P = 1$, elle devient une distance de Manhattan. Et si $P = 2$, elle devient une distance Euclidienne (Z.-H. Zhou, 2021). IQR n'est pas sensible aux valeurs extrêmes. Ces dernières, étant les aberrantes, sont simplement définies par les valeurs qui sont en dessous de $q1 - 1.5IQR$ ou au-dessus de $q3 + 1.5IQR$ (Devore et al., 2021).

2.2 Imputation des valeurs manquantes (missing values imputation)

Dans certains cas, éliminer carrément les unités qui possèdent des données manquantes représente une stratégie consistante. Ce concept est réalisé en utilisant une technique appelée l'analyse de cas complet (CCA : Complete Case Analysis) (Jadhav et al., 2019). Cependant, dans des domaines d'application hautement sensibles, éliminer n'importe quelle donnée pourrait biaiser les modèles développés et mener à des fausses décisions. Pour remédier à ce problème, d'autres stratégies d'imputation sont suivies pour considérer les valeurs manquantes loin de l'élimination. Les méthodes d'imputation sont utilisées pour remplacer les données manquantes par des valeurs pour conserver la structure et les propriétés générales du data-set et garder son aspect significatif. Les problèmes des valeurs manquantes peuvent être classés en trois catégories : (1) le manque au hasard (MAR : Missing At Random), (2) le manque complètement au hasard (MCAR : Missing Completely at Random) et (3) le manque pas au hasard (MNAR : Missing Not at Random) (Jadhav et al., 2019; Kambach et al., 2020; Spratt et al., 2010). Dans la première catégorie, la probabilité de manque est la même dans les parties du data-set définies par les données observées. La deuxième catégorie reflète les cas où la probabilité de manque des données est la même pour l'intégralité du data-set. La troisième catégorie représente le cas où le manque des données dans un data-set n'est reflété ni par MAR ni par MCAR. Elle signifie simplement que la probabilité de manque des données varie pour des raisons inconnues. Les méthodes d'imputation des données manquantes peuvent être classées selon les types des données à considérer. Généralement, ces méthodes peuvent cibler les données numériques, catégoriques ou les deux simultanément. Parmi ces méthodes on peut citer : l'imputation arbitraire des valeurs (Arbitrary Value Imputation), l'imputation par catégorie fréquente (Frequent Category Imputation), l'imputation de moyenne / médiane (Mean / Median Imputation), Imputation par échantillonnage aléatoire (Random Sample Imputation), l'imputation par régression (stochastique et déterministe), l'imputation par « Cold Deck » l'imputation par « Hot Deck » et l'imputation multiple. (Aljuaid & Sasi, 2016; Chhabra et al., 2017; Jadhav et al., 2019; Kambach et al., 2020; Spratt et al., 2010; Thirukumaran & Sumathi, 2012; Z. Zhang, 2016).

2.3 Optimisation des données

L'optimisation de données vise principalement l'allègement des différents processus d'analyse pour réduire le temps nécessaire pour les réaliser. Ce concept englobe différents aspects dont les plus importants sont la déduplication des données. En effet, le concept de déduplication des données est très lié au domaine de stockage des données. Les limites qui caractérisaient les systèmes de stockage traditionnels qu'utilisaient les grandes entreprises ont engendré un intérêt particulier à cet axe de recherche, notamment dans les deux décennies précédentes. L'objectif derrière la déduplication était de permettre aux entreprises une utilisation optimale des ressources de stockage afin de réduire le coût de cette tâche. La déduplication des données peut simplement être définie comme le processus d'identifier

et éliminer les données redondantes au niveau des ressources de stockage. Ainsi, elle permet d'assurer le stockage des instances des données seulement une fois et remplacer leurs copies par des pointeurs à travers lesquels de multiples machines peuvent y accéder. Le processus de sauvegarde des données est aussi optimisé en considérant uniquement les données qui ont changé depuis la dernière sauvegarde. Cela réduit les exigences du stockage ainsi que la charge des réseaux.

La déduplication continue d'attirer la communauté scientifique non seulement pour une fin de stockage, mais aussi pour réduire l'effort d'analyse et développer des modèles performants dans le contexte Big Data. Le processus d'identification des redondances au niveau des data-sets peut se voir comme une tâche de recherche de tous les enregistrements identiques à un autre. Après cela, la seconde tâche de déduplication consiste à garder uniquement une instance pour chaque enregistrement. L'espace de recherche sera ainsi optimisé et le processus d'analyse, par conséquent, le sera. La correspondance et le mappage des schémas (Bellahsene et al., 2011; Do, Hong-Hai, 2006; Sutanta et al., 2016), ainsi que la résolution des entités (Christophides et al., 2020; Efthymiou et al., 2015; Getoor & Machanavajjhala, 2012; Štajner & Mladenić, 2009) sont des techniques très courantes dans le contexte de déduplication.

2.4 Réduction de dimensionalité

La réduction de dimensionnalité a pour objectif de représenter les données à haute dimension par un nombre inférieur de dimensions, tout en préservant au maximum la variance de ces données, afin de faciliter d'autres tâches et opérations telles que l'extraction des variables, l'apprentissage et la visualisation. D'un point de vue mathématique, la réduction de dimensionnalité peut être considérée comme une fonction qui consiste à projeter les données depuis un espace de variables de D dimensions en un autre espace de variables de D' dimensions tel que $D > D'$. Elle présente plusieurs avantages : réduire l'espace de stockage des données, réduire le temps d'exécution, donner plus de lisibilité aux données, éliminer les données redondantes, réduire l'erreur de généralisation et faciliter la visualisation et l'interprétation des données, etc. (Anowar et al., 2021; Sorzano et al., 2014). Elle peut être utilisée sur différents types de données tels que les images (pour la reconnaissance faciale et la classification des images par exemple),les textes (pour la classification et le résumé automatique) et l'analyse des données temporelles (Sorzano et al., 2014). La technique de clustering de k-moyennes peut être considérée comme étant une technique de réduction si l'on garde juste les centres des points et que l'on néglige les autres données. Dans ce cas, le nombre de dimensions égale k.

En effet, plusieurs algorithmes servent à la réduction des données. L'objectif de chacun d'eux est de sélectionner les caractéristiques (variables) essentielles des data-sets. D'où la notion des algorithmes d'Extraction des variables (FEA : Feature Extraction Algorithms). Le concept d'extraction des variables est souvent confondu avec celui de sélection des variables. Bien que l'objectif des deux soit de réduire la dimensionnalité des données (Sorzano et al., 2014), leurs approches et mécanismes ne sont pas les mêmes. La sélection des variables consiste à sélectionner directement un sous-ensemble de variables, étant jugées importantes pour résoudre un problème en question, à partir de leur ensemble original. Quant à l'extraction des variables, elle consiste à dériver des informations à partir de l'ensemble des variables original pour en créer un nouveau sous-ensemble. Les FEAs peuvent être supervisés ou non-supervisé et sont appliqués sur les données linéaires ou non-linéaires (Anowar et al., 2021; Zebari et al., 2020). Parmi ces algorithmes il y a PCA, SVD, tSNE, LDA et LLE. Leurs concepts sont brièvement présentés ci-dessous.

L'Analyse des Composantes principales (PCA : Principal Component Analysis) est une technique linéaire non supervisée qui consiste à produire, à partir d'un grand data-set, un nouveau data-set moins-dimensionnel sans erreur significative (X. Huang et al., 2019; Melit Devassy & George, 2020; Sorzano et al., 2014). Les variables qui composent ce nouveau data-set, construites de combinaisons linéaires

des variables originales, sont appelées les composantes principales. La création de ces dernières, qui sont non-corrélées, doit assurer la préservation de la majorité des informations nécessaires qui existent dans le data-set original. Il existe une extension de PCA, nommée PCA à noyaux (KPCA : Kernel-PCA) qui est utilisée pour réduire la dimensionnalité des données non-linéaires (Anowar et al., 2021; X. Huang et al., 2019).

La décomposition des valeurs singulières (SVD : Singular Value Decomposition) est une technique linéaire non-supervisée qui est utilisée pour la réduction de dimensionnalité, notamment pour la compression des bases de données. Son principe est de factoriser un data-set, représenté par une matrice $A(n \times p)$, en trois matrices afin de calculer les valeurs singulières qui, similaires aux composantes principales de PCA, représentent la majorité des informations du data-set : $U(n \times n)$, $W(n \times p)$ et $V^T(p \times p)$ selon la formule suivante (X. Huang et al., 2019; Sorzano et al., 2014) :

$$A = U.W.V^T$$

Où : U est la matrice des vecteurs propres orthonormés dont les colonnes représentent les vecteurs singuliers de gauche. Ces derniers sont les vecteurs propres de $A \times A^T$. W est une matrice diagonale de valeurs singulières, ces dernières étant les racines carrées des valeurs propres de la matrice $A^T \times A$. V^T est la matrice transposée de V qui continent les vecteurs propres orthonormés de $A^T \times A$. Ses lignes représentent les vecteurs singuliers de droite.

L'analyse discriminante linéaire (LDA) est une technique linéaire supervisée utilisée non seulement pour la réduction, mais aussi pour la classification et la visualisation (Anowar et al., 2021; X. Huang et al., 2019). Elle repose sur la modélisation des différences dans les groupes des données pour maximiser leur séparabilité en identifiant un nouvel espace de variables. Pour cela, elle extrait du data-set, représenté par une matrice X, k nouvelles variables indépendantes à partir de l'ensemble des n variables indépendantes originales de sorte qu'elles maximisent la séparabilité des variables dépendantes. Cela assure que k est inférieur à n, idem pour le nombre de nouvelles variables dépendantes et celles originales.

Locally linear Embedding (LLE) est une technique non-linéaire non-supervisée qui consiste à réduire la dimensionnalité des data-sets tout en préservant leurs propriétés locales qui reflètent leurs caractéristiques géométriques. Elle se sert de KNN pour cet objectif (X. Huang et al., 2019; Sorzano et al., 2014). Les propriétés locales sont représentées par les relations entre les points des échantillons dans le voisinage.

t-Stochastic Neighbour Embedding (tSNE) est une technique de réduction non-linéaire utilisée principalement pour l'exploration et la visualisation des données multidimensionnelles. Elle a différentes applications telles que la bio-informatique, la sécurité des ordinateurs, la réduction des données pour la classification, etc. (Anowar et al., 2021; Melit Devassy & George, 2020; Xyntarakis & Antoniou, 2019). Le package tensor-board de tensorflow utilise cette technique pour visualiser les architectures des modèles développés ainsi que l'évaluation des modèles entrainés. Cette technique a pour objectif de réduire la dimensionnalité des data-sets (essentiellement non-linéaires) en préservant les distances des objets qui le composent. Pour cela, elle calcule la similarité entre les paires des instances dans l'espace de grande dimension et celui de faible dimension pour ensuite optimiser ces deux mesures de similarité via une fonction de coût.

3 Apprentissage automatique dans Big Data

La fouille des données (Data Mining) représente l'essence du processus d'extraction des connaissances à partir des données. Comme mentionné dans (van der Aalst, 2016), elle est définie comme le processus d'analyse des data-sets, ces derniers étant souvent volumineux, pour trouver des relations imprévues et résumer les données de façons nouvelles qui sont à la fois compréhensibles et utiles pour celui qui acquière ces données. Ce processus prend typiquement les données en entrée comme des tables et en résulte des règles, des clusters, des structures d'arborescence, des équations, des patterns, etc. Les défis dans l'ère de Big Data se sont multipliés en raison de différents facteurs tels que la multitude des types des données, la nécessité d'unités de stockage et de traitements plus performantes, etc. A cet égard, aborder le sujet des techniques utilisées pour trouver des solutions aux problèmes de fouille de données mène quasi-forcément aux modèles d'apprentissage automatique. Ces derniers ont montré des performances remarquables dans différentes applications.

L'apprentissage automatique est une branche de l'intelligente artificielle (IA) qui a pour objectif de permettre aux machines d'effectuer leurs tâches en utilisant des programmes intelligents (Mohammed et al., 2017). C'est, en réalité, l'intersection de multiples disciplines à savoir : statistiques, informatique et ingénierie. Il est omniprésent et indispensable pour différents domaines dont le traitement du langage naturel (NLP : Natural Language Processing), le traitement des multimédias (vidéo, audio et image) et la prédiction dans toutes ses applications prennent le plus grand intérêt. Concrètement, l'apprentissage automatique enveloppe un ensemble d'algorithmes qui peuvent apprendre à partir de données observationnelles et faire des prédictions à partir de ces dernières.

On distingue trois types d'apprentissage automatique : supervisé, non-supervisé et semi-supervisé. Tout type est décrit par la suite. **Figure 2-1** montre les différents problèmes de chaque classe d'apprentissage à savoir le supervisé, le semi-supervisé et le non supervisé. Il est à signaler qu'un autre type, appelé l'apprentissage par renforcement (Reinforcement Learning), peut être distingué également. Or, il n'est pas discriminable selon la nature des données, mais plutôt de la manière de collecter ou de percevoir les données, de prendre des décisions afin d'agir dans différentes situations et de sélectionner les techniques adéquates à chaque situation.

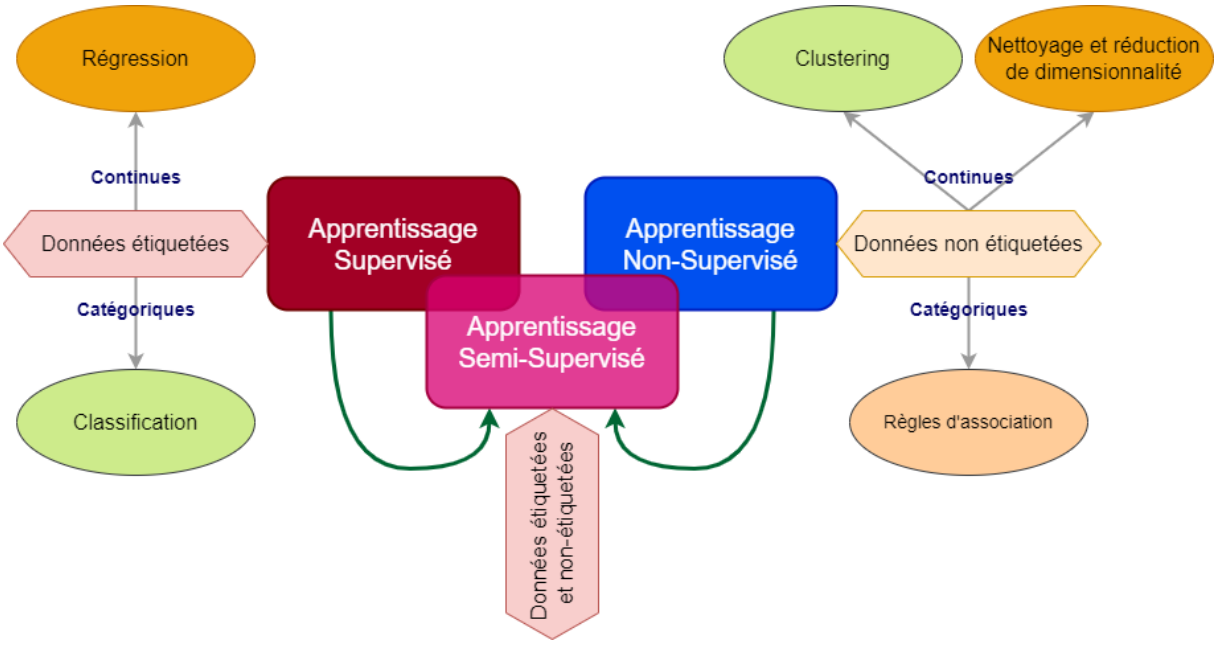

Figure 2-1 : Types d'apprentissage automatique

L'apprentissage supervisé contient l'ensemble des techniques qui se servent des données étiquetées (labelled) d'entrainement et de test (Subasi, 2020). L'étiquetage des données est réalisé via une assistance externe, souvent humaine. Le principe de ces techniques est de définir une fonction qui mappe entre les entrées et les sorties (Mahesh, 2018; Mohammed et al., 2017; van der Aalst, 2016). L'apprentissage supervisé est utilisé pour prédire des valeurs ou classifier des données dans les classes correspondantes.

Contrairement à l'apprentissage supervisé, le non supervisé n'utilise pas des données étiquetées par une supervision, il utilise plutôt des données non-étiquetées (unlabelled) pour découvrir leur structure cachée ou les relations qui peuvent y exister, et appliquer ces dernières sur de nouvelles données (Mahesh, 2018; Mohammed et al., 2017). Il est utilisé pour différentes applications telles que de clustering, l'identification des patterns, la détection de similarité, la recommandation, la segmentation des objets et l'extraction des variables pour réduire les dimensions des données, ce qui aide à les nettoyer et à les préparer pour d'autres tâches d'apprentissage (Bonaccorso & Safari, 2018; Mahesh, 2018; van der Aalst, 2016; Yagang Zhang, 2010). PCA, SVD, LLE et tSNE représentent une illustration de techniques d'apprentissage non-supervisé qui sont utilisées pour la réduction de dimensionnalité.

L'apprentissage semi-supervisé est une combinaison entre l'apprentissage supervisé et le non supervisé. Il est utilisé quand les données sont une mixture de données étiquetées et d'autres non-étiquetées (Mahesh, 2018; Mohammed et al., 2017). Son utilité apparait dans les situations où le volume des données étiquetées dans un data-set est minime par rapport au nombre total des éléments de ce data-set. Les objectifs essentiels de l'apprentissage semi-supervisé peuvent se résumer dans les points suivants (Bonaccorso & Safari, 2018).

- La propagation des labels aux éléments non-étiquetés en se servant d'une structure de graphe qui représente l'intégralité du data-set. Les éléments étiquetés étendent leur influence au voisinage jusqu'à atteindre un point d'équilibre.
- La classification via les données étiquetées et essayer de considérer celles non-étiquetées comme éléments équilibreurs.
- L'utilisation des techniques de réduction de dimensionnalité non-linéaires pour trouver au data-set moins-dimensionnel sans perte significative d'informations.

3.1 Biais, variance, surapprentissage et insuffisance en apprentissage automatique

Le concept de biais dans l'apprentissage automatique peut être défini comme un phénomène qui reflète le processus de fausser le résultat d'un modèle d'apprentissage en faveur ou contre une idée. Concrètement, il représente la différence entre la prédiction moyenne du modèle et les valeurs correctes, ces dernières étant la cible que le modèle tente d'atteindre (Z.-H. Zhou, 2021). Cette différence permet de décrire la qualité d'apprentissage du modèle sur les données d'entrainement. Le biais d'un modèle est en relation inverse avec sa performance ; s'il est faible, la prédiction sera proche des valeurs cibles, sinon, elle en sera loin. Il est défini selon l'équation suivante (Z.-H. Zhou, 2021) :

$$biais(x) = \left(\bar{f}(x) - y\right)^2$$

Tel que : x est un échantillon de test, y est la cible (valeurs réelles), $\bar{f}(x)$ est la prédiction attendue via le modèle f. Dans la pratique, la performance des modèles avec un biais élevé souffre de multiples problèmes tels que l'échec de capturer les propres patterns qui seront très généralisés ou trop simplifiés, ainsi que des taux d'erreur élevés et des situations de surapprentissage, d'insuffisance ou de décalage du domaine de ces modèles. Dans un data-set, le biais correspond au pattern le plus fréquent parmi l'ensemble des objets qui y existent.

La variance représente la variabilité de prédiction d'un modèle d'apprentissage en fonction de l'utilisation de différentes portions du data-set (Z.-H. Zhou, 2021). Une variance élevée d'un modèle, impliquant un faible biais et vice-versa (Z.-H. Zhou, 2021), peut être interprétée par l'existence de bruit dans le data-set, un surapprentissage ou une complexité élevée du modèle.

Le surapprentissage (over-fitting) est un phénomène qui surgit dans le cas où un modèle d'apprentissage fonctionne efficacement durant l'entrainement, mais présente de faibles performances durant le test. Le problème ici est que l'association des données pendant l'entrainement est tellement parfaite que même celles aberrantes ou particulières sont considérées et mémorisées. Ainsi, quand il est testé sur des données non-connues, sa performance baisse et présente une grande erreur de test (Bonaccorso & Safari, 2018; Jabbar & Khan, 2014; Z.-H. Zhou, 2021). Cela signifie que le modèle est incapable de généraliser les patterns. Pour s'en passer, de nombreuses stratégies peuvent être adoptées. Parmi lesquelles il y a les méthodes basées pénalité telles que la validation simple (Hold-out) et la validation croisée (cross-validation), la technique d'arrêt anticipé (Early Stopping) (Jabbar & Khan, 2014; Z.-H. Zhou, 2021) et le choix approprié des paramètres du modèle (model settings) (Z.-H. Zhou, 2021).

L'insuffisance (Under-Fitting), le phénomène contraire de surapprentissage, reflète un modèle dont la performance est si limitée qu'il ne peut saisir la performance montrée par les mêmes données d'apprentissage en raison son incapacité à capturer la variabilité de ces données (Bonaccorso & Safari, 2018; Z.-H. Zhou, 2021). Ceci est dû à un data-set limité dont les modèles d'apprentissage ne peuvent pas identifier des patterns. (Z.-H. Zhou, 2021).

3.2 Régression

La régression est une collection d'algorithmes statistiques qui appartiennent à l'apprentissage supervisé dont l'objectif est de prédire des valeurs continues (Bonaccorso & Safari, 2018; van der Aalst, 2016). Largement considérée pour de nombreuses applications, elle est utilisée pour modéliser et explorer les relations entre les variables dépendantes et indépendantes qui sont liées d'une manière non-déterministe (Montgomery & Runger, 2018).

Il existe dans la littérature des dizaines de dérivées de la régression parmi lesquelles, trois types largement utilisés sont décrits dans ce qui suit, à savoir : régression linéaire, LASSO, de crête. La régression, étant un modèle analytique, peut être, au-delà de son type, simple, multiple ou multivariée. La régression simple est un type d'analyse univariée qui est basée sur l'estimation entre une variable dépendante et une variable indépendante. La régression multiple se manifeste dans les cas où la variable dépendante dépend de plus qu'une seule variable indépendante. La régression multivariée estime la relation entre multiples variables dépendantes avec une ou plusieurs variables indépendantes (Izenman, 2013).

3.2.1 Régression linéaire

La régression linéaire est celle qui estime la relation linéaire entre la(es) variable(s) dépendante(s) qui représentent le vecteur d'entrée et celle(s) indépendante(s) qui représentent le vecteur de sortie (Bonaccorso & Safari, 2018; Z.-H. Zhou, 2021). Cela implique simplement de trouver la ligne qui représente le meilleur ajustement entre elles.

Dans la version simple de la régression linéaire, l'estimation repose sur la variable dépendante avec une seule variable indépendante. Elle est définie par la formule suivante :

$$y = b + w^T * x$$

Tel que : y est la variable dépendante, x est la variable indépendante, b_0 est un constant, b_1 est le coefficient relatif à la variable indépendante.

La régression linéaire multiple est définie par la formule suivante :

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + \cdots + b_n * x_n$$

Tel que : x_1, x_2, …, x_n sont les variables indépendantes et b_1, b_2, …, b_n sont leurs coefficients respectivement. La ligne de meilleur ajustement est retrouvée en minimisant la somme des carrées des résidus. Un résidu signifie la valeur prédite moins la valeur actuelle. L'objectif de ce processus est de minimiser les distances carrées entre les points et la ligne de meilleur ajustement.

3.2.2 Régression de crête

La régression de crête (Ridge Regression) est une régression qui utilise la régularisation pour éviter le problème de surapprentissage. La régularisation emploie un terme de pénalité, appelé la norme $L2$, qui permet d'ajuster les poids et les biais et de façon que le modèle couvre les données de test (Bonaccorso & Safari, 2018; Montgomery & Runger, 2018; Subasi, 2020). Cela pourrait réduire relativement la performance du modèle pendant l'entrainement en raison qu'il devient moins sensible aux changements des variables indépendantes, mais le généralise et le rend certainement plus consistant durant la phase de test. La formule de la régression de crête est la suivante :

$$\mathrm{L} = \| Y - X\bar{\theta} \|_2^2 + \alpha \| \bar{\theta} \|_2^2$$

tel que : X est la matrice qui contient tous les échantillons dont chacun est représenté par une ligne, $\bar{\theta}$ représente le vecteur des poids, c'est la pente de la ligne correspondante au modèle (qui va changer), $||Y - X\bar{\theta}||_2^2$ représente alors la somme des carrés résidus SCR, $\alpha||\bar{\theta}||_2^2$ représente le terme de pénalité.

3.2.3 Régression LASSO

Comme la régression de crête, la régression Lasso vise à traiter le problème de surapprentissage et la sélection des variables en introduisant un terme de pénalité pour la régularisation. La seule différence est que ce dernier, appelé la norme $L1$ cette fois, est calculé en ajoutant la valeur absolue de la pente au lieu son carré (Bonaccorso & Safari, 2018; J. Suzuki, 2021). Sa formule est la suivante :

$$\mathrm{L} = \| Y - X\bar{\theta} \|_2^2 + \alpha \| \bar{\theta} \|_1$$

$\alpha||\bar{\theta}||_1$ est le degré de pénalité de $L1$. Une des caractéristiques de la régression Lasso est qu'elle peut aussi appliquée dans les cas où il y a de multiples variables indépendantes dont l'importance n'est pas déterminée.

3.3 Classification

La classification est un type d'apprentissage supervisé qui est largement considéré dans la communauté scientifique. Elle est utilisée pour différentes applications dans les différents domaines d'apprentissage automatique telles que la reconnaissance des formes, la détection de la parole, la classification des images, etc. (Subasi, 2020; Z.-H. Zhou, 2021). Son principe est d'allouer aux instances (échantillons) des classes spécifiques, ces dernières représentant alors des résultats catégoriques en fonction des variables prédicatives (Subasi, 2020; van der Aalst, 2016; Z.-H. Zhou, 2021). Selon le nombre de résultats discrets qui peuvent exister dans chaque problème, une classification peut être binaire ou multi-classes.

3.3.1 Classification binaire

La classification binaire est le type de classification dont la sortie peut être une des deux catégories distinctes (Bonaccorso & Safari, 2018). Les problèmes soulevés par la classification binaires sont multiples, parmi lesquels on trouve la classification des personnes en fumeurs et non-fumeurs (van der Aalst, 2016), l'analyse binaire des sentiments qui peuvent être positifs ou négatifs (Bonaccorso & Safari, 2018), etc.

3.3.2 Classification multi-classes

La classification multi-classes est celle dont le nombre de classes dans la sortie dépasse deux catégories. Elle a plusieurs applications telles que la catégorisation des documents, la classification des images, la reconnaissance faciale, etc. La résolution d'un problème de classification multi-classes implique généralement sa décomposition en multiples problèmes de classification binaire selon une stratégie adoptée (Z.-H. Zhou, 2021). Dans la phase d'entrainement, chaque classifieur binaire est entrainé séparément des autres. Les sorties collectées depuis tous les classifieurs binaires sont rassemblées dans les prédictions de la classification finale (Z.-H. Zhou, 2021). Parmi les stratégies à suivre pour la décomposition du problème on trouve : (1) une contre une (OvO), (2) une contre le reste (OvR) et (3) plusieurs contre plusieurs (MvM). Le rassemblement des résultats est réalisé en se servant des techniques d'apprentissage ensembliste.

3.3.3 Classification multi-labels

La classification multi-labels est utilisée dans les cas où chaque entrée peut être attribuée par plus qu'une classe ou aucune classe dans la sortie. Un exemple qui illustre ce type de classification est la classification des vêtements dans les images ; une seule image peut contenir à la fois un pantalon, un t-shirt et un chapeau. Un autre exemple est la classification des films selon les genres ; chaque film peut s'inscrire à la fois à plusieurs genres tels que comédie, action et horreur.

Le problème de déséquilibre des classes surgit dans les cas où les données ne sont pas distribuées de manière équitable, ce qui fait que des classes peuvent avoir un nombre important de données tandis que d'autres n'en possèdent qu'un nombre minime (Sagi & Rokach, 2018). Cela affecte le fonctionnement de l'algorithme d'apprentissage qui s'entraine en faveur des classes majeures.

3.3.4 Evaluation des modèles de classification

Il existe plusieurs mesures pour évaluer les modèles de classification, dont l'exactitude (Accuracy) et le taux d'erreur (Error rate) représentent les plus utilisées, mais qui ne sont pas toujours adéquates ou suffisantes à toutes les tâches de classification (Z.-H. Zhou, 2021). Ces mesures peuvent être définies à travers une matrice de confusion.

L'exactitude signifie la proportion des échantillons qui ont été classifiés correctement par rapport à tous les échantillons qui existent dans le data-set. Quant au taux d'erreur, il signifie la proportion des échantillons mal-classifiés par rapport à tous les échantillons du data-set (van der Aalst, 2016; Z.-H. Zhou, 2021). Selon la matrice de confusion, l'exactitude et le taux d'erreur sont définis respectivement par les équations suivantes.

$$Exactitude = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Erreur = \frac{FN + FP}{P + N}$$

La précision est utilisée pour mesurer la proportion des résultats pertinents (van der Aalst, 2016; Z.-H. Zhou, 2021). Sa formule est la suivante.

$$Précision = \frac{TP}{TP + FP}$$

Le rappel (Recall), aussi connu sous le nom sensibilité (sensitivity), est utilisé pour calculer la proportion des positives qui ont été prédites correctement (van der Aalst, 2016; Z.-H. Zhou, 2021). Il est en contradiction avec la précision ; s'il est élevé, la précision est basse, et vice-versa. Il est défini par la formule suivante.

$$Rappel = \frac{TP}{TP + FN}$$

Le score F1 indique la performance du modèle à travers la moyenne harmonique de la précision et le rappel (van der Aalst, 2016). Sa formule est la suivante.

$$F1 = \frac{2TP}{2TP + FP + FN} = \frac{Précision \times Rappel}{Précision + Rappel}$$

La courbe des Caractéristiques de fonctionnement du récepteur (ROC : Receiver Operating Characteristics) est représentée graphiquement par une courbe obtenue en mettant le taux des positives fausses (FPR : FP-rate) dans l'axe x et ce des positives vraies (TPR : TP-rate) dans l'axe y (Bonaccorso & Safari, 2018; Z.-H. Zhou, 2021). Ces taux sont respectivement définis par les formules suivantes.

$$FPR = \frac{FP}{TN + FP}$$

$$TPR = \frac{TP}{TP + FN}$$

L'aire sous la courbe (AUC : Area Under the Curve) est un moyen pour comparer des courbes ROC de classifieurs différents qui sont intersectées. Elle est calculée à travers l'intégrale des aires sous les courbes de ROC (Bonaccorso & Safari, 2018; Z.-H. Zhou, 2021). La valeur 0.5 d'AUC indique que le classifieur est inutile, tandis que 1 en indique la meilleure performance.

Dans la pratique, deux stratégies distinguées peuvent être adoptées pour évaluer les modèles de classification : la micro-moyenne (micro-average) et la macro-moyenne (macro-average) (Z.-H. Zhou, 2021). La première stratégie consiste à mesurer les performances de classification en traitant toutes les classes qui existent à la fois. Tandis que la seconde stratégie s'intéresse à mesurer les performances de classification selon chaque classe, c-à-d. toute classe est considérée indépendamment des autres, puis à calculer la moyenne des résultats. Dans le cas où toutes les classes possèdent le même nombre d'échantillons, le data-set est alors équilibré. Dans ce cas, toutes les deux stratégies donneront les mêmes résultats d'évaluation.

3.3.5 Algorithmes de classification

Régression logistique

La régression logistique est un algorithme populaire qui représente la forme basique de classification qui est utilisé pour résoudre les problèmes de classification linéaire et binaire et qui peut toutefois être généralisé à la classification multi-classes (Subasi, 2020). Cet algorithme consiste à classifier chaque échantillon selon la probabilité que ceci appartient à une classe (Bonaccorso & Safari, 2018), l'ensemble

des classes étant séparable linéairement (Subasi, 2020). En effet, cette probabilité est calculée à partir de la formule de la régression linéaire ($z = w^T x + b$). Cependant, comme les probabilités sont continues et bornées entre 0 et 1 dans le cas de classification binaire, et que le résultat doit prendre une de ces deux valeurs, un seuil est à introduire afin de filtrer la sortie de classification (Bonaccorso & Safari, 2018; Z.-H. Zhou, 2021). Pour cela, la fonction logistique, un type de fonctions sigmoïdes, est utilisée pour approximer les probabilités à 0 ou à 1. Sa formule est la suivante.

$$y = \frac{1}{1 - e^{-z}}$$

Classifieurs bayésiens naïfs

La classe des algorithmes de Bayes est celle qui englobe les algorithmes de classification dont le mécanisme de prédiction est basé sur le théorème de Bayes (Bonaccorso & Safari, 2018), ainsi qu'une hypothèse simplificatrice, dite naïve. Considérant X comme l'ensemble des entrées et y la sortie, la formule de probabilité conditionnelle peut être décrite comme suit (Bonaccorso & Safari, 2018; Z.-H. Zhou, 2021).

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

L'hypothèse simplificatrice signifie que, étant donné la classe y, les variables X sont indépendantes (Z.-H. Zhou, 2021). La probabilité $P(X|y)$ est traduite comme suit :

$$P(X|y) = P(x_1|y) \times P(x_2|y) \times \ldots \times P(x_n|y) = \prod_{i=0}^{n} P(x_i|y)$$

Comme P(X) est considéré comme une constante dans la formule, elle peut être éliminée en y introduisant une proportionnalité.

$$P(y|X) \propto P(X|y)P(y)$$

$$P(y|X) \propto P(y) \times \prod_{i=0}^{n} P(x_i|y)$$

A partir de cette formule, la stratégie du classifieur Bayésien naïf est de choisir la classe y qui a la probabilité maximale. Pour cela, $argmax$, décrite par la formule ci-dessous, est une opération qui a pour objectif de trouver l'argument qui donne la valeur maximale de y.

$$y = argmax_y[P(y) \times \prod_{i=0}^{n} P(x_i|y)]$$

Plusieurs algorithmes de classification appartiennent à la classe de Bayes. Parmi lesquels on trouve : Naïve Bayes, Averaged One-Dependence Estimators (ADDE), Bayesian Belief Networks (BBN) Gaussian Naïve Bayes (GNB), Multinomial Naïve Bayes (MNB) et Bayesian Networks (BN) (Z.-H. Zhou, 2021).

Arbre de décision

L'arbre de décision est un algorithme hiérarchique non-paramétrique qui utilise la structure d'arbre pour décomposer l'espace d'entrées pour la classification (ou la régression) selon le principe de *diviser pour*

régner et prendre une décision à chaque niveau jusqu'aux feuilles de façon récursive (Subasi, 2020). Il est considéré comme une heuristique gloutonne dont l'objectif est de trouver, à chaque nœud, l'attribut qui peut être utilisé pour diviser les données afin de minimiser leur entropie dans l'étape suivante (van der Aalst, 2016; Z.-H. Zhou, 2021). Les feuilles de l'arbre de décisions déterminent les résultats de l'analyse. Ces derniers représentent des étiquettes (labels) qui reflètent les classes qui existent dans le cas d'un problème de classification ou simplement des valeurs numériques dans le cas d'une régression (Subasi, 2020).

Forêt aléatoire

La forêt aléatoire peut simplement être définie comme un ensemble de classifieurs d'arbres de décision où l'on peut en choisir (élire) le meilleur classifieur pour une meilleure décision (Subasi, 2020). Cela permet de bénéficier de la simplicité des arbres de décision en utilisant l'apprentissage ensembliste d'une part, et d'avoir assez de flexibilité et d'efficacité des modèles d'apprentissage pour éviter le problème de surapprentissage d'une autre part (Subasi, 2020; Z.-H. Zhou, 2021). Le principe de cet algorithme peut être résumé en deux phases itératives :

1- Le bagging (bootstrap aggregating) (Subasi, 2020; Z.-H. Zhou, 2021) qui consiste à rééchantillonner aléatoirement les variables pour en passer chacune par un arbre de décision créé. Chaque data-set généré doit avoir la même taille que celle de l'original. Cette sélection aléatoire permet de maximiser la variété des arbres à entrainer. En réalité, le bagging est une technique d'apprentissage ensembliste qui sert à réduire la variance des data-sets pour permettre d'éviter le problème de surapprentissage.
2- La sélection aléatoire de k sous-ensembles des variables pour les classifieurs afin que ces derniers puissent les utiliser dans le processus d'apprentissage loin de la sensitivité des data-sets. Il est acquis que le nombre de variables recommandé pour être utilisé par chaque classifieur est déterminé par la formule (Z.-H. Zhou, 2021) : $k = log_2(d)$ où d est le nombre de classifieurs.

Machine à vecteurs de support (SVM)

Les machines à vecteurs de support sont un ensemble de techniques utilisées pour la classification binaire et multi-classes (Bonaccorso & Safari, 2018; Z.-H. Zhou, 2021), mais aussi pour le clustering. L'objectif des SVM est de trouver parmi les hyperplans qui séparent les données selon les différentes classes le plus approprié (Bonaccorso & Safari, 2018; Subasi, 2020; Z.-H. Zhou, 2021). Cet hyperplan, défini par la fonction de classification, doit assurer la plus grande capacité de généralisation, surtout face aux limites que présentent les data-sets et les valeurs aberrantes qu'ils possèdent (Z.-H. Zhou, 2021). Pour cela, SVM repose sur la maximisation de l'écart entre les données à séparer (Bonaccorso & Safari, 2018; Z.-H. Zhou, 2021). Cela peut se faire à travers l'optimisation convexe (Z.-H. Zhou, 2021).

Par défaut, SVM est plus adapté aux données qui sont linéairement séparables. Afin de pouvoir être utilisé pour les non linéaires, il utilise une fonction appelée l'Astuce de Noyau (Kernel Trick) (Bonaccorso & Safari, 2018; Subasi, 2020; Z.-H. Zhou, 2021) qui consiste à appliquer des transformations sur les données pour les rendre linéairement séparables.

K plus proches voisins (KNN)

L'algorithme de KNN est utilisé pour des prédictions sur de nouvelles données en se basant sur leurs distances avec les données qui sont déjà connues (Z.-H. Zhou, 2021). Il a plusieurs applications aussi bien en classification qu'en régression. Dans le cas de classification, les classes des données de test sont prédites en utilisant le vote sur la classe la plus fréquente dans leur voisinage, tandis qu'en régression

les valeurs de test sont prédites en se basant sur la moyenne de k valeurs voisines les plus proches (Subasi, 2020; Z.-H. Zhou, 2021).

Le processus de KNN pour un problème de classification peut être résumé comme suit :

1- Définir la valeur de k de manière adéquate.
2- Trouver les k plus proches voisins du point à classifier selon la métrique de distance définie.
3- La classe à laquelle appartient ce point est celle qui comprend le plus grand des voisins trouvés.

3.4 Règles d'association

Cette méthode est largement utilisée dans le domaine du marketing, médical et plusieurs autres applications connues sous le nom : Analyse du Panier de Consommation (Market Basket Analysis) (Subasi, 2020). Elle peut simplement être définie comme un ensemble de conditions utilisées pour calculer la probabilité relative aux patterns en se basant sur la corrélation entre eux. Les relations entre les objets sont représentées par des règles. Formellement, les règles d'association sont définies par la formule : $X \rightarrow Y$ où X, appelé l'antécédent et Y, appelé le conséquent, sont des ensembles d'objets (item-set) dans la base de données (van der Aalst, 2016). Deux ensembles sont distingués dans cette méthode : un ensemble $I = \{i_1, i_2, \dots, i_d\}$ d'objets (items) et un ensemble $T = \{t_1, t_2, \dots, t_n\}$ de transactions. I contient tous les objets de X et Y. Tandis que T contient toutes les combinaisons (interactions) entre ces objets.

