UNIVERSITÉ DE BLIDA 1
Faculté des Sciences
Département d'Informatique

THÈSE DE DOCTORAT

Option : Sciences Informatiques et de Données.

FOUILLE DE DONNÉES ÉDUCATIVES.

Par
Benkessirat Selma

devant le jury composé de :

N. Benblidia	**Professeur, U. de Blida 1**	**Présidente**
N. Boustia	**Professeur, U. de Blida 1**	**Directrice de thèse**
N. Rezoug	**Maitre de conférence, U. de Blida 1**	**Co-Directrice de thèse**
D. Bennouar	**Professeur, U. de Bouira**	**Examinateur**
M. Farah	**Maitre de conférence, U. de Blida 1**	**Examinatrice**
F. Boumahdi	**Maitre de conférence, U. de Blida 1**	**Examinatrice**

16 mai 2024

Je dédie cette thèse à mes chers parents.

REMERCIEMENTS

Cette thèse est l'aboutissement de mon voyage de doctorat qui a été comme l'ascension d'un haut sommet, étape par étape, accompagnée d'encouragements, de difficultés, de confiance et de frustration. Lorsque je me suis retrouvée au sommet, éprouvant le sentiment de satisfaction, j'ai réalisé que bien que seul mon nom apparaisse sur la couverture de cette thèse, un grand nombre de personnes, y compris les membres de ma famille, mes directrices de thèse, mes amis ont contribué à accomplir ce travail.

En premier lieu, nous remercions ALLAH, le tout puissant, de nous avoir donné la force pour survivre, ainsi que la volonté, la patience et le courage pour dépasser tous les obstacles et surmonter les moments de faiblesse.

Je remercie ma directrice de thèse Pr. Boustia pour son encadrement et encouragement tout au long de mon parcours doctoral. Je lui suis reconnaissante pour son précieux temps qu'elle m'a souvent accordé, pour ses conseils et son engagement.

Je tiens à exprimer ma profonde gratitude pour ma co-directrice de thèse Dr. Rezoug pour sa disponibilité, son soutien continu et sa patience pendant mon parcours doctoral. Je lui suis reconnaissante pour ses précieux conseils, ses critiques constructives, son appréciation positive et ses avis tout au long de mon parcours qui ont permis de mener à bien ce travail de recherche.

Je tiens à remercier profondément Pr. Benblidia, pour m'avoir fait l'honneur d'accepter de présider mon jury de thèse.

J'exprime toute ma gratitude aux membres du jury Pr. Bennouar, Dr. Farah et Dr. Boumahdi qui ont accepté de lire et évaluer mon modeste travail.

Je suis grandement redevable à mon ami Dr. Riali Ishak, ce travail n'aurait pas été accompli sans ses conseils et son implication, son soutien et ses encouragements quotidiens depuis le début de la thèse jusqu'à aujourd'hui. Grace à lui, j'ai réussi à surmonter de nombreuses difficultés et à apprendre beaucoup de choses.

Un merci très particulier à mon amie Dr. Ykhlef Hadjer pour son soutien qui a contribué à bien démarrer mon parcours.

J'exprime ma gratitude à mes chers parents, pour avoir eu confiance en moi et m'avoir donné la liberté de choisir ce que je voulais. Sans leur énorme soutien, encouragements et prières au cours de toutes mes années d'étude, il me serait impossible d'arriver là.

Un merci très spécial à Amina ma sœur jumelle, mon demie. J'ai tellement de chance d'avoir une sœur qui m'accompagne dans chaque pas, qui réfléchit avec moi et parfois même à ma place, qui m'aide à prendre des décisions de la plus banale à la plus difficile. Sans son implication ce travail n'aurait pas été possible.

Walid, Youcef et Imène, non je ne vous ai pas oublié. Merci pour votre présence à mes coté, votre soutien et encouragements. Je me considère comme la plus chanceuse d'avoir des frères aussi adorables.

Je dois remercier une personne très spéciale, mon époux, Oussama, pour son amour, sa patience et sa compréhension qui ont rendu possible l'achèvement de cette thèse. J'apprécie profondément sa foi en moi.

Mes remerciements les plus sincères vont à ma belle-famille pour leur encouragement et soutien moral.

Merci à mes amis Nesrine et Youcef pour leur amitié et d'être toujours à mon écoute.

À tous ceux qui ont contribué de près ou de loin, MERCI.

Résumé

La fouille de données éducatives est un nouveau domaine de recherche en plein essor, il connait un grand intérêt de la part des chercheurs, ce qui joue un rôle important dans l'amélioration des systèmes éducatifs. Parmi les applications qui contribuent à l'amélioration du processus d'apprentissage des apprenants, nous distinguons les systèmes de recommandations. Les chercheurs de ce domaine se concentrent principalement sur l'amélioration des approches de recommandation de ressources pédagogiques en se basant sur le centre d'intérêt et préférences des apprenants, en créant des systèmes d'apprentissage en ligne qui se base sur la personnalisation. Un tel système adaptatif nécessite l'extraction de bonnes informations qui représentent le profil de l'apprenant. Dans la littérature nous distinguons trois techniques de recommandation de base, à savoir : le filtrage collaboratif, le filtrage basé sur le contenu et le filtrage hybride. Le filtrage collaboratif se base sur la similarité entre les utilisateurs, il recommande à l'utilisateur cible les articles appréciés par les utilisateurs les plus similaires à son profil. La sélection d'utilisateurs similaires est la première étape du processus du filtrage collaboratif.

Au cours de cette thèse, nous nous intéressons à la technique du filtrage collaboratif pour améliorer la pertinence des ressources pédagogiques recommandées aux apprenants. Dans la littérature, plusieurs travaux utilisent le clustering comme étape de présélection d'utilisateurs similaires. La présélection aide le processus à obtenir une meilleure similarité entre les utilisateurs. Dans cette perspective, nous avons proposé un modèle générique $CF-GT$ qui se base sur la théorie des jeux. Notre modèle $CF-GT$ comporte deux phases principales : la première consiste à regrouper les utilisateurs similaires selon un modèle de jeu coopératif et la deuxième applique le processus du filtrage collaboratif sur chaque groupe obtenu. Le modèle a été testé et validé par rapport à la précision des recommandations fournies. Les résultats expérimentaux révèlent l'efficacité de notre contribution par rapport à d'autres modèles.

Afin d'affirmer l'efficacité de notre modèle dans le domaine éducatif, nous avons conçu un nouveau modèle nommé $Edu-CF-GT$. Ce dernier n'est que l'adaptation du modèle $CF-GT$ au domaine éducatif. Le modèle $Edu-CF-GT$ a été testé en utilisant la base de données *EduTest* que nous avons-nous même conçue. Le manque de données éducatives public a motivé la construction de cette base. Les résultats expérimentaux ont affirmé l'efficacité de notre modèle dans un contexte éducatif.

Avec l'évolution de la quantité de données dans les systèmes d'apprentissage informatisés, l'intégration du deep learning traite les problèmes d'interaction utilisateur/item complexes. Nous avons tenté d'optimiser le modèle $Edu-CF-GT$ en intégrant le *CNN* au processus de recommandation. Les résultats expérimentaux révèlent l'efficacité de la proposition.

En dernier lieu, nous proposons quelques perspectives. D'une part, l'amélioration de notre proposition pour fournir des recommandations diversifiées s'impose. D'autre part, nous envisageons de développer une plateforme d'apprentissage informatiser et y intégrer notre modèle de recommandation. La plateforme sera destinée à l'université pour l'intérêt des étudiants et enseignants.

Mots clés : Fouille de données éducatives, système de recommandations, théorie des jeux, deep learning.

Abstract

Educational data mining is a new and growing field of research, and is receiving a lot of attention from researchers, which plays an important role in improving educational systems. Among the applications that contribute to the improvement of the learning process of learners, we distinguish the recommendation systems. Researchers in this field focus mainly on improving approaches to recommending educational resources based on learners' interests and preferences, creating online learning systems that are based on personalization. Such an adaptive system requires the extraction of good information that represents the learner's profile. In the literature we distinguish three basic recommendation techniques: collaborative filtering, content-based filtering and hybrid filtering. Collaborative filtering is based on the similarity between users, it recommends to the target user the articles appreciated by the users most similar to his profile. The selection of similar users is the first step in the collaborative filtering process.

In this thesis, we focus on the collaborative filtering technique to improve the relevance of educational resources recommended to learners. In the literature, several works use clustering as a pre-selection step for similar users. Pre-selection helps the process to achieve better similarity between users. In this perspective, we have proposed a generic $CF-GT$ model based on game theory. Our $CF-GT$ model has two main phases: the first one consists in grouping similar users according to a cooperative game model, and the second one applies the collaborative filtering process on each obtained group. The model has been tested and validated against the accuracy of the recommendations provided. The experimental results reveal the effectiveness of our contribution compared to other models.

In order to affirm the effectiveness of our model in the educational domain, we have designed a new model named $Edu-CF-GT$. The latter is simply an adaptation of the $CF-GT$ model to the educational domain. The $Edu-CF-GT$ model was tested using the *EduTest* database that we designed ourselves. The lack of public educational data motivated the construction of this database. The experimental results affirmed the effectiveness of our model in an educational context.

With the evolution of the amount of data in computerized learning systems, the integration of deep learning addresses complex user/item interaction problems. We attempted to optimize the $Edu-CF-GT$ model by integrating *CNN* into the recommendation process. Experimental results reveal the effectiveness of the proposal.

Finally, we propose some perspectives. On the one hand, the improvement of our proposal to provide diversified recommendations is necessary. On the other hand, we plan to develop a computerized learning platform and integrate our recommendation model. The platform will be intended for the university for the interest of students and teachers.

Keywords: Educational data mining, recommendation system, game theory, deep learning.

ملخص

يعد التنقيب عن البيانات التعليمية مجالًا جديدًا للبحث ، بحيث عرف اهتمامًا كبيرًا من طرف الباحثين في السنوات الاخيرة ، كما اثبت فعاليته في تحسين النظم التعليمية.

من بين تطبيقات التنقيب عن البيانات التي تساهم في تحسين عملية التعلم للمتعلمين، نميز أنظمة التوصية. يركز الباحثون في هذا المجال بشكل أساسي على تحسين مناهج التوصية بالموارد التعليمية بناءً على اهتمامات المتعلمين وتفضيلاتهم ، وإنشاء أنظمة تعلم عبر الإنترنت تستند إلى التخصيص. يتطلب مثل هذا النظام التكيفي استخراج معلومات جيدة تمثل ملف تعريف المتعلم. في الأدبيات العلمية ، نميز بين ثلاث تقنيات توصية أساسية وهي: التصفية التعاونية ، والتصفية القائمة على المحتوى ، والتصفية الهجينة. تعتمد التصفية التعاونية على التشابه بين المستخدمين ، وتوصي المستخدم المستهدف بالمقالات التي يقدرها المستخدمون الأكثر تشابهًا مع ملفه الشخصي. يعد اختيار المستخدمين المماثلين الخطوة الأولى في عملية التصفية التعاونية.

في هذه الاطروحة، نحن مهتمون بتقنية التصفية التعاونية لتحسين ملاءمة الموارد التعليمية الموصى بها للمتعلمين. في الأدبيات العلمية ، تستخدم العديد من أعمال التجميع كخطوة للاختيار المسبق للمستخدمين المماثلين. يساعد الاختيار المسبق على تحقيق تشابه أفضل بين المستخدمين. في هذا المنظور ، اقترحنا نموذجًا عامًا *CF-GT* والذي يعتمد على نظرية **الالعاب**. يتكون نموذج *CF-GT* الخاص بنا من مرحلتين رئيسيتين: الأولى تتكون من تجميع مستخدمين متشابهين وفقًا لنموذج الالعاب التعاونية، بينما تطبق العملية الثانية التصفية التعاونية على كل مجموعة يتم الحصول عليها. تم اختبار النموذج والتحقق من صحته مقابل دقة التوصيات المقدمة. تكشف النتائج التجريبية فعالية مساهمتنا مقارنة بالنماذج الأخرى.

لتأكيد فعالية نموذجنا في المجال التعليمي ، قمنا بتصميم نموذج جديد باسم *Edu-CF-GT*. هذا الأخير هو فقط تكييف نموذج *CF-GT* للمجال التعليمي. تم اختبار نموذج *Edu-CF-GT* باستخدام قاعدة بيانات *EduTest* التي صممناها بأنفسنا. دفع نقص البيانات التعليمية العامة إلى بناء هذه القاعدة. أكدت النتائج التجريبية فعالية نموذجنا في سياق تعليمي.

مع تطور كمية البيانات في أنظمة التعلم القائمة على الكمبيوتر ، ادماج التعلم العميق يساعد على حل مشاكل التفاعل المستخدم / العنصر المعقدة. حاولنا تحسين نموذج *Edu-CF-GT* من خلال دمج *CNN* في عملية التوصية. النتائج التجريبية تكشف فعالية الاقتراح.

أخيرًا ، نقدم بعض المقترحات ، من الضروري تحسين اقتراحنا لتقديم توصيات متنوعة. من ناحية أخرى ، نخطط لتطوير منصة تعليمية محوسبة ودمج نموذج التوصية الخاص بنا فيها. المنصة ستكون مخصصة للجامعة لمصلحة الطلاب والمعلمين.

الكلمات المفتاحية: التنقيب عن البيانات التعليمية ، نظام التوصيات ، نظرية الألعاب ، التعلم العميق.

Table des matières

Table des figures

Liste des tableaux

List of Algorithms

List of Abbreviations

EDM	**E**ducational **D**ata **M**ining
DM	**D**ata **M**ining
ML	**M**achine **L**earning
SR	**S**ystème de **R**ecommandations
FC	**F**iltrage **C**ollaboratif
FBC	**F**iltrage **B**asé sur le **C**ontenu
GT	**G**ame **T**heory
SV	**S**hapley **V**alue
DL	**D**eep **L**earning

Introduction Générale

Motivation et orientation

La fouille de données éducatives ou Educational Data Mining (EDM) est un domaine de recherche multidisciplinaire émergent, dans lequel des techniques de la fouille de données (Data Mining DM) sont déployées pour extraire des connaissances des systèmes d'information éducatifs afin d'aider les responsables à améliorer le processus d'apprentissage et les résultats scolaires des apprenants.

Il existe actuellement de nombreux systèmes d'apprentissage informatisés tel que les systèmes de gestion de l'apprentissage (Learning Management System LMS), les cours en ligne ouverts et massifs (Massive Open Online Courses (MOOC)) et les systèmes de tutorat intelligents (Intelligent Tutoring Systems (ITS)). L'utilisation de ces systèmes entraîne une croissance exponentielle des données générées par les interactions des apprenants. Les données éducatives peuvent également être générées à partir des systèmes de gestion scolaire qui stockent une énorme quantité de données potentielles sur les étudiants, telles que leurs informations académiques et personnelles. La motivation derrière le domaine d'EDM est d'explorer les données générées en y appliquant divers algorithmes d'apprentissage automatique et des techniques de DM, dans le but final de découvrir les informations et les modèles cachés qui serviront à améliorer les résultats de l'apprentissage.

Dans le domaine de la fouille de données éducatives, il existe de nombreuses applications qui sont regroupées sous diverses catégories en fonction de leurs objectifs [2]. L'une de ces catégories est les systèmes d'aide à la décision qui vise essentiellement à aider les parties prenantes à prendre des décisions pour améliorer le processus d'apprentissage. L'aide à la décision en EDM peut être définie [2] comme une technologie bien adapté pour fournir une aide à la décision dans les environnements d'enseignement supérieur, en générant et en présentant des informations et des connaissances pertinentes pour améliorer la qualité des processus d'apprentissage et de la gestion de l'éducation. Les exemples de cette catégorie sont : la sélection de cours et la génération de recommandations, la fourniture de rapports et la création d'alertes pour les parties prenantes. La cible de ces systèmes d'aide à la décision est principalement l'instructeur, mais il peut également s'agir de l'étudiant, des administrateurs ou des chercheurs.

Afin de traiter les différentes applications de l'EDM, de nombreuses méthodes sont disponibles. En général, les méthodes qui existent dans le domaine de l'EDM sont à peu près les

mêmes que celles du domaine DM. Les méthodes d'EDM les plus utilisées sont la prédiction, la classification, la régression, le clustering, l'exploration de relations et la découverte avec modèle [3].

Dans les applications d'aide à la décision, l'utilisation des techniques de la fouille de données et des algorithmes d'apprentissage automatique a augmenté rapidement. Les observations des comportements des étudiants peuvent fournir des instances d'apprentissage qui peuvent être utilisées pour leur appliquer des techniques d'apprentissage automatique et de DM afin de créer un modèle permettant de prédire les actions et caractéristiques ultérieures. De nombreuses techniques peuvent être appliquées dans le contexte d'aide à la décision, comme les réseaux bayésiens, les arbres de décision, les forêts aléatoires, les K plus proches voisins, les machines à vecteurs de support, les réseaux de neurones, les K-means, etc.

Le but des systèmes d'apprentissage informatisés est d'optimiser le processus d'apprentissage des apprenants et d'améliorer leurs rendements pédagogiques. La quantité de ressources pédagogiques chargée dans ces systèmes est très grande. Face à un large éventail de choix, l'apprenant se trouve souvent hésitant à choisir le bon cours qui lui convient, et tombe dans une situation de perte de temps et souvent de parcourir un cours inadéquat à ses besoins/ profil pédagogiques. Pour faire face à cette situation, les systèmes de recommandations ont imposé leurs intégrations dans les systèmes d'apprentissage informatisés et sont devenus une solution incontournable d'EDM. Un système de recommandation garantit à l'apprenant des ressources appropriées à ses besoins pédagogiques et ça répond aux objectifs d'EDMs parce que les ressources éducatives pertinentes recommandées améliorent le processus d'apprentissage de l'apprenant [4].

Dans cette thèse, nous nous intéressons à l'application des systèmes de recommandation dans le domaine de l'éducation en exploitant les méthodes de la fouille de données. Plus précisément, nous nous intéressons à la recommandation de cours et de ressources pédagogiques adéquats au profil et besoin des apprenants. Le fait que les apprenants aient des supports d'apprentissages adéquats à leurs besoins pédagogiques, ça affecte directement et positivement leurs processus d'apprentissage. La recommandation pertinente est considérée comme un facteur essentiel qui améliore les résultats pédagogiques des apprenants.

Depuis la création du World Wide Web en 1991 [5], la quantité de contenu et de ressources générés a augmenté de façon exponentielle. Aujourd'hui, le commerce électronique, les plateformes de streaming et le e-learning, etc. constituent une partie substantielle du trafic du Web. Ces services offrent une vaste quantité variée de contenu à leurs clients : plus de 200 millions de produits sur Amazon.com, 30 millions de chansons sur Spotify, 10000 films sur Netflix, plus de 2200 leçons et 2 millions d'utilisateurs sur la plateforme Khan Acaemy, etc. Dans de telles situations de surcharge de données et d'informations, il est vital d'aider les utilisateurs à explorer et à trouver les ressources qui les intéressent. Les technologies de recommandation qui fournissent des suggestions personnalisées aux utilisateurs se sont avérées la meilleure solution pour obtenir les ressources adéquates à l'utilisateur dans un temps optimal.