La performance des règles d'association repose sur trois métriques essentielles : support, confiance et lift. Ils sont respectivement définis par les formules suivantes (Kaur & Madan, 2015; McNicholas et al., 2008; van der Aalst, 2016; Vijayarani & Sharmila, 2016).

$$Support(X,Y) = \frac{fréquence(X,Y)}{N}$$

$$Confiance(X,Y) = \frac{fréquence(X,Y)}{fréquence(X)}$$

$$Lift(X,Y) = \frac{Support(X \cap Y)}{Support(X) \times Support(Y)}$$

Plusieurs algorithmes peuvent être employés pour implémenter le concept des règles d'associations. Parmi lesquels on trouve AIS, SETM, Apriori avec ses dérivés (AprioriTid, AprioriHybrid), Algorithme de croissance de modèle fréquent (FP), Algorithme génétique, Dclat Algorithm et Eclat Algorithm (Kaur & Madan, 2015; Kumbhare & Chobe, 2014; van der Aalst, 2016; Vijayarani & Sharmila, 2016).

3.5 Clustering

Le clustering contient l'ensemble d'algorithmes les plus populaires dans l'apprentissage non-supervisé (Bonaccorso & Safari, 2018; Z.-H. Zhou, 2021). Il a plusieurs applications modernes telles que la recommandation des films et des applications, le regroupement et la détection des communautés des utilisateurs et des pages dans les réseaux sociaux, (Z.-H. Zhou, 2021), etc. Il peut être défini par le processus d'exploiter les relations entre les données d'un data-set pour en construire des groupes où chacun, appelé cluster, contient les données similaires selon un critère défini (Bonaccorso & Safari, 2018) Formellement, pour un data-set $D = \{x_1, x_2, \dots, x_n\}$ de n échantillons, $x_i \in \mathbb{R}^m$ est un vecteur de dimension m, il s'agit de partitionner D en k clusters non vides $C = \{C_l | l = 1, \dots, k\}$ tels que $D = \bigcup_{l=1}^{k} C_l$ et $C_{l'} \bigcap_{l' \neq l} C_l = \emptyset$ (Bonaccorso & Safari, 2018; Z.-H. Zhou, 2021). **Figure *2-2*** montre des exemples des catégoriques classiques des algorithmes de clustering, à savoir ceux basés (centroïde),

basés densité, hiérarchiques et basés distribution. Aujourd'hui, il existe tant d'autres types de clustering tels que le clustering agglomératif, Birch, Mean-shift, Spectral, Affinity Propagation, etc. Dans ce qui suit est décrit le clustering K-moyennes (KMC : K-means clustering), très courant dans sa classe.

Figure 2-2: Types classiques de clustering

KMC est basé centroïde. Il utilise les données en les représentant par des points. Ensuite, il essaie de diviser ces points en k groupes de sorte que chacun d'eux soit le plus proche de son centre (parmi les k centres). Le principe de cet algorithme peut se résumer dans les points suivants (Mahesh, 2018; J. Suzuki, 2021).

1- Prendre aléatoirement k centres de groupes.
2- Assigner chaque point au centre le plus proche.
3- Recalculer les centres en se basant sur la position moyenne des points de chacun.
4- Répéter les étapes précédentes jusqu'il ne reste plus de changement de centres.

L'idée est alors de minimiser la distance entre les points appartenant au même cluster et de maximiser celle entre les différents clusters. Pour prédire le groupe (cluster) de nouveaux points, il faut juste trouver le centre qui leur est le plus proche.

4 Techniques d'aprentissage automatique avancées

L'émergence rapide d'apprentissage automatique dans les différents domaines de la science des données l'a rendu indispensable comme solution effective à utiliser dans les différentes étapes du processus de KDD, notamment pour la fouille des données. Cela a attiré plus d'intérêt de la communauté scientifique dont la recherche des solutions meilleures et optimisées des modèles d'apprentissage automatique reste préoccupante. Cet intérêt a engendré des nouvelles approches avancées à travers lesquelles est tendue l'envergure d'apprentissage automatique sur d'autres nouveaux domaines et d'autres technologies modernes. Parmi les techniques qui font l'objet de cette section il y a l'apprentissage profond, l'apprentissage ensembliste et l'apprentissage par transfert.

4.1 Apprentissage profond

L'apprentissage profond (Deep Learning) est une classe d'algorithmes d'apprentissage automatique qui utilisent une structure multicouche pour apprendre et extraire à partir des données des caractéristiques de haut-niveau. Il consiste à représenter les données brutes sous la forme d'une hiérarchie imbriquée de variables et les passer par les différentes couches. Le concept de profondeur dans cette classe d'apprentissage automatique fait référence à la multitude des couches que peuvent avoir les modèles d'apprentissage profond pour la transformation des données. Cette multitude implique plus de calcul et rend profond le « chemin » depuis les entrées jusqu'aux sorties.

Le réseau de neurones artificiel (ANN : Artificial Neural Network) représente la base de la plupart des modèles d'apprentissage profond. Il est composé d'un ensemble d'unités interconnectées et organisées

hiérarchiquement appelées les neurones (L. Chen, 2021). Différentes couches qui contiennent ces neurones effectuent de différentes transformations sur les entrées afin d'obtenir des niveaux d'abstraction différents ainsi que d'en extraire les caractéristiques. Ces opérations produisent un modèle entrainable pour la tâche d'apprentissage qui lui est confiée.

4.1.1 Composants d'un ANN

La structure générale de tout réseau de neurones artificiel est constituée de quatre composants dont trois principaux et un optionnel (da Silva et al., 2017; Nanda et al., 2015).

1. Une couche d'entrées qui est aussi connue sous le nom : le vecteur d'entrées. Elle contient une représentation des variables indépendantes
2. Une ou plusieurs couches cachées qui sont dédiées au traitement. Ces couches sont optionnelles en fonction de l'architecture de l'ANN en question.
3. Des connexions pondérées entre les nœuds des couches adjacentes.
4. Une couche de sortie qui représente la variable indépendante. Selon le problème en question, cette couche peut contenir un ou plusieurs éléments.

Figure 2-3 illustre le concept d'un ANN. En effet, ce réseau s'appelle le perceptron multi-couches (MLP : Multi-Layer Perceptron). Il est aussi connu sous le nom de réseau de neurones à propagation en avant (FFN : Feed-Forward Network). MLP représente l'architecture classique des réseaux de neurones qui est composée de multiples couches qui fonctionnent selon la propriété de propagation en avant. Cette propriété signifie que le flux des données qui passent à travers MLP est transféré en avant (depuis couche d'entrées vers la couche de sortie), sans retour en arrière d'une couche vers une autre couche qui la précède (L. Chen, 2021).

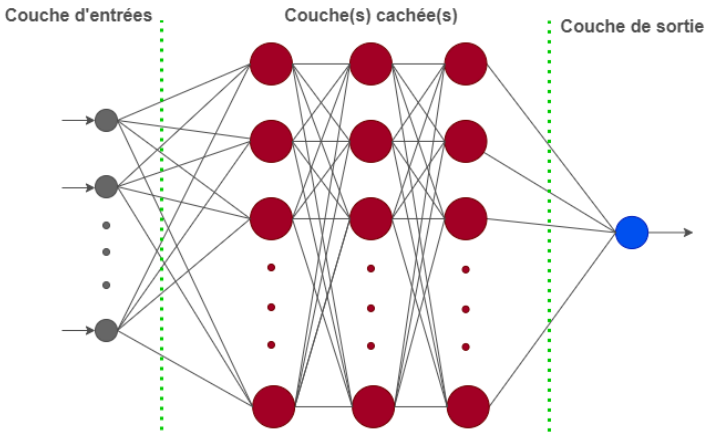

Figure 2-3 : Illustration du concept de réseaux de neurones (inspirée de (Sze et al., 2017))

Comme le montre **Figure 2-4**, un neurone étant l'unité de base dans un réseau de neurones est essentiellement constitué des composants suivants : une entrée, une fonction d'addition, une fonction d'activation, un biais, et une sortie (Nanda et al., 2015; Sze et al., 2017). L'entrée est un vecteur qui représente les variables à passer par le neurone. Ce vecteur est accompagné par un vecteur de poids, ces derniers étant des valeurs dont chacune est multipliée par l'élément du premier auquel elle est attribuée. Le biais est une valeur qui est ajoutée aux entrées pondérées conformément à un seuil défini qu'il ne faut pas excéder.

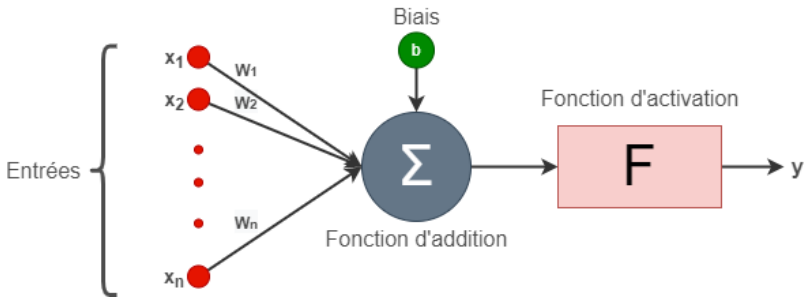

Figure 2-4 : Composants d'un neurone (inspirée de (Nanda et al., 2015; Sze et al., 2017))

La fonction d'activation, aussi appelée la fonction de transformation, est une fonction non-linéaire qui définit comment est transformée la somme pondérée des entrées en sorties des nœuds d'une couche dans un réseau de neurones. Ayant un grand impact sur les performances des réseaux de neurones, la fonction d'activation doit être choisie parmi celles qui existent d'une façon déterminée pour qu'elle soit adéquate avec le fonctionnement du modèle défini. Il existe de multiples fonctions d'activation classiques telles que la fonction sigmoïde et la tangente hyperbolique (tanh), ainsi que modernes telles que l'unité de rectification linéaire (ReLU : Rectified Linear Unit), l'unité exponentielle linéaire (ELU : Exponentional Linear Unit) et Leaky ReLU (LReLU) (Bonaccorso & Safari, 2018; Sze et al., 2017; Z.-H. Zhou, 2021).

4.1.2 Algorithme de rétropropagation du gradient

La rétropropagation du gradient est une méthode itérative d'optimisation dont l'objectif est de minimiser une fonction différentiable définie sur un espace déterminé. A cause de sa nature itérative, l'algorithme de rétropropagation du gradient optimise la fonction de coût en effectuant un des ‘’déplacements’’ via la recherche linéaire vers des valeurs inférieures ou supérieures qui la rapprochent de l'optimalité locale. L'optimalité globale n'est cependant pas garantie. Cet algorithme est un parmi les algorithmes les plus populaires pour l'optimisation dans l'apprentissage automatique, notamment profond. Il est souvent utilisé comme un optimisateur qui fonctionne en boîte noire (Ruder, 2017).

4.1.3 Architectures d'ANN

Le processus de rétropropagation du gradient peut être résumé en deux étapes (L. Chen, 2021; Mohammed et al., 2017). La première étape, appelée l'étape de propagation, consiste à passer les entrées par le réseau. Ce dernier produit la sortie dont deux types sont distingués : les sorties intermédiaires qui sont calculées au niveau des couches cachées, et la sortie finale qui est calculée par la couche de sortie. La seconde étape, appelée la rétropropagation, consiste à propager l'erreur de prédiction qui est calculée au niveau de la couche de sortie vers l'arrière pour ajuster les poids conformément à la somme des carrés de l'erreur. Pour minimiser cette dernière, l'algorithme de rétropropagation utilise l'optimisation déterministe via la méthode du gradient. Le calcul du gradient implique le calcul des dérivées partielles de la fonction d'activation par rapport aux poids des entrées (Calin, 2020).

Il existe de nombreuses architecture d'ANN qui ont été proposées afin de résoudre des problèmes d'apprentissage de manière optimisées. Parmi lesquelles on trouve :

- Les réseaux de neurones convolutifs (CNN : Convolutional Neural Networks) qui sont dédiés aux tâches de la vision par ordinateur (Computer Vision) telles que la classification des images, la reconnaissance d'écriture manuscrite et des formes, la classification des vidéos, la reconnaissance de la parole, etc. (Calin, 2020; L. Chen, 2021; Sze et al., 2017)

- Les réseaux de neurones récurrents (RNN : Recurrent Neural Networks) dont la propagation est faite dans les deux sens (en avant et en arrière) (Calin, 2020; Z.-H. Zhou, 2021). Ils ont montré une haute effectivité les différentes applications telles de NLP y compris l'analyse des sentiments, classification des documents, catégorisation des sujets, prédiction des mots suivants, résumé automatique des textes (ATS : Automatic Text Summarization), ainsi que les données séquentielles en général.
- Les machines de Boltzmann Restreintes (RBM : Restricted Boltzmann Machines) (Calin, 2020; Q. Zhang et al., 2018) qui sont des réseaux de neurones stochastiques génératifs utilisés essentiellement comme solution d'apprentissage non-supervisé pour le filtrage collaboratif, la réduction de dimensionnalité, la modélisation des sujets, l'apprentissage (extraction) des caractéristiques (feature learning, representation learning, feature extraction), etc.
- Les auto-encodeurs qui sont des techniques non-supervisées utilisées pour l'apprentissage des caractéristiques depuis les données étiquetées ou non-étiquetés (W. Liu et al., 2017). Il existe plusieurs types d'auto-encodeur dont les Réseaux Antagonistes Génératifs (GAN : Generative Adversarial Networks) et les Auto-encodeurs Variationnels (VAE : Variational Autoencoders) sont les plus connus.
- Les transformateurs qui utilisent le mécanisme d'attention (Vaswani et al., 2017) et qui ont montré une surperformance dans les tâches de NLP notamment la traduction contextuelle du texte et la détection de similarité entre les données textuelles.

4.2 Apprentissage ensembliste

L'apprentissage ensembliste (Ensemble Learning) consiste à combiner multiples modèles afin d'améliorer leur performance, sélectionner les variables optimales, fusionner les données, etc. (Bonaccorso & Safari, 2018; Dong et al., 2020; Mahesh, 2018; Subasi, 2020). **Figure *2-5*** illustre ce concept. Les modèles utilisés dans ce type d'apprentissage, entrainés par des algorithmes d'apprentissage existants (Z.-H. Zhou, 2021), peuvent être homogènes, c-à-d. du même type tels que de multiples classifieurs, ou hétérogènes telle une combinaison d'un modèle de classification avec un modèle de régression linéaire (Mahesh, 2018).

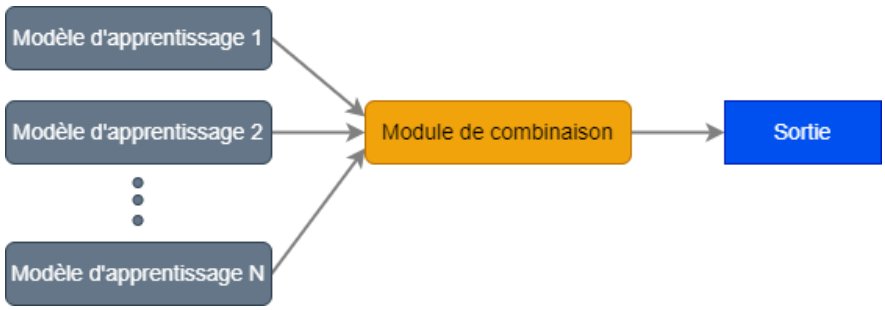

Figure 2-5 : Concept de l'apprentissage ensembliste (Z.-H. Zhou, 2021)

Il existe différentes familles d'apprentissage ensembliste dont les plus populaires sont le Bagging, le Boosting, le Bucket of models et le Stacking (Bonaccorso & Safari, 2018; Dong et al., 2020; Sagi & Rokach, 2018; C. Zhang & Ma, 2012; Z.-H. Zhou, 2021).

4.3 Apprentissage par transfert

L'apprentissage par transfert (Transfer Learning) est un type d'apprentissage automatique qui repose sur le principe d'utiliser les connaissances d'un ou de plusieurs domaines pour la construction et la réalisation d'un nouveau modèle dans un domaine similaire (Kaboli, 2017; Weiss et al., 2016; Zhuang et al., 2021). L'objectif est de bénéficier des avantages que présentent les modèles prédéfinis à combiner pour donner plus de consistance au modèle proposé. **Figure *2-6*** montre le principe de ce type

d'apprentissage. En pratique, il est très répandu ; un modèle peut combiner des techniques d'apprentissage automatique différentes, voire d'apprentissage profond. La régression basée KNN (Mohammed et al., 2017) illustre ce concept. Sans se soucier de la disponibilité de grandes quantités de données, l'apprentissage par transfert permet d'aboutir à des résultats satisfaisants en adaptant les modèles pré-entrainés aux problèmes en question. Son utilité est de remédier à des problèmes liés aux données à savoir l'indisponibilité de celles-ci, le coût élevé de les collecter et de les étiqueter ou leur l'inaccessibilité (Weiss et al., 2016).

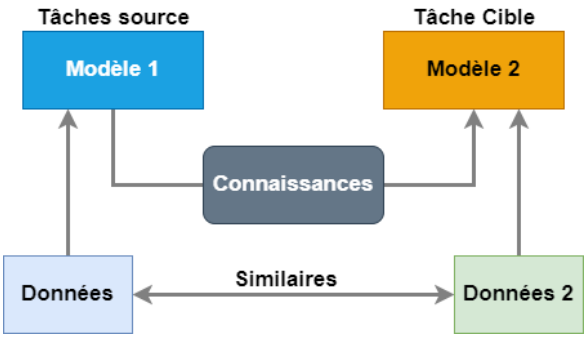

Figure 2-6 : Principe de l'apprentissage par transfert (M. Suzuki et al., 2014)

En effet, l'apprentissage par transfert a attiré l'attention de la communauté scientifique dans différentes applications, notamment les tâches de vison et de NLP. Dans la vision par exemple, un modèle de classification des objets peut être mis à profit pour la classification des scènes. Dans NLP également, un modèle de détection de similarité dans les textes peut être exploité pour la traduction contextuelle. L'apprentissage par transfert peut être homogène ou hétérogène (Weiss et al., 2016; Zhuang et al., 2021). Le transfert homogène est dédié aux situations où le domaine source et le domaine cible sont du même espace de données. Le défi ici est d'adapter leurs distributions. Quant au transfert hétérogène, il est destiné aux situations où ces domaines ont différents espaces de données. L'intérêt ne se limite pas à l'adaptation des distributions des données pour les deux domaines, mais aussi à l'adaptation des domaines de ces données.

Différentes techniques peuvent être utilisées pour transférer l'apprentissage d'un modèle vers un autre. Elles peuvent être catégorisées selon différents critères y compris celui des paramètres des espaces qui les distingue en homogènes et hétérogènes. Parmi ces critères il y a la stratégie d'apprentissage par transfert et la nature des solutions utilisées (Zhuang et al., 2021). Selon la stratégie, toute technique peut appartenir à une des quatre catégories suivantes : (1) le transfert via les instances, (2) le transfert à travers les caractéristiques (features), (3) le transfert à travers les paramètres partagés entre les modèles d'apprentissage du domaine source et ceux du domaine cible et (4) le transfert en se basant sur des relations définies entre le domaine source et le domaine cible (Kaboli, 2017; Weiss et al., 2016; Zhuang et al., 2021). Selon la nature des données, les techniques d'apprentissage par transfert peuvent être (1) transductives, (2) inductives ou (3) non-supervisées. **Figure *2-7*** résume ces catégories.

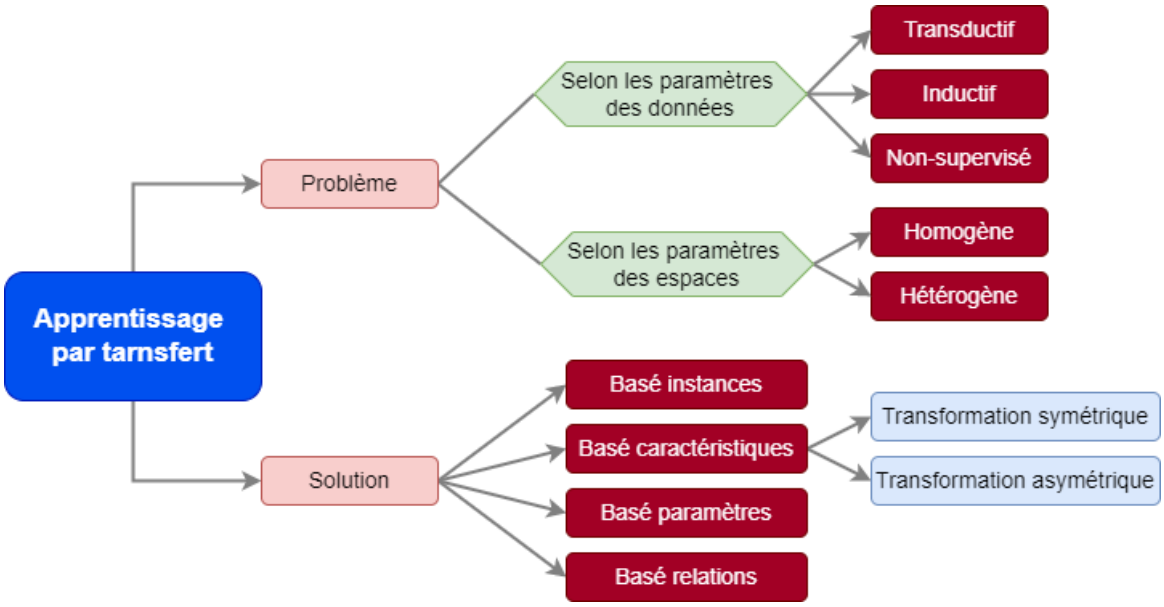

Figure 2-7 : Techniques d'apprentissage par transfert (Zhuang et al., 2021)

Parmi les librairies qui sont largement répandues pour implémenter les différentes solutions mentionnées on trouve SciKit-Learn et Spark MLLib pour l'apprentissage automatique de tout genre, en plus de Theano, Keras, Tensorflow, MXNet et Pytorch pour l'apprentissage profond (Bonaccorso & Safari, 2018; Elshawi et al., 2021; Subasi, 2020; VanderPlas, 2016).

5 Conclusion

Ce chapitre a couvert essentiellement deux étapes importantes dans tout cycle d'extraction des connaissances à partir des données, à savoir le nettoyage des données et la fouille de données. Le nettoyage des données est une étape qui nécessite une attention particulière car il affecte directement le résultat d'analyse et d'analytique des données. Il englobe plusieurs volets, en l'occurrence la détection d'anomalies, l'imputation des valeurs manquantes, l'optimisation des données et la réduction de la dimensionnalité. Différentes techniques et méthodes utilisées pour chacun de ces volets ont été décrites. La fouille des données, étant une étape de la plus grande importance, a été décrite par la suite. L'apprentissage automatique constitue la solution référence aux problèmes de fouille des données. Des algorithmes qui appartiennent à différents types d'apprentissage automatique tels que le supervisé, le non-supervisé, le profond et l'ensembliste ont été abordés.

Chapitre 3: Visualisation interactive des données dans le contexte Big Data

1 Introduction

La visualisation des données est le processus de les présenter sous forme visuelle. Puisque l'être humain est généralement plus à l'aise avec les images qu'avec les chiffres et les textes, la visualisation lui facilite l'exploration et l'analyse des données et lui permet d'obtenir des aperçus sur ces données. Le concept de visualisation est multidimensionnel, et la manière de visualiser les données peut varier d'une application à une autre selon son objectif. Parmi les applications on peut trouver celles dont l'intérêt est juste de mapper les données avec des formes graphiques qui les représentent afin de faciliter leur recherche et leur exploration, celles qui s'intéressent à visualiser les statistiques des données et celles qui visent à présenter visuellement les patterns, c-à-d. les informations et les connaissances cachées à l'intérieur des données. Le dernier cas représente l'analytique visuelle (Visual Analytics) qui consiste à effectuer de l'analytique sur les données via des techniques et méthodes d'extraction des connaissances et à présenter visuellement les résultats. La visualisation, un sujet encore de tendance bien qu'ancien à l'origine, devient de plus en plus indispensable pour la prise de décision dans les différentes industries. Sa portée en termes d'application et sa considération en termes de techniques et outils ne cessent de s'élargir avec la propagation de Big Data. Par conséquent, son marché connait de nos jours une importante croissance qui a atteint la valeur de 4.51 milliards $ en 2017 et qui devrait atteindre les 7.76 en 2023[5].

Le reste de ce chapitre est organisé comme suit : La deuxième section présente les notions générales liées au concept de la visualisation des données, notamment les conventions qui lui sont liées et la latence dans ce contexte. La troisième section met en évidence les types de visualisation selon différents critères. La quatrième section présente les techniques de visualisation classiques et avancés répandues dans les différents domaines d'application. La cinquième section présente des outils de visualisation récents et décrit leurs caractéristiques. La sixième section projette le concept de visualisation sur le contexte Big Data et surligne les axes à reconsidérer lors de cette projection tout en abordant des méthodes avancées de visualisation et d'analytique visuelle dans Big Data. La septième section conclut ce chapitre.

2 Visualisation des données : concepts de base

La visualisation a pour mission de présenter les données graphiquement. Ces données peuvent être de différents types tels que les textes, les images, les fichiers audios, les données numériques, etc. L'objectif principal de la visualisation est de simplifier à l'utilisateur la perception et l'exploration des données granulaires quels que soient leurs types (Davenport, 2014). Un exemple illustratif est celui des données des marchés boursiers qui sont numériques. Sans une présentation graphique, ces données seront difficiles à explorer et à comprendre par les analystes. Par conséquent, leur analyse deviendra un grand défi, ce qui rend difficile de tirer des conclusions et interrompt les tâches subséquentes. Dans certains cas, il ne serait pas pratique de visualiser les données en elles-mêmes, mais plutôt leurs métadonnées. Dans le cas des images et des vidéos, l'objectif de visualisation peut consister à, présenter leurs propriétés. Pour les images, ces propriétés peuvent englober la qualité, la taille, la date de création, etc. Les vidéos peuvent également se doter chacune d'une taille, de la durée, la qualité, le titre, l'auteur, la catégorie, etc. Visualiser les métadonnées pourrait être plus avantageux et moins coûteux que visualiser les informations extraites via des techniques spécialisées telles que la segmentation, la détection des formes, etc. La notion d'interactivité dans le contexte de visualisation est apparue pour introduire l'aspect utilisateur comme un acteur en son sein. L'expérience utilisateur consiste à personnaliser la visualisation en termes de données et la manière de les présenter conformément aux actions de personnalisation (Ali et al., 2016; Dietrich et al., 2015; Godfrey et al., 2016; Lam et al., 2012).

[5] https://www.statista.com/statistics/1003906/worldwide-data-visualization-market-value/

2.1 Méthodologie pour proposer une visualisation

Le développement des méthodes et systèmes de visualisation de données doit être précédé par une étude méthodologique afin qu'ils soient effectifs en termes de compréhensibilité et significativité vis-à-vis des utilisateurs. Cette méthodologie doit couvrir de multiples axes liés à l'application, à savoir les utilisateurs concernés par la visualisation, le(s) type(s) des données à visualiser, les techniques de visualisation à utiliser et les outils à employer pour implémenter ces techniques.

L'aspect utilisateur est primordial car c'est selon les utilisateurs cibles que les visualisations adéquates sont déterminées. En pratique, il peut y avoir différents types d'utilisateur selon le domaine d'application. Parmi ces types on peut trouver les utilisateurs ordinaires, les statisticiens, les analystes, … Un utilisateur ordinaire peut ne pas comprendre des visualisations statistiques telles que les boîtes à moustaches ou les coordonnées parallèles. Ces dernières sont, par contre, compréhensibles par les analystes et les statisticiens. C'est pourquoi il faut spécifier la catégorie des utilisateur ciblés afin de leur sélectionner les données qui susceptibles de les intéresser et les techniques de visualisation adéquates.

La définition à priori des types des données à visualiser permet de gagner du temps lors du développement des solutions de visualisation. Cela est réalisé en tenant compte des techniques qui peuvent être utilisées pour supporter la visualisation de ces types ainsi que les outils appropriés pour réaliser la visualisation sur la base de ces techniques.

Le choix des techniques de visualisation à utiliser n'est pas uniquement lié aux types des données à visualiser, mais aussi d'autres paramètres essentiels, notamment les spécificités des applications et des supports d'affichage vis-à-vis de la nature de visualisation qui reflète les objectifs désignés.

La sélection des outils de visualisation appropriés pour implémenter les techniques repose la considération de multiples facteurs, à savoir leur capacité à fournir ces techniques de visualisation, leur mise en échelle avec la dynamique des données, leur performance et optimalité en termes de ressources de calcul et de latence, ainsi que le rapport qualité prix.

2.2 Latence

Dans le contexte de visualisation, la latence fait référence à la nécessité d'imposer un intervalle de temps à respecter pour offrir la présentation visuelle des données. Généralement, il est acquis que la valeur de latence par défaut est égale à 5 secondes.

2.3 Conventions et contraintes de visualisation

Dans le contexte de la visualisation, l'aspect des contraintes est primordial pour la validation des approches et solutions proposées. Les contraintes sont utilisées pour évaluer l'effectivité et la qualité de visualisation vis-à-vis des objectifs désignés. Elles peuvent être classées en deux catégories, en l'occurrence (1) les contraintes basiques et (2) les contraintes d'interactivité.

2.3.1 Contraintes basiques

L'ensemble des contraintes de visualisation basiques représente celles qui sont nécessaires pour la valider quel que soit le type de cette visualisation. Cet ensemble comprend essentiellement la contrainte d'expressivité, d'efficacité et de non-occlusion de pixels (Erraissi & Belangour, 2020; Godfrey et al., 2016; Kahil et al., 2020).

L'expressivité de visualisation reflète sa capacité à exprimer l'ensemble des données via des formes et des structures graphiques significatives telles que le point, le cercle, le carré, l'arbre, etc. La visualisation doit alors assurer le processus de mappage entre les données et les formes graphiques qui leur correspondent. L'effectivité est liée à la compréhensibilité des formes employées dans la visualisation

par l'utilisateur cible. Dans la visualisation de structure arborescente par exemple, il faut présenter les données dans un ordre logique et significatif selon les composants de l'arbre et les niveaux qui s'y trouvent afin de simplifier la compréhension de cette présentation par l'utilisateur et, par conséquent, lui faciliter l'exploration. La racine peut représenter le pattern principal des données visualisées, les branches peuvent contenir les différents axes qui appartiennent au pattern principal, et les feuilles peuvent véhiculer les informations élémentaires qui sont liées axes présentés au niveau des branches. L'occlusion des pixels (pixel overplot) fait référence à la situation où un pixel est occupé à la fois par différents éléments de données. Avec la multitude des données dans Big Data, cette situation est souvent susceptible d'avoir lieu. La contrainte de non-occlusion des pixels doit alors être vérifiée.

2.3.2 Contraintes d'interactivité

La visualisation interactive implique l'introduction de l'utilisateur au sein du système de visualisation en le considérant comme acteur. Cela signifie que ce système doit offrir à l'utilisateur des fonctionnalités à travers lesquelles il peut personnaliser la visualisation selon ses besoins et intérêts (Dimara & Perin, 2020; Lam et al., 2012). Parmi ces fonctionnalités il peut y avoir les paramètres d'ajustement d'affichage tels que le zoom, la sélection, l'affichage panoramique, des codes pour permettre à l'utilisateur de créer et de personnaliser son propre affichage, etc. ainsi que les paramètre de personnalisation des données tels que le filtrage, recherche, etc. (Dimara & Perin, 2020; Godfrey et al., 2016; Kahil et al., 2020).

3 Types de visualisation

La visualisation est un concept multidimensionnel. Selon son application, elle peut être distinguée en trois catégories principales, à savoir la visualisation des données, la visualisation des informations et la visualisation scientifique. Ci-dessous est mise en évidence la différence entre la visualisation des données et la visualisation des informations d'une part, et entre la visualisation des informations et la visualisation scientifique d'autre part.

3.1 Visualisation des données et analytique visuelle

Bien que les concepts de visualisation des données et d'analytique visuelle (Visual Analytics) soient confondus du fait qu'ils consistent à présenter graphiquement les données, ils sont en réalité différents. La visualisation des données a pour objectif de présenter les données à l'utilisateur d'une manière organisée et efficiente pour lui faciliter l'exploration (Ali et al., 2016; Godfrey et al., 2016), mais sans assurer la pertinence des données présentées. L'analytique visuelle, quant à elle, a pour objectif de visualiser des patterns extraits des données sur lesquels sont tirées des conclusions et prises des décisions. Elle traduit alors les résultats de tout un processus d'analyse et de fouille de données tel que le processus de KDD et ses dérivés, et finit par interpréter graphiquement ces résultats pour de futures utilisations (Fiaz et al., 2016; Lam et al., 2012). Tous les outils et techniques de visualisation ne sont employés qu'à la dernière étape de l'analytique visuelle.

3.2 Visualisation des informations et visualisation scientifique

La visualisation des informations, plus connue récemment sous le nom de l'analytique visuelle, vise à présenter les informations cachées dans les données qui ne sont pas physiques sous forme graphique via différentes techniques de visualisation, notamment celles qui sont basées sur les statistiques (Lam et al., 2012; Nagel, 2006). Parmi les applications de la visualisation des informations on trouve la visualisation textuelle selon la fréquence des mots, la visualisation du flux des données dans un réseau, etc. La visualisation scientifique consiste à produire des représentations par images à partir des données des phénomènes scientifiques afin de faciliter aux spécialistes de les comprendre, d'en avoir un aperçu et de les interpréter d'une manière consistante. Cet objectif n'est généralement pas atteignable via l'utilisation exclusive des méthodes statistiques et les techniques de visualisation des données. La visualisation scientifique, pouvant être 2D ou 3D, est réalisée via une multitude de techniques qui appartiennent à

différentes disciplines, à savoir le traitement du signal, l'infographie, le traitement d'image, l'animation par ordinateur, la simulation, la conception assistée par ordinateur (computer-aided design) et l'interaction homme-machine (Chawla et al., 2018; Nagel, 2006). Elle a plusieurs applications telles que l'écologie, la géographie, la science de la nature, etc.

4 Techniques de visualisation

Les techniques de visualisation peuvent être classées selon différents critères tels que :

1- L'objectif : visualisation des data-sets, visualisation des types des données, visualisation d'un domaine.
2- Les dimensions : 2d, 3d, multidimensionnelles, graphe, etc. (Raghav et al., 2016).
3- Les types de visualisation : comme montré dans la section ci-dessus.

Au-delà de ces critères, ci-dessous sont décrites des techniques de visualisation, basiques et avancées, qui sont répandues dans les différentes applications modernes. Parmi ces techniques les plus populaires sont décrites ci-dessous (Ali et al., 2016; Chawla et al., 2018; Fahad & Yahya, 2018; Raghav et al., 2016).

4.1 Cartes

Il existe plusieurs techniques qui appartiennent à la classe de visualisation basée carte (Map). Parmi lesquelles sont décrites ci-dessous la carte thermique, la carte arborescente et la carte à bulles. Elles sont respectivement illustrées dans **Figure 3-1**, **Figure 3-2** et **Figure 3-3**.

4.1.1 Carte thermique

La carte thermique (Heat map) offre une représentation des données en deux dimensions. Elle désigne une table avec des lignes et de colonnes. Les valeurs des cellules, numériques ou catégoriques, sont encodées par des couleurs qui offrent une visualisation adéquate. Les cartes thermiques permettent de comparer les catégories des data-sets pour déterminer les corrélations entre elles de manière efficace à la base des couleurs plutôt que des chiffres (Camm et al., 2021). Ainsi, elle simplifie aux utilisateur l'exploration des data-sets de manière visuelle.

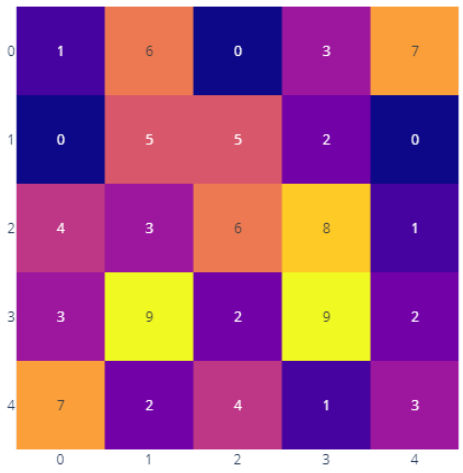

Figure 3-1 : Exemple d'une carte thermique

4.1.2 Carte arborescente

La carte arborescente (Treemap) est une technique qui permet de visualiser les données hiérarchiques sous forme de rectangles imbriqués. Chaque niveau de cette hiérarchie est représenté par un rectangle coloré. Les niveaux supérieurs sont appelés les branches. Ils contiennent chacun d'autres rectangles avec

des tailles inférieures appelés les feuilles. La taille des rectangles est relative aux valeurs quantitatives correspondantes aux points de données. Les cartes arborescentes sont essentiellement utilisées avec les données qui peuvent être agrégées lorsque l'intérêt est d'avoir un aperçu sur ces données (Camm et al., 2021). Elles permettent de visualiser des volumes importants de données hiérarchiques dans un espace contraint.

Figure 3-2 : Exemple d'une carte arborescente

4.1.3 Carte à bulles

La carte à bulles (Bubble maps) est une technique de visualisation qui est basée sur les cartes géographiques pour représenter les données spatiales. Elle utilise des bulles de tailles variables proportionnelles aux valeurs des données discrètes d'un phénomène précis (Camm et al., 2021; Tominski & Schumann, 2020). Elle a différentes applications telles que la visualisation des populations dans les zones géographiques, le dépistage de la propagation d'une maladie, etc.

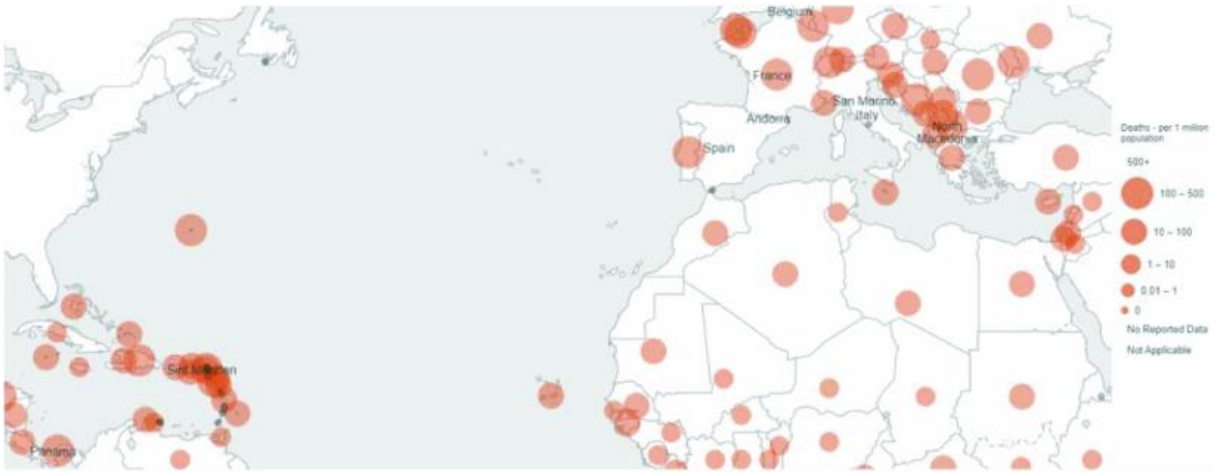

Figure 3-3 : Exemple d'une carte à bulles (Leung et al., 2020)

4.2 Diagramme circulaire

Le diagramme circulaire (Pie chart) est un type de graphique qui visualise les données dans un cercle. Chaque tranche du cercle représente une catégorie et est liée à la taille de cette catégorie par rapport au data-set. L'intégralité du cercle correspond à toutes les données qui composent le data-set. Les tranches du cercle sont alors proportionnelles aux fractions des catégories sur le data-set (Camm et al., 2021).

Les diagrammes circulaires permettent d'avoir un aperçu sur la manière dont un data-set est divisé en groupe de données. **Figure 3-4** montre un exemple d'un diagramme circulaire.