Avec les systèmes de recommandations, l'utilisateur n'a pas à formuler une requête. Sa seule requête est implicite qui peut se formuler par : « quelles sont les ressources adéquates à mes besoins et satisfont mes contraintes et préférences », ce qui constitue un grand avantage de ces systèmes.

Les systèmes de recommandation ont suscité un intérêt croissant dans la communauté universitaire au cours des deux dernières décennies. Cela a donné lieu à une abondance d'algorithmes, de technologies et de logiciels de recommandation. La recherche dans ce domaine a été couverte par de nombreux forums. La conférence ACM sur les systèmes de recommandation, qui a débuté en 2007, peut être considérée comme le principal forum dans ce domaine.

Au départ, les systèmes de recommandation étaient utilisés dans le commerce électronique et, par la suite, ils sont devenus populaires dans presque tous les domaines [6], notamment l'éducation. En fait, l'énorme quantité de données présente dans les systèmes d'enseignement informatisés rend difficile le choix de ressources d'apprentissage appropriées aux étudiants. C'est ainsi que les systèmes de recommandations ont émergé dans le domaine de l'éducation.

Malgré les progrès considérables réalisés dans le domaine des systèmes de recommandation en général et pour le système éducatif en particulier au cours des deux dernières décennies, il est généralement admis qu'il existe encore de nombreux défis à relever et des questions controversées qui affectent l'état actuel des technologies de recommandation et nécessitent des efforts supplémentaires.

L'amélioration de la performance des prédictions fournies par les modèles de recommandations a toujours attiré l'attention de la communauté et reste toujours un défi à relever puisque plus la prédiction est performante, plus l'utilisateur est satisfait et la satisfaction de l'utilisateur procure un impact positif pour l'entreprise. La contribution de notre travail s'inscrit dans le cadre de l'amélioration de la précision des recommandations fournie. Nous nous intéressons particulièrement à l'optimisation de la précision des ressources pédagogiques recommandées aux apprenants.

Objectifs de la recherche

Les systèmes de recommandations peuvent être classés en trois types [7, 8, 9] : filtrage basé sur le contenu, filtrage collaboratif (FC) et filtrage hybride. L'approche basée sur le contenu se base sur la similarité du contenu à recommander avec le contenu déjà apprécié par l'utilisateur. Le filtrage collaboratif fournit des recommandations à un utilisateur sans forcément considérer le contenu des ressources, mais en se basant sur les appréciations de l'utilisateur afin de recommander les ressources qui ont été appréciées par d'autres utilisateurs ayant des profils similaires. Le filtrage hybride combine les deux approches : basé sur le contenu et le filtrage collaboratif.
Au cours de cette thèse, nous nous intéressons au filtrage collaboratif afin de recommander aux apprenants des ressources d'apprentissage personnalisées.

Le processus du FC comporte deux principales étapes : la création de groupe d'utilisateurs similaires et la prédiction des recommandations. La réussite de la première étape est très importante pour l'obtention de recommandations pertinentes. Dans cette perspective, plusieurs travaux [10, 11, 12, 13, 14, 15] ont adopté le clustering comme une étape de présélection d'utilisateurs similaires.

Pour tenter d'améliorer le processus du filtrage collaboratif, nous avons proposé une approche $CF-GT$ qui est constituée de deux principaux module. Le premier module nommé « SimilarUser » qui consiste à regrouper les utilisateurs similaires suivant un modèle de jeu coopératif en utilisant le concept de solutions SV. Une fois les groupes d'utilisateurs similaire obtenus, le deuxième module nommé «CFProcess » intervient. Ce dernier consiste à l'application du processus du filtrage collaboratif sur chaque groupe d'utilisateurs similaires obtenu par le module «SimilarUser » .

En effet, $CF-GT$ repose sur les faiblesses des travaux connexes existants et présente des caractéristiques remarquables : contrairement aux travaux qui prennent le clustering comme étape de présélection, notre approche prend en compte la notion intrinsèque d'un cluster. Le modèle $CF-GT$ est conçu comme une approche générique et a été mise en œuvre comme un prototype, qui peut être utilisé dans de nombreux domaines d'application, notamment le domaine éducatif.

Afin de répondre à l'objectif principal de cette thèse, nous avons adapté le modèle $CF-GT$ au domaine éducatif. Le modèle adapté au domaine éducatif nommé « Edu-CF-GT » utilise une base de données éducatives afin de recommander aux apprenants des ressources pédagogiques.

En raison du volume croissant d'apprenants et d'articles dans les systèmes d'apprentissage, les systèmes de recommandations reçoivent des données de haute dimension. Les ressources d'apprentissage sont souvent accompagnées d'une forme de métadonnées riches telles que du texte non structuré provenant d'un résumé ou d'une description de la ressource et des données catégorielles telles que le domaine éducatif e l'article. Par conséquent, le deep learning peut être utilisé pour extraire des représentations de caractéristiques riches du contenu des articles de manière automatisée. En conjonction avec les interactions non linéaires entre les apprenants et les ressources pédagogiques, une structure plus complexe des préférences des apprenants peut être extraite des données à haute dimension. Pour tenter d'optimiser le modèle « Edu-CF-GT » nous avons intégré le CNN au module « CFProcess ». Les résultats expérimentaux montrent que le modèle optimisé « CNN-Edu-CF-GT » est prometteur.

Contributions

— Notre première contribution [7] est un état de l'art des systèmes de recommandations. Nous avons établi une classification et discussions des différents travaux existants afin d'en tirer les lacunes. L'étude de l'état de l'art nous a permis de constater que l'amélioration de la précision des recommandations reste toujours un défi à relever. Ce constat nous a

motivé à contribuer en tentant d'optimiser les recommandations fournies dans un cadre générique.

— Notre deuxième contribution [8] consiste à l'amélioration des recommandations en utilisant la théorie des jeux. Cette contribution vise à améliorer la précision des recommandations fournies par le filtrage collaboratif en utilisant le concept de solution « la valeur de Shapley » de la théorie des jeux. Les expérimentations réalisées dans un cadre générique ont montré que les résultats obtenus sont prometteurs.

— Afin de répondre à l'objectif de notre thèse et prouver l'efficacité du modèle $CF - GT$ dans un cadre éducatif, nous l'avons adapté et testé avec une base de données pédagogique que nous avons-nous même conçue. Les résultats expérimentaux ont permis de conclure l'efficacité de notre approche dans le domaine éducatif.

— Le gros volume de données généré par les systèmes éducatifs rend difficile leur traitement. Le deep learning est en plein essor, sa capacité à traiter un gros volume de données non structurées à attirer les chercheurs de tous les domaines. Dans cette perspective, nous avons établi une troisième contribution [16] qui consiste en un état de l'art des systèmes de recommandations en deep learning. Le papier comporte nos travaux futurs qui consistent à l'implémentation et validation d'un modèle de filtrage collaboratif en CNN tout en exploitant plusieurs informations sur les étudiants (informations personnelles, pédagogiques, contextuelles, etc.)

— En se basant sur notre étude [16], nous avons optimisé le modèle [8] en intégrant le *CNN*. Les résultats expérimentaux dans un cadre éducatif sont prometteurs. Pour valoriser ce travail, un papier est en cours de rédaction.

Organisation de la thèse

La thèse est composée de cinq chapitres qui sont organisés comme suit :

— **Chapitre 1 « Fouille de données éducatives : méthodes et applications » :** ce chapitre est un état de l'art du domaine EDM. Nous commençons par une introduction générale d'EDM et ses différents objectifs. Ensuite, nous passons à la présentation de certains travaux connexes qui ont été réalisés récemment pour traiter les différentes applications d'EDM.

— **Chapitre 2 « Systèmes de Recommandations »** : dans le deuxième chapitre, nous nous concentrons sur les systèmes de recommandations en général. Nous commençons tout d'abord par introduire les systèmes de recommandations et les différents types de filtrage. Ensuite, nous abordons l'évaluation des systèmes de recommandations. Une synthèse des travaux récents des systèmes de recommandations est réalisée. Le chapitre est clôturé par une conclusion.

— **Chapitre 3 « Approche générique de filtrage collaboratif basée sur la théorie des jeux » :** le troisième chapitre présente les détails de notre contribution. Avant de donner les détails du modèle proposé, nous commençons tout d'abord par introduire la théorie des jeux qui a été exploitée dans le modèle proposé. Enfin, nous validons le modèle $CF-GT$. Les résultats expérimentaux sont présentés, ainsi que le protocole expérimental.

— **Chapitre 4 : « Application du modèle $CF-GT$ dans le domaine éducatif » :** Dans ce chapitre, nous adaptons le modèle $CF-GT$ au domaine pédagogique. Le nouveau modèle $Edu-CF-GT$ a été testé en utilisant une base de données pédagogique *EduTest* que nous avons conçue nous même, le chapitre comporte tous les détails.

— **Chapitre 5 : « Système de recommandation fondé sur la théorie des jeux et le deep learning » :** Dans ce chapitre, nous présentons les détails de l'amélioration du modèle $Edu-CF-GT$ en intégrant le *CNN*.

La thèse est clôturée par une conclusion générale. La conclusion générale récapitule tous nos travaux présentés dans cette thèse et inclut en dernier nos perspectives. La figure 1 sert de guide de lecture qui résume le plan de la thèse en illustrant l'organisation des différents chapitres ainsi que les liens entre eux.

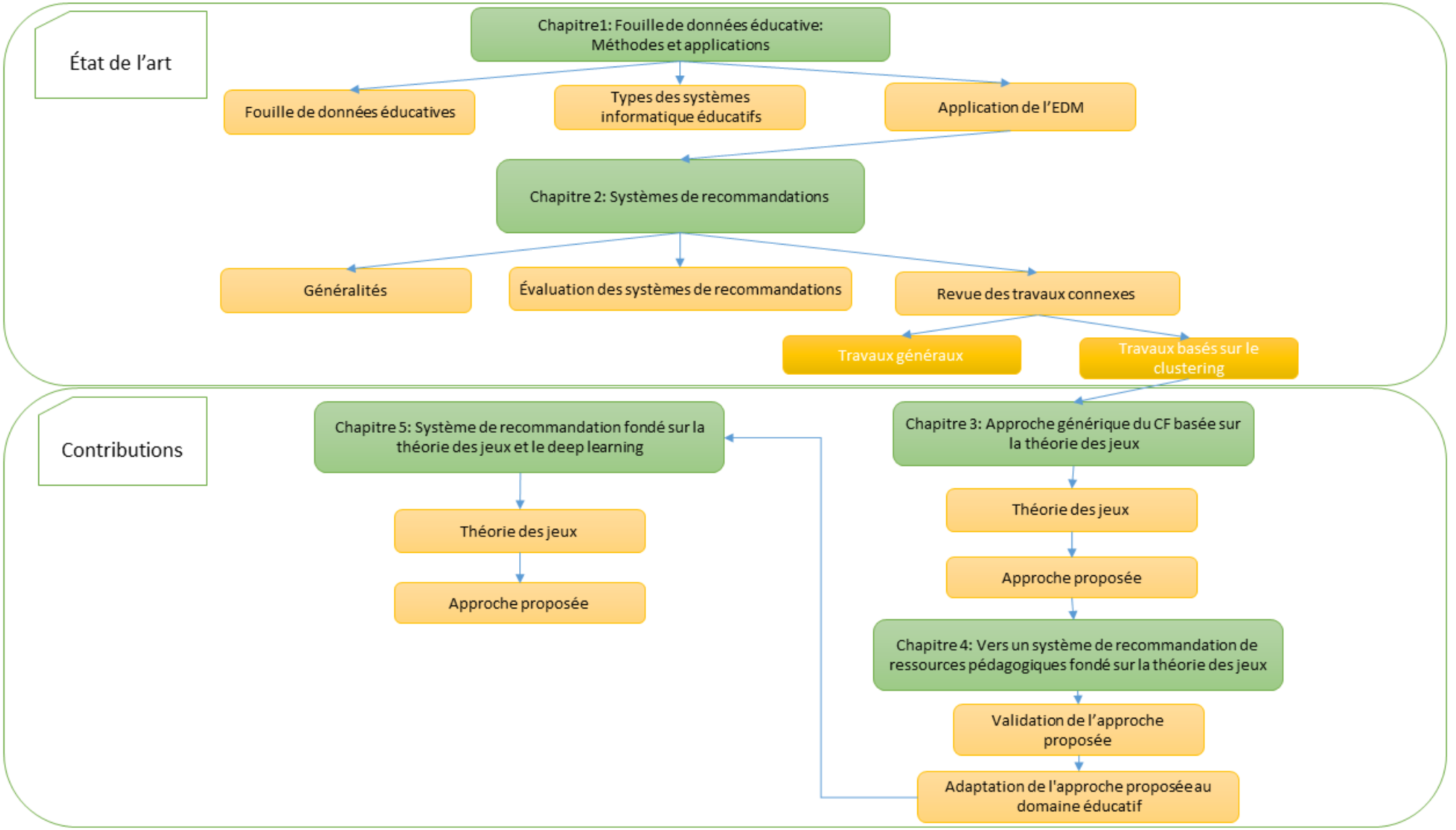

FIGURE 1 – Le synopsis de la thèse

Chapitre 1

Fouille de données éducatives : Méthodes et Applications

1.1 Introduction

La fouille de données éducatives ou Educational Data Mining (EDM) est un domaine de recherche qui vise à appliquer des techniques de la fouille de données aux données générées par les bases de données administratives et les interactions des étudiants avec les systèmes éducatifs, afin de mieux comprendre les comportements des étudiants pendant l'apprentissage et les contextes dans lesquels ils apprennent pour améliorer le processus d'apprentissage. Les bases de données administratives comprennent des informations académiques et démographiques telles que le sexe, l'âge et les notes scolaires. Les données générées par les interactions des étudiants avec les systèmes éducatifs peuvent provenir de différentes sources telles que les fichiers weblog qui enregistrent les comportements des apprenants, les images faciales qui contiennent les expressions faciales des étudiants pendant qu'ils apprennent, et les fichiers texte qui contiennent les opinions des étudiants sur les cours et le système éducatif, et ces opinions peuvent être exprimées dans des forums et des chats.

Le diagramme illustré dans la figure 1.1 montre le résultat obtenu après avoir tapé les mots-clés "educational-data- mining" dans la barre de recherche de la base de données de documents Scopus [1]. Ce diagramme montre le nombre d'articles publiés par an dans le domaine de l'EDM entre 2006 et 2021. Nous pouvons remarquer que le nombre d'articles publiés a considérablement augmenté au cours des dernières années, ce qui signifie qu'il s'agit d'un domaine de recherche très intéressant qui a intéressé et intéresse toujours les chercheurs

L'objectif de l'EDM est d'analyser les données provenant de différents environnements éducatifs afin de comprendre comment les apprenants interagissent avec l'environnement d'apprentissage et quel est l'impact de ces interactions sur leurs processus d'apprentissage. L'EDM peut être appliquée pour atteindre de nombreux objectifs tels que la prédiction et l'évaluation des performances d'apprentissage des étudiants, la compréhension des comportements des étudiants

1. https ://www-scopus-com.eressources.imist.ma/search/form.uri ?display=basic, visité le 12/09/2021

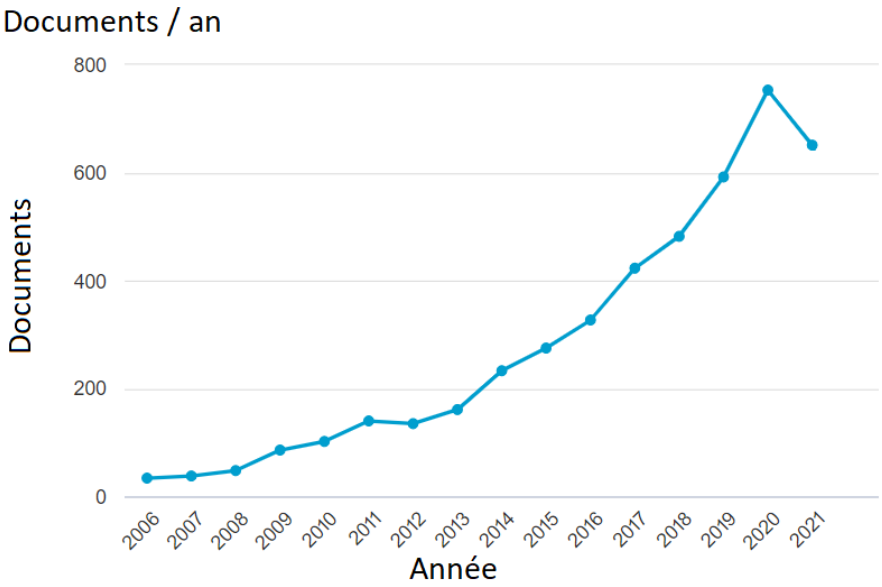

FIGURE 1.1 – Nombre d'articles publiés en EDM par an

et l'adaptation des recommandations d'apprentissage en fonction de ceux-ci, la détection des comportements et des problèmes d'apprentissage indésirables dans le but final d'améliorer le processus d'apprentissage et de guider l'apprentissage des étudiants en comprenant en profondeur les phénomènes éducatifs.

Dans ce chapitre, nous visons à présenter des généralités sur l'EDM et une revue de littérature de ces études. Notre objectif est de présenter les différentes méthodes qui ont été appliquées par les chercheurs dans leurs études ainsi que les applications auxquelles ils se sont intéressés. Fondamentalement, il existe de nombreuses applications dans le domaine de l'EDM qui ont pour objectif commun d'analyser les interactions de l'apprenant avec l'environnement d'apprentissage afin de comprendre l'impact de ces comportements sur les résultats de l'apprentissage.

Le reste de ce chapitre est organisé comme suit : la section 1.2 est consacrée à la définition du domaine EDM et à la présentation de ses objectifs. La section 1.3 présente les différents types d'enseignement assisté par ordinateur, avant de clôturer le chapitre, nous donnons un aperçu sur les différentes applications en EDM ainsi que différents travaux effectués dans la littérature.

1.2 Fouille de données éducatives

1.2.1 Définition

La fouille de données éducatives (EDM) est apparue au cours des deux dernières décennies en raison du grand volume de données éducatives généré par les systèmes éducatifs. Il s'agit de développer et d'appliquer des méthodes de la fouille de données ou Data Mining (DM) pour détecter des modèles dans de grandes quantités de données éducatives, et pour mieux comprendre les étudiants et leurs environnements d'apprentissage [17]. L'EDM utilise un ensemble varié de méthodologies qui ont été développées à l'origine pour la fouille et l'analyse de données [18].

Ces méthodologies sont dans de nombreux cas adaptées à la nature des données éducatives. Les méthodes de prédiction sont parmi les plus utilisées et comprennent la classification, la régression et les méthodes d'estimation des modèles à facteurs latents. Les méthodes de découverte de structures non supervisées, telles que le clustering, l'analyse factorielle et l'analyse des réseaux sociaux, sont également utilisées par l'EDM afin de découvrir les structures émergentes dans divers types de données éducatives. La fouille de règles d'association et d'autres types de méthodes de fouille de motifs sont utilisés pour découvrir les relations entre les différentes variables des données éducatives.

L'EDM est utilisée dans de nombreuses tâches telles que la construction d'un système de tutorat intelligent (STI), où diverses méthodes sont utilisées par le STI pour modéliser l'état actuel des connaissances des étudiants avec prise en compte de leurs différentes compétences. Il propose ensuite aux étudiants des exercices adaptés. L'EDM est également utilisée pour d'autres tâches intéressantes, notamment la conception automatisée de cours basée sur des données, la planification des diplômes universitaires et la recommandation de cours, la compréhension de l'impact du comportement social des étudiants sur leurs résultats scolaires, parmi bien d'autres tâches.

1.2.2 Objectifs d'EDM

Au cours des dernières années, de nombreuses études ont été réalisées dans le domaine d'EDM pour répondre à différents objectifs liés à l'amélioration du processus d'apprentissage. Selon [19] les objectifs d'EDM peuvent être classés en fonction du point de vue du problème traité et de l'utilisateur final (apprenant, éducateur, chercheur et administrateur).

— **Étudiants :** les étudiants peuvent bénéficier de l'EDM en recevant des recommandations personnalisées qui correspondent davantage à leurs exigences et à leurs besoins, ce qui peut contribuer à améliorer le processus d'apprentissage. Dans ce contexte, la modélisation de l'étudiant est une tâche importante de l'EDM qui vise à représenter les caractéristiques de l'apprenant en fonction desquelles les méthodes d'enseignement et le contenu sont adaptés.

— **Les éducateurs :** l'EDM peut les aider à améliorer le processus d'apprentissage en leur fournissant des informations sur les performances et les comportements des élèves et sur les facteurs qui peuvent affecter leurs résultats. Ces informations peuvent aider les éducateurs à superviser le processus d'apprentissage et à le personnaliser en fonction des exigences et des besoins des apprenants. Selon [19], disposer d'informations sur les performances et les comportements des apprenants peut aider les éducateurs à superviser le processus d'apprentissage et à l'adapter aux besoins des apprenants.

— **Les chercheurs :** les chercheurs bénéficient de l'EDM en tant que domaine de recherche émergent pour développer différentes méthodes qui peuvent être utilisées pour évaluer

l'efficacité de l'apprentissage. Ces méthodes peuvent être développées sur la base de techniques de la fouille de données dans le but de déterminer laquelle est la plus appropriée pour traiter chaque tâche ou problème éducatif spécifique.

— **Administrateurs :** le processus d'apprentissage peut être influencé par les administrateurs puisqu'ils sont en charge de la prise de décision et des allocations budgétaires. Ainsi, disposer d'informations significatives sur les étudiants peut aider à la fois les éducateurs à planifier un processus d'enseignement efficace et les administrateurs à allouer les ressources nécessaires à la gestion de ces processus d'enseignement.

Les applications de l'EDM ne se limitent pas à celles mentionnées ci-dessus ; au contraire, il y a encore plus d'applications possibles qui ont été décrites par différents chercheurs. Ci-dessous, quelques applications qui ont été présentées en commun dans différents articles [20, 21, 22]

- **Modélisation de l'étudiant :** la modélisation des étudiants est le composant central de tout système d'apprentissage en ligne adaptatif, car elle représente les caractéristiques des étudiants sur la base desquelles la personnalisation des recommandations de cours et des modes d'enseignement peut être effectuée. Les caractéristiques des étudiants peuvent être récupérées soit de manière statique en utilisant des questionnaires, soit de manière automatique en analysant les comportements des étudiants lors de leur interaction avec le système. Les caractéristiques des étudiants qui ont été largement modélisées comprennent les styles d'apprentissage, les performances de l'étudiant, l'état émotionnel, le niveau de connaissance et les facteurs cognitifs. La modélisation de ces caractéristiques permet aux logiciels de répondre aux différences des étudiants et d'améliorer considérablement leurs apprentissages. Différentes techniques de la fouille de données et des algorithmes d'apprentissage automatique ont été utilisés pour déterminer automatiquement les caractéristiques des étudiants afin d'automatiser la construction de modèles d'étudiants.
- **Prédiction des performances des étudiants :** la prédiction des performances des étudiants est l'une des tâches de l'EDM qui a été largement étudiée par les chercheurs ces dernières années. L'objectif est de prédire les notes finales de l'étudiant ou d'autres types de résultats d'apprentissage (tels que l'efficacité, l'évaluation, la réussite, la compétence, etc.). L'objectif principal de la prédiction de la performance des étudiants est d'estimer la capacité d'un étudiant à accomplir un programme d'apprentissage et à atteindre un objectif d'apprentissage spécifique. En outre, le fait de disposer d'informations sur les facteurs qui affectent les performances des étudiants aidera les établissements universitaires à adopter des stratégies préventives pour améliorer le processus d'apprentissage et réduire le taux d'échec et d'abandon.
- **Formulation de recommandations :** l'objectif est de déterminer le contenu (ou les tâches ou les liens) qui correspond le mieux aux besoins et aux préférences de l'étudiant, afin

d'adapter la recommandation de ces contenus d'apprentissage à chaque étudiant. De nombreuses techniques de la fouille de données ont été utilisées à cette fin.

- **Modélisation du comportement de l'étudiant :** la modélisation du comportement de l'étudiant est l'un des objectifs les plus importants de la modélisation des étudiants. Différents comportements peuvent être modélisés, tels que l'interaction de l'apprenant avec le système, les demandes de renseignements, les demandes d'aide, l'accès de l'apprenant au contenu de l'apprentissage, etc. L'objectif est de décrire ou de prédire différents traits de comportement afin d'adapter le système aux préférences des utilisateurs.
- **Fournir un retour d'information pour soutenir les parties prenantes :** l'objectif est d'aider les réalisateurs de cours/enseignants/administrateurs à analyser et à comprendre les comportements des étudiants pendant les cours. Le fait d'avoir un retour d'information sur le comportement de l'étudiant aidera les parties prenantes (administrateurs de cours et enseignants) à prendre des mesures et des décisions correctives et proactives pour bien organiser les ressources pédagogiques et améliorer le processus d'apprentissage. De nombreuses techniques de la fouille de données ont été utilisées pour extraire des informations cachées et utiles des données pédagogiques.

En général, l'objectif principal de l'EDM est l'acquisition d'informations détaillées sur le processus d'apprentissage afin d'améliorer la qualité de l'enseignement. Mais malgré tous les avantages offerts par l'EDM, elle souffre encore de nombreux inconvénients qui doivent être traités par les chercheurs dans ce domaine. Tout d'abord, pour assurer une pédagogie efficace et améliorer le processus d'apprentissage, il est recommandé aux chercheurs dans le domaine de l'EDM de connecter les méthodes de la fouille de données et d'analyse de données avec la cognition, la métacognition et la pédagogie afin de comprendre le processus d'apprentissage. Cette connexion est une tâche difficile car elle exige que les praticiens du domaine de l'EDM acquièrent de nouvelles compétences pédagogiques pour pouvoir apporter des contributions efficaces. Un autre problème qui affecte l'avancement des chercheurs dans ce domaine est la disponibilité des données éducatives, car elles ne sont pas toujours accessibles. De plus, toutes les données éducatives ne sont pas pertinentes et équivalentes [23]. Par conséquent, le défi consiste à prétraiter les données et à les valider pour qu'elles soient analysées afin de générer des informations utiles en temps réel et prédictives.

En outre, des questions éthiques se posent dans le domaine de l'EDM. Ces questions concernent la confidentialité des données personnelles. Il est donc nécessaire d'anonymiser les données éducatives avant leur utilisation par les chercheurs [24].

Une autre question intéressante et difficile est l'interopérabilité des données, de nombreux chercheurs étaient intéressés par la normalisation des données éducatives et l'amélioration de leur capacité à se déplacer entre différents systèmes éducatifs. Actuellement, les données éducatives ne sont pas effectivement interopérables et cette question nécessite encore des améliorations pour faciliter l'échange de données.

1.3 Type des systèmes éducatifs informatisés

1.3.1 Systèmes de gestion de l'apprentissage

Les systèmes de gestion de l'apprentissage ou Learning Management Systems (LMS) sont des plateformes éducatives basées sur le Web qui permettent la création, l'administration, la documentation, le suivi, l'établissement de rapports et la gestion de la diffusion de contenus éducatifs. Ils fournissent également différents types de forums de discussion et de lieux de travail pour permettre aux étudiants de communiquer et de partager facilement des informations tout en participant à un cours.

Les instructeurs peuvent utiliser le LMS pour créer différents groupes d'apprenants en fonction de leurs niveaux ou de leurs spécialités et leur attribuer un seul cours, ou un test, ou toute une série de matériels appelés parcours d'apprentissage, et suivre facilement les progrès de chaque apprenant pour l'aider à développer son niveau de connaissances.

Un large éventail d'organisations différentes, des écoles secondaires aux grandes entreprises, peuvent bénéficier de l'utilisation de systèmes de gestion de l'apprentissage en réduisant les dépenses liées à la location de salles de classe, au transport et à l'hébergement, au salaire des instructeurs et à l'impression des supports d'apprentissage. Les apprenants peuvent également tirer profit de l'utilisation des LMSs en accédant au matériel d'apprentissage n'importe où, à partir de n'importe quel appareil et à tout moment, afin de rafraîchir leurs connaissances. Ils ont également la possibilité de collaborer avec d'autres apprenants au cours du processus d'apprentissage.

Ces systèmes contiennent des outils de suivi des étudiants qui permettent aux instructeurs de visualiser des données statistiques sur les activités des étudiants. Ils permettent également d'enregistrer les activités historiques des étudiants dans des données log, comme l'écriture, la lecture, les tests, la participation à des forums, le chat et les commentaires sur les événements avec les autres étudiants. En plus des données de log, les LMSs fournissent une base de données relationnelle qui enregistre des informations sur les étudiants dans différents tableaux tels que les données académiques (notes), les informations personnelles (profil) et les données d'interaction des utilisateurs (rapports).

1.3.2 Cours en ligne ouverts et massifs

Les cours en ligne ouverts et massifs ou Massive Open Online Courses (MOOC) sont des cours en ligne ouverts qui peuvent être utilisés pour créer et dispenser des cours en libre accès, souvent dans le cadre de l'enseignement supérieur où le nombre d'étudiants pouvant s'inscrire à ces cours est massif (des centaines de milliers d'étudiants). La massification fait référence à la possibilité d'adapter les cours en fonction du nombre d'étudiants. La massivité résulte du fait que les cours sont présentés sans aucune restriction dans un environnement en réseau, ce qui les rend accessibles à un très grand nombre d'étudiants, dépassant le nombre total d'étudiants inscrits

dans certaines universités. Cet aspect de la massivité différencie les plateformes MOOC des LMSs, car ces derniers ne seraient pas en mesure de gérer un grand nombre d'étudiants. Ainsi, la grande quantité de données générées par les MOOCs nécessite l'utilisation de techniques de DM pour les traiter et les analyser. En plus de la masse, un autre trait qui caractérise les MOOCs est l'ouverture, qui permet aux apprenants d'être libres d'apprendre, de participer et d'accéder aux cours sans contraintes, sans frais de scolarité, sans identification et sans conditions préalables ou certifications. Les MOOCs se distinguent également des autres plateformes en ligne par l'importance qu'ils accordent à la diffusion de contenus au format vidéo.

Les MOOCs peuvent généralement être divisés en deux catégories : les xMOOCs et les cMOOCs. Le cMOOC est le modèle qui a été initialement créé par Downes [25]. Selon l'auteur, les cMOOCs sont "basés sur l'idée que l'apprentissage se fait au sein d'un réseau, où les apprenants utilisent des plateformes numériques telles que les blogs, les wikis, les plateformes de médias sociaux pour établir des connexions avec du contenu, des communautés d'apprentissage et d'autres apprenants pour créer et construire des connaissances".

Cette catégorie est similaire à l'apprentissage mixte où l'enseignement en direct et un environnement de cours virtuel sont utilisés pour fournir le contenu. Dans cette structure, le cours est programmé pour être enseigné à l'université par un professeur. Ce dernier planifie le cours et le construit dans un LMS (page Wiki LMS), puis il invite les étudiants à participer soit en classe, soit à distance. Les étudiants en salle de classe sont normalement inscrits au cours à l'université dans le cadre d'un diplôme accrédité, et participent au cours en ligne. Les participants en ligne, qui étaient probablement à la recherche de tels cours sur Internet, acceptent l'invitation et participent au cours en ligne sans crédit à la fin.

Contrairement au cMOOC qui combine l'enseignement en direct et l'apprentissage en ligne, le xMOOC dépend uniquement du modèle d'apprentissage en ligne/à distance. Seuls les participants en ligne sont présentés dans le xMOOC.

1.3.3 Systèmes de tutorat intelligents

Les systèmes de tutorat intelligents ou Intelligent Tutoring Systems (ITS) se distinguent des systèmes éducatifs en ligne traditionnels par leur capacité à fournir un apprentissage personnalisé aux étudiants en fonction de leurs caractéristiques telles que leur niveau de connaissances, leur style d'apprentissage et leur progression dans le contenu de l'apprentissage.

Selon [26], l'architecture d'un STI se compose principalement de trois modèles architecturaux majeurs. Le modèle de domaine, qui vise à représenter le contenu à apprendre de manière que le système puisse l'utiliser pour le raisonnement. Il existe de nombreuses représentations possibles, notamment l'ontologie, les réseaux sémantiques, les règles de production et les contraintes. Le choix de la représentation à adopter dépend en partie de l'utilisation qui en sera faite. Le modèle de l'étudiant, qui est la composante centrale d'un ITS puisqu'il représente les informations de l'étudiant qui sont utilisées pour adapter les recommandations du système aux besoins individuels. Un grand nombre d'informations peuvent être représenté dans un modèle

d'étudiant, comme le style d'apprentissage, le niveau de connaissances, les facteurs affectifs, les expériences d'apprentissage antérieures et les progrès réalisés. Le modèle de tutorat reçoit des informations des modèles du domaine et de l'étudiant et prend des décisions sur les stratégies et les actions de tutorat. Sur la base de connaissances fondées sur des principes, il doit décider s'il faut intervenir ou non, et si oui, quand et comment. La planification du contenu et de l'enseignement fait également partie des tâches du modèle de tutorat.

1.3.4 Systèmes hypermédias adaptatifs

Les systèmes hypermédias adaptatifs ou Adaptive Hypermedia Systems (AH) sont le résultat de la combinaison des ITSs avec des matériaux d'apprentissage organisés en hypermédia. Les ITSs ont d'abord été conçus pour aider l'étudiant à résoudre des problèmes sans se concentrer sur la fourniture de contenu d'apprentissage. Avec la croissance des capacités informatiques, les chercheurs ont découvert qu'il est possible de développer un système qui, en plus de conserver le principe de personnalisation des ITS, peut également fournir du matériel d'apprentissage sous forme électronique. Ce système, appelé système d'images, de vidéos et d'hypermédia adaptatif (AH), vise à construire un modèle d'étudiant selon lequel l'adaptation de l'hypermédia aux besoins de cet étudiant peut se faire, par exemple, pour adapter le contenu d'une page hypermédia aux connaissances et aux objectifs de l'utilisateur, ou pour lui suggérer les liens les plus pertinents à suivre.

Deux types de techniques ont été largement utilisés par les développeurs pour construire un système hypermédia adaptatif : la présentation adaptative et le support de navigation adaptatif. L'idée derrière les techniques de présentation adaptative est d'adapter le contenu d'une page hypermédia aux connaissances actuelles de l'étudiant, à ses objectifs et à d'autres caractéristiques stockées dans le modèle de l'étudiant. Par exemple, un utilisateur ayant un niveau avancé peut recevoir un contenu plus détaillé et plus développé, tandis qu'un apprenant débutant peut recevoir des descriptions supplémentaires. Le contenu d'une page hypermédia peut-être non seulement un texte comme dans les systèmes hypertextes classiques, mais aussi un ensemble de divers éléments multimédias tels que des images, des vidéos et des graphiques. L'objectif de l'aide à la navigation adaptative est d'adapter la manière de présenter les liens aux objectifs, aux connaissances et aux autres caractéristiques des utilisateurs afin de les aider à trouver leur chemin dans l'hyperespace.

1.4 Applications de la fouille de données éducatives

Comme nous l'avons présenté dans l'introduction, nous distinguons deux applications principales de l'EDM : la modélisation de l'étudiant et le système d'aide à la décision. Chacune des applications comprend des taches. Pour la modélisation de l'étudiant, nous distinguons : prédire la performance, le comportement et les caractéristiques de l'étudiant. L'application de l'aide à la décision comprend : la fourniture d'un feedback aux parties prenantes, la modélisation de

la structure du domaine et l'étude de l'effet des supports pédagogiques sur l'apprentissage des étudiants lors de l'utilisation d'un système d'apprentissage.

Plusieurs études ont été menées dans la littérature sur les différentes applications de l'EDM citons : [2, 18, 22, 27, 28]. Après l'analyse de l'existant, nous avons remarqué que les taches les plus étudiées sont : la prédiction de la performance, comportement et caractéristiques de l'étudiant.

Le but de cette section est la présentation des études d'EDM les plus récentes relatives aux deux principales applications : la modélisation de l'étudiant et le système d'aide à la décision.

1.4.1 Modélisation de l'apprenant

Les dernières études menées pour traiter le problème de la modélisation des étudiants se sont concentrées sur trois contributions principales : l'analyse du comportement de l'étudiant et la prédiction de ses caractéristiques et de ses performances.

1.4.1.1 L'analyse du comportement des étudiants

Dans cette section, nous présentons quelques travaux récents de l'EDM qui traitent l'analyse des comportements des apprenants. Nous avons divisé les travaux en quatre catégories selon l'objectif final :

— **Étudier l'impact des comportements des apprenants sur leurs performances**

Parmi les études qui se sont concentrées sur l'analyse des comportements des étudiants, il y a quelques contributions qui visaient à déterminer le comportement de l'étudiant et ensuite à l'associer à la performance de l'étudiant afin de comprendre la relation entre les comportements des apprenants et leurs réalisations.

Par exemple, dans [29] les auteurs ont construit un framework pour évaluer l'efficacité du processus d'apprentissage des étudiants en analysant leurs comportements, où les comportements des étudiants sont enregistrés sous forme de fichier log et les opérations de l'utilisateur ont été symbolisées sous forme de symboles séquentiels. Après avoir déterminé les séquences d'actions, des méthodes d'exploration de processus ont été appliquées pour révéler les caractéristiques générales. Enfin, les modèles de comportement des étudiants résultant de l'étape d'exploration de processus ont été associés aux scores d'affectation des étudiants afin de comprendre la relation entre les comportements des apprenants et leur effet d'apprentissage. Dans [30], les auteurs ont construit un modèle pour analyser les interactions des étudiants enregistrées dans un fichier log d'une plateforme d'apprentissage virtuel. Tout d'abord, les auteurs ont appliqué un processus de caractérisation pour transformer les fichiers log en informations pertinentes sur le plan académique et ont mesuré des variables qui permettent d'interpréter le comportement de l'étudiant. Ensuite, un algorithme supervisé a été appliqué aux informations codées afin de regrouper les étudiants

en fonction de leurs comportements, et enfin, les informations relatives aux performances académiques des étudiants sont utilisées pour étiqueter les différents groupes obtenus.