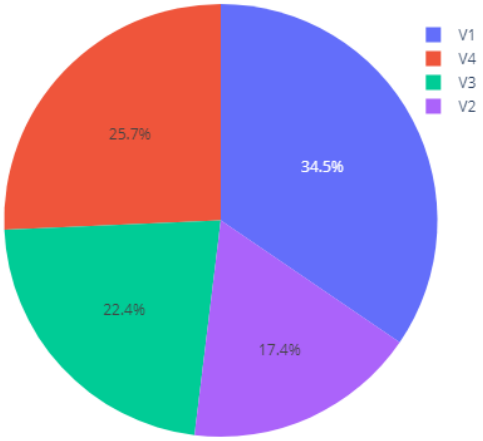

Figure 3-4 : Exemple d'un diagramme circulaire

4.3 Diagramme à barres

Un diagramme à barres (Bar charts ou Bar Graphs), très répandu en statistiques, est une représentation graphique utilisée pour visualiser en barres verticales ou horizontales les données qui sont généralement groupées. Il est alors dédié à visualiser les données catégoriques ou numériques qui sont arrangées dans des intervalles (Camm et al., 2021). **Figure 3-5** illustre un diagramme à barres vertical et un autre horizontal. La signification des diagrammes à barres est déterminée par la longueur de chacune de ces barres qui est liée à la mesure d'un ensemble de données. La largeur des barres, quant à elle, est généralement fixe. L'utilité principale de cette technique de visualisation est de faciliter la comparaison des groupes de données et de visualiser les changements des données selon des critères précis. Les variables catégoriques sont arrangées dans l'axe x et leurs valeurs correspondantes dans l'axe y.

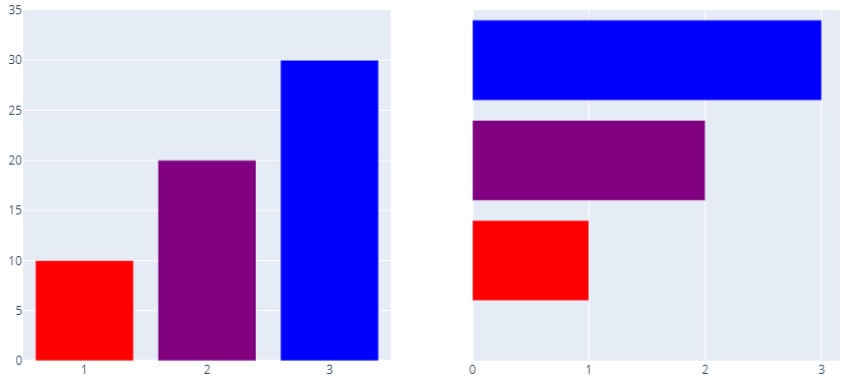

Figure 3-5 : Exemple d'un diagramme à barres vertical et horizontal

Deux dérivées de diagrammes à barres sont distinguées : groupées et empilées (Camm et al., 2021). Elles peuvent être utilisées verticalement ou horizontalement.

1- Un diagramme à barres groupées (clustered bar chart) est utilisé pour représenter les valeurs discrètes de plusieurs objets de la même catégorie.
2- Un diagramme à barres empilées (stacked bar chart), illsutré dans **Figure 3-6** est un diagramme qui divise les barres en différentes parts représentées chacune par une couleur différente et liées chacune à une catégorie.

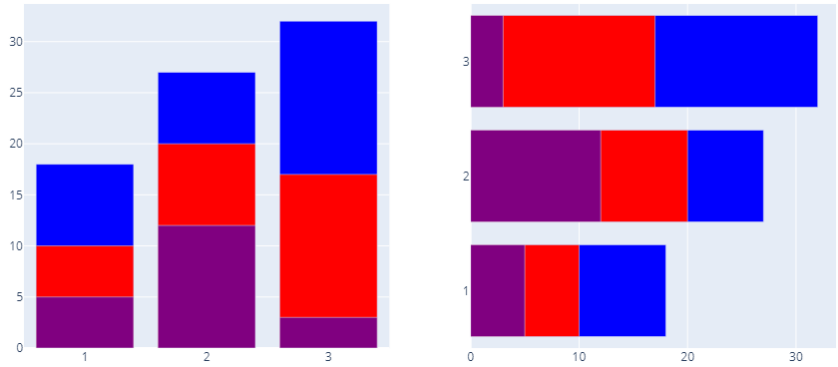

Figure 3-6 : Exemple d'un diagramme à barres empilées

4.4 Histogramme

Un histogramme est un diagramme à barres qui est utilisé pour visualiser les informations statistiques des données continues (numériques) (Camm et al., 2021). Comme montré dans **Figure 3-7**, toutes les barres de l'histogramme sont attachées. Toute barre dans l'histogramme indique le nombre d'observations qui se situent dans la plage des valeurs qui lui correspond.

Figure 3-7 : Exemple d'un histogramme

4.5 Nuage des points (Scatter plots)

Le nuage des points, aussi appelé le diagramme de dispersion, est un type de visualisation qui utilise des points graphiques pour représenter les valeurs de deux variables numériques différentes. Il permet d'observer la relation entre des couples de variables dans les data-sets, déterminer les patterns qu'il peut y avoir et détecter les données aberrantes et les lacunes dans ces data-sets. En pratique, les variables représentées sur l'axe x sont souvent indépendantes, et celles représentées sur l'axe y sont dépendantes. Cette analogie peut être utilisée pour faire des prédictions sur la distribution des données (Camm et al., 2021; Tominski & Schumann, 2020). **Figure 3-8** illustre un nuage de points. Les nuages des points peuvent également supporter la représentation d'une autre variable catégorique en plus des deux variables numériques. Ceci est souvent réalisé en attribuant à chaque catégorie de la variable catégorique une couleur ou une forme graphique. Si la troisième variable est numérique, la meilleure solution est d'utiliser un graphique à bulles pour visualiser toutes les variables.

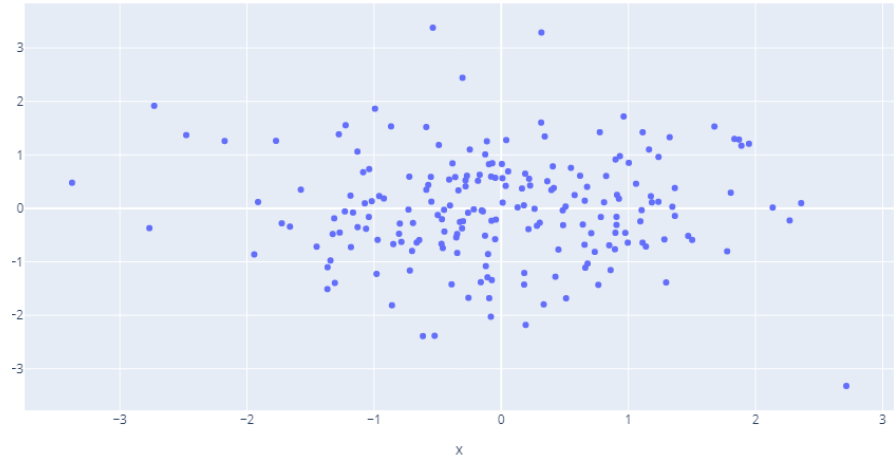

Figure 3-8 : Exemple d'un nuage de points (scatter plot)

4.6 Graphiques à bulles

Comme le montre **Figure 3-9**, les graphiques à bulles sont similaires aux nuages des points, saufs qu'ils sont redimensionnables (Camm et al., 2021). En effet, ils peuvent être utilisés pour visualiser plus de deux variables numériques en 2D à travers x, y, les marqueurs, le pointage et la taille des bulles.

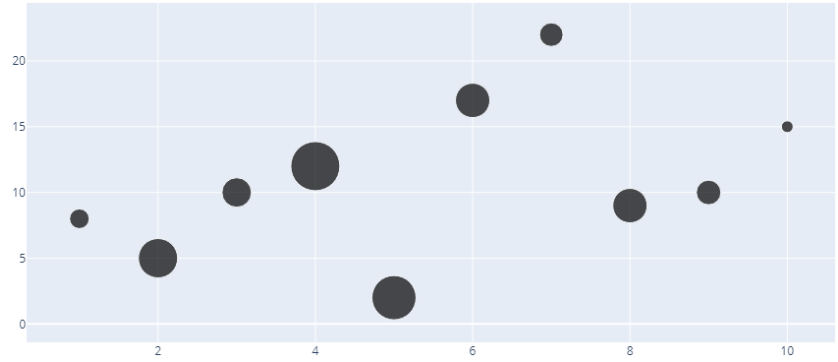

Figure 3-9 : Exemple d'un graphique à bulles

4.7 Graphiques linéaires (Line Charts)

Un graphique linéaire, montré dans **Figure 3-10**, visualise une série de points interconnectés par des segments linéaires. Ces points représentent les valeurs numériques de l'axe y qui changent selon des valeurs de l'axe x (Camm et al., 2021). Ces dernières, souvent de variables temporelles, sont représentées sur l'axe x par des intervalles réguliers. Quant à la variable de l'axe y, souvent représentant des résumés statistiques, elle concerne les valeurs qui correspondent aux intervalles de x.

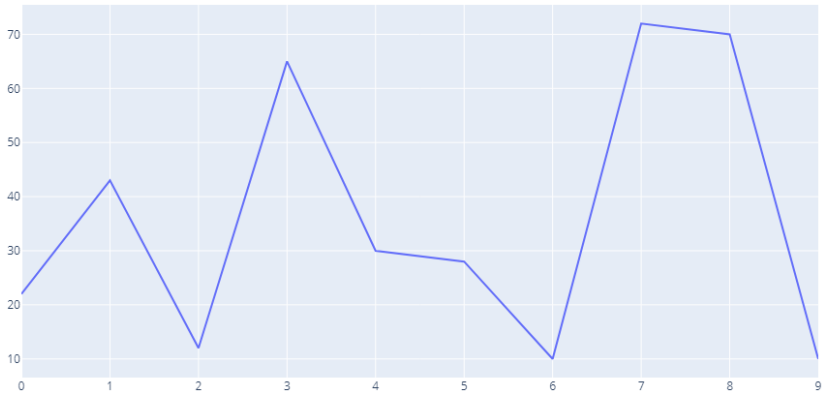

Figure 3-10 : Exemple d'un graphique linéaire

4.8 Boîtes à moustaches

Les boîtes à moustaches (box plots) sont un type de graphique utilisé pour l'analyse exploratoire des statistiques des data-sets. Comme le montre **Figure 3-11**, elles visualisent les distributions des données via leur moyenne et leurs quartiles et montrent cinq paramètres qui résument la distribution des données, à savoir : la valeur minimale, le quartile inférieur (Q1) sous lequel se trouvent approximativement 25% des observations, le quartile Q2 qui représente la médiane qui est approximativement au-dessus de 50% des observations, le quartile Q3 qui se trouve approximativement au-dessus de 75% des observations, la valeur maximale et la moustache (Camm et al., 2021; Montgomery & Runger, 2018). Chacun des quartiles peut ne pas être unique dans un data-set. Ces paramètres permettent de déterminer si les data-sets sont distribués selon la distribution normale ou pas. Ils permettent également d'identifier les valeurs aberrantes dans les data-sets. Les quartiles divisent chaque data-set en quatre segments dont chacun contient approximativement 25% des données qui existent dans le data-set.

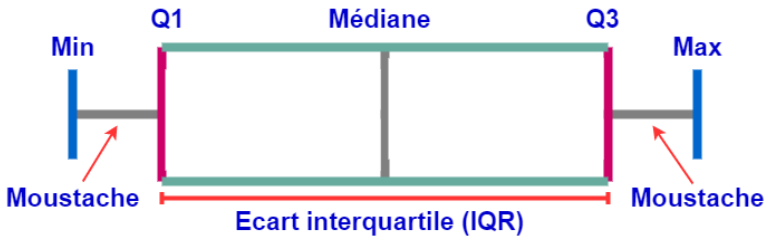

Figure 3-11 : Composants d'une boîte à moustaches (inspirée de (Montgomery & Runger, 2018))

IQR est une mesure de variabilité des data-sets. Chacun de ces quartiles peut ne pas être. Elle est calculé selon la formule suivante (Devore et al., 2021) :

$$IQR = q3 - q1$$

Cette mesure de propagation est résistante aux valeurs extrêmes qui peuvent se trouver dans un data-set (Devore et al., 2021; Montgomery & Runger, 2018).

4.9 Graphiques de distribution

Les graphiques de distribution (distribution plots ou distplots) sont utilisés pour évaluer la distribution des données numériques d'un échantillon par rapport aux valeurs théoriques d'une distribution précise. Ils permettent également d'effectuer des tests d'hypothèse pour déterminer si les données d'un data-set suivent une distribution spécifiée. Un graphique de distribution peut contenir des diagrammes à barres, des graphiques linéaires, des nuages de points, etc. (Camm et al., 2021; Tominski & Schumann, 2020).

4.10 Word Cloud

Word Cloud est une technique de visualisation des textes qui a pour objectif de présenter un nombre de mots dans un texte de manière unique avec une taille et couleur précises pour chaque mot conformément à un critère défini. Ces mots sont groupés ensemble pour former un nuage de mots. La taille de chaque mot détermine son importance dans le texte ; plus il est grand, plus il est important. Le nombre de mots à visualiser joue un rôle important pour la création des nuages de mots. Il doit être défini selon les spécificités de l'application et des textes à visualiser. Les critères de sélection des mots, eux aussi, varient selon le l'application. Un des critères les plus communs est la fréquence des mots dans le texte. Parmi les applications de Word Cloud on peut trouver la visualisation des sujets de tendance dans les pages web et la visualisation des mots fréquents dans les feedbacks des clients. **Figure 3-12** présente un exemple de nuage de mots.

Figure 3-12 : Exemple d'un Nuage à mots (Word Cloud)

5 Outils de visualisation

Il existe plusieurs outils de visualisation des données. Ils peuvent être distingués en librairies, plateformes et services (Agrawal et al., 2015). **Table 3-1** illustre ces trois catégories (Agrawal et al., 2015; Ali et al., 2016; Chawla et al., 2018).

Table 3-1 : Exemples de librairies, plateformes et services de visualisation

Librairies	Plateformes	Services
Matplotlib – Seaborn – Gephi – D3js – Plotly – Dash – Bokeh – TextHero – Igraph – GraphViz – NetworkX – Cytoscape – nodeXL	Tableau – Kibana – PowerBI	CartoDB – Dundas

6 Visualisation et analytique visuelle dans big data

Les données dans Big Data présentent de multiples défis vis-à-vis de la visualisation. Ces défis concernent basiquement le support des données caractérisés par le volume élevé, l'hétérogénéité (Fahad & Yahya, 2018) et la nécessité de les visualiser tout en considérant la satisfaction des contraintes basiques, d'interactivité, de mise en échelle et de structuration (Agrawal et al., 2015; Kahil et al., 2020). De nouveaux horizons sont apparus et font aujourd'hui l'objet de recherche. La visualisation des processus est illustrative de ces horizons ; elle sert à explorer les différentes tâches de traitement et d'analyse afin de les superviser et d'en avoir un aperçu. Parmi les applications de cette nouvelle présentation on trouve la visualisation des flux des données via des outils spécialisés tels que Apache

Ambari (Wadkar & Siddalingaiah, 2014). A travers cette visualisation les flux peuvent être suivis et, par conséquent, contrôlés. De même, l'analytique visuelle dans Big Data a reçu plus de considération que la simple visualisation des données. Ceci est la conséquence de la forte connexité entre elle et la prise de décision qui est basée sur l'analytique des données. Il existe plusieurs outils avancés qui assurent la visualisation et l'analytique visuelle en implémentant les différentes techniques existantes et en garantissant la mise en œuvre des solutions proposées. Ils prennent également en considération les problèmes et les défis relatifs à la visualisation et l'analytique visuelle dans Big Data, notamment l'échelonnabilité et l'interactivité. Parmi ces outils on peut trouver (Ali et al., 2016; Chawla et al., 2018; Fahad & Yahya, 2018) Spotfire, Tableau, IBM Cognos, MicroStrategy, Qlickview, Kibana, Pentaho reporting, Sap Busieness object et ZoomData.

6.1 Contraintes de visualisation et d'analytique visuelle dans le contexte Big Data

En effet, toute solution de visualisation ou d'analytique visuelle dans un contexte Big Data doit satisfaire, en plus des contraintes mentionnées auparavant, d'autre contraintes qui concernent (1) l'échelonnabilité et la (2) structuration. L'ensemble de toutes les contraintes à satisfaire est montré dans **Figure 3-13**.

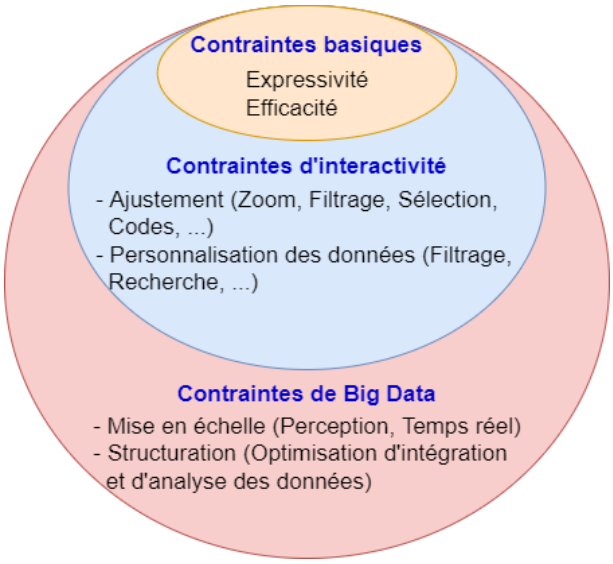

Figure 3-13 : Contraintes de visualisation interactive des données dans big data

6.1.1 Contraintes d'échelonnabilité

Les contraintes d'échelonnabilité reflètent la dynamique de visualisation qui est essentiellement traduite par deux axes, à savoir la mise en échelle avec la perception et la mise en échelle en temps réel (Agrawal et al., 2015; Kahil et al., 2019). Le premier axe implique la capacité des solutions de visualisation à s'adapter aussi bien avec les nouvelles entrées des données qu'avec la croissance du nombre d'utilisateurs. L'aspect du temps réel consiste à visualiser les données en respectant une latence précise. En effet, la mise en échelle est une propriété qui présente encore un défi à considérer dans Big Data. Elle est liée à la capacité des solutions à supporter les données volumineuses et hétérogènes et à offrir leur visualisation à un nombre important d'utilisateurs en temps réel.

6.1.2 Contraintes de structuration

Les contraintes de structuration traduisent l'ensemble des fonctionnalités qui ont pour objectif de faciliter l'opérabilité des différentes étapes de visualisation y compris la simplification de l'intégration et de l'analyse des données (Kahil et al., 2020; Xu et al., 2015). L'enjeu qu'imposent ces contraintes est

essentiellement lié à l'optimisation de ces tâches. Pour les satisfaire, il faut employer les différentes méthodes et techniques d'optimisation dans ces étapes.

6.2 Stratégie de visualisation et analytique visuelle dans Big Data

Afin de proposer des solutions de visualisation et d'analytique visuelle dans un contexte Big Data, différentes voies peuvent être empruntées. Parmi lesquelles il peut y avoir deux approches : contextuelle et applicative. Dans l'approche contextuelle, on peut distinguer en matière de nature la visualisation statique et la visualisation dynamique. La dynamique ici fait référence à la manière dont les données sont acquises et présentées. La visualisation peut également être distinguée, comme est mentionné ci-dessus, selon cette approche en exploratoire et analytique.

Table 3-2 : Exemples d'application de la visualisation selon l'objectif (Kahil et al., 2020)

Objectif	Exemples
Visualisation d'un sujet (topic)	Trafic d'un réseau Secteur de santé Villes intelligentes Education Astronomie Marketing
Visualiser un type de données	Vidéo Audio Image Texte Données temporelles Données catégoriques Données de streaming
Visualiser un data-set	Données médicales Données des réseaux sociaux Données des bourses

Dans l'approche applicative, l'intérêt est de déterminer quoi visualiser conformément à un objectif fixé. A cet égard, il peut y avoir (1) la visualisation d'un type de données, (2) la visualisation d'un data-set ou (3) la visualisation d'un domaine spécifique (Kahil et al., 2019; J. Zhang et al., 2014). La visualisation d'un ou plusieurs types de données, tels que les textes, les vidéos et les images, consiste à choisir les techniques et outils qui supportent proprement ces types de données. La visualisation d'un data-set repose sur la considération de la multidimensionnalité qui peut caractériser les data-sets, l'optimisation du processus de présentation vis-à-vis des types de données que peut avoir le data-set et les techniques d'analyse et de visualisation à employer. La visualisation des domaines spécifiques soulève, en plus des différentes dimensions de la visualisation des data-sets, d'autres enjeux, notamment la dynamique des données, l'intégration de la visualisation dans l'écosystème du domaine précis et la mise en échelle avec le volume et les types des données.

Table *3-2* illustre chacun des trois volets de la dimension applicative de visualisation.

6.3 Multidimensionnalité des données dans Big Data

Pour visualiser un data-set ou un domaine spécifique, la question de multidimensionnalité des données est toujours présente aussi bien dans la visualisation que dans l'analytique visuelle. Le nombre de dimensions à visualiser à tout instant est restreint, ce qui rend impossible de visualiser simultanément l'intégralité des données (Xyntarakis & Antoniou, 2019). Des méthodes sont alors à suivre pour remédier à ce problème.

6.3.1 Méthodes pour considérer les données multidimensionnelles

Il existe de multiples méthodes qui aident à supporter la visualisation des données multidimensionnelles. Toute solution de visualisation dans un contexte Big Data peut adopter la totalité de ces méthodes à la fois ou en sélectionner les pertinentes, selon les spécificités du problème et l'objectif désigné. Puisque les systèmes basés sur les réponses exactes nécessitent parfois considérablement de temps, ces méthodes sont souvent basées sur l'approximation des résultats (Godfrey et al., 2016). Elles peuvent se résumer dans les axes cités et décrits ci-dessous (Bikakis, 2018; Kahil et al., 2020).

Réduction de dimensionnalité

La réduction des dimensions est une manière d'optimiser l'espace de données à visualiser. Elle consiste à limiter ces dernières en en sélectionnant les plus consistantes selon des critères définis. Parmi les techniques utilisées pour réduire la dimensionnalité des données il y a l'échantillonnage, le clustering et les algorithmes d'extraction des variables cités dans chapitre **2** tels que PCA, SVD, etc.

Traitement incrémental et adaptatif

Le traitement incrémental et adaptatif est une méthode très pratique dans l'exploration des données, notamment dans les cas où ces dernières sont dynamiques (Ali et al., 2016; Bikakis, 2018; Dietrich et al., 2015). Cette méthode consiste à fournir à l'utilisateur l'accessibilité à seulement des fragments des données plutôt que leur intégralité sans avoir recours à un pré-traitement. Ensuite, en réponse à l'interaction de l'utilisateur via des techniques d'exploration telles que l'exploration à la volée et le détail à la demande, d'autres fragments de données sont traités et lui sont présentés (Dimara & Perin, 2020).

Présentation progressive des résultats

Comme le processus du traitement et d'analyse des données volumineuses et hétérogènes prend du temps pour être effectué, il affecte négativement la visualisation exploratoire des données qui est contrainte de la latence (temps réel). C'est pourquoi il est nécessaire d'adopter la méthode des résultats progressifs (Dimara & Perin, 2020). Cette dernière consiste à présenter dans la limite de la latence des résultats partiels aux requêtes de l'utilisateur et les raffiner au fil du temps jusqu'à ce l'utilisateur interrompe l'opération de visualisation ou qu'il décide d'en définir une autre.

Visualisation hiérarchique

La visualisation hiérarchique permet de présenter les données d'une manière structurée selon multiples niveaux (Chawla et al., 2018; Kahil et al., 2021a; Raghav et al., 2016). Cette structuration permet à l'utilisateur d'avoir un aperçu sur les données et lui donne une intuition sur leurs différentes parts. De là il procède à l'exploration via les techniques de personnalisation telles que la sélection, le zoom, le filtrage, etc. selon le principe du détail à la demande (Bikakis, 2018; Raghav et al., 2016). Les différents niveaux de hiérarchie sont construits sur la base des méthodes de partitionnement et de clustering des graphes.

Recommandation et assistance de l'utilisateur

Afin de faciliter à l'utilisateur le processus d'exploration des données visualisées, il est pratique de lui présenter ces dernières d'une manière qui lui soit familière et intéressante aussi bien en termes de techniques de visualisation qu'en matière des données présentées. Cela lui simplifie l'exploration des données volumineuses. Le concept de recommandation peut servir à cette fin. En effet, cette dernière est largement considérée dans différentes applications telles que les plateformes d'e-commerce et des films. Elle consiste à sélectionner les données les plus adéquates à chaque profil utilisateur pour les présenter (Davenport, 2014; Dietrich et al., 2015; Kahil et al., In press; Wu et al., 2021). Elle peut assister l'utilisateur dans le processus d'exploration et d'analyse via la sélection des données qui peuvent

l'intéresser selon son comportement d'une part, et de les présenter via les techniques convenables selon leurs types, leurs attributs et les tâches requises d'une autre part.

Mise en cache

La visualisation exploratoire est généralement réalisée via une séquence d'opérations liées les unes aux autres. A cet égard, mettre en cache les données qui sont les plus probables à être explorées par l'utilisateur dans les étapes prochaines permet de réduire le temps de réponse. La mise en cache des données (Caching and Prefetching) aide alors à l'accélération du processus de visualisation et à assurer la mise en échelle (Bikakis, 2018; Schintler & McNeely, 2022). Elle est souvent réalisée via le couplage des techniques de prédiction avec les techniques de supervision du comportement utilisateur et de profilage.

6.4 Méthodes de visualisation et d'analytique visuelle avancées

6.4.1 Coordonnées parallèles

Les coordonnées parallèles (Parallel Corrdinates) sont une technique de visualisation des données multidimensionnelles présentées dans des tables. Cette technique consiste à mapper chaque élément, présenté par une ligne de la table, avec une ligne graphique appelée profil et chaque attribut de cet élément avec un point du profil. Bien que le graphique de cette visualisation soit similaire aux graphiques linéaires comme le montre **Figure 3-14** (Tominski & Schumann, 2020), son interprétation des données est complètement différente. En effet, les valeurs des coordonnées parallèles sont normalisées selon les catégories correspondantes : l'échelle de chaque colonne est totalement différente de celle des autres colonnes (Xyntarakis & Antoniou, 2019). Pour chaque catégorie dans l'axe x, la plus petite valeur correspondante est affectée à 0%, et la plus grande valeur correspondante est affectée à 100%. Les courbes des différentes colonnes ne sont donc pas comparables selon la hauteur de chacune.

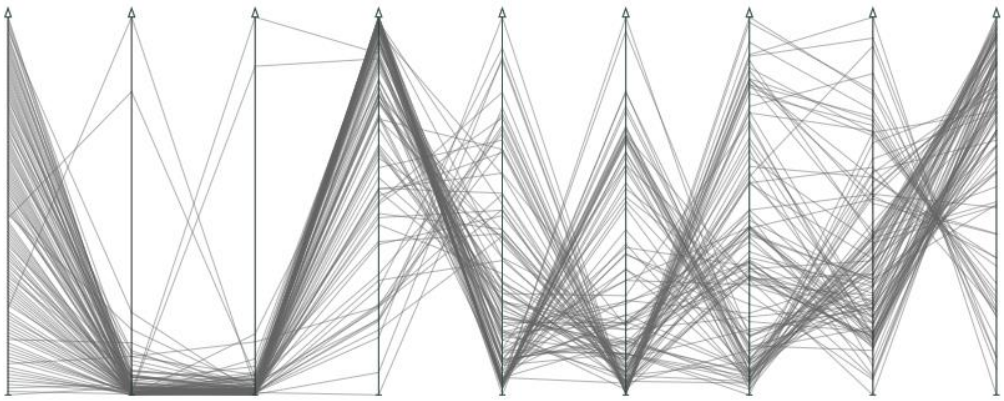

Figure 3-14 : Exemple de visualisation via coodonnées parallèles (Tominski & Schumann, 2020)

6.4.2 Agrégation en corbeilles

L'agrégation en corbeilles (Binned aggregation) est une méthode qui est utilisée pour prétraiter les données et les préparer à la visualisation structurée (Qunchao Fu et al., 2014). Elle consiste à agréger les données dans des corbeilles afin que celles qui ont des propriétés communes soient mises dans la même corbeille et que les plus denses dans toutes les corbeilles soient visualisées en premier. Après ça, la visualisation peut être personnalisée via des fonctionnalités telles que le filtrage afin de présenter les données arrangées dans les corbeilles qui ne sont pas encore visualisées.

6.4.3 Navigation à facettes

La navigation à facettes (Faceted browsing) est une méthode qui facilite l'exploration des données complexes en les présentant en parts et en fournissant à l'utilisateur de multiples filtres, appelés facettes,

à travers lesquels il peut s'interagir avec la visualisation et la personnaliser selon ses orientations vis-à-vis de ces données (Tominski & Schumann, 2020). Les filtres employés peuvent être hétérogènes au sein de la même application. La navigation à facettes est largement adoptée dans différentes applications, notamment l'e-commerce, les plateformes des films et des livres.

7 Conclusion

Ce chapitre a présenté les concepts généraux de visualisation des données et d'analytique visuelle, les conventions qui leur sont liées et les contraintes à satisfaire sur la base desquelles toute solution de visualisation est évaluée et validée. Les techniques de visualisation ainsi que ses types ont également été décrits. Après quoi, la visualisation des données et l'analytique visuelle ont été projetées sur le contexte Big Data tout en mettant en évidence d'autres contraintes à satisfaire dans ce même contexte, à savoir la mise en échelle et la structuration. Ensuite, le problème de multidimensionnalité a été abordé avec des solutions proposées, à savoir la réduction de dimensions des données, le traitement incrémental et adaptatif, la présentation progressive des résultats, la présentation hiérarchique des données, la recommandation et l'assistance de l'utilisateur et la mise en cache des résultats. Ces solutions représentent les axes essentiels à considérer lors d'envisager la résolution des problèmes de visualisation interactive des données dans le contexte Big Data. En effet, c'est à la base des spécificités du problème en question que le chemin à prendre est déterminé, à savoir visualisation ou analytique visuelle de données. Les contributions décrites dans les chapitres suivants visent chacune à résoudre les problèmes qui surgissent dans l'un de ces deux chemins.

Partie II: Contributions

Chapitre 4: Préparation des data-sets à la visualisation multidimensionnelle en utilisant une heuristique gloutonne

1 Introduction

Malgré la multitude des techniques de visualisation des données telles que les nuages de points, les coordonnées parallèles, les cartes et les diagrammes à barres (Ali et al., 2016; Gorodov & Gubarev, 2013; M. Khan & Khan, 2015; Ward et al., 2015), leur application directe sur les grands data-sets est toujours une tâche préoccupante. Ces data-sets contiennent des patterns si nombreux qu'il est impossible de les visualiser tous simultanément. En plus, la visualisation des grands data-sets est soumise à de multiples contraintes, notamment la mise en échelle qui reflète l'interactivité, la réceptivité et le temps réel (Agrawal et al., 2015; Kahil et al., 2019). Une solution consiste à visualiser partiellement les patterns de manière à ce que les plus pertinents soient visualisés en premier tout en fournissant des vues structurées et personnalisables qui pourraient être hiérarchiques dans certains cas. Cela pose le problème de la sélection des patterns de la première vue, car la sélection aléatoire peut ne pas convenir à la visualisation contrairement à ce qu'elle peut l'être dans d'autres cas tels que la recherche (Rama Satish & Kavya, 2019). Les algorithmes d'exploration de données sont bien connus pour être utilisés dans ce but. Parmi eux figurent l'arbre de décision (DT), la forêt aléatoire (RF), l'exploration de règles d'association (ARM : Association Rule Mining), KNN, les SVMs KMC (Cunningham & Delany, 2020; Ziegler & König, 2014). DT utilise la structure arborescente pour présenter les données dans les nœuds. Ainsi, les patterns sont représentés sous la forme d'une série de décisions à chaque niveau de nœuds, nommés chacun points de décision, se terminant par des nœuds feuilles. RF peut être définie comme un ensemble de DT. Elle vise à diviser l'ensemble de données en échantillons non corrélés, à associer chaque échantillon à un DT pour créer son classificateur et à faire des prédictions de classe basées sur le vote afin de trouver la meilleure prédiction. ARM consiste à associer des règles sur des modèles après avoir généré de grandes quantités d'éléments fréquents afin de les visualiser éventuellement, ce qui entraîne une difficulté de processus en raison de la grande complexité résultante. KNN vise à classer chaque pattern en fonction de ses voisins les plus proches. Il utilise la distance euclidienne pour calculer la distance entre eux. Malgré son utilité dans la classification en termes de précision, il est sensiblement lent avec des données volumineuses. Les SVM visent à générer des modèles linéaires et non linéaires afin de fournir une classification binaire des modèles en utilisant deux classes de données étiquetées. Ils présentent une grande efficacité pour la classification de texte. Cependant, comme DT, ils prennent beaucoup de temps pour l'entrainement sur les grands ensembles de données. KMC, qui appartient aux algorithmes d'apprentissage non supervisé, consiste à diviser les collections de données en clusters de sorte que chaque cluster contienne les enregistrements les plus similaires. En ce qui concerne Big Data, son principal avantage est qu'il s'adapte à de grands ensembles de données. Cependant, cet algorithme est dédié à la manipulation de données numériques, ce qui rend difficile son application directe à d'autres types de données. De plus, il est difficile de spécifier le nombre approprié de clusters k, en particulier dans le cas de données dynamiques. À partir de cette brève description des algorithmes de fouille de données, on peut remarquer que le problème commun qu'ils présentent est la grande complexité, en particulier lorsque l'on considère des données massives. Le problème de visualisation des data-sets pourrait toutefois être résolu dans plusieurs cas sans avoir recours à des solutions aussi complexes que ces algorithmes. Par conséquent, une méthode plus abordable serait envisageable à cette fin. Nous proposons donc, à travers cette contribution (Kahil et al., 2021a), une approche gloutonne qui 'résout les problèmes indiqués. En d'autres termes, il s'agit de préparer les grand data-sets avec une faible complexité à la visualisation multidimensionnelle, tout en priorisant la visualisation hiérarchique, en gérant les patterns à visualiser et en tenant compte des contraintes de la visualisation mentionnées ci-dessous.

Le reste de ce chapitre est organisé comme suit : Dans la deuxième section, un ensemble de travaux liés à la visualisation des data-sets volumineux est présenté. La troisième section liste les principaux points que recouvre l'approche proposée et présente ses différentes composantes sous forme d'architecture et

d'algorithmes. La quatrième section présente une expérimentation sur un data-set, discute ses résultats et compare ces derniers avec les travaux existants. La cinquième section conclut le chapitre avec des perspectives concernant certaines pistes liées à ce travail.

2 Travaux connexes de la visualisation des data-sets volumineux

Plusieurs travaux ont été proposés pour répondre aux problèmes de visualisation du Big Data. Comme nous l'avons mentionné (Kahil et al., 2020), chaque travail de visualisation de Big Data vise soit à (1) fournir des fonctionnalités permettant d'améliorer l'interaction avec l'utilisateur, (2) mettre en œuvre des techniques de visualisation de données existantes dans les domaines du Big Data, (3) proposer des approches de visualisation, des modèles et des outils à des domaines spécifiques et/ou (4) accélérer le processus de visualisation des données. Selon ces objectifs, les travaux connexes sont cités ci-dessous.

(Sansen et al., 2017) ont développé un système évolutif basé sur des coordonnées parallèles pour effectuer une exploration visuelle interactive de grands enregistrements de données basés sur HDFS et Hbase pour le stockage, avec Apache Spark et Elasticsearch pour le traitement en temps réel. Pour la préparation des données, ils ont utilisé un algorithme d'apprentissage non supervisé nommé Canopy clustering. L'interactivité est assurée en fournissant à l'utilisateur des options de zoom et de l'affichage panoramique. Un autre système nommé DeepEye [23], basé sur l'algorithme d'apprentissage du classement appelé LambdaMART, a été développé par (Qin et al., 2018) pour sélectionner les techniques de visualisation appropriées en fonction des ensembles de données donnés. Il utilise les évaluations des utilisateurs sur chaque graphique de visualisation et les classifie toutes à l'aide d'un arbre de décision afin de recommander des visualisations appropriées aux autres utilisateurs. (Golfarelli & Rizzi, 2019) ont développé un modèle nommé SkyViz basé sur l'analytique pour fournir différentes formes de visualisation en fonction des préférences des utilisateurs. Il consiste à définir, en fonction des objectifs de chaque utilisateur, le contexte de visualisation à travers des ensembles de coordonnées, où chacune représente un objectif tel que l'utilisateur cible, l'interaction et les cardinalités des données. À partir de ces ensembles, les visualisations appropriées sont déterminées à l'aide de la méthode *skyline* et du framework *toreador*. (Soylu et al., 2013) ont proposé une interface utilisateur appelée OptiqueVQS pour agréger plusieurs applications dans un espace graphique commun multi-widgets. Selon les objectifs de visualisation, chaque widget agrège les applications qui ont un objectif commun. Sa stratégie consiste à combiner différents paradigmes de représentation assurés par les widgets via des ontologies afin de formuler et visualiser les requêtes à l'aide de SPARQL. (Wilkinson, 2018) a proposé un algorithme basé sur un modèle distribué nommé Hdoutliers pour visualiser les valeurs aberrantes du Big Data. Il consiste à paralléliser d'abord l'agrégation des données structurées et non structurées afin d'extraire les valeurs aberrantes ; ensuite, il utilise différentes techniques de visualisation selon les différents cas désignés, à savoir les valeurs aberrantes unidimensionnelles, de faible dimension et de grande dimension. Ce dernier cas, qui présente des problèmes importants lors de l'utilisation de techniques de visualisation inappropriées telles que les histogrammes et les diagrammes de points, est traité à l'aide de différentes techniques statistiques selon les spécificités de l'application. Parmi ces techniques on trouve l'analyse résiduelle dans les modèles de régression, les coordonnées parallèles et les valeurs aberrantes du graphique. (Simonini & Zhu, 2015) ont proposé une approche basée sur la navigation à facettes pour fournir une visualisation interactive efficace dans le contexte Big Data. Cette méthode utilise des filtres dynamiques pour personnaliser la visualisation en temps réel. A cette fin, ils ont utilisé un réseau bayésien afin de représenter les probabilités des enregistrements de données et sélectionner les plus élevées pour les visualiser à l'aide d'*OpenMarkov*. (Dash et al., 2008) ont développé un système basé sur la recherche à facettes dynamique pour visualiser les données structurées et les données textuelles. Adoptant Solr et la représentation bitmap via l'agorithme de compression WAH (Word Aligned Hybrid), ce système sélectionne parmi l'ensemble des attributs ceux les plus importants. L'importance est

déterminée en mesurant l'intérêt des utilisateurs, un concept lié aux entrepôts de données de traitement analytique en ligne (OLAP : OnLine Analytical Processing). (M. L. Huang et al., 2015) ont développé une nouvelle méthode nommée Arc Coordinates Plot (ACP) qui étend les coordonnées parallèles en utilisant les axes géométriques en arc basés en se basant sur SVD pour effectuer l'analytique visuelle des données de grande dimension. (Qunchao Fu et al., 2014) ont utilisé la méthode de *Binned Aggregation* pour prétraiter les données avant la visualisation. Cette méthode consiste à agréger les données dans des corbeilles afin que celles qui ont des propriétés communes soient mises dans la même corbeille et que les plus denses dans toutes les corbeilles soient visualisées en premier. (Im et al., 2013) ont proposé une approche nommée VisReduce pour visualiser de grands ensembles de données. Elle est basée sur le paradigme MapReduce et sert à effectuer une visualisation distribuée incrémentielle avec différentes techniques telles que les cartes thermiques et les coordonnées parallèles. Elle offre la mise à l'échelle via l'ajout des nœuds qui sont gérés par Hadoop MapReduce afin d'accélérer l'exécution des requêtes, de sorte que chaque nœud génère de petits agrégats pour réduire le temps de production et de transmission.

3 Approche proposée (GreedyBigVis) pour la préparation des data-sets volumineux à la visualisation interactive

L'approche proposée, nommée GreedyBigVis (Kahil et al., 2021a), vise à préparer les grands data-sets ayant des schémas hétérogènes, libres de schémas et pouvant contenir plusieurs types de données pour une visualisation à faible latence. Plus précisément, cette approche est composée de (1) algorithmes qui effectuent des opérations sur des ensembles en se basant sur la théorie des ensembles et (2) une architecture qui organise les séquences de tâches liées à ces opérations. Cette approche doit satisfaire les différentes contraintes de visualisation expliquées dans ce qui suit, notamment la mise en échelle qui englobe la perception, l'interactivité et le temps réel.

En revanche, comme il est impossible dans certains cas de visualiser simultanément les patterns très nombreux, il est convenable de les présenter en plusieurs vues, en commençant par les plus pertinents pour les mettre en première vue et en utilisant des techniques de détail à la demande (Bikakis, 2018; Harley, 2015) pour passer aux vues suivantes. La présentation hiérarchique des données, qui donne une visualisation structurée, est dédiée à cet objectif. Cependant, elle n'est pas toujours possible. C'est pourquoi un mécanisme pour vérifier la possibilité d'une visualisation hiérarchique est à proposer. De même, la pertinence des modèles est déterminée en fonction de différents facteurs selon l'application tels que la source des données, leur fraîcheur, le nombre de fois d'accès à ces données et différentes informations statistiques. Les relations cachées doivent également être prises en compte lors du traitement des aspects ci-dessus. Ils font partie de l'extraction de patterns cachés dans le processus de KDD (M. Chen et al., 2009, 2014; Patel & Jain, 2019). Leur considération permet de mieux définir les catégories et les niveaux des données et, par conséquent, d'offrir des visualisations efficientes. À partir data-set, elle extrait et gère les patterns de visualisation, les critères utilisés pour organiser cette dernière et les filtres utilisés pour permettre à chaque utilisateur de la personnaliser sur la base de ses intérêts.

3.1 Explication détaillée de l'approche

Figure 4-1 présente l'architecture de l'approche proposée qui est composée de cinq tâches. Chacune de ces dernières est expliquée en détail ci-dessous. Comme les données d'entrée, définies via la formule : $dataset = \{row_i, i \in [1, n]\}$, sont souvent semi-structurées ou non-structurées, la première tâche consiste à les prétraiter via le nettoyage et la sérialisation. Après cela, elles peuvent être stockées et chargées via le stockage approprié tel que HDFS et les bases de données NoSQL. Le comportement cyclique de ces tâches garantit la mise en échelle perceptuelle. Après le prétraitement, la deuxième tâche consiste à extraire les critères qui définissent les caractéristiques liées au data-set. Ces critères peuvent

être de statistiques, de relations d'inclusion et de types de données. Ils sont classifiés en critères catégoriels et critères hiérarchiques. Le premier type est utilisé pour construire les catégories des données à partir du data-set. Les critères de ce type peuvent être détectés via la formule suivante :

$$\exists row_i, row_j \in dataset, i \neq j, \exists da_{ik}, da_{jl}, k \neq l : da_{ik} = da_{jl} \quad (4.1)$$

tel que : $da_{ik} \in row_i$ et $da_{jl} \in row_j$ sont des attributs qui existent dans ces lignes, $k \in [1, |row_i|]$ et $l \in [1, |row_j|]$. Cette formule signifie que s'il y a deux lignes qui ont des attributs communs, elles acceptent une relation catégorielle. Les critères hiérarchiques sont utilisés pour construire les niveaux de hiérarchie de visualisation. Ils sont détectés via la formule suivante :

$$\exists row_i, row_j \in dataset, i \neq j, \exists da_{ik}, da_{il}, da_{jm}, da_{jn}, k \neq l : da_{ik} = da_{jm}, da_{il} = da_{jn} \quad (4.2)$$

tel que : $da_{ik}, da_{il} \in row_i$ et $da_{jm}, da_{jn} \in row_j$ sont des attributs qui existent dans les lignes row_i et row_j, $k, l \in [1, |row_i|]$ et $m, n \in [1, |row_i|]$. Cette formule signifie que s'il y a des lignes qui ont au moins deux attributs en commun, elles acceptent une relation hiérarchique.

Après l'étape de détection, des critères peuvent être générés en utilisant différentes stratégies. Pour les critères catégoriels, les plus courants sont les types de données. L'extension de leur ensemble revient à définir des propriétés statistiques en fonction du domaine d'application, telles que les sujets les plus consultés, les données les plus pertinentes, les données récentes et le nombre de sous-sujets. Pour les critères hiérarchiques, une matrice carrée ($nbr\_colonnes \times nbr\_colonnes$), nommée matrice d'interaction des colonnes, est proposée. Comme l'indique avec la formule (4.2), sa diagonale est remplie de zéros afin d'éliminer les couples de colonnes identiques dans l'ensemble des critères. Alors que pour les autres interactions, elles prennent chacune la valeur minimale entre chaque couple de colonnes. La formule suivante définit la stratégie du remplissage :

$$M(i,j) = \begin{cases} 0 & si\ i = j \\ \min(v_i, v_j) & sinon \end{cases} \quad (4.3)$$

tel que i et j sont les colonne d'interaction et v_i et v_j son leur valeur correspondantes respectivement.

Sachant que les catégories et les niveaux sont des sous-ensembles du data-set principal, il est à souligner que s'il existe des lignes communes entre les niveaux et les catégories, elles doivent être éliminées des catégories définies afin d'éviter de visualiser certains patterns plus d'une fois et de prioriser la visualisation hiérarchique.

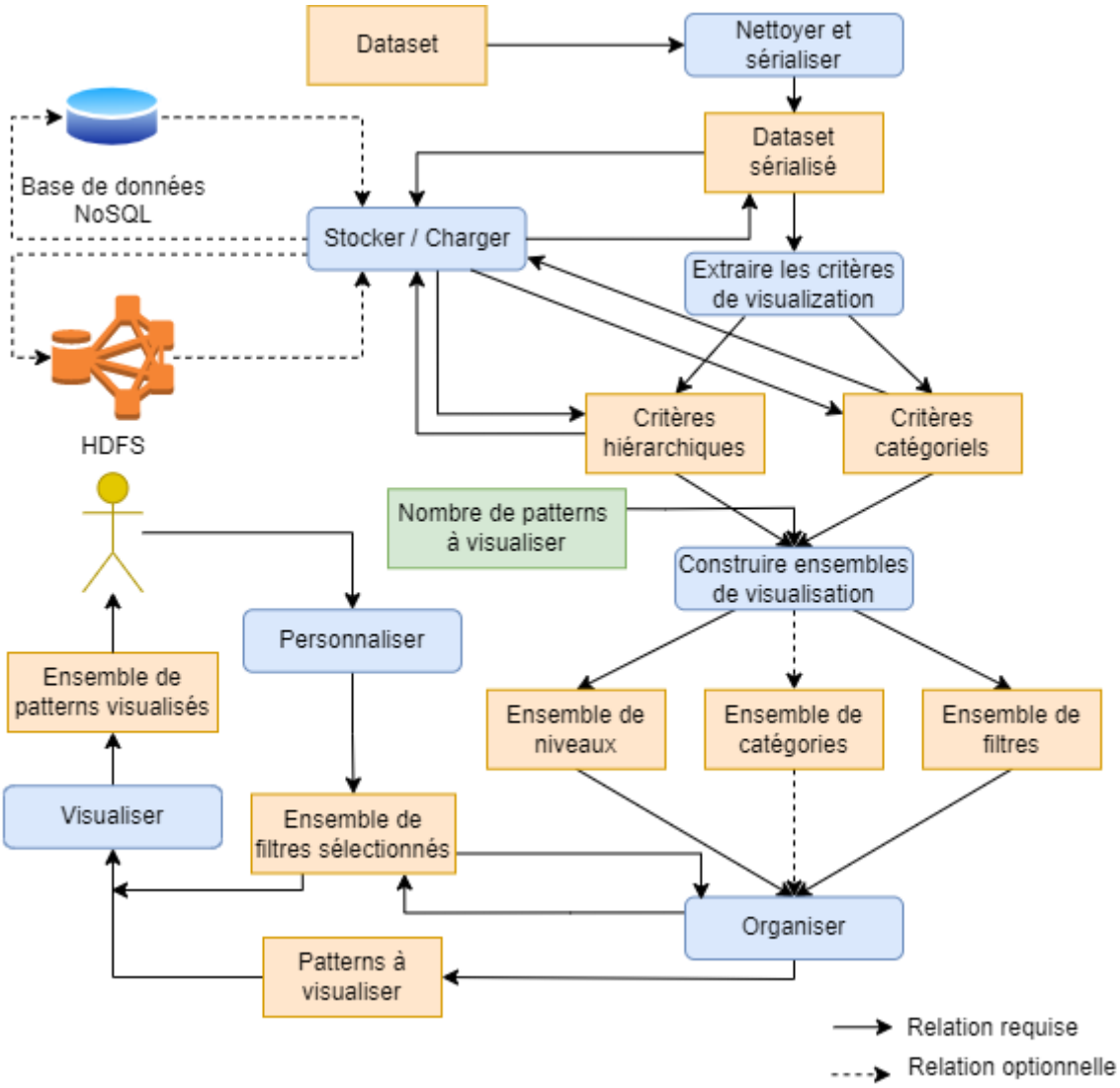

Figure 4-1: Architecture de GreedyBigVis (Kahil et al., 2021a)

La troisième tâche consiste à construire l'ensemble des patterns à visualiser en utilisant les critères extraits dans la deuxième tâche et le nombre autorisé de patterns à visualiser. Ce dernier doit être spécifié soit en fonction des caractéristiques du support visuel soit en tenant compte des conventions du domaine cible. A cet égard, l'approche proposée consiste à utiliser l'heuristique gloutonne pour sélectionner les patterns et les critères en fonction de ce nombre. En effet, cette heuristique a pour but de construire les solutions de manière séquentielle. **Algorithme 4-1** (Bednorz, 2008) présente cette heuristique. Son principe est de construire une solution en partant d'une solution incomplète et, à chaque étape, de faire un choix glouton qui doit être celui localement optimal selon un critère, appelé le critère glouton, jusqu'à ce que la solution soit complète. La raison pour laquelle est utilisée cette heuristique est, outre sa simplicité, sa faible complexité $n \times log(n)$ pour un data-set de n enregistrements par rapport au travaux connexes cités dont la complexité est plus élevée.

Algorithme 4-1 : Heuristique gloutonne (inspiré de (Bednorz, 2008))

Entrée : ds (Dataset)
Sortie : sol (Solution)

Début
  $Sol = \emptyset$
  Tant que sol n'est pas complète et $ds \neq \emptyset$ **faire**
    $x \leftarrow$ Sélectionner un élément de ds selon le critère glouton
    Si l'ajout de x est possible **alors**
      $sol \leftarrow sol \cup \{x\}$
    Fin Si
    $ds \leftarrow ds \backslash \{x\}$
  Fin Tant que
  Renvoyer sol
Fin

Afin d'assimiler la troisième tâche de l'approche proposée à l'heuristique gloutonne, elle est divisée en deux sous-tâches : la première consiste à trier les différents ensembles (l'ensemble des patterns et les ensembles de critères) et la seconde effectue les différentes opérations d'organisation sur ces ensembles. Trier les patterns et les ensembles de critères est l'élément central de la stratégie gloutonne proposée : lorsque ces ensembles sont triés en fonction d'un facteur pertinent, le choix des critères, des filtres et des patterns à visualiser devient évident : les premiers éléments de chaque ensemble sont les plus pertinents. Cela réduit le domaine de recherche (Rama Satish & Kavya, 2019) et donc la latence. La seconde sous-tâche sélectionne les patterns, les filtres et les critères à partir de leurs ensembles. Cette sous-tâche peut être formulée comme le problème du rendu de monnaie qui peut être résolu à l'aide de l'algorithme glouton : à chaque étape, la plus grosse pièce qui fait rapprocher de la solution est choisie. Cependant, il existe de nombreuses différences entre les solutions des deux problèmes. Le problème de rendu de monnaie consiste à trouver une solution avec le nombre minimum de pièces. Pour cela, l'heuristique gloutonne recherche les pièces les plus valorisées pour construire la solution. Cependant, en prenant un exemple simple d'un ensemble de pièces S = {0.10, 0.20, 5.50, 10, 20}, on suppose qu'il y a suffisamment de pièces pour chaque catégorie de monnaie. Si nous recherchons le nombre minimum de pièces pour obtenir 16,50 en utilisant l'algorithme glouton, nous obtenons : $1 \times (10) + 1 \times (5{,}5) + 5 \times (0{,}2)$ (7 pièces). Alors que le résultat optimal est $3 \times (5{,}50)$ (3 pièces), qui ne peut pas être trouvé via cet algorithme. **Table 4-1** résume ces différences. Un algorithme générique (**Algorithme 4-2**) est proposé pour construire l'ensemble des patterns à visualiser à partir des niveaux et des catégories. Pour chaque cas, il considère l'ensemble correspondant et le nombre défini de motifs à visualiser (pnl). Pour cela, il trie l'ensemble donné selon le critère glouton spécifique, sélectionne dans chaque niveau ou catégorie le premier pattern et l'ajoute à l'ensemble des patterns à visualiser tant que le nombre de ces derniers ne dépasse pas pnl. Dans le cas des niveaux, si le nombre de patterns est inférieur à pnl, le reste de ce dernier ($pnl - |ptv|$) peut être ajouté au nombre de patterns de catégories à visualiser dans le cas de la visualisation hiérarchique partielle, ce qui est traité par **Algorithme 4-3**.

Table 4-1 : Comparaison entre le problème de rendu de monnaie et le problème de définition de la séquence des patterns à visualiser (Kahil et al., 2021a)

Problème	Rendu de monnaie	Définition de la séquence des patterns
Objectif	- Un nombre minimum de pièces - Trouver les pièces qui mènent au montant désiré	- Un nombre spécifique de patterns - Trouver les patterns pertinents
Stratégie	- Choisir les pièces les plus valorisées qui approchent du montant spécifié	- Choisir les patterns les plus pertinents jusqu'à atteindre le nombre spécifié
Optimalité	- Pas optimal	- Probablement optimal

Algorithme 4-2 : Construction de l'ensemble de patterns depuis les niveaux ou les catégories (Kahil et al., 2021a)

Entrées :
s : l'ensemble de niveaux ou catégories
pnl : le nombre de patterns à visualiser
Sortie : ptv : l'ensemble de patterns à visualiser

Début
  Trier s et ses patterns selon le critère de tri dans l'ordre décroissant
  Répéter
    $fp \leftarrow$ le premier pattern dans s
    Si $fp \nexists ptv$ **alors**
      $ptv \leftarrow ptv \cup \{fp\}$
    Fin Si
    $s \leftarrow s/\{fp\}$
  Jusqu'à ($|ptv| = pnl$ ou $s = \emptyset$)
  Renvoyer ptv
Fin

La quatrième tâche de GreedyBigVis est dédiée à l'organisation de la visualisation. Afin de prioriser la hiérarchique, il faut gérer les différents cas de visualisation qui se distinguent en trois scénarios décrits ci-dessous.

1- Cas de visualisation exclusivement catégorielle : Dans ce cas, il n'y a pas de critère hiérarchique dans l'ensemble du data-set. Ici, le but est de visualiser un nombre limité de modèles uniquement à partir de catégories.
2- Cas de visualisation exclusivement hiérarchique : Dans ce cas, il existe au moins un lien hiérarchique entre chacun des deux couples vérifiant la formule (4.2). Ici, les critères catégoriels deviennent des filtres afin d'être utilisés pour personnaliser la visualisation.
3- Cas de visualisation hiérarchique partielle : Dans ce cas, le nombre de patterns des niveaux hiérarchiques est inférieur au nombre de patterns à visualiser. Par conséquent, il faut gérer le chevauchement hiérarchie/catégorie tout en privilégiant les patterns hiérarchiques.

Algorithme 4-3 gère ces cas. Sa stratégie dépend du fait qu'une partie de l'ensemble de données accepte des relations hiérarchiques ou pas. Selon cet aspect, un des cas de visualisation est considéré. Cela commence par construire l'ensemble des niveaux sans tenir compte des catégories pour le moment. S'il y a autant ou plus de patterns de niveaux hiérarchiques que le nombre de patterns à visualiser, toute la visualisation est exclusivement hiérarchique. Ainsi, l'algorithme construit ptv uniquement à partir des niveaux. S'il n'y a pas de patterns de niveaux hiérarchiques, toute la visualisation est catégorielle. Dans ce cas, ptv est construit à partir de catégories après avoir construit l'ensemble des catégories. S'il y a

moins de patterns de niveaux que pnl, seule une partie de l'ensemble de données accepte la visualisation hiérarchique. Dans ce cas, ptv est constitué de lsp patterns de niveaux hiérarchiques et ($pnl - lsp$) patterns de catégories, distingués respectivement par les deux clés : lp et cp. Dans tous les cas, l'algorithme construit l'ensemble de filtres à partir de catégories via différentes techniques telles que l'utilisation d'informations statistiques et d'algorithmes d'apprentissage automatique en fonction du type de données de chaque attribut.

Algorithme 4-3 : Gestion des cas de visualisation (Kahil et al., 2021a)

Entrées :
ds : data-set
pnl : nombre de patterns à visualiser
Sorties :
ptv : l'ensemble trié des patterns à visualiser
fs : l'ensemble des filtres

Début
  $hptv \leftarrow \emptyset$
  $cptv \leftarrow \emptyset$
  $ptv \leftarrow \emptyset$
  $fs \leftarrow \emptyset$
  $ls \leftarrow$ Créer l'ensemble de niveaux à partir de ds via la formule (4.2)
  Si $ls > pnl$ **alors**
    $ptv \leftarrow$ Construire ptv à partir de ls via **Algorithme 4-2**
  Sinon
    $cs \leftarrow$ Créer l'ensemble de catégories à partir de ds via la formule (4.1)
    Si $|ls| = 0$ **alors**
      $ptv \leftarrow$ Construire ptv à partir de cs via **Algorithme 4-2**
    Sinon
      $hptv \leftarrow$ Construire $hptv$ à partir de ls via **Algorithme 4-2** avec ($pnl \leftarrow |ls|$)
      $cptv \leftarrow$ Construire cptv à partir de ls via **Algorithme 4-2** avec ($pnl \leftarrow pnl - |ls|$)
      $ptv \leftarrow (lp, hptv) \cup (cp, cptv)$
    Fin Si
  Fin Si
  $fs \leftarrow$ Construire fs à partir de cs
  Renvoyer ptv, fs
Fin

La cinquième tâche de GreedyBigVis, présentée par **Algorithme 4-4**, cible la gestion de l'interaction utilisateur/visualisation afin d'assurer la mise en échelle interactive. Ceci est réalisé en considérant les filtres. Cet algorithme récupère toutes les lignes liées au filtre sélectionné, explore leurs motifs, et ajoute ce dernier à l'ensemble de modèles pour visualiser ptv dans le cas où ils ne sont pas déjà visualisés. De la même manière, l'événement de désélection est géré, sauf que contrairement à l'instruction $ptv = ptv \cup \{motif\}$, l'instruction $ptv = ptv/\{motif\}$ est exécutée sans vérifier la condition $Si\ pattern \in ptv$ ce qui signifie que le motif n'est pas déjà visualisé.

Algorithme 4-4 : Gestion des cas de sélection des filtres (Kahil et al., 2021a)

Entrées :
f : Filtre sélectionné
ptv : Patterns à visualiser
ds : data-set
Sorties :
ptv : L'ensemble modifié des patterns visualisés

Début
  $catégorie \leftarrow$ récupérer les données liées à f depuis ds
  Pour toute $ligne$ de $catégorie$ **faire**
    Pour tout $pattern$ de $ligne$ **faire**
      Si $pattern \in ptv$ **alors**
        $ptv \leftarrow ptv \cup \{pattern\}$
      Fin Si
    Fin Pour
  Fin Pour
Fin

4 Expérimentation

Afin d'évaluer l'approche proposée, elle a été implémentée en utilisant Apache Spark pour effectuer le traitement des tâches. Comparé à Hadoop, Spark est plus rapide et plus efficace en temps réel (Hazarika et al., 2017; Kahil et al., 2020; Maheshwar & Haritha, 2016). *PySpark* (Feng, 2019) est l'interface python de Spark. Ce langage a été choisi car, en plus de sa simplicité et de ses hautes performances, il possède des bibliothèques qui fournissent plusieurs visualisations comme *Plotly* et *Matplotlib*. Un data-set médical[6] formaté en csv a été choisi. Il représente les informations sur les maladies dans villes américaines selon différentes propriétés telles que l'ethnicité, le sexe et le lieu. L'ensemble de données contient 34 492 lignes et 15 colonnes. Conformément à l'approche proposée, **Figure 4-2** présente le processus suivi pour sa mise en œuvre. La lecture du fichier csv se fait avec la dataframe Spark, qui est gérée par le package SparkSQL, afin de la manipuler via les transformations et les actions de la théorie des ensembles Spark de manière structurée (Amghar et al., 2020; Feng, 2019). La pertinence des colonnes est déterminée en comptant le nombre de valeurs nulles que chacune contient ; celles qui en ont un nombre inférieur sont les plus pertinentes. Le résultat du comptage est un dictionnaire qui contient les noms des colonnes et le nombre de valeurs non nulles.

[6] URL: https://bchi.bigcitieshealth.org /indicators/1827/searches/34444

4.1 Application de l'approche

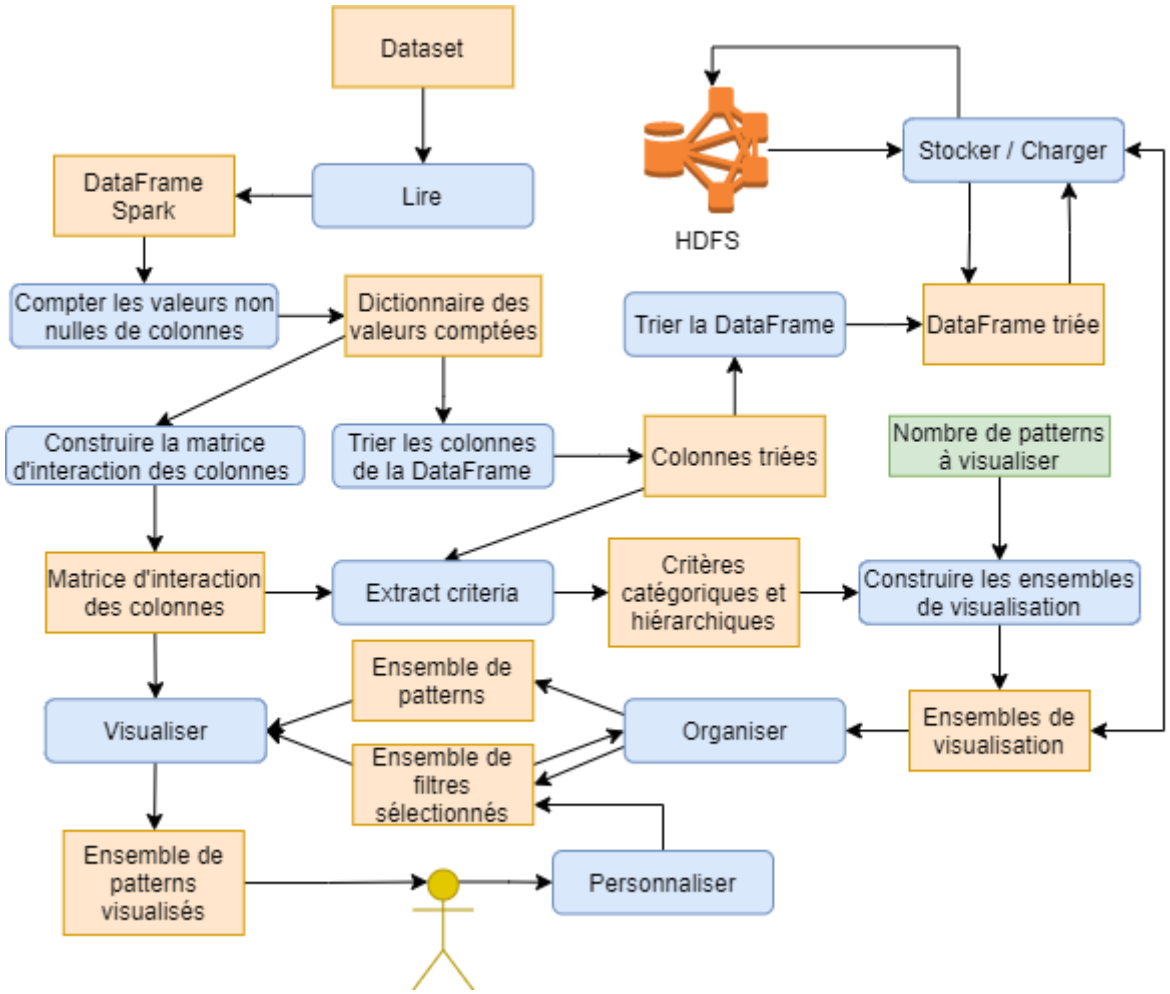

Figure 4-2 : Concrétisation de GreedyBigVis via Spark (Kahil et al., 2021a)

Ce dictionnaire est utilisé pour trier la dataframe à travers colonnes triées et construire la matrice d'interaction des colonnes définie via la formule **(3)**.

Figure 4-3 présente une visualisation 3D de l'interaction de ces colonnes selon la matrice d'interaction construite. Cette présentation offre une vue flexible qui peut être personnalisée via le zoom, l'affichage panoramique et la rotation. Les premiers critères (les plus pertinents) sont présentés de haut en bas. Cette figure montre deux cas de critères selon leurs valeurs : Le premier cas présente un critère moins pertinent (Methods, Value) avec 7887 valeurs non nulles. Alors que le critère (Value, Place) dans le second cas est un exemple de critère pertinent (avec 34 492 valeurs non nulles).

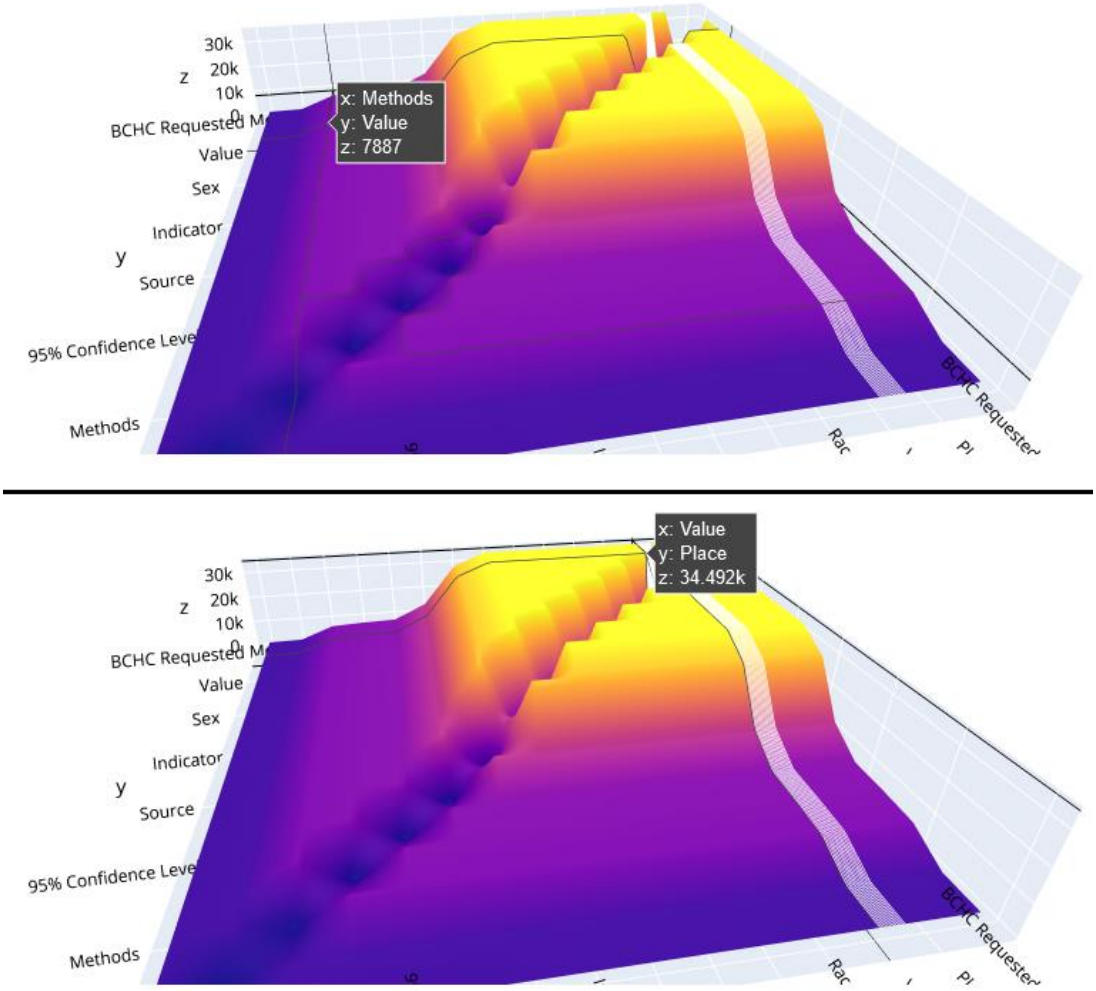

Figure 4-3 : Visualisation 3D de l'interaction des colonnes (Kahil et al., 2021a)

Table 4-2 présente les statistiques des différents ensembles construits de l'approche, à savoir les critères catégoriels, les critères hiérarchiques, les catégories et les niveaux. L'ensemble de l'implémentation peut être consulté dans le référentiel Github nommé spark_dataset_preprocessing[7].

Table 4-2 : Statistiques des critères, niveaux et catégories du data-set (Kahil et al., 2021a)

Lignes	Crtières hiérarchiques	Critères catégoriels	Niveaux hiérarchiques	Catégories
34492	210	15	15	0

A partir de ces statistiques, considérant que le nombre de patterns à afficher est de 20, le cas de visualisation est entièrement hiérarchique (en utilisant **Algorithme 4-3**). Par conséquent, les critères catégoriels sont implémentés en tant que filtres pour personnaliser la visualisation hiérarchique. Leur ensemble est développé selon cinq colonnes : (1) Indicator category, (2) Sex, (3) Year, (4) Race/Ethnicity et (5) Value. Les quatre premiers filtres ont des valeurs discrètes, tandis que le cinquième a des valeurs continues dont l'ensemble a été construit à l'aide de le clustering k-moyennes. Les résultats sont présentés dans **Table 4-3**.

[7] https://github.com/Mus-Kah/spark_dataset_ preprocessing/releases/tag/v0.1

Table 4-3 : L'ensmble de filtres construits (Kahil et al., 2021a)

Colonne	Year	Sex	Value	Race/Ethnicity	Indicator category
Nombre de filtres	8	3	20	9	13
Valeurs	2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017	Male, Female, Both	96, 55,703, 429,732, 658,741, 827,281, 996,847, 1,212,443, 1,513,654, 1,979,341, 2,210,477, 2,541,342, 2,706,312, 3,253,803, 3,923,835, 8,464,338, 313,914,040, 316,128,839, 318,857,056, 321,418,821, 323,127,515	All, American Indian/Alaska Native, Asian/PI, Black, Hispanic, Other, White, Multiracial	Behavioral Health/Substance Abuse, Cancer, Chronic Disease, Demographics, Environment, Food Safety, HIV/AIDS, Infectious Disease, Injury/Violence, Life Expectancy and Death Rate (Overall), Maternal and Child Health, Sexually Transmitted Infections, Social and Economic Factors

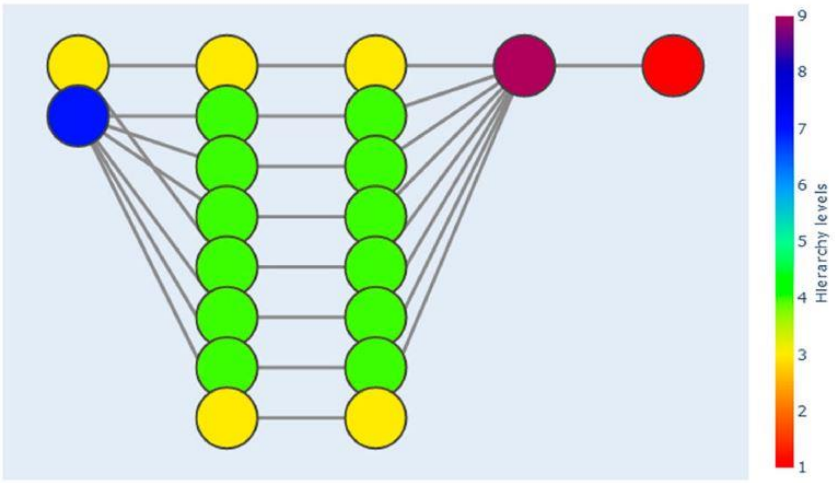

Figure 4-4 : Exemple de visualisation hiérarchique du data-set (Kahil et al., 2021a)

Figure 4-4 montre un exemple de visualisation hiérarchique qui est construite sur la base des critères obtenus par GreedyBigVis en utilisant *plotly* et *networkx*. La valeur de chaque nœud est affichée juste après avoir survolé celui-ci. Les solutions citées sont évaluées selon 8 facteurs désignés et ce afin de les comparer avec l'approche proposée. Ces facteurs sont liés aux contraintes mentionnées ci-dessus. Dans cette évaluation, ils ne sont pas considérés avec le même degré d'importance, mais sont plutôt décrits par ordre d'importance comme mentionné ci-dessous.

1- Les Techniques (Techs) incluent les techniques de visualisation et les algorithmes utilisés pour mettre en œuvre chaque solution. Ils sont fortement liés à la complexité et donnent une idée des domaines du monde réel dans lesquels ils peuvent être appliqués.

2- La complexité (cmpx), pour n lignes et m colonnes, est liée à la latence. Plus la complexité est faible, plus la latence est réduite et plus la visualisation est efficace. elle est plutôt basée sur la complexité des techniques employées pour chaque solution, qui peut ne pas être exacte dans certains cas.
3- Le temps réel (RealT) est l'efficacité de chaque solution pour satisfaire la contrainte de temps réel. Ce facteur repose essentiellement sur les outils utilisés pour mettre en œuvre chaque solution et peut être amélioré par la complexité des méthodes employées pour cette dernière.
4- L'échelonnabilité perceptuelle (Perc) est l'aptitude de chaque solution à gérer les données en temps réel dans les cas de leur modification ou de l'ajout de nouvelles sources de données.
5- L'interactivité (Intr) est liée à la scalabilité interactive, c'est-à-dire la capacité de chaque solution à répondre aux interactions des utilisateurs en temps réel.
6- La multidimensionnalité (MltD) fait référence à la capacité ou non de chacune des solutions existantes à traiter des données de grande dimension provenant de différentes sources et à travers de multiples techniques de visualisation.
7- Hiérarchie (Hierc) indique la capacité de chaque solution à fournir des vues structurées grâce à une visualisation hiérarchique.
8- Les types de données (DataT) supportés par chaque solution donnent une idée des domaines d'application qu'elle peut couvrir.

Table 4-4 : Comparaison entre GreedyBigVis et les travaux connexes (Kahil et al., 2021a)

Approche	Techs	Cmpx	RealT	Perc	Intr	Hierc	MultD	DataT
Méthode (Qunchao Fu et al., 2014)	BA, M	$O(2n^2)$	✓		✓			Sp, Tmp
Système (Dash et al., 2008)	WAH, BT	$O(2n^2+2)$	✓	✓	✓	✓		Tx, Im, VD, Cat
Hashed-cubes	PH, HM, BSP, LH	$O(n^2)$	✓	✓	✓	✓	✓	Sp, Cat, Tmp
Système(Sansen et al., 2017)	PC, CC	$O(nkt)$	✓	✓	✓			Nm, Tx
VisReduce	DA, MR	-	✓	✓	✓		✓	Tx, Im
OptiqueVQS	OWL, DA	-	✓	✓	✓			Tx, Sp
SkyViz	SL, SF	$O(nlog(n))$	✓	✓	✓		✓	Nm, Tx
Faceted Browsing	BN		✓	✓	✓	✓		Tx, Im, Vd, Cat
DeepEye	LR, DT, C	$O(nlog(n))$	✓	✓	✓		✓	Nm, Tx
Architecture (Kahil et al., 2019)	TD, DS	$O(n)$	✓	✓	✓	✓	✓	Nm, Tx, Im
Hdoutliers	P, PC, RR	$O(n)$	✓	✓			✓	Sp, Tx, Tmp
ACP	SVG, PC	$O(mn \times \min(n,m))$	✓	✓	✓			Cat, Nm
TDViz	WC, C	-	✓		✓		✓	Tx
MDV	MDA, HM, C, TD, GD	-	✓		✓	✓	✓	Cat, Im, Vd
GreedyBigVis	GA, ST	$O(nlog(n))$	✓	✓	✓	✓	✓	Im, Vd, Cat, Nm

Table 4-4 montre la comparaison entre les ouvrages existants et l'approche proposée en fonction des facteurs énumérés. Les abréviations utilisées sont listées dans les points ci-dessous.

- Techniques (Techs) : BA : Binned Aggregation, M : Map, BN : Bayesian Network, HM : Hitmaps, BSP : Binned Scatter Plots, PH : Hiérarchie pivot, LH : Linked histograms, PC : Parallel Coordinates, DA : Data aggregation, MR : MapReduce, OWL, S : SPARQL, 3M : 3Dmap, SL : Skyline, SF : SuitabilityFunction, LR : Apprendre à classer (un modèle d'apprentissage automatique), DT : Arbre de décision, BT : Arbre de bits, WAH : Word-Aligned Hybrid, C : Charts, TD : Tree Diagrams, GD : Graph Diagrams, DS : Data Scraping, WC : WordCloud, P : Probabilities, GA : Greedy Algorithm, MDA : Model-driven Architecture, ST : Set Theory, CC : Canopy Clustering, RR : Résidus de régression.
- Types de données (DataT) : T : Tx, Im : Images, Vd : Vidéos, Sp : Spatiales, Cat : Catégoriques, Tmp : Temporelles, Nm : Numériques.

D'après les résultats présentés, et compte tenu des facteurs énumérés, on peut voir que SkyViz, DeepEye et TDViz, bien qu'ils prennent en charge la multidimensionnalité des données, la mise en échelle en termes de perceptivité et d'interactivité, et la contrainte en temps réel, ne fournissent pas de visualisation hiérarchique, ce qui est pris en charge par GreedyBgVis. Parmi les autres solutions, il y a celles qui offrent une visualisation hiérarchique telles que HashedCubes et la solution basée sur la navigation à facettes (Simonini & Zhu, 2015), mais la complexité des techniques qu'ils utilisent est élevée. A partir de cette analyse, on peut juger que l'approche proposée est plus efficace en ce qui concerne les facteurs indiqués. Outre ces avantages, on constate que le nombre de types de données pris en charge est limité : les données spatiales et les données temporelles ne sont pas prises en charge. C'est pourquoi une partie des futurs plans consiste à améliorer la proposition pour couvrir la gestion de plus de types de données.