— **Déterminer les conditions et les exigences d'apprentissage des apprenants**

D'autres auteurs se sont intéressés à la construction de modèles qui analysent et prédisent les comportements des apprenants afin de déterminer leurs conditions d'apprentissage et le contenu d'apprentissage qui correspond à leurs besoins.

Par exemple, les auteurs de [31] ont proposé une approche qui vise à fournir aux apprenants un contenu adapté en fonction de leurs comportements. Pour ce faire, les auteurs ont appliqué le $k-means$ et l'arbre de décision. L'algorithme $k-means$ a été appliqué pour regrouper les étudiants en fonction des fréquences d'accès au contenu électronique. Ensuite, l'arbre de décision a été appliqué aux groupes obtenus afin d'analyser en profondeur les comportements des apprenants pendant l'apprentissage. La compréhension de ces comportements aidera les enseignants à décider quel est le contenu d'apprentissage à fournir aux étudiants pendant qu'ils se préparent à l'examen. Dans [32], les auteurs ont analysé les données du campus afin de construire un modèle qui permet de décrire et de prédire les comportements des étudiants. Le modèle construit peut exploiter efficacement les règles de consommation, les habitudes de vie et les conditions d'apprentissage des étudiants.

— **Découvrir le modèle de comportement d'apprentissage**

Une autre contribution intéressante proposée par de nombreux auteurs est la découverte du modèle de comportement d'apprentissage.

Par exemple, les auteurs de [33] ont cherché à appliquer des techniques de *DM* sur des données générées à la fois à partir de l'ensemble de données académiques et du fichier log d'une plateforme d'apprentissage en ligne afin de créer un modèle pour découvrir des modèles de comportement d'apprentissage. En outre, les auteurs de [34] ont eu pour objectif de déterminer les comportements et les réactions des étudiants afin d'automatiser le feedback dans un environnement d'apprentissage en ligne. Pour ce faire, les auteurs ont combiné le Local Process Mining avec d'autres techniques qui travaillent avec des données non structurées pour découvrir des modèles locaux.

— **Déterminer les comportements indésirables**

Contrairement aux études qui visaient à prédire les comportements d'apprentissage des étudiants, d'autres études se sont concentrées sur la détermination des comportements indésirables.

Par exemple, dans [35], les auteurs ont cherché à mesurer les problèmes de comportement des apprenants en ligne en concevant une échelle de comportement d'apprentissage en ligne. Pour ce faire, les auteurs ont proposé un questionnaire contenant 24 questions, puis ils ont appliqué diverses techniques statistiques aux réponses données par les étudiants

pour évaluer les éventuels problèmes de comportement d'apprentissage des apprenants en ligne.

Le tableau 1.1 résume les travaux suscités.

1.4.1.2 Prédiction des caractéristiques des apprenants

Dans cette section, nous présentons quelques travaux récents d'EDM pour prédire les caractéristiques des apprenants. Le style d'apprentissage et l'état émotionnel font partie des caractéristiques qui ont été largement étudiées par les chercheurs ces dernières années.

— **Analyse des facteurs affectifs des apprenants : application des méthodes d'analyse des sentiments**

Pour analyser les états émotionnels des étudiants, de nombreux chercheurs ont appliqué des méthodes d'analyse de sentiments aux commentaires postés par les apprenants sur différentes plateformes éducatives.

Par exemple, les auteurs de [36] ont cherché à appliquer des techniques de *DM* aux commentaires postés par les étudiants sur des groupes ou des communautés Facebook, afin de déterminer les opinions des étudiants sur le programme d'études de l'université/du collège et sur les activités extrascolaires. Les auteurs ont utilisé l'algorithme $TF-IDF$ pour mesurer la similarité et calculer le sentiment pour chaque mot et décider si le résultat est constitué de messages positifs, de messages négatifs ou de messages neutres. Une autre contribution qui exploite des techniques d'analyse de sentiments est présentée dans [37] où les auteurs ont proposé un classifieur d'analyse de sentiments qui peut déterminer les sentiments des étudiants tout en utilisant des outils de gamification dans un cours éducatif. Pour ce faire, les auteurs ont créé un dictionnaire Arabic Chat Alphabet (ACA) pour convertir les avis anglais-arabe en anglais. Deux ensembles de données ont été utilisés dans cette étude, l'un contient les opinions de 700 étudiants qui ont accepté d'examiner la gamification dans l'apprentissage, tandis que l'autre contient les opinions de 300 étudiants qui ne sont pas intéressés à appliquer la gamification. Après avoir créé le dictionnaire de l'ACA et trouvé les ensembles de données, les auteurs ont appliqué des méthodes d'analyse de sentiments pour déterminer les sentiments des étudiants dans chaque ensemble de données, puis différents classifieurs ont été appliqués pour prédire les sentiments des étudiants. Les expérimentations ont montré que les classifieurs naïfs de Bayes ont obtenu de meilleurs résultats et que l'utilisation de la gamification a un effet positif sur le processus d'apprentissage. Les auteurs de [38] ont combiné un modèle de *ML* et une méthode d'analyse lexicale à haute fréquence pour analyser les commentaires négatifs provenant de MOOC. Les émotions négatives prédites des étudiants ont aidé à analyser les facteurs affectant le développement du MOOC.

— **Analyse des facteurs affectifs des apprenants : application des méthodes de classification**

TABLE 1.1 – Résumé des travaux qui traitent le comportement des apprenants

Référence	Année	Données	Méthodes	Objectif
[29]	2019	Données de log (comportements des apprenants)	Techniques d'exploration de processus, exploration de motifs séquentiels	Comprendre la relation entre les comportements des apprenants et leur effet d'apprentissage.
[30]	2019	Données de log (comportements des apprenants)	Réseaux de neurones auto-organisés	Regrouper les apprenants en fonction de leurs comportements, puis indiquez les résultats scolaires des élèves dans les groupes obtenus.
[31]	2020	Données de log (comportements des apprenants)	K-means et arbre de décision	Trouver le contenu d'apprentissage qui devrait être recommandé à chaque élève en fonction de son comportement.
[32]	2019	Données du campus (consommation des étudiants, résultats scolaires, gestion des présences, emprunt de livres)	Technologie de la fouille de données	Examiner les règles de consommation, les habitudes de vie et les conditions d'apprentissage des apprenants.
[33]	2018	Base de données académiques + données de log (comportements des apprenants)	Arbre de décision et techniques des réseaux bayésiens.	Découvrir le modèle de comportement d'apprentissage.
[34]	2019	Données de log (comportements des apprenants)	Techniques d'exploration des processus locaux	Déterminer les comportements et les réactions des étudiants pour automatiser le retour d'information dans l'environnement d'apprentissage en ligne.
[35]	2020	Réponses aux questionnaires	Analyse statistique	Mesurer les problèmes de comportement des apprenants en ligne

Outre les techniques d'analyse de sentiments, les méthodes de classification sont également très utilisées par de nombreux chercheurs pour traiter les tâches d'analyse de sentiments.

Par exemple, les auteurs [39] ont cherché à comparer l'efficacité du *ML*, deep learning et d'un algorithme évolutionnaire pour prédire les opinions des étudiants sur les performances des enseignants dans leurs différents cours. Pour ce faire, les auteurs ont créé un dictionnaire émotionnel appelé *SentiDict* où les mots ont été étiquetés avec des émotions centrées sur l'apprentissage. Ensuite, deux jeux de données nommés $sentiTEXT$ et *eduSERE*, qui contiennent les opinions des étudiants, ont été créés à partir des commentaires postés sur différentes plateformes éducatives. Après avoir construit le dictionnaire *SentiDic* et les deux ensembles de données, l'étape suivante visait à catégoriser chaque phrase avec des étiquettes positives et négatives pour le corpus $sentiTEXT$ et des étiquettes engagement, excitation, ennui et frustration dans le cas du corpus *eduSERE*. Les résultats expérimentaux ont montré que l'algorithme évolutionnaire EvoMSA et le *DL* ont surpassé les méthodes traditionnelles d'apprentissage automatique pour classer les opinions dans les contextes d'apprentissage. Les auteurs [40] ont construit un système qui permet la classification automatique des commentaires des étudiants. À cette fin, les auteurs ont créé des données de feedback d'étudiants universitaires qui contiennent 5000 phrases, puis les phrases collectées ont été annotées avec 3 étiquettes : positive (POS), négative (NEG) et neutre (NEU). Ils ont ensuite appliqué trois classifieurs, à savoir Naïve Bayes, Maximum Entropy et Support Vector Machine, aux données annotées. Les résultats des expériences ont montré que l'algorithme d'entropie maximale a surpassé les deux autres algorithmes avec le meilleur score de 91,36

— **Proposer des approches automatiques pour déterminer les styles d'apprentissage des apprenants**

En plus de l'état émotionnel, le style d'apprentissage est aussi largement étudié par les chercheurs. De nombreux travaux ont été réalisés pour déterminer le style d'apprentissage automatiquement.

Par exemple, les auteurs [41] ont proposé une approche pour déterminer le style d'apprentissage automatiquement. L'approche proposée se compose de deux étapes principales, à savoir la construction de la table de profil et l'unité de clustering. Dans la première étape, les auteurs se sont appuyés sur le contenu des objets de données sélectionnés par les apprenants afin de trouver les mots-clés pertinents, puis les mots-clés extraits ont été mis en correspondance avec un ensemble de valeurs d'attributs prédéterminées, et enfin, les valeurs d'attributs déterminées ont été combinées pour construire un tableau de profil d'apprenant. Dans la deuxième étape, les auteurs ont créé un ensemble de données d'apprentissage où ils ont attribué un indice de style d'apprentissage à chaque tableau de profil d'apprenant construit sur la base du modèle de style d'apprentissage de Felder et Silverman (FSLSM). Enfin, l'ensemble de données d'apprentissage préparé a été utilisé

pour appliquer la classification NBTree en conjonction avec le classifieur de pertinence binaire afin de construire un modèle de classification qui permet d'attribuer des étiquettes à chaque ligne du tableau de profil de l'apprenant en utilisant les dimensions du FSLSM. Les auteurs [42] ont cherché à révéler les styles d'apprentissage dominants des apprenants adultes qui rejoignent l'université en tant qu'étudiants matures. Dans un premier temps, l'auteur a distribué le questionnaire sur les styles d'apprentissage (LSQ) de Honey et Mumford à un ensemble d'étudiants, puis le logiciel NVIVO et cinq étapes d'analyse qualitative des données ont été appliquées aux réponses générées. Les résultats obtenus à partir de l'analyse du questionnaire LSQ ont montré qu'il existe trois catégories de style d'apprentissage, à savoir les réflecteurs, les théoriciens et les pragmatiques.

— **Évaluation des corrélations entre les styles d'apprentissage des apprenants et leurs comportements**

Une autre contribution présentée dans [43], qui vise à déterminer s'il existe un lien entre les styles d'apprentissage des étudiants et leurs comportements lorsqu'ils interagissent avec les ressources et les activités disponibles dans les environnements d'apprentissage virtuel (EAV). Pour ce faire, les auteurs ont utilisé un questionnaire afin d'obtenir les réponses des étudiants et de les analyser pour identifier les styles d'apprentissage prédominants de chaque étudiant. Après avoir identifié les styles d'apprentissage des étudiants, les auteurs ont consulté la base de données du EAV afin de collecter des données sur les comportements des étudiants, afin d'évaluer les corrélations entre les styles d'apprentissage et le comportement des étudiants en utilisant la technique de régression linéaire. Les expériences ont montré qu'il y avait de faibles corrélations entre les comportements des étudiants et leurs styles d'apprentissage.

— **Recommandation de contenu personnalisé basé sur les styles d'apprentissage des apprenants**

Le style d'apprentissage peut être considéré comme une caractéristique clé en fonction de laquelle la personnalisation de la recommandation de contenu peut être effectuée. Dans ce contexte, les auteurs [44] se sont basés sur les styles d'apprentissage et la taxonomie de Bloom afin de construire des systèmes d'apprentissage en ligne personnalisés basés sur le reinforcement learning où le module d'adaptation est développé en utilisant le deep Q-learning multitâche.

Le tableau 1.2 résume les travaux suscités ci-dessus.

1.4.1.3 Prédiction de la performance des apprenants

Dans cette section, nous présentons quelques travaux récents d'EDM pour prédire la performance des apprenants dans des environnements d'apprentissage par ordinateur.

— **Prédire les notes finales et les résultats des étudiants**

TABLE 1.2 – Résumé des travaux qui traitent la prédiction des caractéristiques des apprenants

Réf	Année	Données	Méthodes	Objectif
[36]	2019	Postes/commentaires des étudiants dans les groupes Facebook	Techniques de fouille de texte	Déterminer l'opinion des étudiants sur le programme d'études de l'université/du collège et sur les activités extra-professionnelles.
[37]	2020	Opinions/feedbacks des étudiants	Techniques d'analyse et de classification des sentiments	Déterminer les sentiments des étudiants lors de l'utilisation d'outils de gamification dans un cours éducatif.
[38]	2020	Commentaires des étudiants	Modèle de machine learning et méthode d'analyse lexicale à haute fréquence.	Déterminer les facteurs qui affectent le développement des MOOC en analysant les commentaires négatifs postés par les étudiants.
[39]	2020	Commentaires d'étudiants extraits de différentes plateformes éducatives.	machine learning, deep learning, et l'algorithme évolutif EvoMSA	Prédire l'opinion des étudiants sur la performance des enseignants dans leurs différents cours.
[40]	2020	Opinions/feedbacks des étudiants	Naïve Bayes, Entropie maximale et Machine à vecteurs de Support.	Classer automatiquement les commentaires des élèves en positifs (POS), négatifs (NEG) et neutres (NEU).
[41]	2009	Les objets d'apprentissage sélectionnés par les apprenants	Algorithme de classification NBTree, classifieur de pertinence binaire.	Proposer une approche pour déterminer automatiquement le style d'apprentissage.
[42]	2020	Réponses au questionnaire sur le style d'apprentissage	Le logiciel NVIVO et l'analyse de données qualitatives	Révéler les styles d'apprentissage dominants des apprenants adultes qui rejoignent l'université en tant qu'étudiants adultes.
[43]	2020	Réponses au questionnaire sur le style d'apprentissage + ensemble de données sur l'environnement d'apprentissage virtuel (interactions des étudiants)	Régression linéaire	Déterminer la relation entre les styles d'apprentissage et le comportement des étudiants lors de l'interaction avec l'environnement d'apprentissage virtuel.
[44]	2020	Base de données du LMS (interactions des élèves)	Reinforcement learning	Utiliser le style d'apprentissage afin de créer un système d'apprentissage en ligne personnalisé basé sur le reinforcement learning

La prédiction de la performance des étudiants est l'une des tâches les plus étudiées au cours de la dernière décennie. L'objectif est d'estimer la valeur d'une variable qui décrit les performances des étudiants au cours d'un processus d'apprentissage. Cette valeur peut être une note finale ou un résultat final.

Un exemple de prédiction du résultat final est présenté dans [45] où les auteurs ont créé un classifieur appelé SPRAR (student prediction performance using Relational Association Rules) pour prédire le résultat final d'un étudiant dans une certaine discipline académique. Les auteurs ont utilisé les notes obtenues par les étudiants pendant le semestre académique comme attributs d'entrée, et la méthode des règles d'association relationnelle pour trouver la relation entre les attributs de données et ensuite déterminer les règles qui peuvent être utilisées pour classer une nouvelle instance soit comme un résultat positif ou négatif. Dans [46], les auteurs ont cherché à construire un modèle de prédiction pour déterminer la note finale de l'étudiant. Les auteurs ont utilisé l'analyse de régression afin de construire le modèle, où les variables d'entrée sont les comportements des étudiants et les variables de sortie sont les notes finales des étudiants. Afin de ne considérer que les variables les plus importantes, les auteurs ont utilisé l'analyse en composantes principales (ACP), où 35 variables ont été considérées, mais après avoir appliqué l'ACP, 9 indicateurs les plus importants ont été trouvés. Les auteurs [47] ont également cherché à prédire les notes finales des étudiants afin d'identifier les étudiants qui pourraient avoir besoin d'aide à un stade précoce du cours.

— **Déterminer les facteurs qui affectent les résultats des étudiants**

Tout en prédisant la performance des étudiants, il est nécessaire de déterminer les facteurs qui affectent davantage leur réussite, à cet égard, de nombreuses contributions ont considéré cette exigence.

Par exemple, les auteurs [48] ont cherché à appliquer diverses techniques de *ML* à un ensemble de données afin d'identifier les facteurs qui influencent le processus d'apprentissage. Les auteurs ont constaté que le nombre de fois où les étudiants ont accès aux ressources disponibles dans l'environnement d'apprentissage virtuel était considéré comme un facteur clé affectant les performances des étudiants. Les auteurs [49] ont constaté que les attributs qui affectent le plus les performances académiques des étudiants appartiennent aux catégories suivantes : données démographiques, informations sur les performances antérieures de l'étudiant, informations sur le cours et l'instructeur, et informations générales sur l'étudiant. Les résultats expérimentaux ont également révélé que l'algorithme Random Forest (RF) est le plus performant pour prédire les performances des étudiants.

— **Étude de la relation entre les performances des étudiants et leurs comportements**

L'étude de la relation entre les performances des étudiants et leurs comportements est une autre tâche à laquelle se sont attelés les chercheurs.

Par exemple, les auteurs [50] ont constaté qu'il existe une forte corrélation entre le comportement des apprenants et leurs performances académiques. Pour ce faire, ils ont proposé une méthodologie qui se compose de cinq phases ; en commençant par la collecte des données du système d'apprentissage en ligne, puis le prétraitement des données collectées, ensuite le mécanisme de discrétisation a été appliqué pour convertir les performances académiques des étudiants de valeurs numériques en valeurs nominales, puis le meilleur ensemble de caractéristiques a été sélectionné en utilisant la méthode de gain d'information, et enfin, cinq classifieurs ont été appliqués pour prédire les performances des étudiants. Dans [51], les auteurs ont appliqué la méthode d'analyse de chemin pour étudier la relation entre le courage, l'auto-efficacité, les objectifs d'orientation vers la réussite et la performance scolaire. L'ensemble de données utilisé dans cette étude a été collecté à partir des réponses des étudiants à une enquête en ligne. Les résultats montrent que la persévérance des étudiants et leurs objectifs prédéterminés d'orientation vers la réussite affectent positivement leurs performances scolaires, tandis qu'il existe une corrélation négative entre les objectifs d'évitement et les performances scolaires.

Le tableau 1.3 résume les travaux suscités.

1.4.2 Aide à la décision

Les applications qui appartiennent à ce groupe visent essentiellement à aider les parties prenantes à prendre des décisions pour améliorer le processus d'apprentissage. Des exemples de ces applications sont la sélection de cours et la génération de recommandations, la fourniture de rapports et la création d'alertes pour les parties prenantes.

1.4.2.1 Recommandation de cours

Il est évident que la sélection des cours et génération de recommandations est une tâche qui est pour l'intérêt des étudiants, puisque cette tâche les aide à recevoir les cours les plus appropriés à leurs profils et besoins.