5 Conclusion

Ce chapitre a présenté GreedyBigVis, une nouvelle approche simple mais prometteuse qui prépare efficacement les grands data-sets multidimensionnels pour une visualisation interactive. Elle vise à donner une vue structurée qui sera efficace en termes de lisibilité et d'expressivité. Afin de l'améliorer, il est envisageable de couvrir d'autres types de données telles que les séries temporelles et les données spatiales. De même, il serait intéressant de cibler l'axe des préférences de l'utilisateur en englobant d'autres aspects tels que la sauvegarde des visualisations souhaitées par les utilisateurs en termes de modèles et de requêtes de recherche. Ces aspects peuvent également être utilisés pour fournir des visualisations efficaces aux nouveaux utilisateurs. Dans ce contexte, l'objectif est de proposer un nouveau framework basé sur cette approche qui offre approximativement à chaque utilisateur sa propre visualisation en fonction de l'historique. Il permet également de sauvegarder les requêtes et, selon les orientations des utilisateurs existants, de donner des visualisations pertinentes aux nouveaux utilisateurs. Des techniques basées sur la recommandation, telles que le filtrage collaboratif, peuvent être utilisées à cette fin. De plus, les filtres de personnalisation seront plus efficaces si leurs valeurs sont regroupées en fonction d'objectifs précis et en utilisant de multiples algorithmes de clustering. Une approche basée sur le clustering multi-perspectives peut être envisagée afin de spécifier parfaitement les groupes de chaque filtre en fonction des types de données et de leurs valeurs.

Chapitre 5: Visualisation des graphes à grande échelle via la détection de communautés

1 Introduction

Les graphes sont largement connus pour être adoptés dans différents domaines tels que les réseaux énergétiques, les réseaux sociaux, les réseaux du trafic, villes intelligentes et beaucoup d'autres systèmes complexes (Kulcu et al., 2016; Lancichinetti & Fortunato, 2009). Leur visualisation permet à l'utilisateur de les explorer efficacement, d'avoir un aperçu et une intuition sur les informations qu'ils peuvent avoir. À l'ère du Big Data, les réseaux sont caractérisés par la complexité en termes de volumes si importants et relations si nombreuses et variées qu'ils nécessitent des solutions modernes permettant le traitement à grande échelle. Cette haute échelonnabilité émerge de l'interconnexion de différentes technologies qui produisent de très grandes quantités de données (Kahil et al., 2020) et, par conséquent, des réseaux plus complexes en termes de nœuds et de relations. En conséquence, de nouvelles caractéristiques et défis qui sont apparus, tels que les changements dynamiques des réseaux, doivent être considérés. La visualisation de ces graphes est devenue, elle-aussi, un enjeu préoccupant du fait que la présentation graphique d'un nombre important de nœuds avec des relations complexes est difficilement réalisable (Kahil et al., 2021b). Cela mène à générer une vue brouillée difficile à comprendre par l'utilisateur. Une solution pour une visualisation efficace et effective des graphes est de se baser sur le concept de détection des communautés afin de diminuer la densité des nœuds à visualiser et, par conséquent, offrir une visualisation plus lisible et plus compréhensible. La détection de communauté est un enjeu très considéré dans les graphes (Teng, 2016). Ce problème peut être simplement défini comme le processus de découvrir des sous-graphes à partir d'un graphe donné, de sorte que chacun de ceux-ci contienne des nœuds qui ont une densité plus élevée que celle entre les autres nœuds du même graphique. Cette densité reflète la forte connexion entre des nœuds appartenant à la même communauté (Rozemberczki et al., 2019). Le problème de détection des communautés est répandu dans différents domaines tels que les réseaux sociaux et les systèmes biologiques (Bedi & Sharma, 2016). La définition d'un tel problème est flexible ; elle peut être déterminée en fonction des relations entre les nœuds. Dans un contexte de réseaux sociaux, les utilisateurs étant les nœuds peuvent avoir différents types de connexions tels que : ami, follower, membre d'un groupe, réaction à un poste et ainsi de suite. Cet aspect a attiré l'attention de la communauté scientifique. C'est une partie des défis les plus ciblés dans les réseaux liés à Big Data qui ont introduit le problème de modularité (Resolution Limit Problem) (Duan et al., 2014; Shao et al., 2015). Ce dernier est lié aux réseaux à grande échelle face au nombre élevé de nœuds, différentes tailles de communautés, distribution hétérogène des degrés des nœuds et les relations que peuvent avoir ces nœuds. De multiples solutions ont été proposées pour résoudre ce problème. Ces dernières utilisent différentes méthodes dont les algorithmes d'apprentissage automatique tels que les voisins les plus proches (KNN) et le clustering. La présente contribution propose une nouvelle approche pour résoudre le problème de détection des communautés dans les réseaux à grande échelle d'une manière efficace, en s'inspirant de l'architecture d'Apache Spark dont le principe de cette dernière est de distribuer le calcul sur plusieurs machines afin d'accélérer le traitement. Cet framework contient une bibliothèque nommée GraphX qui est dédiée au traitement des graphes pour simplifier et améliorer le processus de détection des communautés.

La présentation de cette contribution (Kahil et al., 2019, 2021b) se fera comme suit : La deuxième section présente une définition des notions générales sur la théorie des graphes englobant des définitions relatives à ce concept, des problèmes fondamentaux répandus et une définition détaillée du problème de détection des communautés dans les graphes (réseaux). La quatrième section décrit l'approche proposée pour résoudre le problème de détection des communautés dans les graphes à grande échelle, ainsi que leur visualisation interactive. La cinquième section mène une évaluation et une discussion des résultats de l'approche proposée. La sixième section conclut ce chapitre.

2 Concepts généraux sur les graphes

Les graphes sont omniprésents dans différents domaines, y compris ceux liés à Big Data. Ils peuvent représenter des abstractions de différents systèmes complexes tels que les réseaux sociaux, les systèmes de parrainage, les réseaux de distribution d'énergie, etc.

2.1 Définition des graphs

D'un point de vue informatique et de recherche opérationnelle, un graphe G est une structure de données qui est constituée d'un ensemble de nœuds V liés par un ensemble d'arcs (ou d'arêtes) E (Knauer, 2019). Ces derniers représentent des liaisons entres les nœuds qui peuvent être homogènes ou hétérogènes.

2.2 Types des graphes

Les graphes peuvent être distingués selon différents critères dont le plus commun est l'orientation. Selon cette dernière, les graphes peuvent être orientés ou non-orientés. Un graphe orienté est un graphe qui contient des arcs (Knauer, 2019). Les relations sont alors définies selon l'orientations de ces derniers et elles sont unidirectionnelles, c-à-d. pour deux nœuds A et B : $A \rightarrow B$ n'implique pas $B \rightarrow A$.

Un graphe non orienté est un graphe où les nœuds ne sont liés que par de arêtes (Knauer, 2019). Ces dernières représentent des relations bidirectionnelles, c-à-d. pour deux nœuds A et B liés par une arête : $A \rightarrow B$ et $B \rightarrow A$.

3 Détection des communautés dans les graphes : Aperçu

3.1 Définition

La détection des communautés est une méthode de décomposition qui a pour objectif de trouver des groupes au sein des systèmes complexes qui sont représentés par des graphes (Allman et al., 2018). En effet, cette méthode trouve les sous-réseaux qui contiennent les nœuds qui ont significativement et statistiquement plus de liaisons au sein d'un groupe que les nœuds en dehors. **Figure 5-2** illustre le concept de détection des communautés dans son cas trivial.

3.2 Techniques de détection des communautés et mesures d'évaluation

Il existe de nombreuses méthodes qui visent à résoudre le problème de détection des communautés dans les graphes. (Kim & Lee, 2015) les ont classifiées selon le nombre de couches que contient le réseau. Selon ce critère, on distingue les réseaux (1) monocouches (à une couche), (2) à deux couches et (3) multicouches. Pour la première catégorie, différents algorithmes peuvent être employés pour la détection des communautés, parmi lesquels il y a les algorithmes de partitionnement des graphes, les algorithmes basés modularité (la recherche gloutonne et le recuit simulé), les algorithmes de partitionnement spectral et les algorithmes de définition des structures (Shao et al., 2015). Le principe de ces algorithmes est de diviser le graphe selon les sommets jusqu'à ce que chaque partition atteinte une taille minimale. Pour les réseaux à deux couches, la détection des communautés est généralement basée sur les algorithmes « d'expansion » des clusters (cluster expansion algorithm) comme les SVM, les techniques basées modèle comme le modèle probabiliste Bayésien, les techniques de fusion des graphes comme KNN et les techniques de fouille des patterns (pattern mining techniques) comme la corrélation structurelle (Structural Correlation Pattern Mining). Pour les réseaux multicouches, différentes techniques basées sur la factorisation matricielle et la fouille des patterns peuvent être utilisées telles que DFS et la factorisation matricielle de rang inférieur (Low-Rank Matrix Factorization). Une autre classification des méthodes de détection des communautés proposées par (Rozemberczki et al., 2019) les distingue en méthodes basées voisinages, multi-échelles et des méthodes d'intégration sensibles à la communité (community-aware embedding). Parmi les algorithmes qui ont montré une bonne performance dans la détection des communautés dans les réseaux il y a l'algorithme de Markov Cluster, LPA, la fouille des

sous-graphes fréquents (Frequent Sub-graph Mining), la prédiction des liens (Link Prediction), la classification telle que la classification des graphes et la classification des sommets (Vertex classification), etc. Afin d'évaluer les solutions de détection des communautés, de nombreuses mesures peuvent être utilisées parmi lesquelles il y a la F-mesure (F-measure), l'indice de Jaccard et l'information mutuelle (Duan et al., 2014).

4 Approche proposée pour la détection et la visualisation des communautés dans les graphes à grande échelle

L'approche proposée (Kahil et al., 2021b) véhicule une nouvelle méthode de visualisation des graphes à grande échelle de manière consistante et optimisée. L'idée est de visualiser à tout moment une partie du graphe au lieu de son intégralité de façon à ce que cette partie soit représentative d'un certain aspect dans ce graphe. Une solution pour obtenir des parties significatives lors de diviser le graphe est la détection des communautés. Comme est mentionné ci-dessus, toute communauté d'un graphe englobe l'ensemble de nœuds qui sont fortement connexes selon un critère précis. Le critère est défini selon les connections que peuvent avoir les arêtes dans un graphe non-orienté ou les arcs dans un arc orienté. Dans le contexte des réseaux sociaux, ces liaisons peuvent être des relations entre les personnes telles que : ami, follower, etc., des réactions à des publications telles que : aimer, ne pas aimer, etc… Après la détection des communautés, elles sont présentées visuellement selon l'intérêt d l'utilisateur. Ce dernier peut personnaliser la présentation en en visualisant d'autres via des paramètres qui lui sont offerts, notamment le filtrage et la sélection. Ainsi, le bruit de visualisation des graphes est considérablement minimisé et leur exploration devient plus effective.

L'approche proposée est abstraitement conçue pour couvrir cinq étapes séquentielles, à savoir :

1- l'extraction des propriétés (labels) à partir des relations traduites par les arcs ou les arêtes,
2- la construction des sous-graphes de façon à ce que chaque sous-graphe contienne tous les nœuds qui sont liés via une des propriétés extraites,
3- la détection des communautés depuis chacun des sous-graphes via l'algorithme LPA (Label Propagation Algorithm),
4- la fusion et optimisation des communautés détectées afin d'éliminer les duplications entre elles,
5- la présentation des communautés via une visualisation interactive et personnalisable par l'utilisateur via des mécanismes qui lui sont offerts.

Comme les graphes à grande échelle ont des structures complexes en termes de nœuds et des relations qu'ils peuvent avoir, la détection des communautés au sein d'eux est un processus lourd qui nécessite une puissance de calcul pour être exécuté dans un temps acceptable. Une solution pour accélérer ce processus est de le paralléliser. A cet égard, le processus de détection et de visualisation des communautés adopte l'architecture offerte par le framework GraphX de Spark. Pour cela, le flux abstrait de l'approche proposé est concrétisé par l'architecture illustrée dans **Figure 5-1**. L'ensemble des étapes qui la composent son processus sont décrites ci-dessous.

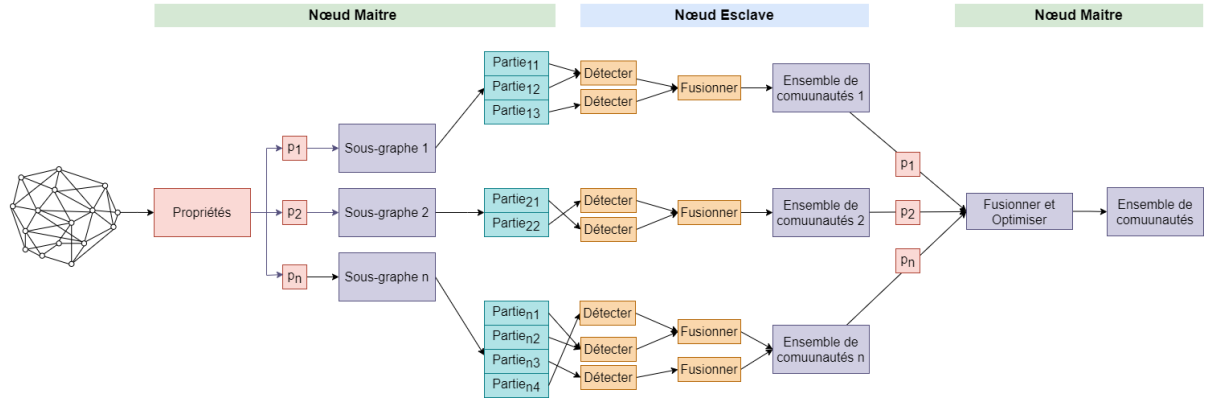

Figure 5-1: Processus détaillé de l'approche proposée (Kahil et al., 2021b)

L'étape d'extraction des propriétés consiste à les trouver dans le graphe en récupérant tous les arcs ou les arêtes qui y existent et en construisant l'ensemble des attributs P qu'ils contiennent. La formule suivante représente cette étape.

$$P = \{p = e \in E\} \quad (5.1)$$

A partir de cette formule on peut déduire que $|P| \leq |E|$, c-à-d. le nombre de propriétés extraites est inférieur ou égal au nombre de connexion dans le graphe.

4.1 Construction des sous-graphes

L'ensemble des sous-graphes est construit à partir de chacune des propriétés extraites dans la première étape. La construction est faite via une fonction de filtrage qui est définie par la formule suivante :

$$sg_p = f(G, p) \quad (5.2)$$

où : G est le graphe et sg_p, le sous-graphe relatif à la propriété p et f est la fonction de filtrage. En effet, f peut être assimilée à la fonction de sélection dans SQL dont l'expression est la suivante : $SELECT * FROM\ G\ WHERE\ p$. L'ensemble de tous les sous-graphes GSG construits vérifie l'équation $|GSG| = |P|$ qui signifie que chaque sous-graphe est relatif à une propriété unique. **Algorithme 5-1** résume le processus d'extraction des sous-graphes.

Algorithme 5-1 : Extraction des Sous-graphes (Kahil et al., 2021b)

Entrées :
$G(V, E)$: le graphe
P : l'ensemble des propriétés qui définissent les types de connexions entre les nœuds
Sortie :
GSG : l'ensembles des sous-graphes de G

Début
  $GSG = \emptyset$
  Pour $p \in P$ **faire** :
    $Vp =$ l'ensemble des nœuds relatifs à p
    $Ep =$ l'ensemble des arcs ou des arêtes relatifs à p
    Construire le sous-graphe $g(Vp, Ep)$ via la formule **(5.2)**
    $GSG = GSG \cup \{(p, g)\}$
  Fin Pour
Fin

Comme les nœuds peuvent chacun avoir plus d'un seul type de relations avec les autres, les sous-graphes peuvent avoir des nœuds communs à cette phase. On peut distinguer deux cas selon $|P|$.

1- $|P| = 1$: cela signifie qu'il y a une seule propriété dans l'intégralité du graphe, c-à-d. chaque nœud peut ou ne peut pas être connecté aux nœuds de graphe via un seule label de liaison. Ce cas, illustré dans **Figure 5-2** représente le simple cas des graphes à grande échelle et vérifie l'équation $GSG = G$.

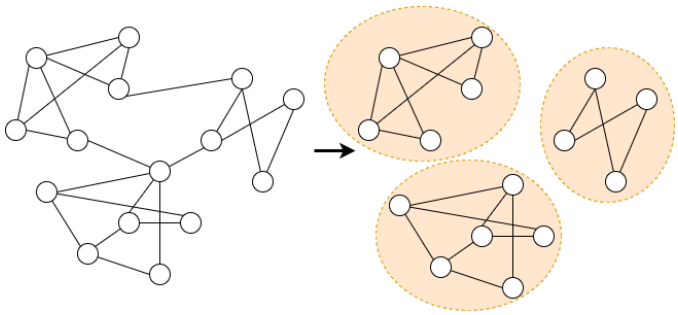

Figure 5-2 :Communautés extraites depuis un graphe avec une seule propriété (Kahil et al., 2021b)

2- $|P| > 1$: il y a plus d'une propriété dans le graphe. **Figure 5-3** illustre cette situation. Deux cas peuvent être distingués selon le nombre de propriétés qui peuvent être sélectionnées simultanément :

   a. Si seule une propriété peut être sélectionnée à la fois, la détection des communautés est exécutée uniquement à partir du sous-graphe qui correspond à la propriété sélectionnée. Dans ce cas, la formule **(5.3)** est vérifiée. Elle signifie qu'à chaque moment, toute propriété existe exclusivement dans un seul sous-graphe.

$$\forall E_i, E_j \subset E, i \neq j: E_i \cap E_j = \emptyset \qquad (5.3)$$

   tel que : E_i et E_j sont respectivement les ensembles des arcs ou des arêtes de sg_i et sg_j.

   b. S'il est possible de sélectionner plusieurs propriétés simultanément, la détection des communautés est faite sur tous les sous-graphes qui correspondent aux propriétés sélectionnées. Dans ce cas, il peut y avoir des chevauchements entre les communautés détectées qui sont traduits par la formule (5.4) et qui doivent être traités. Ils font l'objet de la sous-section 4.3.

$$\exists E_i, E_j \subset E, i \neq j: E_i \cap E_j \neq \emptyset \qquad (5.4)$$

En pratique, les différents cas de la valeur de $|p|$ mentionnés sont spécifiés selon l'application en question ainsi que l'intérêt de l'utilisateur. Ces derniers peuvent sélectionner une ou plusieurs propriétés qui sont représentées par des critères manipulés via différents mécanismes tels que les filtres de visualisation. Comme le montre **Algorithme 5-1**, tout sous-graphe doit être attribué par un label qui l'identifie afin de le préserver après la division et de l'utiliser dans les étapes subséquentes.

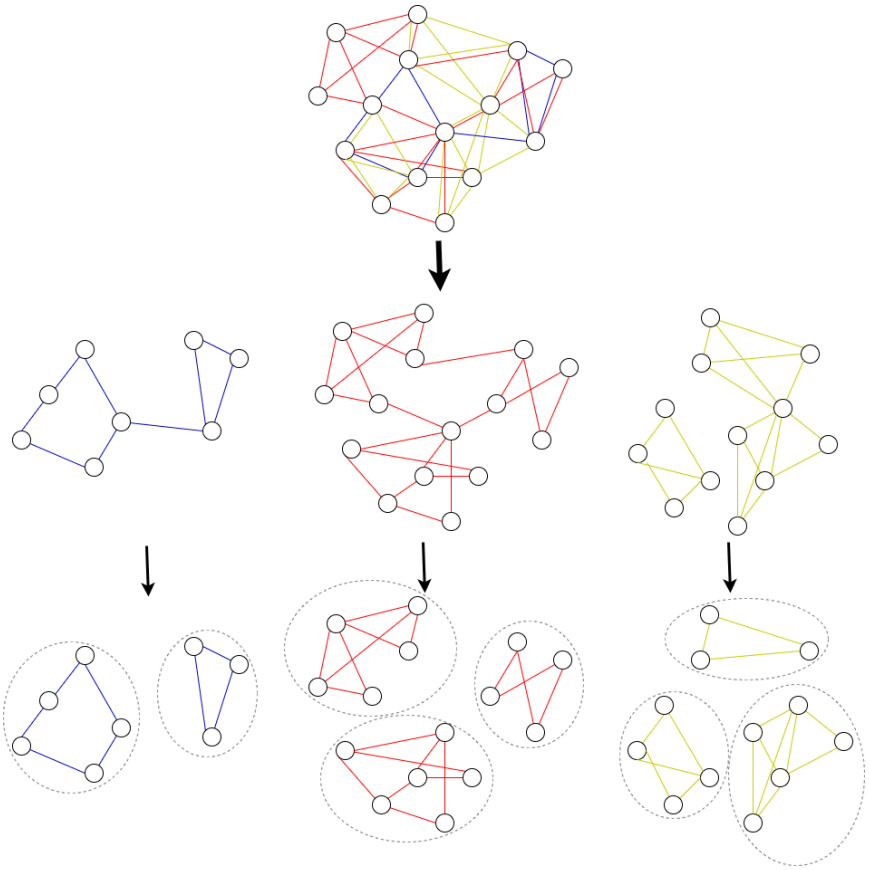

Figure 5-3 : Extraction des communautés dans un graphe avec multiples propriétés (Kahil et al., 2021b)

4.2 Détection des communautés

Dans cette étape, le processus de détection des communautés s'exécute de manière parallèle sur la base de la distribution de Spark. Bien que le principe de cette dernière paraisse similaire à celui du paradigme MapReduce, ils diffèrent en réalité dans la stratégie de division et de collecte des données. En MapReduce, la division des données est faite exclusivement sur la base de leur taille. Quant à l'approche proposée, elle construit d'abord les sous-graphes et divise chacun d'eux pour les distribuer. Ainsi, tout sous-graphe est traité indépendamment des autres et les tailles des fragments peuvent varier d'un sous-graphe à un autre. Cette stratégie, consistant à traiter les graphes selon un seul critère à chaque moment, garantit le non-chevauchement et aide à faciliter à l'utilisateur le processus d'exploration de la visualisation. Afin de mettre en œuvre cette stratégie, il faut prendre en considération les spécificités du graphe, à savoir le nombre de nœuds et le nombre des arcs ou des arêtes pour mettre en place les machines avec la capacité nécessaire. Ensuite, le programme de détection des communautés est exécuté sur chaque machine. En effet, ce programme est l'implémentation de l'algorithme itératif LPA. Il a été choisi pour sa rapidité dans la détection des communautés comparé à d'autres algorithmes tels que Walktrap et Infomap (Yang et al., 2016). Le principe de LPA peut être résumé dans les trois points suivants :

1- En supposant que les nœuds d'un graphe G sont chacun initialisé par un label de communauté unique, choisir un nœud.
2- Déplacer aléatoirement par ses voisins et mettre à jour son label par celui du nœud qui a plus de connections avec les autres nœuds.
3- Répéter ces étapes jusqu'à ce que chaque nœud ait le label de la majorité de ses voisins.

Comme les communautés peuvent être considérées comme des sous-graphe, chacune d'elles est définie comme suit : $C = (C_V, C_E)$ tel que C_V est l'ensemble des nœuds qu'elle contient et C_E est l'ensemble d'arcs ou d'arêtes. **Algorithme 5-2** présente la version étendue de LPA sur la base de l'approche proposée. La fonction f dans cet algorithme se charge de donner attribuer chaque nœud au label de la majorité de ses voisins.

Algorithme 5-2 : Extraction des communautés via LPA (Kahil et al., 2021b)

Entrées :
$G(V, E)$: le graphe
Sortie :
C : l'ensembles des communautés détectées

Début
  $GSG =$ Extraire les sous-graphes de G via **Algorithme 5-1**
  Pour $(p, g) \subset GSG$ **faire** :
    Assigner chaque nœud de g par un label aléatoire unique
    $it = 1$ // le nombre d'itérations
    $V = g_V$ // stocker les nœuds de g dans le vecteur V
    Pour $v \in V$ **faire :**
      $c_v(t) = f(v_l, v_{ul})$ // v_l : les voisins de v dont les labels sont déjà mis à jour, v_{ul} : les voisins de v dont les labels ne sont pas encore mis à jour
    Fin Pour
    Si condition **alors**
      Rassembler tous les nœuds qui partagent le même label dans une communauté
      Séparer toutes les communautés connectées
      $C_p = \{toutes\ les\ communautés\ détectées\}$
      $C = C \cup \{(p, C_p)\}$
      Renvoyer C
    Fin Si
    $it = it + 1$
  Fin Pour
Fin

4.3 Fusionnement et optimisation des communautés

Cette étape consiste à optimiser les communautés détectées qui possèdent des nœuds communs comme illustré dans **Figure 5-4**. Ces nœuds doivent appartenir à une seule communauté afin de rendre efficaces la visualisation et l'exploration. A cet égard, il faut définir une stratégie pour déterminer la communauté la plus consistante parmi celles qui partagent des nœuds communs. Pour cela, la formule **(5.5)** la trouve en mesurant la consistance sur la base du nombre de connexions que chaque communauté contient.

$$Conisit_c = \{\, c_i \subset C, \forall c_j \subset C, i \neq j : |E_{Ci}| > |E_{Cj}|\} \qquad (5.5)$$

tel que : E_{Ci} et E_{Cj} sont respectivement les ensembles des arcs ou des arêtes de c_i et c_j.

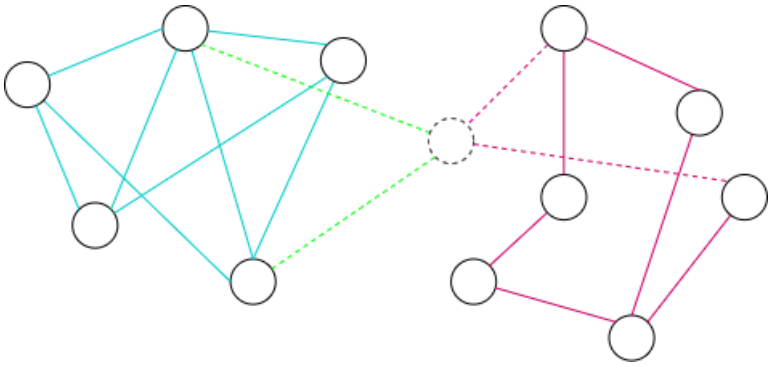

Figure 5-4 : Exemple d'un nœud commun entre deux communautés détectées (Kahil et al., 2021b)

4.4 Visualisation interactive des communautés

La dernière étape consiste à présenter les communautés à l'utilisateur à travers une interface graphique qui lui est interactive via des paramètres à travers lesquels il peut la personnaliser selon ses intérêts. La personnalisation est essentiellement traduite par le filtrage des communautés à visualiser selon des critères précis. Ces derniers comprennent un critère par défaut qui est défini selon le domaine en question et d'autres critères qui reflètent les propriétés extraites dans la première étape de l'approche proposée. L'ensemble des critères peut alors être défini par la formule suivante :

$$Cr = P \cup Dcr \qquad (5.6)$$

tel que : P est l'ensemble des propriétés extraites dans la première phase et Dcr est l'ensemble des critères liés aux communautés construites dans la troisième étape.

Les communautés, attribuées par l'ensemble des critères, sont visualisées conformément à une architecture proposée. En effet, cette architecture, montrée dans **Figure 5-5**, n'est pas destinée uniquement à visualiser les graphes, mais aussi les data-sets construits par la collecte des données depuis différentes sources. Ces data-sets peuvent être statiques ou dynamiques. Cette architecture a pour objectifs de :

- Assurer à l'utilisateur une visualisation interactive tout en respectant les contraintes listées dans le chapitre **3** à savoir les contraintes de visualisation basiques, les contraintes d'interactivité, les contraintes de mise en échelle et les contraintes de structuration. Cette architecture offre des formes graphiques significatives et compréhensibles par l'utilisateur. En plus, elle permet de fournir des fonctionnalités pour personnaliser la visualisation tels que le zoom et la sélection.
- Respecter l'homogénéité des formes graphiques pour assurer une prise et familiarité rapides à l'utilisateur.
- Assurer la disponibilité des données relatives aux patterns visualisés.
- Fournir des critères pour adapter la visualisation aux besoins de l'utilisateur en termes de niveaux de visualisation et de données.

Comme le montre **Figure 5-5**, cette architecture est composée de quatre modules non séquentiels qui assurent la réalisation des tâches de collecte des données, les stocker et les mettre à jour, gérer les critères de visualisation et visualiser ces données en assurant l'interactivité avec l'utilisateur. L'intégralité de cette architecture prend en considération toutes les contraintes citées ci-dessus.

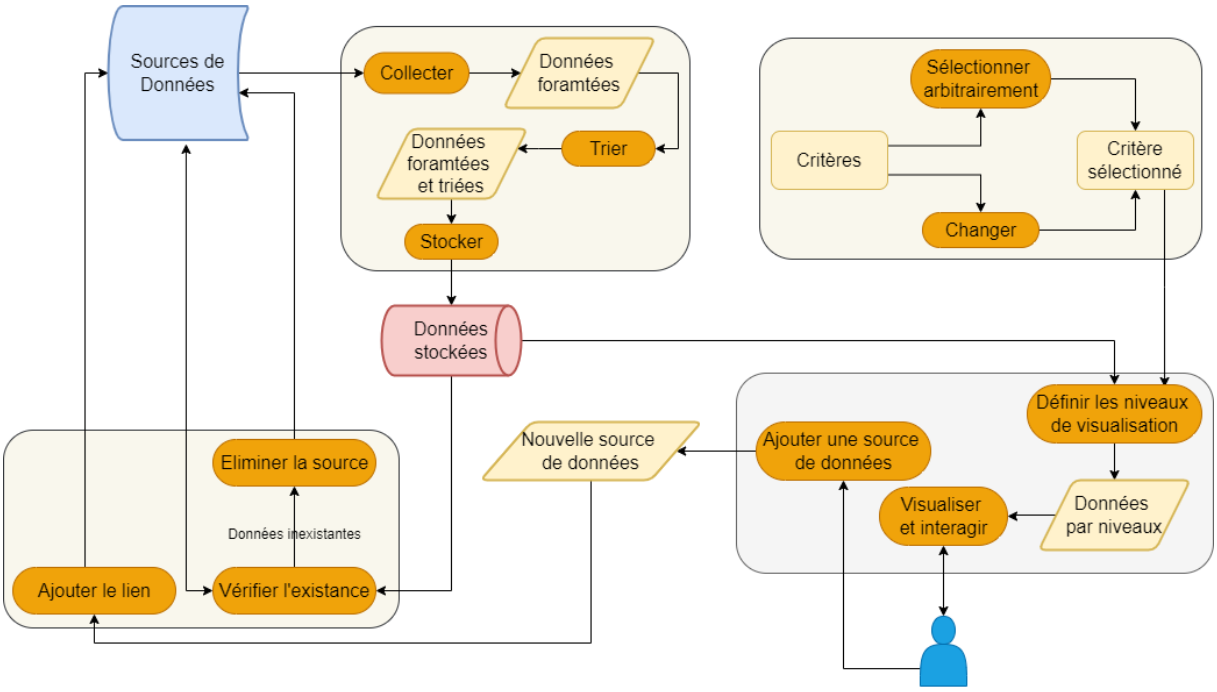

Figure 5-5: Architecture pour la visualisation munti-niveaux (Kahil et al., 2019)

4.4.1 Module de sérialisation et de stockage

Après la collecte des données depuis les différentes sources à savoirs les réseaux sociaux, les sites web, les forums, etc., via un des outils dédiés à cette fin tels que le framework Scrapy, le service web Data Miner et la plateforme Parse-Hub, et les sérialiser, elles sont triées selon un critère défini par défaut. Après quoi elles sont stockées selon une stratégie de stockage assignée telle qu'une base de données NoSQL.

4.4.2 Module de gestion des critères

Les critères permettent de définir l'ordre à suivre lors de la présentation des données. L'ensemble des critères est défini selon le domaine d'application. Il suit une stratégie précise telle que la pertinence des données, leur récence, les statistiques relatives aux données comme le nombre de vues d'une vidéo, …

4.4.3 Module de visualisation Interactive

Ce module assure deux tâches : la définition des niveaux de visualisation hiérarchique des données et leur présentation visuelle en réponse à l'interaction de l'utilisateur avec cette visualisation. C'est à ce niveau que sont définies les règles de hiérarchie : les relations entre les niveaux sont déterminées après la définition des niveaux de visualisation, ainsi que les patterns qui les composent comme indiqué dans **Algorithme 5-3**. Après cela, les patterns sont visualisés via un outil spécialisé en spécifiant le nombre de niveaux à afficher. Dans ce module, l'interaction avec l'utilisateur comprend : (1) l'ajout des sources de données en les envoyant au module de mise à jour, (2) la génération des requêtes pour modifier les critères de visualisation, (3) changer la visualisation en réponse aux requêtes utilisateur de personnalisation des options de visualisation telles que le zoom et la sélection et (4) la génération des requêtes de recherche et la visualisation de leurs résultats.

Algorithme 5-3 : Construction des Niveaux de Visualisation (Kahil et al., 2019)

Entrées :
$dataset$: le data-set contenant les données standardisées
Sortie :
$pattern\_levels$: l'ensembles des niveaux de la visualisation hiérarchique

Début
  Pour $pattern$ dans $dataset$ **faire**
    Si $pattern$ a sous-patterns **alors**
      Créer un nouveau niveau pl $(pattern - sous\_pattern)$
      $pattern_level$s= $pattern\_levels \cup \{pl\}$
    Fin Si
  Fin Pour
  Renvoyer $pattern\_levels$
Fin

4.4.4 Module de mise à jour des données

Ce module est chargé de deux tâches essentielles : (1) la vérification de la fraîcheur des données : Comme indiqué dans **Algorithme 5-4**, afin de vérifier si les données relatives aux patterns visualisés existent toujours, il récupère les patterns à partir des données formatées et triées, génère pour chacun de ces patterns une requête et l'exécute sur les sources de données. Si un pattern n'existe plus (obsolète), une requête de suppression est générée et exécutée sur les sources des données auquel elles sont relatives. (2) Ajout de données : Si de nouvelles sources sont à ajouter (à travers le module de visualisation), une requête d'ajout est générée au niveau de ce module pour les ajouter à l'ensemble des sources des données. Après un des deux cas, ces données sont mises à jour. Le formatage des données ainsi que la visualisation doivent donc être refaits en réexécutant les deux modules qui leur sont associés (modules 1 et 3).

Algorithme 5-4 : Mise à jour des sources de données (Kahil et al., 2019)

Entrées :
$dataset$: le data-set contenant les données standardisées
ds : l'ensemble des sources de données
Sortie :
C : l'ensembles des communautés détectées

Début
  Pour chaque pattern dans $dataset$ **faire** :
    Générer une requête de recherche Q sur pattern
    Exécuter Q dans ds
    Si pas de résultat **alors**
      Eliminer pattern depuis $dataset$
    Fin Si
  Fin Pour
Fin

Pour appliquer cette architecture sur la visualisation des communautés, le module de gestion des critères et le module de visualisation sont directement employés pour gérer les critères liés aux communautés et visualiser ces dernières en assurant l'interactivité avec l'utilisateur.

5 Expérimentation

Pour évaluer l'approche proposée, cinq data-sets qui représentent des graphes ont été sélectionné : (1) artist-edges et (2) new-sites qui ont été proposés par (Rozemberczki et al., 2019), (3) MathSciNet et (4) DBLP (Rossi & Ahmed, 2015), ainsi qu'un data-set étendu depuis DBLP pour introduire plus qu'une propriété sur le graphe. Ces data-sets sont tous formatés en csv. **Table 5-1** montre leurs statistiques, à savoir le nombre de nœuds, les nombre des arcs et le nombre de propriétés dont chaque data-set dispose.

Table 5-1 : Statistiques des data-sets (Kahil et al., 2021b)

Data-set	Nombre de nœuds	Nombre d'arcs	Nombre de propriétés
Artist_edges	50515	819306	1
New_sites	27917	206259	1
MathSciNet	311284	820644	1
DBLP	317080	1049866	1
DBLP étendu	317080	1049866	4

Apache Spark a été choisi pour l'implémentation de l'approche proposée. La raison est ce framework offre le traitement parallèle pour accélérer le traitement. En outre, il dispose d'une libraire appelée GraphX qui, comme mentionné dans chapitre **1**, fournit un mécanisme puissant qui assure le traitement des graphes en se basant sur une structure appelée *GraphFrame* (Andersen & Zukunft, 2016; Dave et al., 2016). Afin d'évaluer l'approche proposée en termes de temps d'exécution et de nombre de communautés détectées, elle est comparée avec l'algorithme LPA standard dont l'implémentation est offerte par la librairie python *networkx*.

Table 5-2 : Statistiques des communautés détectées (Kahil et al., 2021b)

Data-set	Nombre d'itérations LPA	Nombre de communautés		Temps d'exécution	
		LPA standard	Méthode proposée	LPA standard	Méthode proposée
Artist_edges	55	531	352	21.916	199.528
New_sites	55	972	986	7.400	118.203
MathSciNet	55	46933	48418	1616.033	313.164
DBLP	55	43185	39595	1423.490	333.995
DBLP étendu	55	186120	184004	3637.127	548.377

Les statistiques des résultats sont montrées dans **Table 5-2** conformément au nombre d'itérations qui est fixé à 55. Le nombre de communautés détectées par chaque algorithme est également visualisé dans **Figure 5-6**, **Figure 5-7**, **Figure 5-8**, **Figure 5-9** et **Figure 5-10** (Kahil et al., 2021b). A partir de ces figures, le nombre de communautés détectées se stabilise à partir d'environ la dixième itération pour tous les data-sets. Cela peut être expliqué par la relativité du processus de détection des communautés à la complexité du graphe.