Dans cette section, nous présentons quelques travaux récents d'EDM pour la sélection de cours et la génération de recommandations.

Citons comme exemple de ce domaine d'application [52] où les chercheurs ont appliqué plusieurs techniques basées sur le DM et l'analyse de l'apprentissage pour développer un modèle de recommandation pour sélectionner les cours appropriés aux profils des étudiants. L'ensemble de données utilisé dans l'étude est les informations personnelles et pédagogiques des étudiants inscrit entre 2015 et 2019 dans leur université.

Un autre travail a été mené dans [53] où les auteurs ont construit un framework pour aider les étudiants à sélectionner les cours appropriés sur la base de leurs sujets précédemment suivis et des notes obtenues. L'idée est de calculer la similarité entre les cours précédemment étudiés et les cours optionnels actuels en utilisant les titres et les descriptions des cours. Une contribution

TABLE 1.3 – Résumé des travaux qui traitent la prédiction de la performance des apprenants

Référence	Année	Données	Méthodes	Objectif
[45]	2019	Caractéristiques académiques (notes)	Règles d'association relationnelles	Prédire le résultat final d'un étudiant dans une discipline académique donnée.
[46]	2018	Données de log (comportements des apprenants)	Fouille de données avec analyse de régression	Déterminer la note finale de l'étudiant
[47]	2020	Base de données académiques + Données de log (comportements des apprenants)	Techniques de classification	Prédire les notes finales des étudiants afin d'identifier les étudiants qui pourraient avoir besoin d'aide.
[48]	2021	Caractéristiques démographiques + Base de données de l'environnement d'apprentissage virtuel (interactions des étudiants)	Modèle basé sur les arbres et réseaux neuronaux artificiels	Identifier les facteurs qui influencent le processus d'apprentissage.
[49]	2020	Base de données du système d'information (caractéristiques personnelles, démographiques et académiques)	Sept algorithmes d'apprentissage supervisé	Déterminer les facteurs qui affectent le plus les performances des élèves.
[50]	2020	Base de données du LMS (caractéristiques démographiques, antécédents académiques et caractéristiques comportementales).	Techniques de classification	Proposer un nouveau modèle de prédiction des performances des étudiants.
[51]	2020	Réponses à une enquête en ligne	Méthode d'analyse des chemins	Étudier la relation entre le courage, l'auto-efficacité, les objectifs d'orientation vers la réussite et la performance scolaire.

TABLE 1.4 – Résumé des travaux qui traitent la recommandation de cours

Réf.	An.	Données	Méthode	Objectif	Limites
[52]	2021	Données personnelles et résultats pédagogiques des étudiants de l'université.	Techniques de DM et l'analyse 'apprentissage, collaborative filtering	Recommandations de cours appropriés aux profils des étudiants.	Le jeu de données de test est très petit. Aucune comparaison réalisée.
[53]	2020	Cours et notes des étudiants du département d'ingénierie informatique de leur université	Traitement automatique des langues, collaborative filtering	Aider les étudiants à choisir les cours appropriés en fonction des matières qu'ils ont déjà suivies et des notes obtenues.	Le modèle n'a pas été testé.
[55]	2017	Support d'apprentissage (document) de la plateforme e-learning de l'université	Clustering (K-means)	Aidez les élèves à retrouver les connaissances appropriées.	La précision des prédictions et la pertinence des recommandations n'ont pas été testées.
[54]	2016	Les inscriptions antérieures de l'étudiant à des cours dans une université au Canada.	Processus d'analyse hiérarchique	Prédire les inscriptions aux cours en respectant les préférences des étudiants.	Le modèle proposé n'est pas scalable.

similaire est présentée dans [54] où les auteurs ont cherché à prédire les inscriptions aux cours en respectant les préférences des étudiants. Les préférences des étudiants ont été extraites des sources disponibles dans les systèmes institutionnels d'information sur les étudiants et analysées à l'aide du processus d'analyse hiérarchique.

Un autre travail qui traite la sélection et la recommandation de cours est présenté dans [55]. Dans ce travail, les auteurs ont eu pour objectif de proposer une approche qui permet de sélectionner et d'annoter les supports d'apprentissage, puis d'organiser les supports d'apprentissage sélectionnés pour retrouver ceux qui sont utiles à recommander. L'approche proposée se compose de deux modules, à savoir le module serveur et le module client. Le module serveur a pour but de collecter et d'annoter les ressources d'apprentissages, puis de les indexer et de les structurer sous la forme d'une représentation multidimensionnelle des connaissances à l'aide de XML et d'une ontologie. Le module client a pour but de récupérer les informations utiles en faisant correspondre la requête de l'utilisateur et les ressources pédagogiques indexées.

Le tableau 1.4 résume les travaux suscités.

TABLE 1.5 – Résumé des travaux qui traitent la présentation de rapport

Référence	Année	Données	Méthodes	Objectif
[19]	2013	Données des quiz	Algorithme d'extraction de règles d'association	Découvrir les règles et fournir aux instructeurs des informations intéressantes.
[56]	2019	Données institutionnelles	Techniques de fouille de données.	Découvrir des connaissances et réaliser de multiples tâches académiques en fonction des besoins des différentes parties prenantes.

1.4.2.2 Présentation de rapport

Dans cette sous-section, nous présentons deux travaux sur l'EDM qui traitent l'application de présentation de rapports. L'objectif de cette catégorie d'applications est de fournir aux éducateurs et aux administrateurs un retour d'information et des renseignements utiles pour les aider à prendre des décisions.

Un exemple de cette catégorie d'applications est présenté dans [19] où les auteurs ont appliqué un algorithme de fouille de règles d'association sur des données de quiz afin de découvrir des règles et de fournir aux instructeurs des relations intéressantes qui peuvent être utiles pour mettre à jour et améliorer les quiz et les cours. Dans [56], les auteurs ont proposé un framework qui utilise des données institutionnelles vérifiées et d'autres sources d'information disponibles publiquement, afin de découvrir des connaissances et de fournir de multiples tâches académiques en fonction des besoins des différentes parties prenantes (éducateurs et administrateurs).

Le tableau 1.5 résume les deux travaux examinés ci-dessus.

1.4.2.3 Création d'alertes pour les parties prenantes

Dans cette section, nous présentons quelques travaux récents d'EDM pour la création d'alertes pour les parties prenantes.

— **Construction d'un modèle de prédiction du décrochage scolaire des étudiants**

Cette catégorie d'applications partage les mêmes objectifs que ceux présentés dans la modélisation des étudiants, puisqu'elle vise à prédire les caractéristiques des étudiants et à déterminer les comportements indésirables. Un exemple de situation qui nécessite une alerte précoce est le décrochage scolaire des étudiants. De nombreuses études ont été réalisées dans ce contexte.

Par exemple, les auteurs de [57] ont cherché à construire un modèle de prédiction du décrochage des étudiants afin d'aider les fournisseurs d'enseignement en ligne à mieux gérer les risques de décrochage. De même, dans [58], les auteurs ont proposé un algorithme intelligent parallèle pour prédire le décrochage des étudiants d'une université.

— **Déterminer les facteurs qui influencent le décrochage des étudiants**

Outre la construction de modèles prédictifs de décrochage, de nombreux chercheurs se sont également intéressés à la détermination des facteurs qui influencent la performance et le décrochage des étudiants.

Par exemple, les auteurs de [59] ont cherché à répondre à deux questions principales : quels sont les facteurs qui influencent le taux d'abandon dans un programme d'ingénierie système d'une université privée colombienne, et quelle est la meilleure technique de fouille de données qui peut être utilisée pour déterminer ces facteurs. Les expériences ont montré que la forêt aléatoire donnait de meilleurs résultats. Les auteurs de [60] ont cherché à construire deux modèles prédictifs pour identifier les étudiants à risque de décrochage parmi les étudiants de premier cycle d'une institution d'enseignement supérieur. Les modèles prédictifs permettent également d'identifier les principaux facteurs à l'origine du décrochage scolaire. Dans [61] l'objectif est d'appliquer des méthodes de fouille de données pour construire un modèle qui prédit le décrochage des étudiants et déterminer les facteurs qui influencent la performance des étudiants.

— **Comparaison des performances des algorithmes de machine learning pour prédire le décrochage des étudiants.**

Afin de construire le modèle de prédiction du décrochage scolaire le plus efficace, une étude comparative entre différents algorithmes a été réalisée par des chercheurs.

Par exemple, les auteurs de [62] ont comparé la performance de trois algorithmes de machine learning pour étudier la corrélation entre les variables démographiques et de performance académique et le décrochage des étudiants, puis les trois algorithmes sont combinés pour créer une méthode de classification d'ensemble pour optimiser les résultats de précision. Dans [63], les auteurs ont cherché à comparer les performances des techniques de fouille de processus et de séquence pour prédire le décrochage des étudiants. Dans ce travail, les auteurs exploitent les interactions des étudiants avec le MOOC afin de comprendre leur processus d'apprentissage et de prédire si un étudiant va abandonner le cours. Les auteurs ont constaté que les techniques d'exploration de processus fournissent des outils plus efficaces pour l'analyse descriptive, tandis que l'exploration de séquences traite mieux les données non structurées et fournit des outils utiles à des fins prédictives. Les auteurs de [64] ont utilisé des techniques de classification pour construire un modèle prédictif afin de prévoir le décrochage des étudiants. Une comparaison des algorithmes utilisés a été faite, et il a été constaté que les prédicteurs d'ensemble surpassent les algorithmes de classification standard dans la prédiction des abandons.

Le tableau 1.6 résume les travaux étudiés ci-dessus.

TABLE 1.6 – Résumé des travaux qui traitent la création d'alertes

Réf	Année	Données	Méthodes	Objectif
[58]	2020	Base de données du collège (informations démographiques et académiques)	Techniques de classification	Prédire le décrochage scolaire dans une université.
[59]	2018	Base de données du collège (informations démographiques et académiques)	Classification (arbres de décision, régression logistique, Naïve Bayes et foret aléatoire)	Sélectionner la meilleure technique de fouille de données pour déterminer les facteurs qui influencent les taux du décrochage scolaire
[60]	2020	Base de données du collège (informations démographiques et académiques)	Forêt aléatoire, machines à vecteurs de support et réseaux de neurones artificiels.	Identifier les étudiants à risque de décrochage parmi les étudiants de premier cycle d'un établissement d'enseignement supérieur.
[61]	2019	Base de données du collège (informations démographiques et académiques)	Algorithmes de fouille de données	Prédire le décrochage scolaire des étudiants et déterminer les facteurs qui influencent leurs performances.
[62]	2019	Base de données du collège (informations démographiques et académiques)	Techniques de classification (K-Nearest Neighbor (KNN), Naïve Bayes (NB) et arbre de décision (DT))	Étudier la corrélation entre les variables démographiques et les performances scolaires dans un modèle de prédiction du décrochage scolaire.
[63]	2017	Base de données du MOOC (interactions des étudiants)	Techniques d'exploration de processus et de séquences	Comparez les performances des techniques d'exploration de processus et de séquences pour prédire le décrochage des étudiants.
[64]	2021	Base de données du collège (informations démographiques et académiques)	Techniques de classification	Construire un modèle prédictif pour prévoir le décrochage scolaire.

1.5 Discussion

1.5.1 Tendance de la recherche

Comme nous avons vu précédemment, la figure 1.1 montre que le nombre de papiers publiés en EDM depuis 2006 est en croissance permanente. La communauté de recherche s'attend à la continuité de cette croissance dans les années à venir [65, 66]. Les chercheurs en EDM se sont concentrés sur des études d'analyse réelle, ce qui est conforme aux objectifs de l'EDM, à savoir la détection et l'extraction automatisées de modèles et d'informations significatifs à partir de grandes collections de données dans des contextes éducatifs [19]. En plus des outils de base du DM, les algorithmes du ML (tels que les méthodes Naive Bayes, SVM, réseau de neuron) ont été utilisés et étudiés en EDM. Cependant, le nombre de recherches en EDM exploitant les techniques Ml et DL n'est pas surprenant. En tant que recherche interdisciplinaire, l'EDM peut certainement attirer plus l'attention des chercheurs en ML et DL. Si l'on exclut les études menées dans le cadre des MOOCs, plus de la moitié des études sur l'EDM étaient axées sur l'enseignement supérieur plutôt que sur l'environnement du collège, moyen et secondaire. La totalité des travaux ont été réalisés avec des ensembles de données relativement restreints, ce qui pourrait résulter du manque d'ensembles de données public issu des systèmes éducatifs. Il est connu que le terme "big data" a des attributs spécifiques - volume, vélocité, variété, variabilité et complexité[67]. Bien que les chercheurs aient souvent utilisé le terme "big data" dans le titre de l'article, la majorité des études pourraient ne pas répondre strictement à cette définition.

1.5.2 Discussion sur les principaux sujets de recherche

1.5.2.1 Prédiction de la performance

La prédiction de la performance des apprenants est toujours le sujet de recherche le plus dominant en EDM [17, 18, 21]. En raison de la capacité des systèmes d'apprentissage en ligne à suivre et à stocker les activités en ligne, les comportements en ligne sont devenus la source de données la plus réalisable pour prédire les performances des apprenants en EDM [68]. Plus récemment, avec le développement rapide des techniques de fouille de données textuelles, les chercheurs ont commencé à utiliser les discussions en ligne pour prédire les performances d'apprentissage [69]. À présent, peu de chercheur en EDM ont adopté la combinaison des comportements en lignes avec des données textuelles pour optimiser les performances de prédictions. Par conséquent, la combinaison des comportements en ligne avec des données textuelles pour la prédiction de la performance des apprenants est essentielle pour améliorer les résultats de la prédiction.

De nombreux chercheurs considèrent qu'un bon engagement des apprenants et une bonne cohérence sont des indicateurs importants pour la prédiction de la performance [69, 70]. Cependant, ces deux critères restent un concept abstrait. La manière de "qualifier" l'engagement ou

le niveau de cohérence avec une définition claire et simple doit faire l'objet d'une étude plus approfondie.

Bien que les techniques de DL aient montré des avantages prometteurs dans divers domaines, peu d'études ont utilisé ces techniques prometteuses pour la prédiction de la performance. Le manque d'ensembles de big data appropriés disponibles dans de nombreux environnements éducatifs peut être un obstacle. D'autre part, bien que l'internet des objets se soit développé rapidement, les variables d'entrée pour la prédiction de la performance des apprenants sont encore fortement dépendantes des LMSs ou des ITSs. Les études futures pourraient devoir envisager d'intégrer de nouvelles sources de données traçables pour obtenir une compréhension holistique de l'état de l'apprentissage [71].

Enfin, on constate également que la majorité de ces études visent à fournir une liste d'apprenants à risque, et qu'une petite partie d'entre elles pourraient fournir des suggestions qui aident les instructeurs à concevoir manuellement des interventions. Les instructeurs peuvent concevoir des interventions générales plutôt que des interventions personnalisées. Par conséquent, l'analyse automatique de raisons spécifiques à risque et la génération de programmes d'intervention plus efficaces sont attendus dans les travaux futurs, ce qui pourrait être plus conforme aux exigences des ratios élevés apprenants/instructeur dans l'apprentissage en ligne.

1.5.2.2 Détection des comportements et modélisation de l'apprenant

En raison de l'absence des plateformes d'apprentissage, aucun résultat cohérent ne peut être extrait des études sur les modèles comportementaux menées dans la littérature (les travaux cités dans la section 1.4.1). De ce fait, les résultats de la recherche se sont limités aux applications d'environnements d'apprentissage spécifiques plutôt que d'être généralisés. Par conséquent, l'identification des "schémas d'apprentissage" classiques qui peuvent être observés dans différents contextes serait attendue dans les travaux futurs.

Bien que l'étude des modèles d'apprentissage des apprenants soit le courant dominant, plusieurs chercheurs ont commencé à étudier l'utilisation de certains outils numériques par les enseignants afin de comprendre le processus de conception pédagogique [72, 73]. Étant donné que des activités pédagogiques bien conçues peuvent grandement améliorer l'efficacité de l'apprentissage, nous en déduisons qu'un nombre important de chercheurs prêteront attention à la détection des modèles d'enseignement dans les années à venir. Cependant, ces études n'ont pas examiné comment aider automatiquement les enseignants et les apprenants à améliorer leur enseignement et leur apprentissage, par exemple comment fournir des suggestions automatiques de conception pédagogique ou d'ajustement des stratégies d'apprentissage.

1.5.2.3 Aide à la décision pour les enseignants et les apprenants

Sur la base de la définition de l'EDM, fournir une aide à la décision aux enseignants et aux apprenants reste un objectif primordial.

Tout d'abord, les études qui ont adopté la visualisation de données pour fournir une aide à la décision se sont souvent concentrées sur la conception et le développement de techniques de visualisation pour fournir des aperçus intuitifs [74]. Cependant, des recherches supplémentaires sont nécessaires pour révéler si ces résultats visualisés sont bénéfiques ou non pour promouvoir l'efficacité de l'enseignement et de l'apprentissage.

Deuxièmement, un grand nombre d'études existantes sur l'aide à la décision ont fourni des résultats descriptifs comme principal moyen d'analyse[75]. Cependant, les enseignants peuvent avoir des difficultés à interpréter ces résultats de manière à ce qu'ils puissent en tirer profits pour intervenir d'une manière efficace en cours. De simples alertes précises (comme les alertes de cours à l'Université Purdue [76]) pourraient être les plus souhaitées pour soutenir la prise de décision pratique. Par conséquent, les chercheurs pourraient recommander des actions de suivi aux enseignants.

D'autre part, comme l'enseignement et l'apprentissage sont fortement liés par les activités de soutien à l'enseignement, l'amélioration des performances d'apprentissage pourrait également être obtenue par la recherche sur le tutorat. Certains chercheurs se sont penchés sur l'évaluation de la qualité du tutorat [77], mais fournir des conseils pratiques sur la façon d'améliorer le processus d'enseignement est également attendu dans les travaux futurs.

Enfin, la recommandation de cours et de supports d'apprentissages aux apprenants est une tâche importante qui contribue d'une manière directe à l'amélioration du processus d'apprentissage. Une bonne recommandation aide l'apprenant à optimiser son temps de recherche et à améliorer ses compétences et par conséquent optimiser ses résultats. Bien que cette tâche soit d'une grande importance, nous avons constaté une négligence de la part de la communauté de recherche en EDM. Le peu d'études qui ont abordé les systèmes de recommandation en EDM se basent particulièrement sur les outils basiques du DM et n'adoptent pas les techniques connues des SRs à savoir le filtrage collaboratif et le filtrage basé sur le contenu. Cette tâche devrait attirer plus l'attention des chercheurs en EDM. L'intégration des techniques de filtrage aux systèmes d'apprentissages informatisés serait une solution efficace pour améliorer le processus d'apprentissage des apprenants.

1.6 Conclusion

Dans ce chapitre, nous avons présenté une analyse de la littérature sur l'utilisation de l'EDM dans les systèmes éducatifs. Nous avons commencé par décrire le domaine d'EDM ses définitions connues dans le domaine, et ses différents objectifs avant de passer à la présentation de différents types de systèmes éducatifs informatisés. Ensuite, nous avons étudié les applications récentes d'EDM (2015-2021) en considérant les données d'entrée pour l'analyse, les méthodes utilisées et les objectifs de l'analyse.