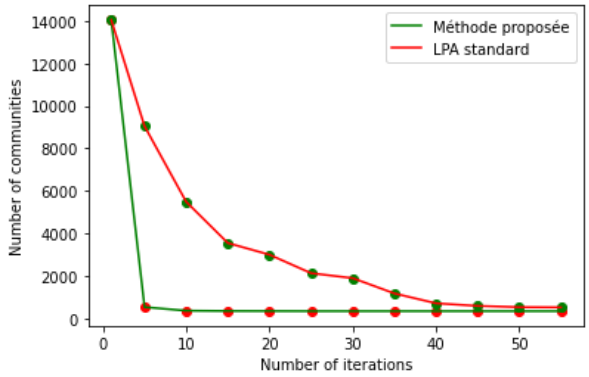

Figure 5-6 : Nombre de communautés détectées pour le data-set Artist_edges (Kahil et al., 2021b)

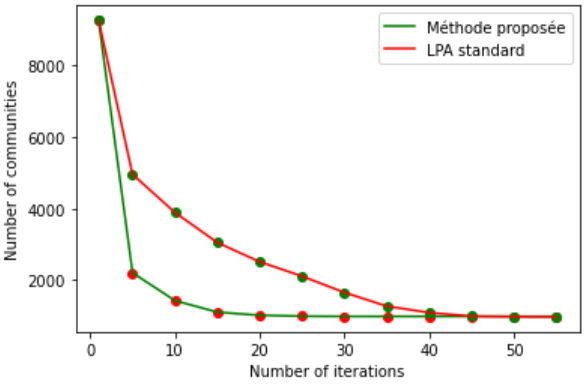

Figure 5-7 : Nombre de communautés détectées pour le data-set News_sites (Kahil et al., 2021b)

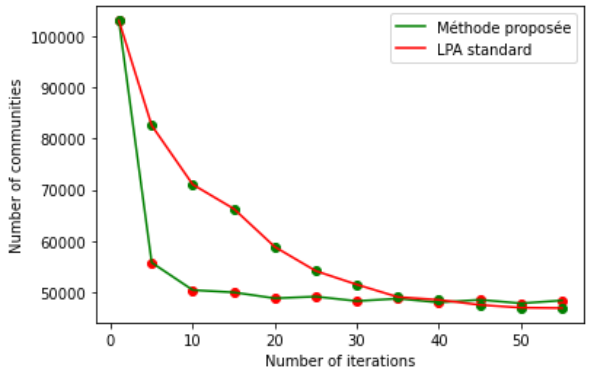

Figure 5-8 : Nombre de communautés détectées pour le data-set MathSciNet (Kahil et al., 2021b)

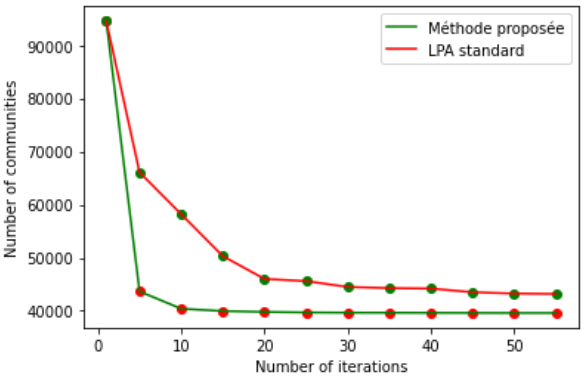

Figure 5-9 : Nombre de communautés détectées pour le data-set DBLP (Kahil et al., 2021b)

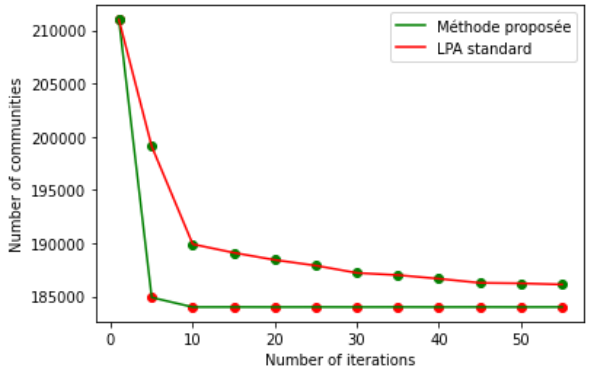

Figure 5-10 : Nombre de communautés détectées pour le data-set DBLP étendu (Kahil et al., 2021b)

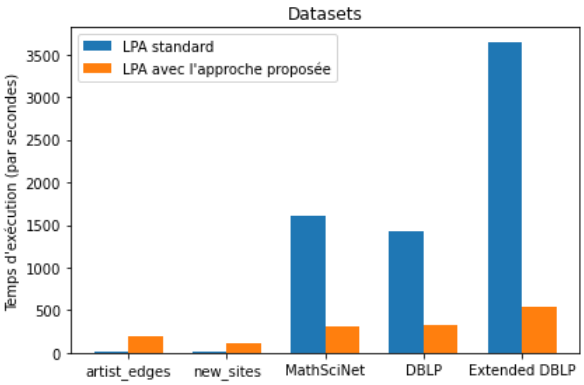

Figure 5-11 : Temps d'exécution du programme LPA standard et du programme de l'approche proposée (Kahil et al., 2021b)

D'autre part, selon **Figure 5-11**, le temps de détection des communautés est relativement inférieur via l'algorithme standard LPA que via l'approche proposée pour les deux premier data-sets (artist-edges et new-sites), mais plus élevé pour les autres data-sets. Ces résultats sont justifiés par le fait que les deux premiers data-sets ont un nombre faible de nœuds et de connexions, tandis que les autres data-sets en ont des nombres importants. Cela affirme que :

1- La solution proposée prend un peu plus de temps pour diviser les graphes, y compris les non-complexes, et distribuer les tâches de détection sur les différentes machines, d'où son ralentissement avec les deux petits data-sets.
2- Après la phase de division et de distribution, le processus de détection s'accélère.
3- La solution proposée est plus effective aves les graphes à grande échelle.

6 Conclusion

Ce chapitre a apporté une solution pour une visualisation interactive consistante des graphes à grande échelle sur la base de la détection des communautés. Cette solution est une combinaison de deux contributions complémentaires dont la première consiste à détecter les communautés dans les réseaux à grande échelle de manière distribuée. Pour cela, elle repose sur le framework Spark qui, d'une part, assure la distributivité à travers son architecture et, d'autre part, dispose d'une librairie dédiée au traitement des graphes qui implémente de multiples algorithmes de manipulation, y compris LPA qui a été sélectionné pour la détection des communautés dans l'approche proposée. Ce processus permet de diviser les graphes en communautés moins denses et facilite, par conséquent, leur visualisation exploratoire. La seconde contribution traite de la visualisation des communautés détectées. Pour cela, une architecture a été proposée pour présenter les graphes et offrir à l'utilisateur un mécanisme d'interactivité à travers lequel il peut personnaliser les critères et, par conséquent, la visualisation selon ses intérêts. Ainsi, l'exploration est moins brouillée et plus effective. En effet, l'architecture de visualisation proposée n'est pas exclusive aux graphes ; elle peut être utilisée pour collecter les données depuis différentes sources, les standardiser et les visualiser hiérarchiquement sur la base des critères qui varient selon l'application. Les modules qui la composent se chargent chacun d'une mission, à savoir la collecte et la sérialisation des données, la mise à jour de ces dernières, la gestion des critères, la visualisation des données et la personnalisation de cette visualisation via l'interaction avec l'utilisateur.

Chapitre 6: Considération du problème d’exploration visuelle comme un problème de recommandation

1 Introduction

L'exploration basée sur la visualisation et l'analytique visuelle peut devenir plus efficace si elle est présentée selon l'orientation de chaque utilisateur. Cette orientation peut être exprimée explicitement ou implicitement. Une façon d'améliorer l'exploration visuelle afin de l'adapter aux besoins et aux intérêts des utilisateurs est d'utiliser les systèmes de recommandation (RS : Recommender Systems). En effet, le but de ce concept est de filtrer les informations pertinentes (Quijano-Sánchez et al., 2020), c'est-à-dire de sélectionner parmi de nombreux éléments les plus pertinents pour chaque utilisateur (S. Zhang et al., 2019). Dans ce contexte, RS présente une solution au fameux problème du big data, à savoir la valeur (Tsai et al., 2015). RS est rapidement devenu populaire dans différents domaines tels que le stockage en ligne des items multimédias, les achats, le tourisme, les bibliothèques, la musique, les films, la recherche d'emploi, etc. (Bobadilla et al., 2013; B.Thorat et al., 2015; Kunaver & Požrl, 2017; Park et al., 2012; Singhal et al., 2017; Yera & Martínez, 2017). De nos jours, presque tous les cas réels de RS peuvent être traités en utilisant, en plus des techniques classiques, l'apprentissage profond (Cheng et al., 2016; S. Zhang et al., 2019). L'objectif de cette contribution est de développer une approche basée sur RS pour améliorer la visualisation des données volumineuses, tout en considérant les données hétérogènes qui peuvent être structurées, non structurées ou semi-structurées (Kahil et al., 2020), et l'aspect interaction utilisateur (Kahil et al., 2019) qui doit être assuré par la visualisation. Pour cela, la présente proposition apporte une contribution (Kahil et al., In press) qui consiste en trois mécanismes alternatifs pour fournir des recommandations aux utilisateurs, à savoir : les mécanismes basés sur la factorisation matricielle (MF : Matrix Factorization), basés sur les moindres carrés alternés (ALS) et sur les réseaux de neurones profonds (DNN), sous une architecture proposée pour assurer le profilage des utilisateurs. Une comparaison des résultats est présentée afin de montrer la meilleure technique à utiliser pour le problème de visualisation de données volumineuses. De même, une autre comparaison est opérée avec les solutions de l'état de l'art pour montrer l'efficacité des deuxième et troisième solutions proposées.

Le reste de ce chapitre est organisé comme suit : La deuxième section présente un aperçu sur les RS et les différentes techniques utilisées pour résoudre les problèmes qui leur sont liés. La troisième section décrit l'approche proposée avec ses trois solutions alternatives. La quatrième section comprend la mise en œuvre de ces solutions, suivie d'une comparaison avec des travaux connexes. La cinquième section conclut ce chapitre.

2 Systèmes de recommandation : aperçu et état de l'art

Bien qu'il ne s'agisse pas d'un sujet nouveau, les RS attirent encore l'attention dans le milieu scientifique et engendrent des défis jusqu'à nos jours. Cela est dû à la diffusion de leur application sur plusieurs domaines à l'ère du big data. Parmi les exemples concrets bien connus figurent YouTube (S. Zhang et al., 2019) qui recommande des vidéos, Google Play (Cheng et al., 2016) pour recommander des applications et des jeux aux utilisateurs d'Android, et Netflix (Gomez-Uribe & Hunt, 2015; Y. Zhou et al., 2008), le fameux système qui produit et recommande les séries et les films. Les enjeux du RS, qui sont fondamentaux pour les rendre plus efficaces, peuvent être résumés en (Verma et al., 2015) :

1- transparence et justification
2- contrôle des utilisateurs sur le RS
3- manque de diversité
4- problèmes de démarrage à froid
5- acquisition et représentation des informations contextuelles

Chacun de ces défis sera examiné quand l'approche proposée abordera les différents problèmes. Le processus RS est divisé, après la recherche d'information (Cheng et al., 2016), en deux tâches (Yera & Martínez, 2017) :

1- la prédiction qui consiste à estimer la note de l'item de l'utilisateur
2- la recommandation qui, basée sur la prédiction, sélectionne les meilleurs items qui intéressent chaque utilisateur

Les informations peuvent être récupérées à partir de nombreuses sources. Ces dernières peuvent être explicites telles que la collecte des évaluations des utilisateurs et des classements d'articles, ou implicites comme la supervision des comportements des utilisateurs ou des articles consultés par des utilisateurs spécifiques. La recommandation visuelle assure la transparence aux utilisateurs, un enjeu important de RS (Verma et al., 2015). Cela garantit que ces derniers fonctionnent en mode boîte noire. Cependant, ils peuvent être développés en utilisant des techniques de boîte noire ou de boîte blanche. Quelques exemples des deux types sont présentés après la description des techniques de filtrage RS existantes.

Les deux techniques classiques de filtrage de RS sont (Park et al., 2012; Verma et al., 2015) : le filtrage basé sur le contenu et le filtrage collaboratif. Le premier consiste à recommander à chaque utilisateur les éléments similaires à ceux qu'il a déjà "appréciés" (très bien notés). Les mesures de similarité sont définies à l'aide de plusieurs techniques dont les plus couramment utilisées sont le produit scalaire et la technique de similarité cosinus (Cremonesi et al., 2011), qui, pour deux items a et b sont définis par les formules (6.1) et (6.2).

$$S(\overrightarrow{a_i}, \overrightarrow{b_i}) = \sum_i a_i b_i \qquad (6.1)$$

$$S(\vec{a}, \vec{b}) = \frac{\sum_i a_i b_i}{|\vec{a}| \times |\vec{b}|} \qquad (6.2)$$

Le filtrage basé sur le contenu nécessite essentiellement deux matrices : la matrice des caractéristiques des éléments et la matrice d'évaluation des éléments par l'utilisateur. La première matrice contient toutes les caractéristiques des items, tandis que la seconde contient les évaluations des éléments pour chaque utilisateur. Le filtrage collaboratif est basé sur les évaluations de multiples utilisateurs. Il consiste à définir chaque utilisateur et chaque élément par des point d-dimensionnel et à les représenter dans l'espace latent. Ce type de filtrage peut être défini comme une matrice d'évaluation $n \times m$, où n représente le nombre d'utilisateurs et m représente le nombre d'éléments. Ensuite, le résultat de la recommandation est n matrices pondérées. Cela engendre un produit cartésien massif dans le cas d'un très grand nombre d'utilisateurs et d'éléments. Ainsi, la matrice utilisateurs-éléments devient très clairsemée (sparse matrix). Une solution à ce problème consiste à utiliser MF. Le filtrage hybride était simplement basé sur ces deux algorithmes. Cependant de nos jours, le filtrage contextuel (context-aware filtering) (Bobadilla et al., 2013) y est ajouté pour améliorer le processus de recommandation. Comme le montre **Figure 6-1**, ce type de filtrage n'est pas basé sur l'évaluation mais utilise plutôt des informations locales et personnelles provenant de différentes sources telles que le temps disponible (Verbert et al., 2012; Yujie & Licai, 2010), information démographiques (âge de l'utilisateur, nationalité, sexe, etc.), informations sociales (abonnés et publications), localisation GPS, signaux en temps réel dans le domaine de santé, et autres informations liées à l'internet des objets... Sachant que le filtrage contextuel peut être pré-filtrage ou post-filtrage, le choix de celui qui convient diffère d'un domaine d'application à l'autre. Dans tous les cas, la sortie de tout RS est une matrice de notation qui contient les scores des combinaisons utilisateur-item.

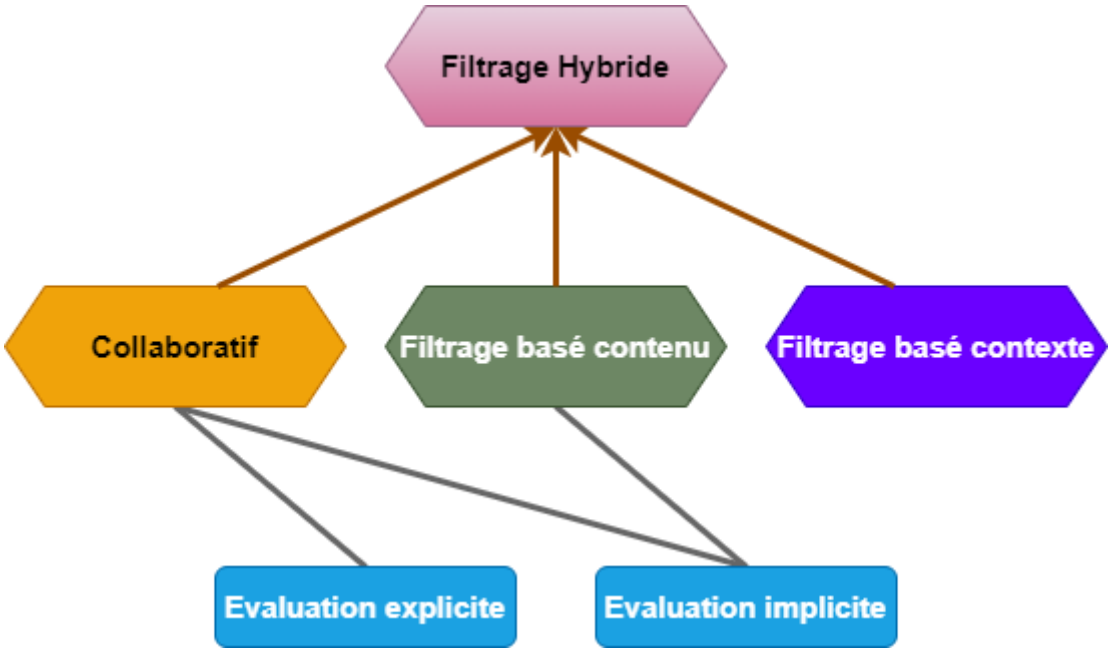

Figure 6-1 : Types de filtrage et techniques de collecte des feedbacks pour chaque type

Table 6-1 résume les principes, les algorithmes utilisés, les avantages et les inconvénients de chaque type (Ajesh et al., 2016; B.Thorat et al., 2015). Les DNN sont les mécanismes de boîte noire les plus connus pour développer RS. Alors que parmi les techniques de boîte blanche, il peut y avoir : les DT, KNN, k-means clustering, MF (Bobadilla et al., 2013; Kunaver & Požrl, 2017; S. Zhang et al., 2019), et l'exploration des règles d'association (Schafer et al., 2007; Shaw et al., 2010; Sobhanam & Mariappan, 2013).

Table 6-1 : Comparaison des techniques de filtrage (Kahil et al., In press)

	Filtrage basé contenu	Filtrage collaborative	Filtrage basé context
Algorithmes utilisés	- Corrélation - Classifieurs linéaires - Méthodes probabiliste	- ALS - Weighted ALS (WALS) - KNN - PCA - SVD - Latest Semantic Index (LSI) - Règles d'association	- Base sur les préférences de l'utilisateur - Supervision du comportement utilisateur
Avantages	- Aucune nécessité des données concernant l'utilisaeur - Pas de problème de démarrage à froid - Les articles de niche sont recommandés	- Users and items' similarities are used simultaneously - No domain knowledge is required (it can learn outside the user's interests) - Easier implementation of the recommendation	- No interaction data needed - High-fidelity data from user's self-reporting
Inconvénients	- La connaissance du domaine est requise. - La proposition de nouveaux domaines aux utilisateurs est impossible	- Problème de démarrage à froid dans les cas de nouveaux utilisateurs et de nouveaux contenus - Sparsité	- Les données utilisateur sont requises - Problèmes de confidentialité

Types d'évaluation	Seulement l'évaluation implicite	Evaluation implicite et explicite	Pas d'évaluation

Le reste de cette section liste certaines solutions pertinentes d'état de l'art qui répondent à plusieurs problèmes de recommandation. (Kbaier et al., 2017) ont proposé une nouvelle technique basée sur l'arbre de décision pour gérer le processus de recommandation hybride en utilisant un nouveau critère de division pour réduire la complexité de la construction de l'arbre. (Subramaniyaswamy & Logesh, 2017) ont proposé un framework adaptatif basé sur un algorithme KNN pour fournir aux utilisateurs des recommandations top-N basées sur un filtrage collaboratif en fonction de leurs différentes classes. En plus de l'algorithme KNN, ce framework utilise sémantiquement des données historiques afin de fournir aux anciens utilisateurs des données qui leur sont pertinentes. (H.-R. Zhang & Min, 2016) ont combiné la forêt aléatoire avec un support de décision à trois voies (Tree-way decision support) pour développer un framework dont le but est de minimiser le coût du processus de recommandation en termes de classification des utilisateurs. (Xue et al., 2017) ont proposé un modèle DNN pour gérer la MF afin de sélectionner les top-N recommandations. De même, (Dziugaite & Roy, 2015) ont développé un framework appelé Neural Network Matrix Factorization (NNMF) sur la base de DNN pour fournir une recommandation basée sur le filtrage collaboratif en utilisant une MF. (Paradarami et al., 2017) ont proposé un modèle basé sur DNN pour prédire la recommandation basée à la fois sur le filtrage basé contenu et sur le filtrage collaboratif. (P. Li et al., 2017) ont proposé un modèle basé sur un réseau de neurones (NN) pour gérer le processus de recommandation à l'aide d'un filtrage collaboratif. Ce modèle est composé de deux tâches : la première, nommée Neural Rating Regression (NRR) vise à prédire les évaluations en se basant sur la régression, tandis que la seconde génère des astuces abstraites sur les items en fonction des interactions d'évaluation utilisateur-item. (Cremonesi et al., 2011) ont proposé un modèle appelé Neural Factorization Machines (NFM) pour résoudre le problème de matrice clairsemée (Sparcity) lié aux variables discrètes qui caractérisent aussi bien les attributs des utilisateurs et que des items à l'aide de la régression. (Juan et al., 2016) ont développé l'approche de machines factorielles basées sur le champ (Field-aware Facorization Machine) (FFM) a été développée pour gérer la prédiction selon le taux de clics (CTR : Click-Through Rate) à l'aide d'une méthode de gradient stochastique optimisé. (Guo et al., 2017) ont développé un modèle nommé deep factorisation machine (DeepFM) pour résoudre le problème de CTR afin de calculer la probabilité qu'un item soit choisi par un utilisateur donné. Il utilise un réseau de neurones feed-forward (FFN) pour apprendre des interactions utilisateur-item de la machine de factorisation (FM). (Cheng et al., 2016) ont développé un modèle appelé apprentissage large et profond (WDL) pour améliorer le processus de recommandation, en particulier pour résoudre le problème de matrice clairsemée, en combinant un modèle linéaire avec DNN. (Lian et al., 2018) ont développé un modèle appelé XDeepFM en utilisant une combinaison de FFN et d'un réseau d'interaction compressé (CIN : compressed interaction network) afin d'améliorer la recommandation en se basant sur des évaluations implicites et explicites. (Rendle, 2010) a combiné un modèle SVM avec des FM clairsemées afin d'évaluer les items dans les situations de sparsity. (R. Wang et al., 2017) ont développé un modèle profond et inter-réseaux (DCN) via un modèle FFN et un modèle de régression logistique distribuée afin d'automatiser l'extraction des caractéristiques du CTR et de prédire les évaluations sur les items. (Shambour, 2021), a utilisé une architecture d'auto-encodeur profond pour proposer un nouvel algorithme nommé AEMC qui considère les recommandations multicritères. Cet algorithme consiste essentiellement à appliquer le principe de l'auto-encodeur sur la matrice utilisateur-item, c-à-d. à chaque itération, encoder les évaluations de chaque utilisateur par un vecteur et décoder ce dernier par un autre vecteur avec la même dimension. (Ayundhita et al., 2019) ont

développé un système basé sur une ontologie pour recommander des ordinateurs portables en fonction de leurs spécifications ainsi que des besoins des utilisateurs. Il représente les spécifications des produits par des exigences fonctionnelles à travers les relations sémantiques de l'ontologie afin de cibler essentiellement les utilisateurs qui ne sont pas familiers avec ces spécifications. Cela a donné une meilleure précision en termes d'ordinateurs portables recommandés par rapport aux recommandations du commerce électronique. (Fang et al., 2020) ont proposé un nouveau modèle basé sur MF appelé Fusion Probability Matrix Factorization (FPMF) pour remédier à la rareté des données dans le filtrage collaboratif. Ce modèle utilise la similarité multifactorielle utilisateur-item qui est basée sur : (1) la similarité linéaire qui est liée à la corrélation entre les utilisateurs et (2) la similarité comportementale extrême qui traite les évaluations clairsemées. Le résultat de la similarité multifactorielle permet de construire la matrice voisine qui est fusionnée avec la matrice d'évaluations originale pour construire la matrice de prédiction. (Xin et al., 2020) ont proposé deux frameworks nommés Self-Supervised Q-learning (SQN) et Self-Supervised Actor-critic (SAC) qui utilisent l'apprentissage par renforcement pour améliorer les tâches de recommandation. Ces frameworks résolvent principalement les problèmes suivants : le manque d'évaluations négatives qui sont souvent implicites dans de nombreux cas, le manque de données pour entrainer efficacement les modèles supervisés et l'apprentissage de RS sans interaction avec les utilisateurs. Dans (K. Zhou et al., 2020) ont proposé une approche qui utilise des graphes de connaissances (KG : Knowledge Graphs) pour représenter sémantiquement les composants RS conversationnels à savoir : les items et les textes. Cette approche améliore les conversations multi-tours (dialogues) avec les utilisateurs afin de leur recommander les meilleurs éléments qui les intéressent. Ceci est réalisé en alignant les espaces sémantiques des éléments et des textes après avoir amélioré leur représentation grâce à KG. (Margaris et al., 2020) ont proposé un algorithme pour enrichir la recommandation par filtrage collaboratif. Son principe est d'associer effectivement à chaque avis textuel une note chiffrée qui servira à générer des recommandations. Il peut être appliqué à des systèmes qui utilisent des avis textuels comme mécanisme d'évaluation tels que les réseaux sociaux et certaines plateformes d'e-commerce.

3 Approche proposée dans l'assistance de l'utilisateur via la recommandation pour l'exploration des données

En considérant le problème de l'exploration et de la visualisation du big data comme un problème de recommandation, l'objectif est d'associer à chaque utilisateur l'ensemble de patterns qu'il est censé avoir besoin d'explorer. Cependant, contrairement à RS, les évaluations explicites ne sont pas la principale base sur laquelle repose la visualisation des données ; les utilisateurs n'ont pas nécessairement à évaluer explicitement les patterns afin de les explorer. Par conséquent, les évaluations doivent être définies implicitement et une stratégie doit être employée à cette fin. Cette stratégie peut se résumer dans les points ci-dessous (Kahil et al., In press).

1- Collecte et stockage des données qui concernent les attributs ou propriétés des articles (genre, domaine, prix, etc.), les comportements de l'utilisateur (évaluations, items visités, achats, types d'articles qui l'intéressent, etc.), les données démographiques des utilisateurs (âge, lieu, sexe, éducation, etc.).
2- Le filtrage des données qui consiste à sélectionner uniquement les attributs utiles. A ce stade, ces attributs sont sélectionnés en fonction du domaine d'application de l'utilisateur. C'est pourquoi un expert du domaine pourrait être requis dans cette phase.
3- Extraction des informations relatives aux interactions utilisateur-item.
4- Construction et évaluation du modèle de recommandation.

Le choix de la technique de collecte de données est basé sur les feedbacks des utilisateurs qui peuvent être implicites ou explicites (Schafer et al., 2007) selon le domaine d'application. Dans les deux cas, les mécanismes utilisés pour la collecte des données doivent être fiables vis-à-vis de la technique choisie et la source de feedbacks. Les évaluations explicites ne sont pas nécessaires dans la visualisation. La collecte de données doit alors adopter une stratégie de définition des évaluations implicites afin de recommander efficacement les items aux utilisateurs de manière unifiée selon le profil de chacun. Trois types de ce profil sont distingués. Ils sont expliqués dans les points ci-dessous et résumés dans **Figure 6-2** et **Figure 6-3** :

1- Si l'utilisateur est inconnu, c-à-d. il n'y a pas d'informations sur l'utilisateur.
2- Si l'utilisateur est inconnu mais accepte de donner des informations sur ses préférences.
3- Si l'utilisateur fait partie du système de visualisation, c-à-d. il est enregistré dans la base de données du système.

Pour chacun de ces cas, une technique est choisie pour remplir la matrice d'évaluation. Dans le premier cas, les patterns de données sont présentés selon leur critère de pertinence par défaut dans l'ensemble du système. Il y a différents exemples qui illustrent ce cas tels que les items les plus consultés, ceux liés aux mots-clefs les plus recherchés, les sujets récents, etc. Les items ici sont filtrés en fonction des informations contextuelles. Ces dernières peuvent être collectées via les méthodes citées dans **Table 6-1**. Les étapes couvertes pour traiter ce cas sont résumées dans les points suivants :

- Définir le critère par défaut.
- Associer à l'utilisateur toutes les évaluations des items selon le critère défini de manière à ce que la valeur la plus élevée du critère corresponde à la valeur de la note d'évaluation la plus élevée, idem pour la valeur la plus basse.
- Selon le comportement de l'utilisateur, changer le critère par défaut : le filtrage effectué à ce moment par lui, les données liées à sa recherche, etc.

Le deuxième cas peut faire référence au filtrage contextuel ; les informations sont collectées de différentes manières telles que les sondages, la fourniture d'informations sur les orientations, etc. Sur la base de ces informations, la matrice d'évaluation est remplie. Dans le troisième cas, les informations sont collectées via différentes techniques qui ciblent le comportement de l'utilisateur telles que la durée de visionnage de la vidéo, les caractéristiques d'impression (Cheng et al., 2016), les statistiques sur l'historique d'accès aux données, etc. Deux matrices sont utilisées afin d'effectuer les différentes opérations :

1- matrice d'items ($items \times attributs$)
2- matrice d'interaction utilisateur-élément ($utilisateurs \times items$).

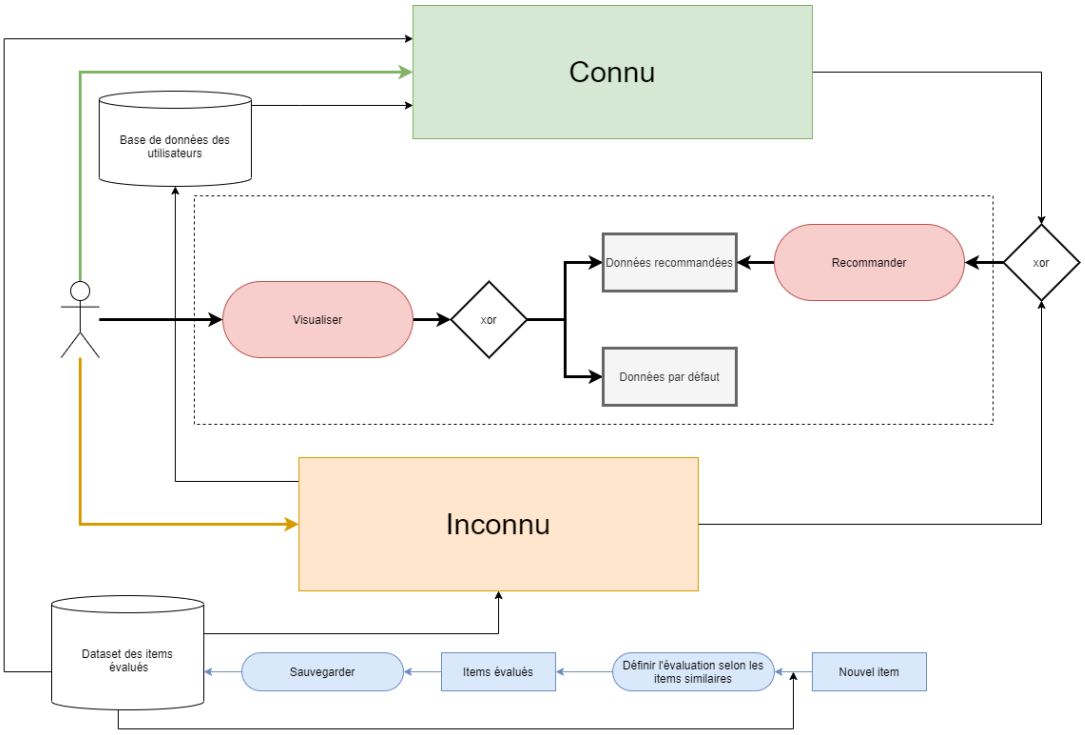

Figure 6-2 : Architecture abstraite de l'approche proposée (Kahil et al., In press)

Dans la tâche de recommandation, les patterns de données sont considérés en respectant les contraintes suivantes :

- Masquer les items qui ont déjà été complètement explorés par l'utilisateur.
- Considérer les défis cités auparavant.

Le processus de recommandation repose sur les données présentées ainsi que les scénarios liés à l'utilisateur ciblé selon les cas indiqués ci-dessus. À cette fin, l'approche proposée est basée sur la MF pour garantir la gestion de la matrice clairsemée (sparsity), la recommandation de données de manière transparente et la prise en compte des autres défis mentionnés ci-dessus. Cette approche utilise trois techniques alternatives :

1- MF basée sur une méthode brute intuitive
2- Une méthode de factorisation basée sur la recommandation via ALS
3- Une régression basée sur un modèle DNN.

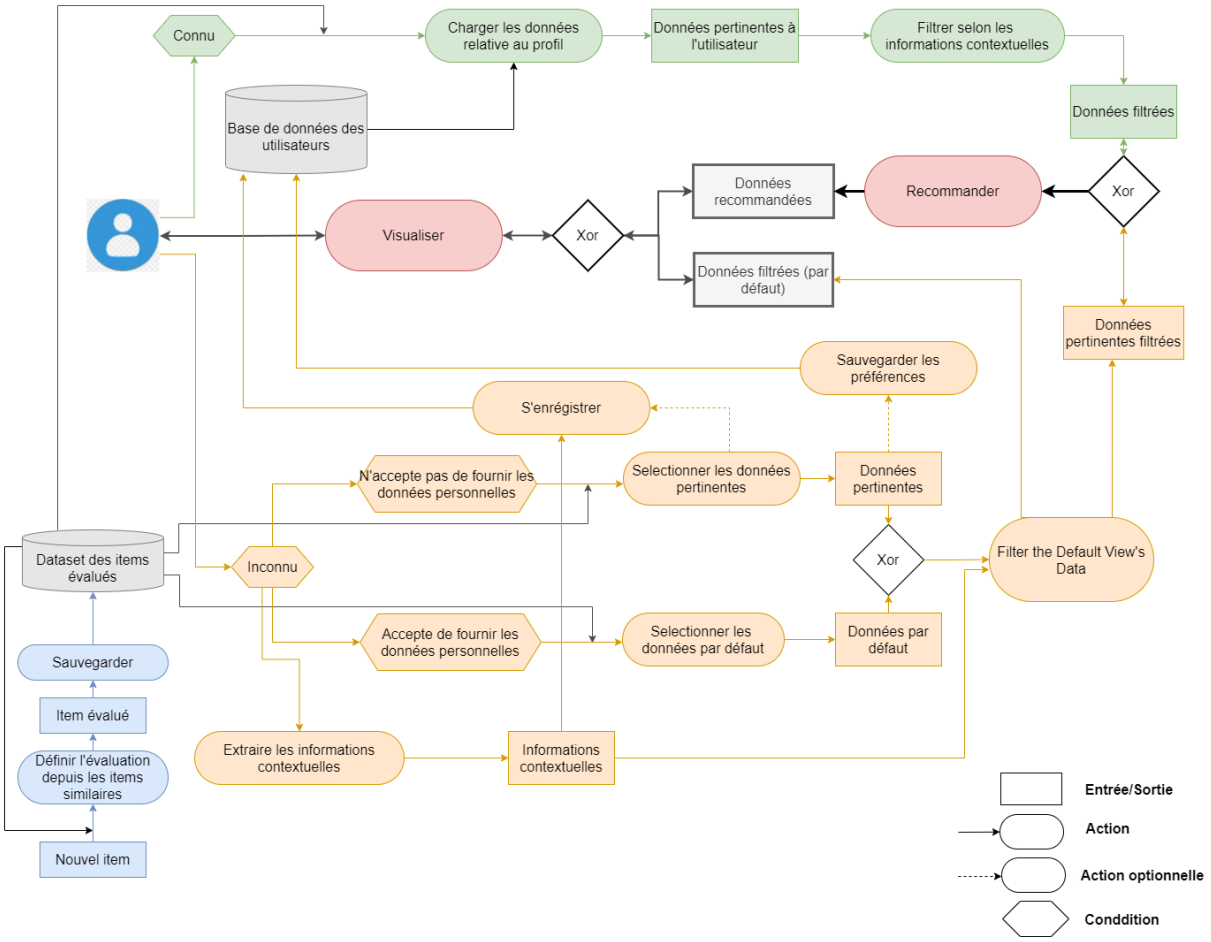

Figure 6-3 : Sélection et recommandation selon le profil utilisateur (Kahil et al., In press)

3.1 Méthode brute

La solution classique intuitive proposée utilise la recommandation basée sur le contenu. Elle emploie la matrice d'évaluation définie précédemment. Dans le cas présent, pour n utilisateurs avec u_a attributs factoriels, et m items avec i_a attributs factoriels, cette matrice $r(k \times l)$ est définie comme suit :

- Le nombre d'attributs factoriels est inférieur ou égal au nombre d'attributs aussi bien pour les utilisateurs que les items.
- $k = n \times m$ (nombre de lignes de r) est le nombre de toutes les interactions entre les utilisateurs et les éléments.
- $l \geq u_a + i_a + 1$ (nombre de colonnes de r) est le nombre d'attributs factoriels des utilisateurs et des éléments. Certaines colonnes, étant non numériques, doivent soit être encodées selon une stratégie définie, soit en ajoutant de nouvelles colonnes numériques qui les représentent. La dernière colonne est la colonne des évaluations. Chacune de ces évaluations est calculée selon la formule (6.3).

$$r_{il} = \begin{cases} \sum_{j=0}^{nbr_{fct_{attr}}} w_j a_j & Si\ \exists a_x, a_y \in \{a_z, z \in [1,l]\}: a_x \Re a_y \\ 0 & Sinon \end{cases} \quad (6.3)$$

tel que :

  - $nbr\_fct\_attr \leq l$ est le nombre de colonnes factorielles (attributs) en excluant les colonnes numériques, d'utilisateurs et d'éléments.

- w_j est le poids assigné à l'attribut a_j qui doit être défini par un expert de domaine.

- Pour tous les utilisateurs, les lignes qui leur sont associées sont triées en fonction de la colonne des évaluations par ordre décroissant.
- Dans le cas d'un nouvel utilisateur, toutes les interactions avec les éléments sont ajoutées à la matrice et calculent les évaluations selon la formule (6.3).

3.2 ALS

Cette solution est basée sur le filtrage collaboratif. Elle utilise, elle-aussi, MF pour effectuer le processus d'évaluation, mais à partir d'une matrice d'évaluation non complète. Sa réalisation implique l'utilisation de l'algorithme ALS. Ce dernier est choisi plutôt que le Gradient Stochastique (SGD : Stochastic Gradient Descent) après les avoir comparés selon différents critères tels que le parallélisme et la vitesse de traitement. La comparaison est présentée dans **Table 6-2**.

Table 6-2 : Avantages et inconvénients d'ALS et SGD (Kahil et al., In press)

	ALS	SGD
Avantages	- Parallèle - Rapide - Supporte facilement les paires d'interaction non-observées	- Parallèle - Flexible
Inconvénients	- Utilisé uniquement pour les moins carrés	- Difficile pour supporter les paires d'interaction non-observées

L'algorithme de moindres carrés alternés (ALS : Alternating Least Square) consiste simplement à fixer un utilisateur précis pour lequel les recommandations doivent être trouvées, et vice-versa. Ce processus, exprimé par **Algorithme 6-1** (Aberger, 2014; Y. Zhou et al., 2008), peut se résumer en deux phases : la première consiste à initialiser la matrice d'évaluation selon une stratégie définie, telle que l'attribution des premières valeurs à la moyenne des évaluations, suivie de la deuxième phase qui comprend un processus répétitif qui fixe alternativement les utilisateurs et résout les items, et vice versa, en utilisant un algorithme de minimisation des erreurs tel que la technique de l'erreur quadratique moyenne (MSE : Mean Squared Error) et SGD, jusqu'à la convergence. La formule (6.4) est utilisée pour calculer l'évaluation.

$$z = x - \frac{\mu}{\sigma} \qquad (6.4)$$

tel que :

- z est la valeur d'évaluation
- μ est la moyenne
- σ est l'écart type.

Algorithme 6-1 : ALS standard (Aberger, 2014; Y. Zhou et al., 2008)

- Étape 1 Initialisez la matrice V en attribuant la note moyenne pour ce film comme première ligne et de petits nombres aléatoires pour les entrées restantes.
- Étape 2 Fixez V et résolvez U en minimisant la fonction RMSE.
- Étape 3 Fixez U, résolvez V en minimisant la fonction RMSE de la même manière.
- Étape 4 Répétez les étapes 2 et 3 jusqu'à convergence.

3.3 Réseaux de neurones

La méthode basée sur DNN est composée de deux tâches principales :

1- la présentation des données
2- la recommandation.

La première tâche consiste à définir une présentation unifiée des données d'entrée qui contiennent des colonnes factorielles d'utilisateur et d'élément (appelés données d'intégration dans le cas des DNN). Comme présenter l'intégralité des données est moins utile et plus compliqué lors de l'utilisation des DNN, en particulier lors de l'introduction de l'aspect contextuel, les filtrer est nécessaire pour conserver uniquement les attributs utiles qui seront utilisés pour la recommandation. Par exemple, un attribut d'images descriptives n'est pas nécessaire pour la recommandation. Pour cela, il faut définir les colonnes factorielles et les colonnes non factorielles. Le premier type est lié à la matrice d'évaluation, c'est-à-dire les colonnes qui contiennent les informations nécessaires à la recommandation. Dans ce cas : $user\_id$, $item\_id$, $user\_item\_rating$, ... Quant au second type, il concerne des informations facultatives telles que les noms scientifiques des produits. Ces informations n'affectent pas la qualité de la recommandation mais sont utilisées pour d'autres tâches comme la recherche et la correspondance des entités. Cette représentation peut être utilisée pour gérer le filtrage basé sur le contenu, collaboratif et contextuel. La deuxième tâche, chargée de la recommandation, est proposée sur la base d'un modèle DNN. L'idée est de prendre aléatoirement des utilisateurs avec la liste des items qu'ils ont déjà évalués, d'explorer les utilisateurs ayant des similitudes en termes d'évaluation et d'extraire les items qu'ils ont évalués le mieux, afin de prédire les prochains items susceptibles d'intéresser chacun d'eux. Étant donné que ce problème peut être formulé comme une prédiction de valeurs numériques d'évaluations, et afin d'éviter le problème de surapprentissage, la régression multiple est choisie. Chaque interaction utilisateur-item dans la MF représente l'évaluation que l'utilisateur est plus susceptible d'attribuer à cet item. Cette solution peut être assimilée à NRR proposé dans (P. Li et al., 2017), où la régression a été utilisée pour prédire les valeurs d'évaluations en fonction des interactions utilisateur-item existantes. Cependant, sa particularité est que toutes les colonnes factorielles pour les utilisateurs et les éléments sont incluses dans la matrice d'évaluation. Afin de considérer à la fois le filtrage collaboratif et le filtrage contextuel, les données d'intégration d'entrée propres à chacun selon les colonnes factorielles sont à définir. En se basant sur le produit scalaire, cela peut être traduit par la formule **(6.5)** responsable du premier filtrage et la formule **(6.6)** responsable du second.

$$r'(u,v) = f'\left(\sum_{i=0}^{nu'} uc_i' \times \sum_{j=0}^{nv'} vc_j' + ub' + vb'\right) \quad (6.5)$$

$$r''(u,v) = f''\left(r'(u,v) \times \left(\sum_{i=0}^{nu''} uc_i'' \times \sum_{j=0}^{nv''} vc_j''\right) + ub'' + vb''\right) \quad (6.6)$$

tel que :

- u est l'utilisateur
- v est l'item
- chaque uc_i' et vc_i' représente une colonne factorielle liée respectivement à l'utilisateur u et l'item v dans les couches du premier filtrage
- chaque uc_i'' et vc_i'' représente une colonne factorielle liée respectivement à l'utilisateur u et l'item v dans les couches du second filtrage
- ub' et vb' sont respectivement les biais des couches du premier filtrage
- ub'' et vb'' sont respectivement les biais des couches du second filtrage

- A supposer que, ufc et vfc sont respectivement les colonnes factorielles des utilisateurs et des items, que $ufc' = \{uc_i'\}$ et $vfc' = \{vc_i'\}$ sont respectivement les ensembles des colonnes factorielles des utilisateurs et des items dans le premier filtrage, $ufc'' = \{uc_i''\}$ et que $vfc'' = \{vc_i''\}$ sont respectivement les ensembles des colonnes factorielles des utilisateurs et des items dans le second filtrage, les contraintes relatives aux colonnes factorielles peuvent être résumées dans les points suivants :
  a. $ufc' \cap ufc'' = \emptyset$
  b. $|ufc'| + |ufc''| = |ufc|$
  c. $vfc' \cap vfc'' = \emptyset$
  d. $|vfc'| + |vfc''| = |vfc|$
- f' et f'' sont respectivement les fonctions d'activation des couches du premier filtrage et du second filtrage.

Les deux types de filtrage peuvent se précéder selon le cas d'application, puisque le filtrage contextuel peut être utilisé comme pré-filtrage ou post-filtrage. Dans le modèle proposé, le premier type a été choisi puisque l'implémentation du cas d'utilisation a été réalisée sur cette base. Lors du développement du modèle DNN, nous proposons d'utiliser plus d'un type de fonction d'activation afin de distinguer les couches responsables du filtrage collaboratif de celles responsables du filtrage sensible au contexte. Pour cela, $tanh$ (formule **(6.7)**) est choisie pour les couches du premier filtrage et la sigmoïde (formule **(6.8)**) pour le second.

$$\tanh(x) = \frac{\exp(2x-1)}{\exp(2x+1)} \quad (6.7)$$

$$sigmoid(x) = \frac{1}{1-\exp(-x)} \quad (6.8)$$

4 Expérimentation

Cette expérimentation comprend deux phases, à savoir (1) la génération de données par feedback implicite et (2) la recommandation. La présente section est divisée en trois sous-sections. La première présente la configuration expérimentale qui comprend la construction et le prétraitement du data-set en fonction du cas de chaque solution. La deuxième présente les mesures d'évaluation des résultats. La troisième présente les résultats des trois solutions alternatives, les discute et compare les résultats de la dernière solution avec les travaux connexes.

4.1 Configuration expérimentale

Afin d'évaluer l'approche proposée, elle a été appliquée sur le domaine académique. Plus précisément, le but désigné est de recommander à chaque utilisateur (universitaires et chercheurs) les revues et conférences qui l'intéressent. A cet effet, deux data-sets sont sélectionnés :

1- Le data-set principal qui est lié au classement des revues scientifiques (SJR). Il contient des informations sur les revues indexées SCOPUS, les conférences, les séries de livres et les revues spécialisées. Les informations comprennent l'index H, le nombre d'articles de chaque revue, le nombre de citations, etc.
2- Un data-set artificiel contenant 17 902 enregistrements d'utilisateurs, générés de manière aléatoire et contenant les attributs : identifiant de l'utilisateur, sexe, âge, domaine académique, langues et pays.

Pour considérer les trois solutions alternatives, la matrice d'évaluation est construite deux fois :

1- Une matrice complète pour la première solution, qui contient toutes les interactions utilisateur-item, c'est-à-dire qu'elle est construite via le produit cartésien des utilisateurs avec les items.
2- Une matrice partielle pour les deuxième et troisième solutions, qui ne contient que quelques interactions utilisateur-item.

Dans tous les cas, l'attribution des valeurs d'évaluation pour les interactions existantes se fait selon la stratégie proposée suivante : pour chaque utilisateur, si l'item (dans ce cas : article, conférence, série de livres ou revue spécialisée) avec lequel il interagit appartient au moins à une de ses catégories, attribuer la valeur d'évaluation à $\sum_{j=1}^{nbr\_fctr\_attr} w_j a_j$ (comme l'équation de la méthode brute), sinon lui attribuer la valeur 0. Avant de mettre en œuvre les trois solutions alternatives, le data-set principal doit être prétraité. Ce pré-traitement consiste principalement à supprimer les lignes ayant des informations manquantes ou fausses afin d'éviter les mauvaises recommandations. De même, la colonne $rank$ est définie comme la colonne id, car elle représente chaque ligne par un numéro unique. Le nombre de lignes après le prétraitement est : 30 882. **Figure 6-4** présente une visualisation par carte thermique des corrélations des colonnes.

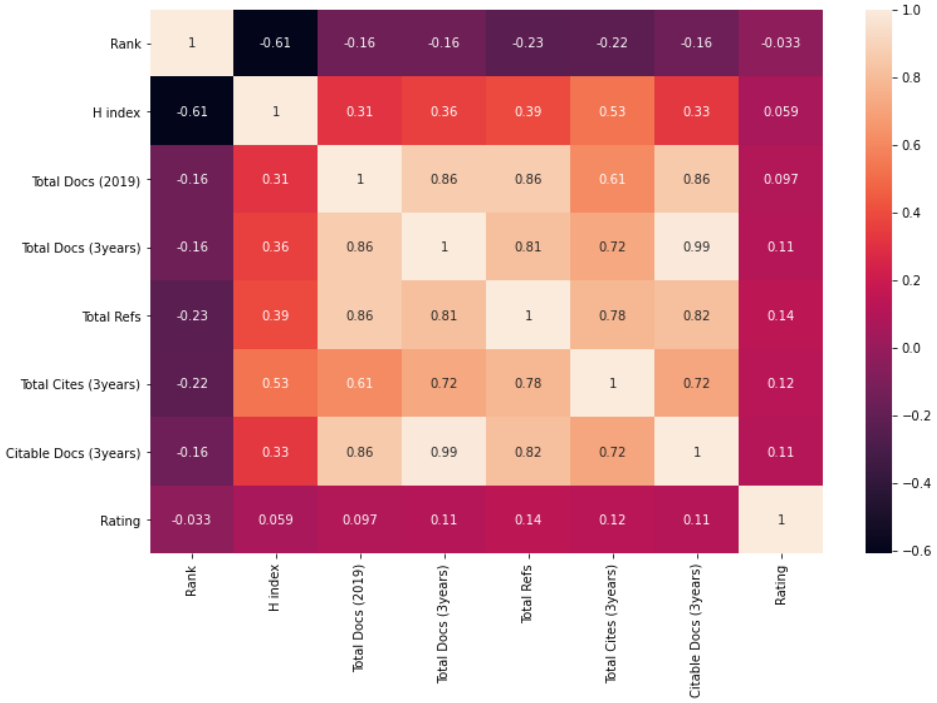

Figure 6-4 : Corrélation des colonnes du data-set (Kahil et al., In press)

4.1.1 Méthode brute

La première solution a été implémentée via Apache Spark. Ce framework fournit un traitement rapide par rapport à d'autres frameworks tels que Hadoop MapReduce, en particulier dans le cas du traitement en mémoire (Kahil et al., 2020). Après avoir prétraité le data-set des journaux, sa jointure au data-set des utilisateurs a été effectué à travers la méthode $crossJoin$ du package Spark SQL. Cela permet d'obtenir une interaction complète entre tous les utilisateurs et les éléments. Après cela, la nouvelle colonne d'évaluation a été ajoutée. Les colonnes factorielles sont ensuite sélectionnées. Pour les utilisateurs, la colonne $user\_id$ est la seule colonne factorielle. Quant aux items, il y a : $rank$, $H\ index$, $total\ docs.(2019)$, $total\ doc.(3\ ans)$, $total\ ref.$, $total\ citations\ (3\ ans)$ et $citable\ docs.(3\ années)$. Pour appliquer la formule (6.3), la condition $\mathcal{R}$ doit être définie. Dans ce cas : x est l'intervalle des valeurs de la colonne $academic\ domain$ qui représente le domaine

académique de l'utilisateur et y est l'intervalle des catégories liées à l'item. $\Re$ signifie qu'au moins une des catégories de revues fait partie des intérêts de l'utilisateur. Formellement : $a_x \cap a_y \neq \varnothing$. Sur la base de cette condition, et compte tenu des pondérations d'attributs choisies par les auteurs, comme indiqué dans **Table 6-3**, la colonne des notes est remplie.

Table 6-3 : Poids proposées pour chaque colonne (Kahil et al., In press)

Attribut	H index	Total docs. (2019)	Total docs. (3 years)	Total refs.	Total cites (3 years)	Citable docs.
Poids	1	1	1/3	1	1/3	1

4.1.2 ALS

Afin d'implémenter la deuxième solution, la bibliothèque *MLlib* d'Apache Spark est utilisée. Elle implémente l'algorithme *ALS* pour effectuer la recommandation avec les deux cas : évaluations implicites et explicites. Le data-set d'entrée contient trois colonnes : $user\_id$, $journal\_id$ et $rating$. Cette dernière colonne contient les valeurs définies selon la stratégie citée ci-dessus. Trois colonnes sont ajoutées : $mean\_rating$, std_rating et $scaled\_rating$. Un $crossJoin$ est effectué sur les deux premières colonnes avec le data-set. Après cela, la troisième colonne est ajoutée selon la formule **(6.4)**. Ensuite, après avoir divisé le data-set en données d'entrainement (0,8%) et en données de test (0,2%), les paramètres ALS sont fixés : 30 pour les itérations maximales, 0,1 pour le paramètre de régularisation et "$drop$" pour la stratégie de démarrage à froid (cold start). Le dernier paramètre permet de prédire les recommandations tout en éliminant les valeurs non numériques de la dataframe. Ainsi, une fois le modèle créé, la matrice d'évaluation complète est obtenue.

4.1.3 DNN

Le framework *TensorFlow2* est choisi pour implémenter le modèle proposé basé sur le DNN. La raison est que, en plus de ses performances élevées avec les petits et les grands data-sets, cette version prend en charge les graphes dynamiques, ce qui réduit le temps d'apprentissage, en particulier avec les grands data-sets. Le modèle proposé basé sur DNN est un FFN qui est composé de sept couches :

1- la couche d'entrée qui contient toutes les colonnes factorielles utilisateurs et éléments ainsi que les évaluations existantes
2- cinq couches cachées : trois couches activées par sigmoïde responsable de la recommandation basée sur le filtrage collaboratif et deux autres, avec la tangente hyperbolique, qui filtrent les résultats en fonction du contexte
3- la couche de sortie qui contient le résultat.

Ce FFN est défini par le modèle séquentiel de Keras avec l'optimiseur Adam. Le filtrage contextuel a été choisi pour faire le pré-filtrage des données selon les langues des utilisateurs. Il filtre tous les éléments en fonction de la langue de l'utilisateur cible. Cela permet d'éviter le cas de recommander des articles écrits dans des langues autres que celles des utilisateurs.

4.2 Mesures d'évaluation

Deux comparaisons sont faites pour évaluer les solutions proposées. La première concerne les trois solutions selon les critères généraux, à savoir : le problème de démarrage à froid, la taille du data-set nécessaire, la rapidité de la solution, si des experts du domaine peuvent être nécessaires pour chaque solution et la procédure de chaque solution lors de l'introduction de nouveaux items et de nouveaux événements liés aux utilisateurs. La seconde comparaison consiste à évaluer les deuxième et troisième solutions avec les travaux connexes en excluant la première solution proposée, car cette dernière n'est

pas basée sur l'apprentissage automatique. Pour cette comparaison, il est nécessaire de définir des métriques d'évaluation. Parmi ces métriques, nous avons choisi la perte (Loss), MSE et l'erreur absolue moyenne (MAE : Mean Absolute Error) comme principales métriques d'évaluation des modèles de régression (Baccianella et al., 2009; Spuler et al., 2015), avec le score carré R^2, aussi appelé le coefficient de détermination. Ils sont respectivement définis par les formules **(6.9)**, **(6.10)** et **(6.11)**. Pour la fonction de perte, l'entropie croisée catégorielle est sélectionnée.

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n} \quad (6.9)$$

$$MAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \quad (6.10)$$

$$R^2 = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (6.11)$$

tel que :

- n est le nombre de points de données
- y_i est la valeur observée
- $\hat{y}_i$ est la valeur prédite
- $\bar{y}$ est la moyenne

Toutes ces mesures sont déjà implémentées dans Tensorflow.

4.3 Résultats et discussion

Table 6-4 montre les résultats de la première comparaison. Il est constaté que la méthode brute ne nécessite pas de phase d'apprentissage et n'est pas affectée par le problème de démarrage à froid, puisque la valeur d'évaluation de chaque item est calculée par la formule **(6.3)**. Cependant, elle nécessite une grande quantité de données depuis l'état initial pour fournir un processus de recommandation rapide. De plus, elle nécessite un expert du domaine afin de définir les règles de recommandation. Alors que les deux solutions basées sur l'apprentissage automatique nécessitent moins de données (seulement 0,8% par rapport à la solution brute). De plus, bien que leur processus d'apprentissage soit lent, elles tiennent compte du problème de démarrage à froid pour les nouveaux utilisateurs et les éléments. En outre, dans le cas de nouveaux items ou utilisateurs, le nombre d'interactions est limité uniquement aux concernés, ce qui réduit le temps de recommandation. Par ailleurs, la réalisation de ces solutions ne nécessite pas d'experts du domaine.

Pour la seconde comparaison, les résultats des deux solutions d'apprentissage automatique sont comparés à ceux des travaux connexes après les avoir appliqués sur le data-set. Les travaux connexes sont : NNMF ((Dziugaite & Roy, 2015), NRR (R. Wang et al., 2017), NFM (Cremonesi et al., 2011), DeepFM (Guo et al., 2017), FM (Rendle, 2010), FFM (Juan et al., 2016), DCN (R. Wang et al., 2017), WDL (Cheng et al., 2016) et XDeepFM (Lian et al., 2018). Les résultats sont présentés dans **Table 6-5**. À partir de ces résultats, on peut voir que la factorisation matricielle utilisant la solution des moindres carrés alternés (MF-ALS) présente des valeurs d'erreur acceptables, en particulier celle de MSE. MF-NN, qui le surpasse, présente les meilleures valeurs d'erreur pour la perte, MAE et MSE parmi les travaux connexes. Cela confirme ses bonnes performances. Néanmoins, bien que la valeur R^2 soit également bonne, ce n'est toujours pas la meilleure. Par conséquent, le modèle proposé doit alors être amélioré. Une manière envisageable est d'optimiser les fonctions des différentes couches. Cette optimisation peut englober les fonctions d'entrée afin de spécifier les colonnes factorielles des

utilisateurs et des items dans un ordre spécifique. L'utilisation d'une architecture DNN avancées, telles que RNN et CNN, pourrait améliorer la solution proposée. Les LSTM, qui appartiennent à RNN, sont caractérisés par la mémoire. Cela pourrait être utile si les évaluations communes sont considérées comme des entrées séquentielles. De même, CNN pourrait être utilisé pour distribuer la matrice d'évaluation de sorte que chaque partie alimente des couches de convolution, ce qui pourrait améliorer le processus d'apprentissage si la matrice d'évaluation est distribuée conformément aux interactions existantes.

Table 6-4 : Comparaison des trois solutions alternatives (Kahil et al., In press)

Critère	MF-CM	MF-ALS	MF-NN
Cold start	-	✓	✓
Nombre de lignes dans l'état initial	552849564	44231818	44231818
Nombre de lignes dans les données d'entrainement	-	39808249	39808249
Nombre de lignes dans les données de test	-	4423569	4423569
Vitesse de calcul des évaluation	Fast	Slow learning process	Slow learning process
Exige un expert de domaine	✓	-	-
Évènement de nouveaux utilisateurs ou éléments	Toutes les interactions doivent être tenues en compte	Seulement quelques interactions sont suffisantes	Seulement quelques interactions sont suffisantes

Table 6-5 : Comparaison des deux solutions basées apprentissage automatique et les travaux connexes (Kahil et al., In press)

Solution	Perte	MAE	MSE	R carré
NNMF (Dziugaite & Roy, 2015)	1.1997	0.0244	0.3254	0.8285
NRR (P. Li et al., 2017)	0.0059	0.0006	0.0059	0.8937
NFM (Cremonesi et al., 2011)	1.5438	0.7310	0.6503	0.7005
DeepFM (Guo et al., 2017)	0.9610	0.7765	0.7441	0.7913
FM (Rendle, 2010)	1.2007	1.6235	1.7018	0.6247
FFM (Juan et al., 2016)	0.03269	1.7655	1.3504	0.7237
DCN (R. Wang et al., 2017)	2.0102	0.8603	0.7500	0.4904
WDL (Cheng et al., 2016)	0.2439	1.3239	1.8750	0.5217
XDeepFM (Lian et al., 2018)	0.0012	0.4800	0.0752	0.4048
MF-ALS	-	0.5149	0.1590	0.7353
MF-NN	0.0016	0.00026	0.00028	0.7953

En résumé, les trois solutions traitent efficacement le problème de matrice clairsemée qui présente le principal problème interne dans le contexte de la recommandation, ainsi que le démarrage à froid, la transparence de la recommandation et les problèmes d'acquisition et de représentation des informations contextuelles de l'utilisateur qui reflètent l'environnement externe de recommandation. Cela garantit l'efficacité de visualisation en termes de disponibilité des données à recommander et, par conséquent, à visualiser. Les solutions basées sur l'apprentissage automatique sont plus performantes que la solution brute en termes de consommation de mémoire et de qualité des recommandations. La solution basée sur

DNN présente des résultats prometteurs qui améliorent la valeur de visualisation, mais elle pourrait être plus efficace en l'améliorant en adoptant des architectures DNN optimisées au lieu de FFN pour atteindre de meilleures performances.

5 Conclusion

L'objectif de cette contribution était d'aborder le problème de la visualisation du Big Data sous l'angle de la recommandation. Pour cela, trois approches alternatives ont été proposées : brute, basée ALS et basée DNN. La solution classique présente une efficacité en termes de recommandation, mais nécessite plus de ressources de calcul en raison de la grande quantité de données dont elle a besoin. Les résultats de la comparaison des deux autres solutions avec les travaux connexes sur le data-set construit ont montré leur efficacité, avec une supériorité sur celle basée DNN. De plus, elles traitent le problème de démarrage à froid. La solution basée sur DNN, dont les résultats étaient meilleurs que celle basée sur ALS, présente les meilleurs scores MSE et MAE, qui sont les principales mesures d'évaluation du problème actuel. Cependant, elle ne présente pas le meilleur R^2 par rapport à certaines solutions connexes. C'est pourquoi le prochain travail à faire est d'améliorer le modèle proposé en utilisant des architectures DNN avancées telles que CNN et RNN afin de le rendre flexible et d'améliorer l'apprentissage et donc la recommandation. Il pourrait également être plus avantageux que ces architectures proposées englobent, en plus de prédire les évaluations, le processus de profilage des utilisateurs en fonction des différents cas indiqués dans l'architecture globale proposée. Plusieurs travaux sont dédiés à la gestion de la recommandation en fonction du statut de l'utilisateur, comme la recommandation basée sur la session, qui peut être utilisée pour le problème de visualisation exploratoire dans Big Data. Un autre aspect qui pourrait être considéré sur la base du DNN est de sélectionner les techniques de visualisation appropriées en fonction des types de données et des besoins de l'utilisateur. A cet égard, une structure pour définir les techniques de visualisation est envisageable afin de relier ces dernières aux data-sets et d'effectuer l'apprentissage pour présenter les patterns de la manière la plus adéquate.

Conclusion générale et perspectives

Cette thèse a abordé la visualisation interactive des données dans un contexte Big Data. Big Data représente le phénomène où les données sont basiquement caractérisées par les 3Vs qui traduisent le volume, la variété et la vélocité. Les données dans Big Data sont le résultat de l'utilisation quotidienne des données par les individus, les corporations et les entreprises d'une part, et de l'interconnexion des technologies modernes d'autre part. La visualisation des données consiste à les présenter graphiquement de manière efficace et significative qui permet de faciliter à l'utilisateur le processus d'analyse, d'exploration et de recherche. Or, en Big Data, les données présentent de nombreux enjeux qui touchent différents axes, y compris la visualisation. Parmi ces enjeux il y a le support des données volumineuses et hétérogènes pour les présenter graphiquement dans un temps raisonnable, la manière de sélectionner depuis les grands data-sets les données consistantes pour les visualiser, la visualisation des informations extraites via un processus d'analytique visuelle et la réduction de complexité.

En matière de production scientifique trois contributions ont été proposées pour remédier aux problèmes essentiels dans le contexte de cette thèse. La première contribution présente une méthode qui vise à préparer les grands data-sets à la visualisation multidimensionnelle tout en assurant la réduction de complexité de cette tâche. Cette contribution garantie la visualisation structurée des data-sets qui peut être personnalisée par l'utilisateur selon des fonctionnalités qui lui sont fournies telles que la sélection et le filtrage. La deuxième contribution propose une nouvelle approche qui a pour objectif de visualiser les graphes à grande échelle de manière efficace en limitant le nombre de nœuds à visualiser à un moment donné. Pour cela, elle se sert de la détection des communautés afin de visualiser une partie du graphe au lieu de son intégralité, tout en offrant un mécanisme qui assure la personnalisation de la visualisation afin de permettre à l'utilisateur d'explorer les parties dont il a besoin. La troisième contribution présente une nouvelle approche qui a pour objectif d'améliorer la visualisation des données en considérant l'aspect de préférences des utilisateurs. Pour cela, elle considère le problème de visualisation exploratoire comme un problème de recommandation. Ce dernier a différentes applications telles que la recommandation des films, des produits dans le domaine d'e-commerce, etc. Il consiste à présenter à chaque utilisateur les données susceptibles de l'intéresser. Ce concept est adopté par l'approche proposée pour recommander à l'utilisateur parmi les données nombreuses celles qui pourraient l'intéresser.

Comme perspectives, les approches proposées à travers les trois contributions peuvent être améliorées en considérant les aspects particuliers relatifs à chacune d'elles. La première approche peut être étendue pour couvrir davantage de types de données afin de la rendre plus générique. Pour cela, il sera nécessaire d'intégrer des mécanismes de pré-traitement des autres types de données dans l'architecture de l'approche GreedyBigVis. Dans la deuxième approche, la visualisation de multiples communautés dans les graphes à grande échelle, assurée par la sélection de plusieurs filtres, peut être améliorée en définissant une stratégie plus adéquate pour la gestion du chevauchement des communautés. Déterminer la consistance des communautés en fonction de nombre de liaisons qu'elle contient peut ne pas être la manière optimale de sélection. D'autres critères sont alors envisageables. Pour la troisième contribution, le travail à venir sera d'appliquer des architectures avancées de DNN afin d'améliorer le processus de recommandation.

Les données textuelles prennent une place considérable dans l'ère de Big Data ; elles se trouvent dans les réseaux sociaux, les blogs, les sites web des informations, les forums de discussion, etc. La visualisation de ce type de données devrait recevoir plus de considération afin de rendre son exploration et son analyse plus efficaces. Avec leur volume important, les données textuelles présentent des défis de visualisation importants, notamment la présentation des informations et patterns consistants de

manière interactive et échelonnable, ainsi que l'exploration sémantique de cette présentation. Une solution envisageable serait de résumer les textes avant de les visualiser. Cela permet de réduire le volume des données et, par conséquent, la complexité de visualisation et d'analytique visuelle de données textuelles. Il existe des techniques de résumé automatique des textes qui peuvent être extractives ou abstractives. Ces techniques s'inscrivent généralement dans l'apprentissage automatique. De multiples modèles, notamment d'apprentissage automatique, peuvent être employés à cette fin. Parmi ces modèles on peut citer Fasttext, Doc2vec, BERT (Bidirectional Encoder Representations from Transformers), etc. Ces modèles permettent de représenter les données textuelles par des vecteurs tout en gardant leur sémantique. Ils peuvent alors être utilisés comme mécanisme de pré-traitement pour préparer les textes à la tâche de résumé automatique (ATS).

D'une manière générale, les contributions réalisées, tout comme les autres travaux de visualisation interactive des données dans un contexte Big Data, peuvent être améliorées en ciblant l'optimisation. Cette dernière concerne aussi bien le processus de visualisation lui-même que les requêtes de recherche et d'exploration en se servant de solutions spécialisées telles que les bases de données NoSQL basées graphe, les différentes méthodes d'optimisation des requêtes, etc. Finalement, le nettoyage des données joue un rôle important dans tout processus de visualisation et d'analytique visuelle. A cet égard, l'automatisation du nettoyage des données sera énormément bénéfique pour ces processus. Il peut être considéré selon multiples facteurs tels que les types de données, leur volume et dimensionnalité, l'objectif de visualisation, ….

Production scientifique

- Kahil, M. S., Bouramoul, A., & Derdour, M. (2021). GreedyBigVis–A greedy approach for preparing large datasets to multidimensional visualization. *International Journal of Computers and Applications (TJCA)*, 1-10. [https://www.tandfonline.com/doi/full/10.1080/1206212X.2021.1920670]
- Kahil, M. S., Bouramoul, A., & Derdour, M. (2019) "Mutual Progress of Big Data and Interactive Visualization", *In 2nd Conference on Informatics and Applied Mathematics*, IAM'2019, Guelma, Algeria.
- Kahil, M. S., Bouramoul, A., & Derdour, M. (2019, July). Big data and interactive visualization: Overview on challenges, techniques and tools. In *International Conference on Advanced Intelligent Systems for Sustainable Development* (pp. 157-167). Springer, Cham. [http://link.springer.com/10.1007/978-3-030-36674-2_17]
- Kahil, M. S., Bouramoul, A., & Derdour, M. (2019, June). Towards a new architecture for data multilevels interactive visualization in big data domains. In *2019 International Conference on Networking and Advanced Systems (ICNAS)* (pp. 1-7). IEEE. [https://ieeexplore.ieee.org/document/8807847/]
- Kahil, M. S., Bouramoul, A., & Derdour, M. (2021, September). Multi Criteria-Based Community Detection and Visualization in Large-scale Networks Using Label Propagation Algorithm. In *2021 International Conference on Recent Advances in Mathematics and Informatics (ICRAMI)* (pp. 1-6). IEEE. [https://ieeexplore.ieee.org/document/9585964]
- Kahil, M. S., Bouramoul, A., & Derdour, M. (In press). Big data visual exploration as a recommendation problem. Inrernational Journal of Data Mining, Modelling and Management. [https://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=ijdmmm]

Références

Aberger, C. R. (2014). Recommender: An Analysis of Collaborative Filtering Techniques. *Personal and Ubiquitous Computing Journal*, 5.

Adhikari, A., & Adhikari, J. (2015). *Advances in Knowledge Discovery in Databases* (Vol. 79). Springer International Publishing. https://doi.org/10.1007/978-3-319-13212-9

Agrawal, R., Kadadi, A., Dai, X., & Andres, F. (2015). Challenges and opportunities with big data visualization. *Proceedings of the 7th International Conference on Management of Computational and Collective IntElligence in Digital EcoSystems - MEDES '15*, 169–173. https://doi.org/10.1145/2857218.2857256

Ajesh, A., Nair, J., & Jijin, P. S. (2016). A random forest approach for rating-based recommender system. *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 1293–1297. https://doi.org/10.1109/ICACCI.2016.7732225

Ali, S. M., Gupta, N., Nayak, G. K., & Lenka, R. K. (2016). Big data visualization: Tools and challenges. *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, 656–660. https://doi.org/10.1109/IC3I.2016.7918044

Aljuaid, T., & Sasi, S. (2016). Proper imputation techniques for missing values in data sets. *2016 International Conference on Data Science and Engineering (ICDSE)*, 1–5. https://doi.org/10.1109/ICDSE.2016.7823957

Allman, A., Tang, W., & Daoutidis, P. (2018). Towards a Generic Algorithm for Identifying High-Quality Decompositions of Optimization Problems. In *Computer Aided Chemical Engineering* (Vol. 44, pp. 943–948). Elsevier. https://doi.org/10.1016/B978-0-444-64241-7.50152-X

Alsunaidi, S. J., Almuhaideb, A. M., Ibrahim, N. M., Shaikh, F. S., Alqudaihi, K. S., Alhaidari, F. A., Khan, I. U., Aslam, N., & Alshahrani, M. S. (2021). Applications of Big Data Analytics to Control COVID-19 Pandemic. *Sensors*, *21*(7), 2282. https://doi.org/10.3390/s21072282

Amghar, S., Cherdal, S., & Mouline, S. (2020). Storing, preprocessing and analyzing tweets: Finding the suitable noSQL system. *International Journal of Computers and Applications*, 1–10. https://doi.org/10.1080/1206212X.2020.1846946

Andersen, J. S., & Zukunft, O. (2016). Evaluating the Scaling of Graph-Algorithms for Big Data Using GraphX. *2016 2nd International Conference on Open and Big Data (OBD)*, 1–8. https://doi.org/10.1109/OBD.2016.8

Andrienko, G., Andrienko, N., Drucker, S., Fekete, J.-D., Fisher, D., Idreos, S., Kraska, T., Li, G., Ma, K.-L., Mackinlay, J., Oulasvirta, A., Schreck, T., Schmann, H., Stonebraker, M., Auber, D., Bikakis, N., Chrysanthis, P., Papastefanatos, G., & Sharaf, M. (2020). *Big Data Visualization and Analytics: Future Research Challenges and Emerging Applications*. 9.

Anowar, F., Sadaoui, S., & Selim, B. (2021). Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE). *Computer Science Review*, *40*, 100378. https://doi.org/10.1016/j.cosrev.2021.100378

Arulkumar, V., Charlyn, P. L., & Daniel, D. J. (2019). Concept of implementing Big data in smart city: Applications, Services, Data Security in accordance with Internet of Things and AI. *International*

Journal of Recent Technology and Engineering, *8*(3), 6819–6825. https://doi.org/10.35940/ijrte.C5782.098319

Atzeni, P., Bugiotti, F., Cabibbo, L., & Torlone, R. (2020). Data modeling in the NoSQL world. *Computer Standards & Interfaces*, *67*, 103149. https://doi.org/10.1016/j.csi.2016.10.003

Ayundhita, M. S., Baizal, Z. K. A., & Sibaroni, Y. (2019). Ontology-based conversational recommender system for recommending laptop. *Journal of Physics: Conference Series*, *1192*, 012020. https://doi.org/10.1088/1742-6596/1192/1/012020

Azevedo, A., & Santos, M. F. (2008). KDD, SEMMA AND CRISP-DM: A PARALLEL OVERVIEW. *IADS-DM*, 6.

Aziz, K., Zaidouni, D., & Bellafkih, M. (2018). Big Data Processing using Machine Learning algorithms: MLlib and Mahout Use Case. *Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications*, 1–6. https://doi.org/10.1145/3289402.3289525

Baaziz, A., & Quoniam, L. (2014). *How to use Big Data technologies to optimize operations in Upstream Petroleum Industry*. 10.

Baccianella, S., Esuli, A., & Sebastiani, F. (2009). Evaluation Measures for Ordinal Regression. *2009 Ninth International Conference on Intelligent Systems Design and Applications*, 283–287. https://doi.org/10.1109/ISDA.2009.230

Bajaber, F., Elshawi, R., Batarfi, O., Altalhi, A., Barnawi, A., & Sakr, S. (2016). Big Data 2.0 Processing Systems: Taxonomy and Open Challenges. *Journal of Grid Computing*, *14*(3), 379–405. https://doi.org/10.1007/s10723-016-9371-1

Bedi, P., & Sharma, C. (2016). Community detection in social networks: Community detection in social networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *6*(3), 115–135. https://doi.org/10.1002/widm.1178

Bednorz, W. (2008). *Greedy algorithms*. InTech. http://www.intechweb.org/books/show/title/greedy_algorithms

Bellahsene, Z., Bonifati, A., & Rahm, E. (Eds.). (2011). *Schema Matching and Mapping*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-16518-4

Berti-Equille, L. (2019). Learn2Clean: Optimizing the Sequence of Tasks for Web Data Preparation. *The World Wide Web Conference on - WWW '19*, 2580–2586. https://doi.org/10.1145/3308558.3313602

Bhogal, J., & Choksi, I. (2015). Handling Big Data Using NoSQL. *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, 393–398. https://doi.org/10.1109/WAINA.2015.19

Bikakis, N. (2018). Big Data Visualization Tools. *ArXiv:1801.08336 [Cs]*. http://arxiv.org/abs/1801.08336

Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, *46*, 109–132. https://doi.org/10.1016/j.knosys.2013.03.012

Bonaccorso, G., & Safari, an O. M. C. (2018). *Machine Learning Algorithms—Second Edition*. https://www.safaribooksonline.com/library/view//9781789347999/?ar

B.Thorat, P., M. Goudar, R., & Barve, S. (2015). Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. *International Journal of Computer Applications*, *110*(4), 31–36. https://doi.org/10.5120/19308-0760

Calin, O. (2020). *Deep Learning Architectures: A Mathematical Approach*. Springer International Publishing. https://doi.org/10.1007/978-3-030-36721-3

Camm, J. D., Cochran, J. J., Fry, M. J., & Ohlmann, J. W. (2021). *Data visualization: Exploring and explaining with data* (1e ed.). Cengage Learning.

Cattell, R. (2011). Scalable SQL and NoSQL data stores. *ACM SIGMOD Record*, *39*(4), 12. https://doi.org/10.1145/1978915.1978919

Chawla, G., Bamal, S., & Khatana, R. (2018). Big Data Analytics for Data Visualization: Review of Techniques. *International Journal of Computer Applications*, *182*(21), 37–40. https://doi.org/10.5120/ijca2018917977

Chen, L. (2021). *Deep Learning and Practice with MindSpore*. Springer Singapore. https://doi.org/10.1007/978-981-16-2233-5

Chen, M., Ebert, D., Hagen, H., Laramee, R. S., van Liere, R., Ma, K.-L., Ribarsky, W., Scheuermann, G., & Silver, D. (2009). Data, Information, and Knowledge in Visualization. *IEEE Computer Graphics and Applications*, *29*(1), 12–19. https://doi.org/10.1109/MCG.2009.6

Chen, M., Mao, S., & Liu, Y. (2014). Big Data: A Survey. *Mobile Networks and Applications*, *19*(2), 171–209. https://doi.org/10.1007/s11036-013-0489-0

Cheng, H.-T., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X., Shah, H., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., & Chai, W. (2016). Wide & Deep Learning for Recommender Systems. *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems - DLRS 2016*, 7–10. https://doi.org/10.1145/2988450.2988454

Chhabra, G., Amity School of Institute Technology, Amity University, Noida – 201313, Uttar Pradesh, India, Vashisht, V., Department of Computer Science and Engineering, Amity School of Engineering, Amity University, Noida – 201313, Uttar Pradesh, India, Ranjan, J., & Institute of Management Technology, Ghaziabad – 201001, Uttar Pradesh, India. (2017). A Comparison of Multiple Imputation Methods for Data with Missing Values. *Indian Journal of Science and Technology*, *10*(19), 1–7. https://doi.org/10.17485/ijst/2017/v10i19/110646

Christophides, V., Efthymiou, V., Palpanas, T., Papadakis, G., & Stefanidis, K. (2020). End-to-End Entity Resolution for Big Data: A Survey. *ArXiv:1905.06397 [Cs]*. http://arxiv.org/abs/1905.06397

Cremonesi, P., Tripodi, A., & Turrin, R. (2011). Cross-Domain Recommender Systems. *2011 IEEE 11th International Conference on Data Mining Workshops*, 496–503. https://doi.org/10.1109/ICDMW.2011.57

Cunningham, P., & Delany, S. J. (2020). k-Nearest Neighbour Classifiers: 2nd Edition (with Python examples). *ArXiv:2004.04523 [Cs, Stat]*. http://arxiv.org/abs/2004.04523

Cuzzocrea, A., Bellatreche, L., & Song, I.-Y. (2013). Data Warehousing and OLAP over Big Data: Current Challenges and Future Research Directions. *Proceedings of the Sixteenth International Workshop on Data Warehousing and OLAP*, 67--70.

da Silva, I. N., Hernane Spatti, D., Andrade Flauzino, R., Liboni, L. H. B., & dos Reis Alves, S. F. (2017). Artificial Neural Network Architectures and Training Processes. In I. N. da Silva, D. Hernane Spatti, R. Andrade Flauzino, L. H. B. Liboni, & S. F. dos Reis Alves, *Artificial Neural Networks* (pp. 21–28). Springer International Publishing. https://doi.org/10.1007/978-3-319-43162-8_2

Dash, D., Rao, J., Megiddo, N., Ailamaki, A., & Lohman, G. (2008). Dynamic faceted search for discovery-driven analysis. *Proceeding of the 17th ACM Conference on Information and Knowledge Mining - CIKM '08*, 3. https://doi.org/10.1145/1458082.1458087

Dave, A., Jindal, A., Li, L. E., Xin, R., Gonzalez, J., & Zaharia, M. (2016). GraphFrames: An integrated API for mixing graph and relational queries. *Proceedings of the Fourth International Workshop on Graph Data Management Experiences and Systems - GRADES '16*, 1–8. https://doi.org/10.1145/2960414.2960416

Davenport, T. (2014). *Big data at work: Dispelling the myths, uncovering the opportunities* [Data set]. Harvard Business Review Press. https://doi.org/10.1287/8943f842-86f8-4d42-9a64-9a7cd07b31f5

Davoudian, A., Chen, L., & Liu, M. (2018). A Survey on NoSQL Stores. *ACM Computing Surveys*, *51*(2), 1–43. https://doi.org/10.1145/3158661

De Jonge, Edwin & Van Der Loo, Mar. (2013). An introduction to data cleaning with R. *Statistics Netherlands Heerlen*, 53.

Deepa, N., Pham, Q.-V., Nguyen, D. C., Bhattacharya, S., Prabadevi, B., Gadekallu, T. R., Maddikunta, P. K. R., Fang, F., & Pathirana, P. N. (2022). A survey on blockchain for big data: Approaches, opportunities, and future directions. *Future Generation Computer Systems*, *131*, 209–226. https://doi.org/10.1016/j.future.2022.01.017

Demchenko, Y., de Laat, C., & Membrey, P. (2014). Defining architecture components of the Big Data Ecosystem. *2014 International Conference on Collaboration Technologies and Systems (CTS)*, 104–112. https://doi.org/10.1109/CTS.2014.6867550

Devore, J. L., Berk, K. N., & Carlton, M. A. (2021). *Modern Mathematical Statistics with Applications*. Springer International Publishing. https://doi.org/10.1007/978-3-030-55156-8

Dietrich, D., Heller, B., Yang, B., & EMC Education Services (Eds.). (2015). *Data science & big data analytics: Discovering, analyzing, visualizing and presenting data*. Wiley.