À travers les travaux étudiés, nous avons remarqué que les chercheurs étaient plus intéressés à traiter la tâche de modélisation des étudiants, et plus précisément la prédiction des performances et des caractéristiques des étudiants. Nous avons constaté que l'EDM peut être utilisé pour prédire la performance des étudiants, en particulier la note finale et le résultat final. Elle peut également aider à déterminer les facteurs qui influencent le plus les résultats des apprenants. Nous avons également constaté que l'état émotionnel et le style d'apprentissage faisaient partie des caractéristiques largement étudiées par les chercheurs, et que la classification, le clustering, la régression, les règles d'association et l'exploration de texte faisaient partie des techniques d'exploration de données couramment utilisées dans le domaine d'EDM. Parmi les applications les plus intéressantes aussi, c'est l'application des systèmes de recommandations, destinée aux étudiants directement, elle vise à les aider à avoir les ressources d'apprentissages adéquates à leurs profils, ce qui contribue directement à l'amélioration du processus d'apprentissage et des résultats finaux. Au cours de cette thèse, nous nous intéressons à l'application des systèmes de recommandations.

Les systèmes de recommandations font partie d'un domaine à part entière, dans le chapitre suivant, nous nous focalisons sur ce domaine. Le chapitre a pour objectif d'introduire les systèmes de recommandation, les différents types de recommandation et les limites de chaque type. Avant de clôturer le chapitre, nous présentons une étude des travaux existants. L'étude est menée en suivant la méthodologie SLR [78] pour identifier, expliquer et évaluer les travaux de recherche.

Chapitre 2

Systèmes de recommandations

2.1 Introduction

Dans le chapitre précédent, nous avons présenté en détail le domaine de la fouille de données éducatives. Nous avons défini la notion EDM ainsi que ses objectifs. Comme deuxième point principal, nous avons abordé les différents types de systèmes éducatifs informatisés tels que le LMS et le MOOC. Le troisième point abordé est d'une grande importance : les applications de l'EDM. Nous avons divisé les applications de l'EDM en deux catégories, à savoir : la modélisation de l'étudiant (qui comporte : l'analyse du comportement des apprenants, la prédiction des caractéristiques et performances des apprenants) et le système d'aide à la décision (qui comporte : la recommandation de cours, la présentation de rapport et la création d'alertes). Avant de conclure le chapitre, nous avons mené une discussion générale des travaux d'état de l'art et on a constaté que les chercheurs du domaine EDM prêtent beaucoup d'attention à la tâche de la prédiction des performances et des caractéristiques des apprenants. En ce qui concerne la recommandation de cours, nous avons constaté un grand manque de travaux traitant ce problème. L'intégration des systèmes de recommandations au sein des systèmes éducatifs informatisés permet de personnaliser le contenu proposé à l'apprenant en fonction de ses besoins, intérêts et ses objectifs. La tâche de recommandation contribue d'une manière directe à l'amélioration du processus d'apprentissage et des compétences de l'apprenant. Bien qu'elle soit de cette importance, la recommandation de cours connait une négligence dans le milieu de la communauté de recherche, c'est ce qui nous a motivé à travailler sur les systèmes de recommandations dans le domaine éducatif au cours de cette thèse.

Un système de recommandation est un outil et un ensemble de techniques qui fournit des suggestions d'article, parmi un large choix de (produits, vidéos, musique ou d'autres ressources) aux utilisateurs. Les suggestions sont personnalisées pour chaque utilisateur.

L'utilité des systèmes de recommandations devient présente dans les situations où il y a une surcharge d'informations, en d'autres termes, quand l'utilisateur est confronté à une grande sélection d'articles et devient perplexe à faire son choix. L'assistance personnalisée dans l'exploration et la découverte du contenu qu'offrent les systèmes de recommandation s'est révélée

être un moyen efficace d'accroître la satisfaction des utilisateurs et d'améliorer les revenus de nombreuses plateformes de commerce électronique et de diffusion de médias en continu comme Amazon, Netflix, Youtube ou Spotify et de réseaux sociaux comme Facebook ou Twitter.

Les méthodes de recommandation se basent sur un ensemble de théorie et d'algorithmes multidisciplinaires provenant de plusieurs domaines, citons entre autre : la recherche d'informations et le machine learning. En recherche, les systèmes de recommandations ont pris de l'ampleur ces dernières années et sont en croissance permanente.

Ce chapitre a pour but d'introduire les systèmes de recommandations dans un cadre général.Tout d'abord, nous commençons par donner des généralités sur les systèmes de recommandation. Ensuite, nous présentons les méthodes de recommandations conventionnelles. Nous abordons également la manière d'évaluer les systèmes de recommandations. Avant de conclure le chapitre, nous passons en revue quelque travaux effectués dans le domaine.

2.2 Généralité

Les systèmes de recommandations sont considérés comme la solution principale de la surcharge d'informations [79]. L'objectif d'un système de recommandations est défini de manières légèrement différentes dans la littérature et selon différentes perspectives.

Dans le domaine du commerce, l'objectif principal d'un système de recommandation est de stimuler les ventes de produits afin d'augmenter les bénéfices. En proposant aux utilisateurs des articles soigneusement sélectionnés, les systèmes de recommandation apportent des articles pertinents aux utilisateurs correspondants, ce qui entraîne une augmentation du volume des ventes et des bénéfices pour l'entreprise [80].

Du point de vue du consommateur, l'objectif d'un système de recommandation est de filtrer les articles non pertinents pour faciliter le choix d'un article qui convient à ses préférences.

Les SRs sont définis comme un ensemble de logiciels, d'outils et de techniques destinés à fournir des suggestions d'articles utiles à l'utilisateur [7, 81]. L'intégration des systèmes de recommandations dans les services en ligne est d'un grand intérêt tant pour l'utilisateur que pour le fournisseur de services. Le SR permet d'offrir des articles spécifiques et personnalisés qui correspondent mieux aux besoins et aux attentes de l'utilisateur, ce qui renforcera son lien avec le fournisseur du service et augmentera les bénéfices de ce dernier.

Dans la littérature, nous distinguons trois grandes catégories d'algorithme de recommandations [7, 8, 9, 82] :

- **Le filtrage basé sur le contenu :** basé sur la similarité entre de nouveaux items et les items déjà appréciés par l'utilisateur.
- **Le filtrage collaboratif :** basé sur la relation entre les utilisateurs et les items.
- **Le filtrage hybride :** combine le filtrage collaboratif et le filtrage basé sur le contenu.

En raison de sa grande importance, les SRs sont considérés comme un domaine de recherche à part entière [83].

2.3 Filtrage basé sur le contenu

Le filtrage basé sur le contenu recommande les articles à l'utilisateur cible en se basant uniquement sur les articles précédemment appréciés par cet utilisateur [9, 82]. Le profil d'utilisateur est créé lorsque l'utilisateur commence à interagir avec le système et crée son compte. Au fur et à mesure que l'utilisateur interagit avec le système, son profil devient plus riche en informations.

Dans ce type de filtrage, les informations de l'utilisateur cible sont suffisantes plutôt que celle d'autres utilisateurs similaires [1, 84]. Ce type de filtrage a pour avantage de pouvoir recommander les nouveaux articles, contrairement au filtrage collaboratif qui ne recommande jamais les nouveaux articles et en particulier les articles jamais évalués par les utilisateurs. Les algorithmes basés sur le contenu sont toutefois très dépendants du domaine de recommandation, ce qui contraste avec la généralité des méthodes du filtrage collaboratif. Un autre inconvenant connu de ce filtrage est la surspécialisation des éléments recommandés [7], en d'autres termes les éléments recommandés sont très similaires et n'apporte rien de nouveau par rapport à ce que l'utilisateur a déjà apprécié. La description des articles joue un rôle important dans le filtrage basé sur le contenu [85], ce qui le rend dépendant de la disponibilité d'informations suffisantes et précises sur les caractéristiques des articles, ce qui est parfois coûteux à obtenir. Le filtrage basé sur le contenu utilise des techniques de divers domaines telles que : la recherche d'information, le Web sémantique et l'apprentissage automatique.

En résumé, les systèmes de recommandation qui utilise le filtrage basé sur le contenu analysent la description d'un ensemble d'élément précédemment évalué par l'utilisateur cible U et construisent son profil sur la base des caractéristiques des objets évalués par cet utilisateur [86]. Le profil est une représentation structurée des intérêts de l'utilisateur, adoptée pour recommander de nouveaux éléments susceptibles de l'intéresser. Le processus de recommandation consiste à faire correspondre les attributs du profil de l'utilisateur avec les attributs des objets candidats [1]. Le résultat indique la pertinence qui représente le niveau d'intérêt de l'utilisateur pour cet objet. Les éléments recommandés sont les éléments les plus similaires à son profil.

2.3.1 Architecture des systèmes à base du filtrage basé sur le contenu

Les systèmes de filtrage basés sur le contenu nécessitent des techniques appropriées pour représenter les éléments et construire le profil de l'utilisateur, ainsi que des stratégies pour comparer le profil de l'utilisateur avec la représentation des éléments. L'architecture de haut niveau d'un système de recommandation basé sur le contenu [1] est décrite dans la figure 2.1.

Le processus de recommandation s'effectue en trois étapes [1], chacune d'entre elles étant gérée par un module distinct :

— **Module d'analyse de contenu :** un prétraitement est primordial pour extraire des informations structurées et pertinentes, surtout dans le cas des données non structurées telles que du texte. La fonctionnalité de ce module est la représentation du contenu des éléments (des documents, des pages web, etc.) sous une forme adéquate avec les étapes suivantes.

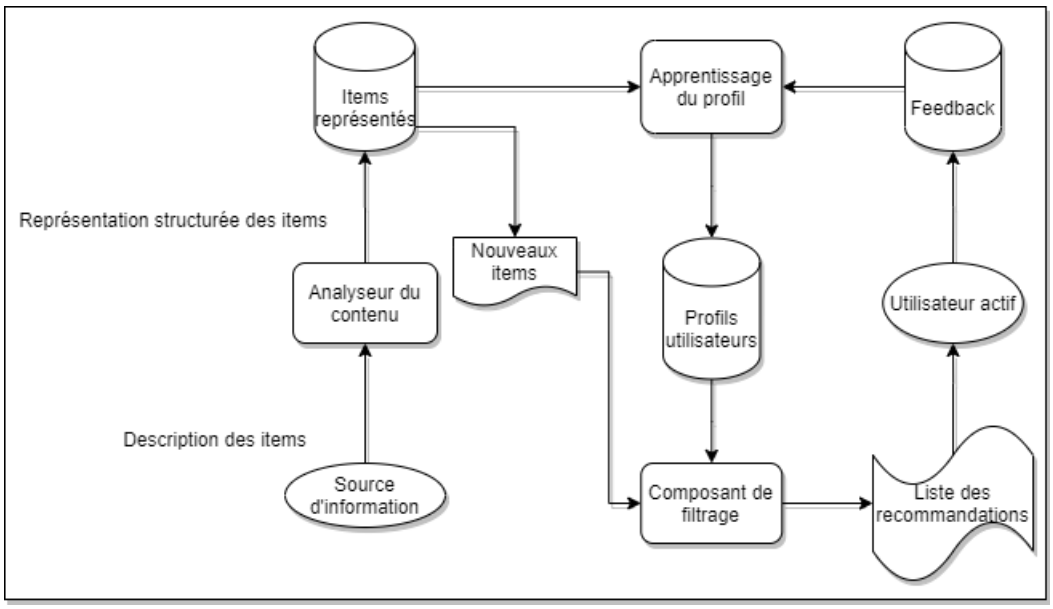

FIGURE 2.1 – Une architecture haut niveau d'un système de recommandation basé sur le contenu [1]

Des techniques d'extractions de caractéristiques sont utilisées pour analyser les éléments de données dans le but de déplacer la représentation de l'élément de l'espace d'information d'origine vers l'espace cible (par exemple : sous forme de vecteurs de mots clés). Cette représentation est l'entrée des modules suivants : module d'apprentissage de profil et module de filtrage.

— **Module d'apprentissage de profil :** la fonctionnalité principale de ce module et la construction du profil de l'utilisateur. Le module collecte des données représentatives des préférences de l'utilisateur et tente de généraliser ces données, afin de construire le profil. Pour généraliser les données, généralement des techniques de machine learning sont utilisées [87, 88] (citons entre autres : les arbres de décisions, les réseaux de neurones et la classification naïve de Bayes), ces techniques sont aptes de déduire un modèle des intérêts de l'utilisateur à partir des éléments appréciés ou non précédemment.

— **Module de filtrage :** la fonctionnalité de ce module est l'exploitation du profil de l'utilisateur afin suggérer des éléments pertinents en faisant correspondre la représentation du profil à celle des éléments à recommander. La pertinence des éléments est calculée à l'aide de certaines métriques de similarité [89]. Plus la similarité de l'élément et le profil est grande, plus cet élément a une chance d'être recommandé.

2.3.2 Avantages et limites

Le filtrage basé sur le contenu présente des avantages, citons :

— **L'indépendance de l'utilisateur :** le profil de l'utilisateur est construit uniquement sur la base des préférences de l'utilisateur cible. Contrairement au filtrage collaboratif qui exploite les préférences des utilisateurs les plus similaires à l'utilisateur actif.

— **Considération du nouvel élément :** la recommandation se base sur la description des éléments, quand un nouvel élément est introduit dans le système, il a toutes les chances d'être recommandé. Contrairement au filtrage collaboratif qui recommande des éléments obligatoirement évalués par les utilisateurs similaires à l'utilisateur cible. Dans [90], il a été démontré qu'en filtrage collaboratif, il faut que l'élément ait reçu 20 évaluations au minimum pour qu'il puisse être recommandé.

Cependant, le filtrage basé sur le contenu souffre de limites, citons :

— **Analyse du contenu limité :** le filtrage basé sur le contenu souffre d'une limite naturelle concernant le nombre et le type de caractéristique associés aux items candidats. La connaissance du domaine d'application est nécessaire, par exemple pour la recommandation de film, le système doit connaitre les réalisateurs, acteurs...etc., parfois, des ontologies du domaine sont nécessaires. Le filtrage basé sur le contenu ne peut pas fournir des recommandations précises si le contenu analysé ne contient pas suffisamment d'information pour distinguer les éléments qu'aime des éléments que n'aime pas l'utilisateur cible [1]. Contrairement au filtrage collaboratif qui traite les éléments même si aucune description n'est fournie.

— **Sur-spécialisation :** le filtrage basé sur le contenu ne recommande à l'utilisateur actif que les éléments similaires à son profil. L'utilisateur est souvent confronté à des recommandations similaires, identiques à celles précédemment appréciées [7, 91]. Les éléments non adaptés au profil de l'utilisateur ne seront pas proposés même s'ils sont les plus appropriés pour l'utilisateur. La diversité des recommandations fournies par le système est considérée comme un critère d'évaluation du système [92]. L'utilisateur préfère des recommandations pertinentes et diversifiées. Par exemple, il n'est pas intéressant de recommander tous les livres de Paulo Coelho à un utilisateur qui a aimé l'un de ses livres.

— **Problème du nouvel utilisateur :** il faut collecter suffisamment d'évaluations pour qu'un système de recommandation basé sur le contenu puisse comprendre les préférences des utilisateurs et fournir des recommandations précises [81]. Par conséquent, lorsque peu d'évaluations sont disponibles, comme pour un nouvel utilisateur, le système ne sera pas en mesure de fournir des recommandations fiables, ce problème est connu sous le nom du démarrage à froid pour les utilisateurs (user cold start problem).

2.4 Filtrage collaboratif

La technique du filtrage collaboratif ou collaboratif filtering *FC* est la technique la plus utilisée dans la littérature [82, 93]. Cette technique exploite les préférences et évaluation d'un

groupe d'utilisateur pour fournir des recommandations à l'utilisateur cible U. Par exemple, les utilisateurs X, Y et Z aiment les produits A et B et on a U qui aime le produit A va surement aimer le produit B.

Les systèmes de recommandations à base du FC fonctionnent de manière totalement indépendante par rapport au contenu des articles ou leurs caractéristiques. Ces systèmes recueillent les préférences des utilisateurs de manière explicite ou implicite. La méthode explicite consiste à donner la main et guider l'utilisateur à évaluer l'article en attribuant une note ou exprimer son appréciation par un "$LIKE$". La méthode implicite se base sur le comportement de l'utilisateur (achat d'un article, clics sur l'article, durée sur une page, etc.).

Le principal avantage de la technique FC est son indépendance par rapport au domaine d'application. Nous distinguons deux techniques du filtrage collaboratif [94] : la technique basée sur la mémoire (aussi appelée méthode basée sur le voisinage) et la technique basée sur le modèle.

2.4.1 Technique basée sur la mémoire

Cette approche se base sur le calcul de la similarité entre les utilisateurs [95] elle utilise une matrice de vote (utilisateur/ produit) contenant la note attribuée à chaque élément de la part de chaque utilisateur pour générer une prédiction. Des techniques statistiques sont utilisées pour trouver un ensemble d'utilisateurs, appelés voisins, qui ont un profil similaire au profil de l'utilisateur cible [96]. Une fois l'ensemble de voisins formé, ces systèmes utilisent différents algorithmes pour combiner les préférences des voisins afin de produire une prédiction pour l'utilisateur cible.

Mesure de similarité entre les utilisateurs

Afin de mesurer la similarité, nous voulons trouver la corrélation entre deux utilisateurs. Ce qui donne une valeur de -1 à 1 qui détermine la similarité entre deux utilisateurs. La valeur 1 signifie qu'ils évaluent tous les deux exactement de la même manière, tandis que la valeur -1 signifie qu'ils évaluent des choses exactement opposées (c'est-à-dire que l'un est élevé, l'autre faible ou vice versa).

Nous distinguons plusieurs algorithmes de mesure de similarité [96, 97], citons entre autres : $pearsoncorrelation$, $cosinevectorsimilarity$, $adjustedcosinevectorsimilarity$.

— **Corrélation de Pearson :** est considéré comme l'algorithme de similarité de base pour les systèmes à base de notation. Il mesure, suivant la formule (2.1) la corrélation linéaire entre deux vecteurs d'évaluation.

$$sim(u,v) = \frac{\sum_{C \in I_{uv}} (R_{u,c} - A_u)(R_{j,c} - A_v)}{\sqrt{\sum_{C \in I_{uv}} (R_{u,c} - A_u)^2 \sum_{C \in I_{uv}} (R_{v,c} - A_v)^2}} \tag{2.1}$$

Où $R_{u,c}$ est l'évaluation de l'utilisateur u au produit c, A_u est la moyenne des évaluations attribuées par l'utilisateur u à tous les éléments évalués et I_{uj} est l'ensemble des éléments évalués à la fois par l'utilisateur u et l'utilisateur v.