Dimara, E., & Perin, C. (2020). What is Interaction for Data Visualization? *IEEE Transactions on Visualization and Computer Graphics*, *26*(1), 119–129. https://doi.org/10.1109/TVCG.2019.2934283

Do, Hong-Hai. (2006). *Schema matching and mapping-based data integration* [Interdisciplinary Center for Bioinformatics and Department of Computer Science - University of Leipzig]. https://ul.qucosa.de/api/qucosa%3A16447/attachment/ATT-0/

Dong, X., Yu, Z., Cao, W., Shi, Y., & Ma, Q. (2020). A survey on ensemble learning. *Frontiers of Computer Science*, *14*(2), 241–258. https://doi.org/10.1007/s11704-019-8208-z

Duan, L., Street, W. N., Liu, Y., & Lu, H. (2014). Community detection in graphs through correlation. *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1376–1385. https://doi.org/10.1145/2623330.2623629

Dziugaite, G. K., & Roy, D. M. (2015). Neural Network Matrix Factorization. *ArXiv:1511.06443 [Cs, Stat]*. http://arxiv.org/abs/1511.06443

Efthymiou, V., Stefanidis, K., & Christophides, V. (2015). Big data entity resolution: From highly to somehow similar entity descriptions in the Web. *2015 IEEE International Conference on Big Data (Big Data)*, 401–410. https://doi.org/10.1109/BigData.2015.7363781

Eine, B., Jurisch, M., & Quint, W. (2017). Ontology-Based Big Data Management. *Systems*, *5*(3), 45. https://doi.org/10.3390/systems5030045

El Arass, M., & Souissi, N. (2018). Data Lifecycle: From Big Data to SmartData. *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*, 80–87. https://doi.org/10.1109/CIST.2018.8596547

Elshawi, R., Wahab, A., Barnawi, A., & Sakr, S. (2021). DLBench: A comprehensive experimental evaluation of deep learning frameworks. *Cluster Computing*, *24*(3), 2017–2038. https://doi.org/10.1007/s10586-021-03240-4

Erraissi, A. (2017). *A Comparative Study of Hadoop-based Big Data Architectures*. *9*(4), 9.

Erraissi, A., & Belangour, A. (2020). *An Approach Based On Model Driven Engineering For Big Data Visualization In Different Visual Modes*. *9*(01), 9.

Fahad, S. K. A., & Yahya, A. E. (2018). Big Data Visualization: Allotting by R and Python with GUI Tools. *2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*, 1–8. https://doi.org/10.1109/ICSCEE.2018.8538413

Fan, J., Han, F., & Liu, H. (2014). Challenges of Big Data analysis. *National Science Review*, *1*(2), 293–314. https://doi.org/10.1093/nsr/nwt032

Fang, W., Jiang, J., Lu, S., Gong, Y., Tao, Y., Tang, Y., Yan, P., Luo, H., & Liu, J. (2020). A LSTM Algorithm Estimating Pseudo Measurements for Aiding INS during GNSS Signal Outages. *Remote Sensing*, *12*(2), 256. https://doi.org/10.3390/rs12020256

Fegaras, L. (2016). Incremental Query Processing on Big Data Streams. *IEEE Transactions on Knowledge and Data Engineering*, *28*(11), 2998–3012. https://doi.org/10.1109/TKDE.2016.2601103

Feng, W. (2019). *Learning Apache Spark with Python*.

Fiaz, A. S. S., Asha, N., Sumathi, D., & Navaz, A. S. S. (2016). *Data Visualization: Enhancing Big Data More Adaptable and Valuable*. *11*(4), 4.

Getoor, L., & Machanavajjhala, A. (2012). Entity resolution: Theory, practice & open challenges. *Proceedings of the VLDB Endowment*, *5*(12), 2018–2019. https://doi.org/10.14778/2367502.2367564

Giceva, J., & Sadoghi, M. (2018). Hybrid OLTP and OLAP. In S. Sakr & A. Zomaya (Eds.), *Encyclopedia of Big Data Technologies* (pp. 1–8). Springer International Publishing. https://doi.org/10.1007/978-3-319-63962-8_179-1

Godfrey, P., Gryz, J., & Lasek, P. (2016). Interactive Visualization of Large Data Sets. *IEEE Transactions on Knowledge and Data Engineering*, *28*(8), 2142–2157. https://doi.org/10.1109/TKDE.2016.2557324

Golfarelli, M., & Rizzi, S. (2019). A model-driven approach to automate data visualization in big data analytics. *Information Visualization*, 147387161985893. https://doi.org/10.1177/1473871619858933

Gomez-Uribe, C. A., & Hunt, N. (2015). The Netflix Recommender System: Algorithms, Business Value, and Innovation. *ACM Transactions on Management Information Systems*, *6*(4), 1–19. https://doi.org/10.1145/2843948

Gorodov, E. Y., & Gubarev, V. V. (2013). Analytical Review of Data Visualization Methods in Application to Big Data. *Journal of Electrical and Computer Engineering*, *2013*, 1–7. https://doi.org/10.1155/2013/969458

Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. *ArXiv:1703.04247 [Cs]*. http://arxiv.org/abs/1703.04247

Harley, A. W. (2015). An Interactive Node-Link Visualization of Convolutional Neural Networks. In G. Bebis, R. Boyle, B. Parvin, D. Koracin, I. Pavlidis, R. Feris, T. McGraw, M. Elendt, R. Kopper, E. Ragan, Z. Ye, & G. Weber (Eds.), *Advances in Visual Computing* (Vol. 9474, pp. 867–877). Springer International Publishing. https://doi.org/10.1007/978-3-319-27857-5_77

Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Ullah Khan, S. (2015). The rise of "big data" on cloud computing: Review and open research issues. *Information Systems*, *47*, 98–115. https://doi.org/10.1016/j.is.2014.07.006

Hazarika, A. V., Ram, G. J. S. R., & Jain, E. (2017). Performance comparision of Hadoop and spark engine. *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 671–674. https://doi.org/10.1109/I-SMAC.2017.8058263

Huang, M. L., Lu, L. F., & Zhang, X. (2015). Using arced axes in parallel coordinates geometry for high dimensional BigData visual analytics in cloud computing. *Computing*, *97*(4), 425–437. https://doi.org/10.1007/s00607-014-0383-z

Huang, X., Wu, L., & Ye, Y. (2019). A Review on Dimensionality Reduction Techniques. *International Journal of Pattern Recognition and Artificial Intelligence*, *33*(10), 1950017. https://doi.org/10.1142/S0218001419500174

Im, J.-F., Villegas, F. G., & McGuffin, M. J. (2013). VisReduce: Fast and responsive incremental information visualization of large datasets. *2013 IEEE International Conference on Big Data*, 25–32. https://doi.org/10.1109/BigData.2013.6691710

Iqbal, R., Doctor, F., More, B., Mahmud, S., & Yousuf, U. (2016). Big Data analytics: Computational intelligence techniques and application areas. *International Journal of Information Management*, S0268401216303309. https://doi.org/10.1016/j.ijinfomgt.2016.05.020

Izenman, A. J. (2013). Multivariate Regression. In G. Casella, S. Fienberg, & I. Olkin (Eds.), *Modern Multivariate Statistical Techniques* (pp. 159–194). Springer New York. https://doi.org/10.1007/978-0-387-78189-1_6

Jabbar, H. K., & Khan, R. Z. (2014). Methods to Avoid Over-Fitting and Under-Fitting in Supervised Machine Learning (Comparative Study). *Computer Science, Communication and Instrumentation Devices*, 163–172. https://doi.org/10.3850/978-981-09-5247-1_017

Jadhav, A., Pramod, D., & Ramanathan, K. (2019). Comparison of Performance of Data Imputation Methods for Numeric Dataset. *Applied Artificial Intelligence*, *33*(10), 913–933. https://doi.org/10.1080/08839514.2019.1637138

Jagadish, H. V., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J. M., Ramakrishnan, R., & Shahabi, C. (2014). Big data and its technical challenges. *Communications of the ACM*, *57*(7), 86–94. https://doi.org/10.1145/2611567

Jing Han, Haihong E, Guan Le, & Jian Du. (2011). Survey on NoSQL database. *2011 6th International Conference on Pervasive Computing and Applications*, 363–366. https://doi.org/10.1109/ICPCA.2011.6106531

Juan, Y., Zhuang, Y., Chin, W.-S., & Lin, C.-J. (2016). Field-aware Factorization Machines for CTR Prediction. *Proceedings of the 10th ACM Conference on Recommender Systems*, 43–50. https://doi.org/10.1145/2959100.2959134

Kaboli, M. (2017). A Review of Transfer Learning Algorithms. *Technische Universität*, 68.

Kahil, M. S., Bouramoul, A., & Derdour, M. (2019). Towards a New Architecture for Data Multilevels Interactive Visualization in Big Data Domains. *2019 International Conference on Networking and Advanced Systems (ICNAS)*, 1–7. https://doi.org/10.1109/ICNAS.2019.8807847

Kahil, M. S., Bouramoul, A., & Derdour, M. (2020). Big Data and Interactive Visualization: Overview on Challenges, Techniques and Tools. In M. Ezziyyani (Ed.), *Advanced Intelligent Systems for Sustainable Development (AI2SD'2019)* (Vol. 1105, pp. 157–167). Springer International Publishing. https://doi.org/10.1007/978-3-030-36674-2_17

Kahil, M. S., Bouramoul, A., & Derdour, M. (2021a). GreedyBigVis – A greedy approach for preparing large datasets to multidimensional visualization. *International Journal of Computers and Applications*, 1–10. https://doi.org/10.1080/1206212X.2021.1920670

Kahil, M. S., Bouramoul, A., & Derdour, M. (2021b). Multi Criteria-Based Community Detection and Visualization in Large-scale Networks Using Label Propagation Algorithm. *2021 International Conference on Recent Advances in Mathematics and Informatics (ICRAMI)*, 1–6. https://doi.org/10.1109/ICRAMI52622.2021.9585964

Kahil, M. S., Bouramoul, A., & Derdour, M. (In press). Big data visual exploration as a recommendation problem. *International Journal of Data Mining, Modelling and MAnagement*, 21.

Kambach, S., Bruelheide, H., Gerstner, K., Gurevitch, J., Beckmann, M., & Seppelt, R. (2020). Consequences of multiple imputation of missing standard deviations and sample sizes in meta-analysis. *Ecology and Evolution*, *10*(20), 11699–11712. https://doi.org/10.1002/ece3.6806

Kaur, J., & Madan, N. (2015). Association Rule Mining: A Survey. *International Journal of Hybrid Information Technology*, *8*(7), 239–242. https://doi.org/10.14257/ijhit.2015.8.7.22

Kbaier, M. E. B. H., Masri, H., & Krichen, S. (2017). A Personalized Hybrid Tourism Recommender System. *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, 244–250. https://doi.org/10.1109/AICCSA.2017.12

Khan, M., & Khan, S. S. (2015). Data and Information Visualization Methods, and Interactive Mechanisms: A Survey. *International Journal of Computer Applications*, *34*, 14.

Khan, N., Alsaqer, M., Shah, H., Badsha, G., Abbasi, A. A., & Salehian, S. (2018). The 10 Vs, Issues and Challenges of Big Data. *Proceedings of the 2018 International Conference on Big Data and Education - ICBDE '18*, 52–56. https://doi.org/10.1145/3206157.3206166

Kim, J., & Lee, J.-G. (2015). Community Detection in Multi-Layer Graphs: A Survey. *ACM SIGMOD Record*, *44*(3), 37–48. https://doi.org/10.1145/2854006.2854013

Knauer, U. (2019). *Algebraic Graph Theory: Morphisms, Monoids and Matrices*. De Gruyter Studies in Mathematics.

Kolajo, T., Daramola, O., & Adebiyi, A. (2019). Big data stream analysis: A systematic literature review. *Journal of Big Data*, *6*(1), 47. https://doi.org/10.1186/s40537-019-0210-7

Konys, A. (2017). Ontology-Based Approaches to Big Data Analytics. In S. Kobayashi, A. Piegat, J. Pejaś, I. El Fray, & J. Kacprzyk (Eds.), *Hard and Soft Computing for Artificial Intelligence, Multimedia and Security* (Vol. 534, pp. 355–365). Springer International Publishing. https://doi.org/10.1007/978-3-319-48429-7_32

Krishnan, S., Franklin, M. J., Goldberg, K., Wang, J., & Wu, E. (2016). ActiveClean: An Interactive Data Cleaning Framework For Modern Machine Learning. *Proceedings of the 2016 International Conference on Management of Data*, 2117–2120. https://doi.org/10.1145/2882903.2899409

Kulcu, S., Dogdu, E., & Ozbayoglu, A. M. (2016). A survey on semantic Web and big data technologies for social network analysis. *2016 IEEE International Conference on Big Data (Big Data)*, 1768–1777. https://doi.org/10.1109/BigData.2016.7840792

Kumbhare, T. A., & Chobe, S. V. (2014). *An Overview of Association Rule Mining Algorithms*. *5*, 4.

Kunaver, M., & Požrl, T. (2017). Diversity in recommender systems – A survey. *Knowledge-Based Systems*, *123*, 154–162. https://doi.org/10.1016/j.knosys.2017.02.009

Lam, H., Bertini, E., Isenberg, P., Plaisant, C., & Carpendale, S. (2012). Empirical Studies in Information Visualization: Seven Scenarios. *IEEE Transactions on Visualization and Computer Graphics*, *18*(9), 1520–1536. https://doi.org/10.1109/TVCG.2011.279

Lancichinetti, A., & Fortunato, S. (2009). Community detection algorithms: A comparative analysis. *Physical Review E*, *80*(5), 056117. https://doi.org/10.1103/PhysRevE.80.056117

Lee, I. (2017). Big data: Dimensions, evolution, impacts, and challenges. *Business Horizons*, *60*(3), 293–303. https://doi.org/10.1016/j.bushor.2017.01.004

Leung, C. K., Chen, Y., Hoi, C. S. H., Shang, S., Wen, Y., & Cuzzocrea, A. (2020). Big Data Visualization and Visual Analytics of COVID-19 Data. *2020 24th International Conference Information Visualisation (IV)*, 415–420. https://doi.org/10.1109/IV51561.2020.00073

Li, P., Wang, Z., Ren, Z., Bing, L., & Lam, W. (2017). Neural Rating Regression with Abstractive Tips Generation for Recommendation. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 345–354. https://doi.org/10.1145/3077136.3080822

Li, Y., Wang, Z., & Hao, Y. (2018). A Hierarchical Visualization Analysis Model of Power Big Data. *IOP Conference Series: Earth and Environmental Science*, *108*, 052064. https://doi.org/10.1088/1755-1315/108/5/052064

Lian, J., Zhou, X., Zhang, F., Chen, Z., Xie, X., & Sun, G. (2018). xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1754–1763. https://doi.org/10.1145/3219819.3220023

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, *234*, 11–26. https://doi.org/10.1016/j.neucom.2016.12.038

Liu, X., Iftikhar, N., & Xie, X. (2014). Survey of real-time processing systems for big data. *Proceedings of the 18th International Database Engineering & Applications Symposium on - IDEAS '14*, 356–361. https://doi.org/10.1145/2628194.2628251

LTIM, Erraissi, A., Belangour, A., & Tragha, A. (2017). A Big Data Hadoop building blocks comparative study. *International Journal of Computer Trends and Technology*, *48*(1), 36–40. https://doi.org/10.14445/22312803/IJCTT-V48P109

Mahesh, B. (2018). *Machine Learning Algorithms—A Review*. *9*(1), 6.

Maheshwar, R. C., & Haritha, D. (2016). Survey on high performance analytics of bigdata with apache spark. *2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, 721–725. https://doi.org/10.1109/ICACCCT.2016.7831734

Margaris, D., Vassilakis, C., & Spiliotopoulos, D. (2020). What makes a review a reliable rating in recommender systems? *Information Processing & Management*, *57*(6), 102304. https://doi.org/10.1016/j.ipm.2020.102304

Marx, V. (2013). The big challenges of big data. *Nature*, *498*(7453), 255–260. https://doi.org/10.1038/498255a

McAfee, A., Brynjolfsson, E., Davenport, Thomas H, Patil, DJ, & Barton, Dominic. (2012). Big Data: The Management Revolution. *Harvard Business Review*, *90*(10), 60--68.

McNicholas, P. D., Murphy, T. B., & O'Regan, M. (2008). Standardising the lift of an association rule. *Computational Statistics & Data Analysis*, *52*(10), 4712–4721. https://doi.org/10.1016/j.csda.2008.03.013

Melit Devassy, B., & George, S. (2020). Dimensionality reduction and visualisation of hyperspectral ink data using t-SNE. *Forensic Science International*, *311*, 110194. https://doi.org/10.1016/j.forsciint.2020.110194

Mohammed, M., Khan, M. B., & Bashier, E. B. M. (2017). *Machine learning: Algorithms and applications*. CRC Press, Taylor & Francis Group.

Montgomery, D. C., & Runger, G. C. (2018). *Applied statistics and probability for engineers*.

Nagel, H. R. (2006). *Scientific Visualization versus Information Visualization*. 4.

Nair, L. R., Shetty, S. D., & Shetty, S. D. (2016). *INTERACTIVE VISUAL ANALYTICS ON BIG DATA: TABLEAU VS D3.JS*. *12*(4), 12.

Nanda, S. B., Kalha, A. S., Jena, A. K., Bhatia, V., & Mishra, S. (2015). Artificial neural network (ANN) modeling and analysis for the prediction of change in the lip curvature following extraction and non-extraction orthodontic treatment. *Journal of Dental Specialities*, *3*(2), 217. https://doi.org/10.5958/2393-9834.2015.00002.9

Oussous, A., Benjelloun, F.-Z., Ait Lahcen, A., & Belfkih, S. (2018). Big Data technologies: A survey. *Journal of King Saud University - Computer and Information Sciences*, *30*(4), 431–448. https://doi.org/10.1016/j.jksuci.2017.06.001

Padgavankar, M. H., & Gupta, D. S. R. (2014). *Big Data Storage and Challenges*. *5*, 6.

Paradarami, T. K., Bastian, N. D., & Wightman, J. L. (2017). A hybrid recommender system using artificial neural networks. *Expert Systems with Applications*, *83*, 300–313. https://doi.org/10.1016/j.eswa.2017.04.046

Park, D. H., Kim, H. K., Choi, I. Y., & Kim, J. K. (2012). A literature review and classification of recommender systems research. *Expert Systems with Applications*, *39*(11), 10059–10072. https://doi.org/10.1016/j.eswa.2012.02.038

Patel, A., & Jain, S. (2019). Present and future of semantic web technologies: A research statement. *International Journal of Computers and Applications*, 1–10. https://doi.org/10.1080/1206212X.2019.1570666

Patgiri, R. (2019). A Taxonomy on Big Data: Survey. *ArXiv:1808.08474 [Cs]*. http://arxiv.org/abs/1808.08474

Pouchard, L. (2016). Revisiting the Data Lifecycle with Big Data Curation. *International Journal of Digital Curation*, *10*(2), 176–192. https://doi.org/10.2218/ijdc.v10i2.342

Qi, C. (2020). Big data management in the mining industry. *International Journal of Minerals, Metallurgy and Materials*, *27*(2), 131–139. https://doi.org/10.1007/s12613-019-1937-z

Qin, X., Luo, Y., Tang, N., & Li, G. (2018). DeepEye: An automatic big data visualization framework. *Big Data Mining and Analytics*, *1*(1), 75–82. https://doi.org/10.26599/BDMA.2018.9020007

Quijano-Sánchez, L., Cantador, I., Cortés-Cediel, M. E., & Gil, O. (2020). Recommender systems for smart cities. *Information Systems*, *92*, 101545. https://doi.org/10.1016/j.is.2020.101545

Qunchao Fu, Wanheng Liu, Tengfei Xue, Heng Gu, Siyue Zhang, & Cong Wang. (2014). A big data processing methods for visualization. *2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems*, 571–575. https://doi.org/10.1109/CCIS.2014.7175800

Raghav, R. S., Pothula, S., Vengattaraman, T., & Ponnurangam, D. (2016). A survey of data visualization tools for analyzing large volume of data in big data platform. *2016 International*

Conference on Communication and Electronics Systems (ICCES), 1–6. https://doi.org/10.1109/CESYS.2016.7889976

Rama Satish, K. V., & Kavya, N. P. (2019). A framework for big data pre-processing and search optimization using HMGA-ACO: A hierarchical optimization approach. *International Journal of Computers and Applications*, *41*(3), 183–194. https://doi.org/10.1080/1206212X.2017.1417768

Rendle, S. (2010). Factorization Machines. *2010 IEEE International Conference on Data Mining*, 995–1000. https://doi.org/10.1109/ICDM.2010.127

Rossi, R. A., & Ahmed, N. K. (2015). *The Network Data Repository with Interactive Graph Analytics and Visualization*. 2.

Roy, S., Sarkar, D., & De, D. (2020). Entropy-aware ambient IoT analytics on humanized music information fusion. *Journal of Ambient Intelligence and Humanized Computing*, *11*(1), 151–171. https://doi.org/10.1007/s12652-019-01261-x

Rozemberczki, B., Davies, R., Sarkar, R., & Sutton, C. (2019). GEMSEC: Graph Embedding with Self Clustering. *ArXiv:1802.03997 [Cs]*. http://arxiv.org/abs/1802.03997

Ruder, S. (2017). An overview of gradient descent optimization algorithms. *ArXiv:1609.04747 [Cs]*. http://arxiv.org/abs/1609.04747

Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *WIREs Data Mining and Knowledge Discovery*, *8*(4). https://doi.org/10.1002/widm.1249

Sagiroglu, S., & Sinanc, D. (2013). Big data: A review. *2013 International Conference on Collaboration Technologies and Systems (CTS)*, 42–47. https://doi.org/10.1109/CTS.2013.6567202

Sansen, J., Richer, G., Jourde, T., Lalanne, F., Auber, D., & Bourqui, R. (2017). Visual Exploration of Large Multidimensional Data Using Parallel Coordinates on Big Data Infrastructure. *Informatics*, *4*(3), 21. https://doi.org/10.3390/informatics4030021

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative Filtering Recommender Systems. In *The Adaptive Web* (p. 34).

Schintler, L. A., & McNeely, C. L. (Eds.). (2022). *Encyclopedia of Big Data*. Springer International Publishing. https://doi.org/10.1007/978-3-319-32010-6

Shafique, U., & Qaiser, H. (2014). *A Comparative Study of Data Mining Process Models (KDD, CRISP-DM and SEMMA)*. *12*(1), 6.

Shambour, Q. (2021). A deep learning based algorithm for multi-criteria recommender systems. *Knowledge-Based Systems*, *211*, 106545. https://doi.org/10.1016/j.knosys.2020.106545

Shao, J., Han, Z., Yang, Q., & Zhou, T. (2015). Community Detection based on Distance Dynamics. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1075–1084. https://doi.org/10.1145/2783258.2783301

Shaw, G., Xu, Y., & Geva, S. (2010). Using Association Rules to Solve the Cold-Start Problem in Recommender Systems. In M. J. Zaki, J. X. Yu, B. Ravindran, & V. Pudi (Eds.), *Advances in*

Knowledge Discovery and Data Mining (Vol. 6118, pp. 340–347). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-13657-3_37

Simonini, G., & Zhu, S. (2015). Big data exploration with faceted browsing. *2015 International Conference on High Performance Computing & Simulation (HPCS)*, 541–544. https://doi.org/10.1109/HPCSim.2015.7237087

Sinaeepourfard, A., Garcia, J., Masip-Bruin, X., & Marín-Torder, E. (2016). Towards a comprehensive data lifecycle model for big data environments. *Proceedings of the 3rd IEEE/ACM International Conference on Big Data Computing, Applications and Technologies*, 100–106. https://doi.org/10.1145/3006299.3006311

Singh, D., & Reddy, C. K. (2015). A survey on platforms for big data analytics. *Journal of Big Data*, *2*(1). https://doi.org/10.1186/s40537-014-0008-6

Singh, R., & Kaur, P. J. (2016). Analyzing performance of Apache Tez and MapReduce with hadoop multinode cluster on Amazon cloud. *Journal of Big Data*, *3*(1), 19. https://doi.org/10.1186/s40537-016-0051-6

Singhal, A., Sinha, P., & Pant, R. (2017). Use of Deep Learning in Modern Recommendation System: A Summary of Recent Works. *International Journal of Computer Applications*, *180*(7), 17–22. https://doi.org/10.5120/ijca2017916055

Sobhanam, H., & Mariappan, A. K. (2013). Addressing cold start problem in recommender systems using association rules and clustering technique. *2013 International Conference on Computer Communication and Informatics*, 1–5. https://doi.org/10.1109/ICCCI.2013.6466121

Sorzano, C. O. S., Vargas, J., & Montano, A. P. (2014). A survey of dimensionality reduction techniques. *ArXiv:1403.2877 [Cs, q-Bio, Stat]*. http://arxiv.org/abs/1403.2877

Soylu, A., Giese, M., Jimenez-Ruiz, E., Kharlamov, E., Zheleznyakov, D., & Horrocks, I. (2013). OptiqueVQS: Towards an ontology-based visual query system for big data. *Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems - MEDES '13*, 119–126. https://doi.org/10.1145/2536146.2536149

Spratt, M., Carpenter, J., Sterne, J. A. C., Carlin, J. B., Heron, J., Henderson, J., & Tilling, K. (2010). Strategies for Multiple Imputation in Longitudinal Studies. *American Journal of Epidemiology*, *172*(4), 478–487. https://doi.org/10.1093/aje/kwq137

Spuler, M., Sarasola-Sanz, A., Birbaumer, N., Rosenstiel, W., & Ramos-Murguialday, A. (2015). Comparing metrics to evaluate performance of regression methods for decoding of neural signals. *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 1083–1086. https://doi.org/10.1109/EMBC.2015.7318553

Štajner, T., & Mladenić, D. (2009). Entity Resolution in Texts Using Statistical Learning and Ontologies. In A. Gómez-Pérez, Y. Yu, & Y. Ding (Eds.), *The Semantic Web* (Vol. 5926, pp. 91–104). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-10871-6_7

Storl, U., Hauf, T., Klettke, Meike, & Scherzinger, Stefanie. (2015). Schemaless NoSQL Data Stores – Object-NoSQL Mappers to the Rescue? *Datenbanksysteme Fur Business, Technologie Und Web (BTW 2015)*, 21.

Subasi, A. (2020). Machine learning techniques. In *Practical Machine Learning for Data Analysis Using Python* (pp. 91–202). Elsevier. https://doi.org/10.1016/B978-0-12-821379-7.00003-5

Subramaniyaswamy, V., & Logesh, R. (2017). Adaptive KNN based Recommender System through Mining of User Preferences. *Wireless Personal Communications*, *97*(2), 2229–2247. https://doi.org/10.1007/s11277-017-4605-5

Sutanta, E., Wardoyo, R., Mustofa, K., & Winarko, E. (2016). Survey: Models and Prototypes of Schema Matching. *International Journal of Electrical and Computer Engineering (IJECE)*, *6*(3), 1011. https://doi.org/10.11591/ijece.v6i3.9789

Suzuki, J. (2021). *Statistical Learning with Math and Python: 100 Exercises for Building Logic*. Springer Singapore. https://doi.org/10.1007/978-981-15-7877-9

Suzuki, M., Sato, H., Oyama, S., & Kurihara, M. (2014). Transfer learning based on the observation probability of each attribute. *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 3627–3631. https://doi.org/10.1109/SMC.2014.6974493

Sze, V., Chen, Y.-H., Yang, T.-J., & Emer, J. S. (2017). Efficient Processing of Deep Neural Networks: A Tutorial and Survey. *Proceedings of the IEEE*, *105*(12), 2295–2329. https://doi.org/10.1109/JPROC.2017.2761740

Tariq RS, N. T. (2015). Big Data Challenges. *Computer Engineering & Information Technology*, *04*(03). https://doi.org/10.4172/2324-9307.1000133

Teng, S.-H. (2016). Scalable Algorithms for Data and Network Analysis. *Foundations and Trends® in Theoretical Computer Science*, *12*(1–2), 1–274. https://doi.org/10.1561/0400000051

Thirukumaran, S., & Sumathi, A. (2012). Missing value imputation techniques depth survey and an imputation Algorithm to improve the efficiency of imputation. *2012 Fourth International Conference on Advanced Computing (ICoAC)*, 1–5. https://doi.org/10.1109/ICoAC.2012.6416805

Tominski, C., & Schumann, H. (2020). *Interactive visual data analysis*. CRC Press/Taylor & Francis Group.

Tsai, C.-W., Lai, C.-F., Chao, H.-C., & Vasilakos, A. V. (2015). Big data analytics: A survey. *Journal of Big Data*, *2*(1). https://doi.org/10.1186/s40537-015-0030-3

Uzunkaya, C., Ensari, T., & Kavurucu, Y. (2015). Hadoop Ecosystem and Its Analysis on Tweets. *Procedia - Social and Behavioral Sciences*, *195*, 1890–1897. https://doi.org/10.1016/j.sbspro.2015.06.429

van der Aalst, W. (2016). *Process Mining*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-49851-4

VanderPlas, J. (2016). *Python Data Science Handbook*. 548.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. *ArXiv:1706.03762 [Cs]*. http://arxiv.org/abs/1706.03762

Verbert, K., Manouselis, N., Ochoa, X., Wolpers, M., Drachsler, H., Bosnic, I., & Duval, E. (2012). Context-Aware Recommender Systems for Learning: A Survey and Future Challenges. *IEEE Transactions on Learning Technologies*, *5*(4), 318–335. https://doi.org/10.1109/TLT.2012.11

Verma, J. P., Patel, B., & Patel, A. (2015). Big Data Analysis: Recommendation System with Hadoop Framework. *2015 IEEE International Conference on Computational Intelligence & Communication Technology*, 92–97. https://doi.org/10.1109/CICT.2015.86

Vijayarani, S., & Sharmila, S. (2016). Comparative analysis of association rule mining algorithms. *2016 International Conference on Inventive Computation Technologies (ICICT)*, 1–6. https://doi.org/10.1109/INVENTIVE.2016.7830203

Vohra, D. (2016). *Practical Hadoop Ecosystem*. Apress. https://doi.org/10.1007/978-1-4842-2199-0

Wadkar, S., & Siddalingaiah, M. (2014). *Pro Apache Hadoop: Analyze large volumes of data in amazingly short wall-clock intervals* (2. ed). Apress.

Wang, J., Yang, Y., Wang, T., Sherratt, R. S., & Zhang, J. (2020). Big Data Service Architecture: A Survey. *Journal of Internet Technology*, 14.

Wang, R., Fu, B., Fu, G., & Wang, M. (2017). Deep & Cross Network for Ad Click Predictions. *ArXiv:1708.05123 [Cs, Stat]*. http://arxiv.org/abs/1708.05123

Ward, M. O., Grinstein, G., & Keim, D. (2015). *Interactive Data Visualization*. 106.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big Data*, *3*(1), 9. https://doi.org/10.1186/s40537-016-0043-6

Wilkinson, L. (2018). Visualizing Big Data Outliers Through Distributed Aggregation. *IEEE Transactions on Visualization and Computer Graphics*, *24*(1), 256–266. https://doi.org/10.1109/TVCG.2017.2744685

Wu, A., Wang, Y., Shu, X., Moritz, D., Cui, W., Zhang, H., Zhang, D., & Qu, H. (2021). AI4VIS: Survey on Artificial Intelligence Approaches for Data Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 1–1. https://doi.org/10.1109/TVCG.2021.3099002

Xin, X., Karatzoglou, A., Arapakis, I., & Jose, J. M. (2020). Self-Supervised Reinforcement Learning for Recommender Systems. *ArXiv:2006.05779 [Cs]*. http://arxiv.org/abs/2006.05779

Xu, Y., Zhou, W., Cui, B., & Lu, L. (2015). Research on performance optimization and visualization tool of Hadoop. *2015 10th International Conference on Computer Science & Education (ICCSE)*, 149–153. https://doi.org/10.1109/ICCSE.2015.7250233

Xue, H.-J., Dai, X., Zhang, J., Huang, S., & Chen, J. (2017). Deep Matrix Factorization Models for Recommender Systems. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 3203–3209. https://doi.org/10.24963/ijcai.2017/447

Xyntarakis, M., & Antoniou, C. (2019). Data Science and Data Visualization. In *Mobility Patterns, Big Data and Transport Analytics* (pp. 107–144). Elsevier. https://doi.org/10.1016/B978-0-12-812970-8.00006-3

Yagang Zhang. (2010). *Machine Learning* (In-Tech). INTECH Open Access Publisher. intechweb.org

Yang, Z., Algesheimer, R., & Tessone, C. J. (2016). A Comparative Analysis of Community Detection Algorithms on Artificial Networks. *Scientific Reports*, *6*(1), 30750. https://doi.org/10.1038/srep30750

Yera, R., & Martínez, L. (2017). Fuzzy Tools in Recommender Systems: A Survey. *International Journal of Computational Intelligence Systems*, *10*(1), 776. https://doi.org/10.2991/ijcis.2017.10.1.52

Yujie, Z., & Licai, W. (2010). Some challenges for context-aware recommender systems. *2010 5th International Conference on Computer Science & Education*, 362–365. https://doi.org/10.1109/ICCSE.2010.5593612

Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., & Saeed, J. (2020). A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction. *Journal of Applied Science and Technology Trends*, *1*(2), 56–70. https://doi.org/10.38094/jastt1224

Zgraggen, E., Galakatos, A., Crotty, A., Fekete, J.-D., & Kraska, T. (2017). How Progressive Visualizations Affect Exploratory Analysis. *IEEE Transactions on Visualization and Computer Graphics*, *23*(8), 1977–1987. https://doi.org/10.1109/TVCG.2016.2607714

Zhang, C., & Ma, Y. (Eds.). (2012). *Ensemble Machine Learning*. Springer US. https://doi.org/10.1007/978-1-4419-9326-7

Zhang, H.-R., & Min, F. (2016). Three-way recommender systems based on random forests. *Knowledge-Based Systems*, *91*, 275–286. https://doi.org/10.1016/j.knosys.2015.06.019

Zhang, J., Huang, M. L., Wang, W. B., Lu, L. F., & Meng, Z.-P. (2014). Big Data Density Analytics Using Parallel Coordinate Visualization. *2014 IEEE 17th International Conference on Computational Science and Engineering*, 1115–1120. https://doi.org/10.1109/CSE.2014.219

Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2018). A survey on deep learning for big data. *Information Fusion*, *42*, 146–157. https://doi.org/10.1016/j.inffus.2017.10.006

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep Learning based Recommender System: A Survey and New Perspectives. *ACM Computing Surveys*, *52*(1), 1–38. https://doi.org/10.1145/3285029

Zhang, Z. (2016). Multiple imputation for time series data with Amelia package. *Annals of Translational Medicine*, *4*(3), 10.

Zhou, K., Zhao, W. X., Bian, S., Zhou, Y., Wen, J.-R., & Yu, J. (2020). Improving Conversational Recommender Systems via Knowledge Graph based Semantic Fusion. *ArXiv:2007.04032 [Cs]*. http://arxiv.org/abs/2007.04032

Zhou, Y., Wilkinson, D., Schreiber, R., & Pan, R. (2008). Large-Scale Parallel Collaborative Filtering for the Netflix Prize. In R. Fleischer & J. Xu (Eds.), *Algorithmic Aspects in Information and Management* (Vol. 5034, pp. 337–348). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-68880-8_32

Zhou, Z.-H. (2021). *Machine Learning*. Springer Singapore. https://doi.org/10.1007/978-981-15-1967-3

Zhu, D., Zhang, H., Sun, Y., & Qi, H. (2021). Injury Risk Prediction of Aerobics Athletes Based on Big Data and Computer Vision. *Scientific Programming*, *2021*, 1–10. https://doi.org/10.1155/2021/5526971

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, *109*(1), 43–76. https://doi.org/10.1109/JPROC.2020.3004555

Ziegler, A., & König, I. R. (2014). Mining data with random forests: Current options for real-world applications: Mining data with random forests. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *4*(1), 55–63. https://doi.org/10.1002/widm.1114

Annexes

Liste d’abréviations

A

Artificial Neural Network
ANN, 49
Association Rule Mining
ARM, 76
Automatic Text Summarization
ATS, 51

B

Bidirectional Encoder Representations from Transformers
BERT, 131
Bootstrap Aggregating
bagging, 46

C

Click-Through Rate
CTR, 115
Complete Case Analysis
CCA, 33
compressed interaction network
CIN, 115

E

Exponentional Linear Unit
ELU, 51

F

Feature Extraction Algorithms
FEA, 35
Feed-Forward Network
FFN, 49

G

Generative Adversarial Networks
GAN, 52

I

Internet of Things
IoT, 10
Interquartile Range
IQR, 33

K

Kernel-PCA
KPCA, 36
KMC
K-means clustering, 48
Knowledge Discovery from Databases
KDD, 28
Knowledge Graphs
KG, 116

L

Label Propagation Algorithm
LPA, 98
Leaky ReLU
LReLU, 51
Locally Linear Embedding
LLE, 36

M

Matrix Factorization
MF, 111
Mean Absolute Error
MAE, 126
Mean Squared Error
MSE, 121
Multi-Layer Perceptron
MLP, 49

N

Natural Language Processing
NLP, 37

O

OnLine Analytical Processing
OLAP, 78

P

Principal Component Analysis
PCA, 35

R

Recommeder Systems
RS, 111
Rectified Linear Unit
ReLU, 51
Restricted Boltzmann Machine
RBM, 52

S

Singular Value Decomposition
    SVD, 36
Stochastic Gradient Descent
    SGD, 120

T

tangente hyperbolique
    tanh, 50
t-Stochastic Neighbour Embedding
    tSNE, 36

V

Variational Autoencoders
    VAE, 52

W

Word Aligned Hybrid
    WAH, 78

Liste des synonymes et de traduction

Terme	Synonymes
Feature learning, feature extraction	Apprentissage de caractéristique, Extraction des caractéristiques, apprentissage de représentation
Factor column	Colonne factorielle
Pattern	Item, pattern
Sparsity	Matrice clairesemée
Scalability	Mise en échelle, échelonnabilité
Feature	Caractéristique, variable
Rating	Evalution
Machine Learning	Apprentissage automatique
Ensemble learning	Apprentissage ensembliste
Transfert learning	Apprentissage par transfert
Reiforcement learning	Apprentissge par renforcement
Embedding	Intégration
Greedy algorithm	Algorithme glouton, heuristique gloutonne
Clustering	Regroupement, partitionnement
Entity resolution	Résolution des entités
Context-aware filtering (knowledge-aware filtering)	Filtrage contextual
Matrix factorization	Factorisation matricielle
Vertical scaling (scale-up)	Mise en échelle verticale
Horizontal Scaling (scale-out)	Mise en échelle horisontale
Precision	Précision
Accuracy	Exactitude
Recall	Rappel
Data-set	Jeu de données
Cold start	Démarrage à froid
Dynamic graph	Graphe dynamique
Loss	Perte
Principal Component Analysis	Analyse en composantes principales
Singular Value Decomposition	Décomposition en valeurs singulières
Alternating least squares	Moindres carrés alternés
Stochastic gradient descent	Algorithme du gradient stochastique
Linear Discriminent Analysis	Analyse discriminante linaire
Heat map	Carte thermique
Treemap	Carte arborescente
Bubble map	Carte à bulles
Pie charte	Diagramme circulaire
Scatter plot	Nuage des points
Bubble chart	Graphique à bulles
Word cloud	Nuage de mots
Parallel coordinates	Coordonnées parallèles
Distplot (Distribution plot)	Graphique de distribution
Faceted Browsing	Navigation à facettes
Binned aggregation	Agrégation en corbeilles