— **Similarité de Cosine :** également appelé similarité vectorielle [98]. Cet algorithme, selon la formule (2.2), mesure l'angle entre deux vecteurs d'évaluations, un angle plus petit étant considéré comme impliquant une plus grande similarité.

$$sim(u,v) = \sum_{i \in I_{uv}} \frac{r_{ui}}{\sqrt{\sum_{K \in I_u} r_{uk}^2}} \frac{r_{vi}}{\sqrt{\sum_{K \in I_v} r_{vk}^2}} \tag{2.2}$$

Où I_{uv} est l'ensemble des éléments évalués par l'utilisateur u et v, I_u est l'ensemble des éléments évalués par l'utilisateur u et I_v est l'ensemble des éléments évalués par l'utilisateur v.

— **Similarité vectorielle de Cosine ajustée :** c'est une forme modifiée de la similarité vectorielle [98]. Cet algorithme prend en compte la différence de l'échelle d'évaluation par chaque utilisateur ; en d'autres termes, certains utilisateurs attribuent une note élevée aux éléments en général, et d'autres peuvent donner des notes plus basses. Pour éliminer cet inconvénient de la similarité vectorielle, cette méthode normalise les évaluations en considérant l'écart d'une évaluation d'utilisateur par rapport à sa moyenne. Les évaluations moyennes de chaque utilisateur sont soustraites de l'évaluation de chaque utilisateur pour la paire d'éléments en question. La similarité est calculée comme le montre la formule (2.3).

$$sim(u,v) = \frac{\sum_{i \in I_{uv}} (R_{uc} - A_c)(R_{vc} - A_c)}{\sqrt{\sum_{i \in I_{uv}} (R_{uc} - A_c)^2 * \sum_{i \in I_{uv}} (R_{vc} - A_c)^2}} \tag{2.3}$$

Où R_{uc} est l'évaluation de l'élément c par l'utilisateur u, A_c est la moyenne des notes attribuées par l'utilisateur et I_{uv} est l'ensemble des éléments évalués à la fois par l'utilisateur u et l'utilisateur v.

— **La similarité Jaccard :** La mesure du coefficient de similarité Jaccard entre deux ensembles de données est le résultat de la division entre le nombre de caractéristiques communes divisé par le nombre de propriétés totales. La similarité de Jaccard entre deux utilisateurs u et v est calculée comme suit [98] :

$$sim(u,v) = \frac{|I_u \cap I_v|}{|I_u \cup I_v|} \tag{2.4}$$

où I_u est l'ensemble des éléments évalués par l'utilisateur u et I_v est l'ensemble des éléments évalués par l'utilisateur v.

Prédiction de votes

Une fois la similarité entre les utilisateurs calculée, les notes des produits non évalués par l'utilisateur cible a sont calculées. Les notes sont calculées comme un agrégat des notes de certains autres utilisateurs (généralement, les N les plus similaires) pour les mêmes éléments, comme suit [6, 82] :

$$p(a,i) = \overline{r_a} + \frac{\sum_{i=1}^{n}(r_{u,i} - \overline{r_u} * sim(a,u)}{\sum_{i=1}^{n} s(a,u} \tag{2.5}$$

Où $p(a,i)$ désigne la prédiction de la note attribuée par l'utilisateur cible a au produit i, $sim(a,u)$ désigne la similarité entre l'utilisateur u et a,$r_{u,i}$ est la note attribuée à i par l'utilisateur u, $\overline{r_a}$ est la moyenne des notes attribuées par a et n désigne le nombre total d'item.

2.4.2 Technique basée sur le modèle

Contrairement à la technique basée sur la mémoire, la technique basée sur un modèle n'a pas besoin de calculer explicitement les similarités entre les utilisateurs et les articles. Les systèmes basés sur le modèle fournissent des recommandations en utilisant un modèle conçu à partir des votes des utilisateurs [99, 100]. La phase de la construction du modèle est séparée de la phase de prédiction.

Les techniques basées sur le modèle dépendent d'une phase d'apprentissage, dans laquelle un modèle descriptif des préférences des utilisateurs basés sur les données observées est construit pour faire des prédictions. Ces méthodes s'inspirent de techniques d'apprentissage automatique telles que les réseaux de neurones artificiels, les réseaux bayésiens et les modèles à facteurs latents [80]. Parmi ces approches, les modèles de facteurs latents sont les techniques basées sur des modèles les plus étudiées et les plus répandues. Ces techniques effectuent une réduction de la dimensionnalité de la matrice utilisateur-item R dans laquelle un ensemble de variables latentes est utilisé pour expliquer les préférences des utilisateurs à des fins de recommandation. Ces techniques comprennent la factorisation matricielle, l'analyse sémantique latente probabiliste et l'allocation de Dirichlet latente.

Factorisation matricielle

La factorisation matricielle est l'une des techniques les plus connues et utilisées [9] et elle consiste à obtenir deux matrices, une matrice d'utilisateurs et une matrice d'items, $P \in R^{|u|,k}$ et $Q \in R^{|i|,k}$ qui représentent tous les utilisateurs et articles dans un espace vectoriel latent à k dimensions, où k est généralement beaucoup plus petit que le nombre d'utilisateurs ou d'articles. Ces matrices d'utilisateurs et d'items sont obtenues en minimisant une certaine fonction d'erreur ou de perte $L(P,Q)$ par rapport aux observations d'une matrice d'utilisateurs/items R par des méthodes variées telles que la descente de gradient stochastique [101, 102], les moindres carrés alternatifs [103] ou la factorisation de matrice à marge maximale [104]. La figure 2.2 illustre une simple factorisation matricielle d'une matrice utilisateur/item.

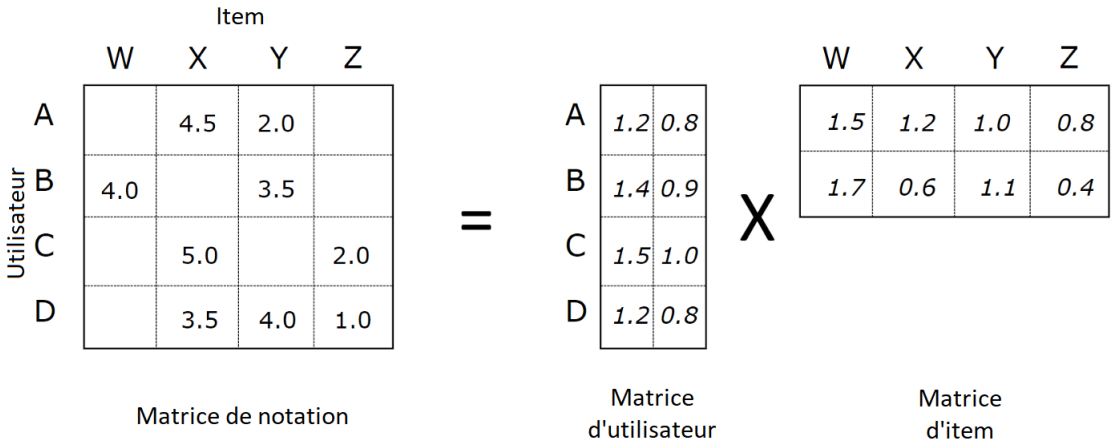

FIGURE 2.2 – Exemple de factorisation matricielle

Une fois que P et Q sont calculées, les recommandations sont déterminées par les scores générés par la multiplication des vecteurs utilisateur et item [105] :

$$p(u,i) = P_u.Q_i^t \tag{2.6}$$

où P_u est le vecteur de ligne de P qui correspond à l'utilisateur u et Q_i le vecteur de ligne de Q qui décrit l'élément i dans l'espace vectoriel latent, pour plus de détails voir [9, 105].

2.4.3 Avantages et limites

Le filtrage basé sur le contenu présente des avantages, citons :

— **Aucune connaissance du domaine n'est nécessaire :** le filtrage collaboratif ne requiert aucune connaissance du domaine d'application ni les caractéristiques des éléments à recommander. La recommandation se base uniquement sur les notes attribuées par les utilisateurs aux éléments.

— **Effet de surprise (serendipity) :** le filtrage collaboratif peut aider les utilisateurs à découvrir de nouveaux centres d'intérêt, c'est ce qui est souhaitable par les utilisateurs. C'est ce qu'on appelle les recommandations à effet surprise. Par exemple, si un utilisateur v est similaire à un utilisateur u du fait qu'ils apprécient tous les deux les livre du genre science-fiction, et que l'utilisateur u apprécie également un autre livre d'un autre genre, ce livre peut plaire à l'utilisateur v même qu'il soit loin de ses lectures habituées, donc sa recommandation serait une surprise.

Cependant, le filtrage collaboratif souffre de limites, citons :

— **Démarrage à froid :** pour construire un bon profil utilisateur, il doit y avoir suffisamment d'autres utilisateurs déjà dans le système pour trouver une correspondance. Le filtrage collaboratif dépend totalement des voisins similaires dans le système, mais si ces voisins similaires ne sont pas disponibles dans le système dans la phase initiale, ça engendre

le problème du démarrage à froid (cold start problem). Ce problème peut être évité en demandant à l'utilisateur d'évaluer au départ un certain nombre d'éléments ou par la recommandation des éléments les plus populaires. L'approche hybride en est une solution aussi.

— **La parcimonie (Sparsity) :** la parcimonie est le problème du manque d'information [106]. Généralement dans un système, on trouve le volume d'élément à recommander très grand. Généralement, les utilisateurs du système n'évaluent qu'un petit nombre d'éléments, même si le nombre d'utilisateurs est important, on distingue une matrice d'évaluation (utilisateur/item) creuse avec un taux de note manquante pouvant aller jusqu'à 95 % [107]. Le problème réside dans le fait d'avoir le nombre de notes à prédire très supérieur au nombre de notes connues. Les approches qui réduisent la dimension de la matrice d'évaluation peuvent être une solution à ce problème.

— **Mouton gris (Gray sheep problem) :** parfois, on distingue des utilisateurs à goûts uniques par rapport à tous les autres utilisateurs. Le système ne peut pas recommander des articles à quelqu'un qui a des goûts uniques. Ce problème est connu sous le nom du gray sheep. Il peut être résolu par l'approche hybride (filtrage collaboratif+ filtrage basé sur le contenu).

2.5 Filtrage hybride

Pour obtenir de meilleures recommandations, certains systèmes combinent plusieurs approches. Le filtrage collaboratif et basé sur le contenu sont considérés comme des approches complémentaires [91]. Combiner les deux filtrages permet de surmonter les limites de chaque modèle indépendamment [7]. Dans [108] l'auteur a identifié 7 façons pour combiner ses deux approches :

— **Hybridation pondérée (Weighted) :** L'hybridation pondérée combine les prédictions résultantes du filtrage collaboratif et filtrage basé sur le contenu pour générer une recommandation prédictive. Au départ, le poids de chaque approche de recommandation est le même. Ensuite, en utilisant différentes évaluations, le poids est ajusté en fonction des besoins. [109] est un exemple de ce type de système.

— **Hybridation alternée (switching) :** Le système passe d'une technique de recommandation à l'autre en fonction d'une heuristique reflétant la capacité du modèle à produire une bonne prédiction. Ainsi, ce type de système évite les inconvénients d'un seul type d'approche en permutant entre différentes approches. L'exemple de ce type de système est dans [110] qui utilise à la fois le filtrage basé sur le contenu et le filtrage collaboratif de manière alternée.

— **Hybridation mixte (Mixed) :** En hybridation mixte chaque technique prédit les notes manquantes, en suite les prédictions sont classées en ordre décroissant, les meilleurs seront sélectionnées. Dans [111] les auteurs utilisent une hybridation mixte. Il utilise des

techniques de contenu basées sur des descriptions textuelles d'émissions de télévision et des informations collaboratives sur les préférences d'autres types d'utilisateurs similaires. En utilisant l'hybridation mixte, il évite le problème des nouveaux articles par le filtrage basé sur le contenu plutôt que le filtrage collaboratif. On peut compter sur le composant basé sur le contenu pour recommander de nouvelles émissions sur la base de leurs descriptions, même si elles n'ont été évaluées par personne.

— **Hybridation en cascade (Cascade) :** Dans cette hybridation, les recommandations d'une technique sont affinées par une autre technique de recommandation. En d'autres termes, une première technique de recommandation est appliquée pour produire un premier classement des éléments candidats, ensuite une deuxième technique est appliquée sur la sélection résultante pour produire des recommandations affinées. Dans [112] les auteurs utilisent l'hybridation en cascade.

— **Hybridation par augmentation de caractéristique (feature-augmentation) :** cette hybridation utilise la sortie d'une technique comme caractéristique d'entrée pour une autre technique. Dans [113] les auteurs utilisent cette hybridation, le système produit des recommandations de livres en utilisant le filtrage basé sur le contenu à partir de données trouvées sur Amazon.com en utilisant un classifieur de texte (classifieur naive de Bayes).

— **Hybridation par combinaison de caractéristique (feature-combination) :** Cette hybridation combine les caractéristiques de différentes sources de données de recommandation dans un seul algorithme de recommandation. L'exemple de ce type d'approche est dans [114], qui a utilisé les évaluations du filtrage collaboratif dans un système basé sur le contenu comme caractéristique pour recommander des films.

— **Hybridation en définissant un niveau méta (meta-level) :** Dans cette hybridation, le modèle appris par une technique de recommandation est utilisé comme entrée pour une autre. L'avantage de la méthode de méta-niveau, en particulier pour l'hybride contenu/collaboratif, est que le modèle appris est une représentation précise de l'intérêt d'un utilisateur, et un mécanisme collaboratif qui suit peut opérer sur cette représentation dense en informations plus facilement que sur des données de notation brutes [9]. [115] est un exemple de ce type d'approche, dans lequel l'apprentissage instantané permet de créer un profil d'utilisateur en utilisant le filtrage basé sur le contenu qui est ensuite utilisé par le filtrage collaboratif.

2.6 Évaluation des systèmes de recommandations

L'évaluation des systèmes de recommandation fait l'objet de recherches actives dans ce domaine. Depuis l'avènement des premiers systèmes de recommandation, la performance de la recommandation est généralement assimilée à la précision de la prédiction des notes, c'est-à-dire que les notes estimées sont comparées aux notes réelles.

En termes d'utilité effective des recommandations pour les utilisateurs, on se rend de plus en plus compte que la qualité (précision) d'un classement d'éléments recommandés peut être

plus importante que la précision de la prédiction de valeurs d'évaluation spécifiques. Par conséquent, les mesures orientées vers la précision sont de plus en plus prises en compte dans le domaine, et un grand nombre de travaux récents se sont concentrés sur l'évaluation des listes de recommandations classées dans le $top-N$ avec le type de mesures ci-dessus.

Dans cette section, nous présentons un aperçu des mesures, protocoles et méthodologies d'évaluation dans le domaine des systèmes de recommandation.

2.6.1 Paradigmes d'évaluation

Deux paradigmes d'évaluation des SRs sont distingués dans la littérature [116] : évaluation en ligne et évaluation hors ligne.

— **Évaluation en ligne :** l'évaluation en ligne consiste à entrer en contact avec les utilisateurs et à leur demander d'essayer le système, de donner leur avis en direct ou de répondre à un questionnaire. Dans ce cas, l'évaluation se fait par rapport à l'acceptation des recommandations par les utilisateurs ($top-N$). En d'autres termes, ce paradigme ne mesure pas la précision des prédictions. L'acceptation est le plus souvent mesurée par le taux de clics (CTR), c'est-à-dire le ratio de recommandations cliquées. Par exemple, si un système affiche 10000 recommandations et que 120 sont cliquées, le CTR est de $1,2$ %. Pour comparer deux algorithmes, des recommandations sont créées en utilisant chaque algorithme et le CTR des algorithmes est comparé (test A/B). Les évaluations en ligne mesurent implicitement la satisfaction des utilisateurs et peuvent être directement utilisées pour estimer les revenus si les systèmes de recommandations appliquent un système de paiement au clic.

— **Évaluation hors ligne :** les évaluations hors ligne utilisent des ensembles de données hors ligne pré-compilés. Ensuite, les algorithmes de recommandation sont évalués par rapport à leur capacité à faire une prédiction précise des notes manquantes et par la suite fournir des recommandations pertinentes.

Dans la littérature, les systèmes de recommandation sont testés principalement hors ligne [54]. Plusieurs chercheurs ont marqué la littérature par leurs contributions dans le domaine, comme C. Lee Giles et ses co-auteurs [117, 118, 119, 120, 121, 122] ils peuvent effectuer des tests réels sur leur moteur de recherche universitaire CiteSeer, mais ils choisissent plutôt d'effectuer des évaluations hors ligne. Une raison pourrait être la simplicité d'effectuer une évaluation hors ligne en quelques minutes ou heures au lieu de quelques jours ou semaines pour un test en ligne. Un autre facteur peut être distingué dans de nombreux cas, les tests dans des scénarios hors ligne donnent de meilleurs résultats et sont plus pratiques que les tests en ligne [54].

2.6.2 Métriques d'évaluation

Un système de recommandations est conçu pour un objectif bien précis : recommandations d'items, optimisation d'utilité et prédiction des évaluations [123]. La métrique d'évaluation du

système est choisie selon l'objectif de ce dernier. Il est important de bien choisir la métrique, afin de garantir la fiabilité de l'évaluation. Nous présentons ci-dessous un ensemble de métriques d'évaluation qui ont été suggérées dans la littérature pour les systèmes de recommandation. Pour chacune de ces métriques, nous identifions ses propriétés importantes et expliquons pourquoi elle est la plus appropriée pour l'objectif donné. Pour chaque objectif, nous expliquons également un scénario d'évaluation possible qui peut être utilisé pour évaluer les différents algorithmes.

2.6.2.1 Prédiction d'évaluations

Pour cette tâche, le système doit fournir un ensemble d'évaluations prédites. L'exactitude des prédictions doit être évaluée en comparant les prédictions fournies par le système avec les choix que l'utilisateur aurait fait dans un cas réel. Nous distinguons plusieurs métriques d'évaluation de l'exactitude des prédictions, citons entre autres : erreur absolue moyenne (Mean Absolute Error (MAE)) et écart quadratique moyen(Root Mean Squared Error (RMSE)).

— **MAE :** MAE est une mesure de précision statistique qui évalue la précision d'un modèle de recommandation. Cette métrique mesure l'écart entre les recommandations prédites et les choix réels faits par les utilisateurs. Plus la valeur du MAE est faible, meilleure est la prédiction. Elle est calculée comme suit [124] :

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|p_{i,j} - q_{i,j}| \tag{2.7}$$

où $p_{i,j}$ est la prédiction d'évaluation de l'item j par l'utilisateur i, $q_{i,j}$ est l'évaluation réelle et n et le nombre total d'évaluations. Une petite valeur de MAE indique une meilleure performance du système.

— **RMSE :** RMSE calcule la valeur moyenne de toutes les différences au carré entre les évaluations réelles et prédites, puis calcule la racine carrée du résultat [125]. Par conséquent, des erreurs importantes peuvent affecter considérablement la valeur RMSE, ce qui rend la métrique RMSE plus précieuse lorsque des erreurs importantes ne sont pas souhaitées. L'erreur quadratique moyenne entre les évaluations réelles et les évaluations prédites est donnée par [124] :

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(p_{i,j} - q_{i,j})^2} \tag{2.8}$$

où $p_{i,j}$ est la prédiction d'évaluation de l'item j par l'utilisateur i, $q_{i,j}$ est l'évaluation réelle et n et le nombre total d'évaluations.

2.6.2.2 Recommandation d'articles

Pour la tâche de recommandation d'articles, nous ne sommes généralement intéressés que par des évaluations binaires, c'est-à-dire que l'article a été sélectionné (1) ou non (0). Pour déterminer l'efficacité du système, nous distinguons deux métriques [123] précision et rappel. Ces métriques mesurent la fréquence à laquelle un système de recommandation prend des décisions correctes ou incorrectes sur la pertinence d'un élément.

— **Précision :** détermine la probabilité qu'un élément recommandé soit pertinent. La précision est le rapport entre le nombre d'éléments pertinents sélectionnés et le nombre total d'éléments sélectionnés, elle est calculée comme suit :

$$Precision = \frac{TR}{(TR+FR)} \tag{2.9}$$

FR est le nombre de prédictions pertinentes fausses, tandis que TR est le nombre de prédictions pertinentes vraies. La précision est définie comme la probabilité qui montre la pertinence d'un élément sélectionné.

— **Rappel :** détermine la probabilité qu'un élément pertinent soit recommandé. Le rappel est le rapport entre le nombre d'éléments pertinents sélectionnés et le nombre total d'éléments pertinents disponibles, il est calculé comme suit :

$$Rappel = \frac{TR}{(TR+FN)} \tag{2.10}$$

où FR est le nombre de fausses prédictions pertinentes,TR est le nombre de vraies prédictions pertinentes et FN est le nombre de fausses prédictions non pertinentes. Le rappel signifie la probabilité qu'un élément pertinent soit choisi.

2.7 Revue des travaux connexes

Dans cette section, nous donnons un aperçu des travaux les plus récents et les plus référencés sur les systèmes de recommandations. Nous suivons la méthodologie SLR [78] pour identifier, expliquer et évaluer les travaux de recherche. Cette méthodologie est principalement composée de trois phases, à savoir la planification, l'exécution et la comparaison de l'examen, comme illustré dans la figure 2.3 et détaillé dans les sections suivantes.

2.7.1 Planification de l'étude

Cette étape vise à proposer un protocole d'examen qui met en évidence les principales hypothèses et les objectifs de l'examen et qui identifie clairement les principales questions de recherche (QR). En fait, dans notre étude, nous présentons et discutons les travaux récents publiés à travers quelques questions de recherche bien étudiées. Nous nous concentrons principalement sur :

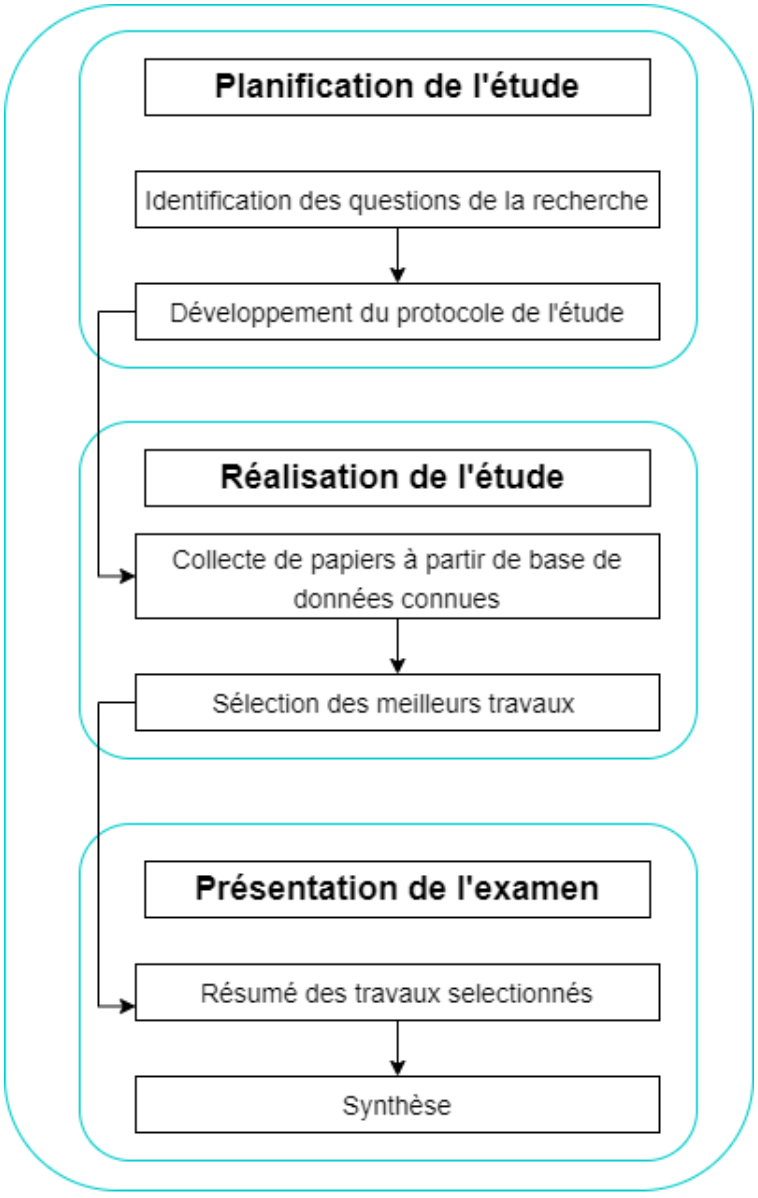

FIGURE 2.3 – Processus d'analyse systématique de la littérature

— **QR1 :** Quels domaines ont bénéficié des avantages des systèmes de recommandation ?

— **QR2 :** Quels modèles ont été adaptés dans chaque système ?

— **QR3 :** Comment les auteurs évaluent-ils leur système ?

2.7.2 Réalisation de l'étude

Le but ultime de cette phase est de sélectionner un ensemble de documents qui répondent à nos objectifs. En effet, dans notre stratégie de sélection, plusieurs bases de données et moteurs de recherche ont été utilisés tels que ACM Digital Library, Google Scholar, IEEE Xplore Digita Library, Science Direct, Web of Science et Elsevier afin de trouver des études publiées de haut niveau.

Notre stratégie de recherche est basée sur un ensemble de mots clés "systèmes de recommandation" "filtrage collaboratif", "filtrage basé sur le contenu", "systèmes de recommandation basés sur les $k-means$", etc. Après avoir collecté un ensemble d'articles basés sur notre stratégie de recherche, l'étape suivante consiste à filtrer cet ensemble en considérant uniquement les contributions pertinentes et récentes. En effet, nous ne considérons que les journaux, les conférences et les chapitres de livres publiés et datant de 2005 à 2020 afin de faire une analyse critique de la littérature.

2.7.3 Présentation de l'examen

Cette étape vise à résumer les articles sélectionnés pour répondre à nos questions de recherche et à les discuter en profondeur afin d'identifier les lacunes de la recherche.

Synthèse des papiers sélectionnés

Plusieurs systèmes de recommandation ont été proposés dans la littérature. Certains de ces systèmes ont été proposés comme des systèmes généraux, qui peuvent être adaptés dans n'importe quel domaine d'application. D'autres systèmes ont été conçus pour des domaines spécifiques tels que : la recommandation de film, de livre, d'endroit à visiter, etc. En effet, un SR peut améliorer considérablement les revenus des entreprises présentes dans le web, par exemple pour fidéliser les clients d'une marque donnée, promouvoir le tourisme dans un tel pays, améliorer la qualité de l'enseignement à distance, etc. Par conséquent, de nombreuses applications Web ont été développées, notamment dans le domaine du commerce électronique, de l'apprentissage en ligne, du tourisme en ligne, etc (réponse à la QR1). Ces applications ont été développées en utilisant un ensemble de techniques comme le filtrage collaboratif, le filtrage basé sur le contenu, la logique floue et surtout les techniques du machine learning telles que les approches basées sur le clustering (réponse de la QR2). Dans cette vision, nous classons les approches étudiées en deux catégories, les approches basées sur le clustering qui exploitent le clustering et les approches qui ont été proposées comme solution pour un domaine spécifique. Dans ce qui suit, nous présentons

ces deux catégories en commençant par les approches basées sur le clustering. Il est à noter que chaque approche présentée est discutée sur la base des modèles exploités et de leur validation (réponse à la QR3).

Modèles basés sur le clustering

Les approches basées sur le clustering prennent en compte la similarité entre les utilisateurs en deux niveaux différents. Le clustering permet d'avoir des groupes d'utilisateurs similaires ; il s'agit d'une présélection qui prend en compte la similarité entre les utilisateurs. Après avoir obtenu des clusters d'utilisateurs, un processus de recommandation est appliqué à chaque groupe indépendamment des autres. De nombreux chercheurs ont intégré le clustering au modèle de recommandation [10].

Dans [11], les auteurs intègrent le clustering dans un modèle de recommandation. L'algorithme $k-means$ est utilisé comme une étape de prétraitement pour une bonne formation du voisinage. Comme étape de présélection des voisins, ils ont utilisé la distance entre l'utilisateur et divers centroïdes. Les résultats expérimentaux indiquent que la structure recommandée peut améliorer considérablement la précision de la prédiction et le problème de l'évolutivité (scalability). Cependant, les ensembles de données utilisés pour l'expérimentation sont mineurs pour évaluer le problème d'évolutivité.

Dans [12], un algorithme a été conçu pour explorer séparément l'espace du contexte et l'espace des éléments et crée un algorithme qui intègre le clustering des éléments et l'agrégation des informations relatives à l'espace du contexte. Le modèle proposé manque de test et de validation.

Son et al. [13] ont présenté une étude qui intègre la réduction de dimension et le regroupement d'utilisateurs dans un modèle de filtrage collaboratif. Une phase de prétraitement est employée pour regrouper les utilisateurs similaires. Après cette phase, la technique du filtrage collaboratif est appliquée pour chaque groupe. La qualité du modèle n'a pas été évaluée. Une évaluation et une comparaison avec les approches conventionnelles seraient nécessaires pour valider le modèle proposé. Les résultats du temps de traitement ont montré que les performances du système développé sont nettement améliorées.

Zarzour et al. [14] ont développé un nouvel algorithme de recommandation par filtrage collaboratif en considérant la réduction de la dimensionnalité ainsi que les techniques du clustering. Les résultats expérimentaux ont montré que la méthode proposée a amélioré de manière significative les performances de recommandations et a maintenu les valeurs les plus basses de la courbe RMSE dans toute la gamme de voisins.

La technique présentée dans [15] est basée sur le clustering hiérarchique. Les clusters sont formés en utilisant l'algorithme de clustering hiérarchique *Chameleon*. Selon les résultats expérimentaux, le modèle présenté produit moins d'erreurs par rapport au système de recommandation basé sur le $k-means$. Cependant, une complexité de temps d'exécution plus faible a été constatée avec l'approche basée sur le $k-means$ par rapport à la méthode proposée.

Modèles généraux

Conçu par Beel et al. [126], Docear est considéré comme un nouveau système qui englobe différentes applications scientifiques. En particulier, Docear est considéré comme un SR dédié à la littérature scientifique. Le module suggéré par Docear utilise des algorithmes FC et BC. Ce dernier peut déterminer les intérêts des utilisateurs, offrant ainsi des recommandations très pertinentes. Le travail présente une limite, le système n'a pas été évalué ni comparé. Une évaluation serait nécessaire pour estimer la fiabilité du modèle.

Dans le domaine du e-tourisme, une approche de recommandations basée sur la logique floue et la classification associative a été proposée par Lucas et al. [127]. L'approche proposée prend en compte le filtrage collaboratif. L'analyse relative à une simulation réalisée de situations critiques réelles a démontré que la logique floue et la classification associative peuvent être utilisées avec le filtrage collaboratif. Ainsi, les problèmes de parcimonie, d'évolutivité et de mouton gris ont été considérablement allégés. Cependant, le système a été évalué manuellement avec un nombre limité d'utilisateurs, cela doit être revu pour valider les capacités réelles du modèle.

Pour le domaine de Technology-Enhanced Learning (TEL) un système de recommandation a été proposé dans [128]. Les auteurs visaient à faire face au problème du mouton gris et à améliorer la pertinence de recommandations des objets d'apprentissage. Le système développé étudie la possibilité d'adapter le style d'apprentissage de l'utilisateur calculé et les fonctionnalités de la page Web afin de fournir les meilleures ressources pédagogiques à chaque utilisateur. Le système a été évalué par un groupe d'étudiants et validé sur la base de certaines hypothèses. TEL est un domaine très intéressant. Une validation fiable du système développé serait intéressante pour passer à une utilisation efficace dans ce domaine.

Pour le domaine éducatif, Salehi et all. [129] propose un SR pour les ressources d'apprentissage basé sur un modèle d'information multidimensionnel et un algorithme génétique. Le système comprend deux modules de recommandation clés : un module qui tient compte des attributs explicitement collectés et une matrice de préférence proposée pour contrôler les intérêts de l'apprenant en ce qui concerne les caractéristiques explicites des ressources d'apprentissage pour un espace multidimensionnel. Le deuxième module est centré sur les attributs recueillis implicitement et un algorithme génétique est utilisé pour extraire ces attributs intégrés. La technique du filtrage collaboratif a été utilisée pour dériver les deux modules. Une liste de ressources est suggérée par chaque module, puis une combinaison linéaire est utilisée concernant le filtrage collaboratif en considérant les caractéristiques explicites et le filtrage collaboratif en considérant les attributs implicites, qui sont utilisés pour la recommandation finale. Ce système a été évalué et les résultats montre que l'approche proposée surpasse en précision les algorithmes conventionnels et atténue les problèmes de démarrage à froid et de parcimonie. Cependant, dans le domaine de l'éducation, il est crucial d'intégrer les connaissances des apprenants afin de proposer une recommandation personnalisée ainsi que l'adéquation avec son niveau d'éducation. Ces

TABLE 2.1 – Comparaison des approches existantes

Ref.	Domaine d'application	Modèles utilisés	Cluste-ring	Propriétés intrinsèque	Validation
[126]	Application scientifique	FC, CB	✕	✕	Modèle non validé
[127]	E-tourism	FC, logique floue, classification associative	✕	✕	Non satisfaisant
[128]	TEL	Approche formelle	✕	✕	Non satisfaisant
[129]	Domaine éducatif	FC, algorithme générique, modèle d'information multidimensionnel	✕	✕	Très satisfaisant
[11]	Approche générique	$k-means$	✔	✕	Satisfaisant
[12]	Approche générique	Contextual bandit algorithm	✔	✕	Modèle non validé
[13]	Approche générique	FC, réduction de dimensionnalité	✔	✕	Modèle non validé
[14]	Approche générique	FC, réduction de dimensionnalité SVD, $k-means$	✔	✕	Très satisfaisant
[15]	Approche générique	Clustering hièrarchique	✔	✕	Satisfaisant

✔ : critère pris en compte, ✕ : critère non pris en compte

informations manquent dans la procédure de recommandation suggérée.

2.7.4 Synthèse

Après avoir étudié les approches sélectionnées, nous suggérons une étude comparative selon plusieurs critères qui sont liés aux questions de recherche proposées.

Le tableau 2.1 présente un résumé des travaux connexes susmentionnés, en fonction de plusieurs critères. Dans le tableau 2.1 :

- **Domaine d'application :** Ce critère indique si les approches sont conçues comme des approches génériques ou pour un domaine d'application spécifique.
- **Modèles utilisés :** ce critère détermine les modèles et techniques utilisés dans les approches proposées.
- **Clustering :** ce critère vérifie si les approches proposées utilisent une technique de clustering comme étape de présélection des utilisateurs similaires.
- **Propriété intrinsèque :** ce critère détermine si les approches proposées prennent en compte la notion intrinsèque d'un cluster.
- **Validation :** dans ce critère, nous cherchons à étudier la qualité de la validation dans chaque approche proposée.

À partir de ce tableau comparatif, nous distinguons quelques lacunes, qui sont décrites ci-dessous.

— Nous remarquons que la plupart des approches proposées utilisent l'algorithme FC.
— Nous remarquons également que la plupart de ces approches tirent profit du clustering pour regrouper les utilisateurs similaires avant d'appliquer l'algorithme FC.
— La similarité est calculée uniquement par rapport à une paire d'utilisateurs pour toutes les approches.
— La relation entre une paire d'utilisateurs et les autres utilisateurs du même groupe n'est pas prise en compte.
— Pour la plupart des approches, la validation n'est pas satisfaisante.

Dans cette perspective, notre objectif principal est de combler ces lacunes en proposant une nouvelle approche basée sur la théorie des jeux et le FC, qui prend en considération la notion intrinsèque d'un cluster (i.e. prendre en compte la relation entre une paire d'utilisateurs et les autres utilisateurs du même groupe).

2.8 Conclusion

Dans ce chapitre, nous avons donné un aperçu complet sur les méthodes de recommandation. En résumé, nous distinguons trois méthodes principales, à savoir : le filtrage collaboratif, le filtrage basé sur le contenu et le filtrage hybride. Nous avons également donné un aperçu des travaux les plus récents et les plus référencés sur les systèmes de recommandations. En suivant la méthodologie SLR [78] nous avons identifié, expliqué et évalué les travaux de recherche.

Au cours de ce chapitre, nous avons résumé des articles sélectionnés afin de répondre à nos questions de recherche, nous avons discuté ces articles et d'identifié les lacunes de la recherche. Sur la base de nos conclusions de recherche, nous avons constaté que la sélection des utilisateurs similaire est une étape très importante dans un processus de recommandation. Pour une meilleure sélection, plusieurs auteurs ont inclus une phase de présélection d'utilisateurs similaires comme première étape du processus FC. La présélection se base sur les méthodes conventionnelles de clustering telle que le $k-means$. Les résultats de leurs expérimentations affirment que la présélection améliore d'une manière significative les méthodes sans présélection.

Les méthodes classiques du clustering prennent en compte la relation entre une paire d'utilisateurs uniquement. La prise en compte d'une paire d'utilisateur et sa relation par rapport à l'ensemble du groupe pourrait améliorer encore plus l'étape de présélection, et cela n'est pas pris en compte par les méthodes conventionnelles. Pour combler cette lacune tirée, l'intégration de la théorie des jeux comme présélections du voisinage dans un processus de filtrage collaboratif serait une solution. En effet, l'intégration de la théorie des jeux permet de considérer la notion intrinsèque d'un cluster (i.e. prendre en compte la relation entre une paire d'utilisateurs et les autres utilisateurs du même groupe).

Dans le chapitre suivant, nous allons décrire une nouvelle approche de recommandation basée sur la théorie des jeux que nous avons nommée $CF-GT$. Avant cela, nous présentons les bases de la théorie des jeux en se focalisant sur les notions utilisées pour notre modèle.

Chapitre 3

Approche générique de filtrage collaboratif basée sur la théorie des jeux

3.1 Introduction

Dans les deux premiers chapitres de cette thèse, nous avons présenté un état de l'art qui a donné les différentes issues de recherche permettant de tirer profits des données volumineuses générées par les systèmes éducatifs informatisés, plus précisément :

(i) Le chapitre I a présenté une analyse de la littérature sur l'utilisation de l'EDM pour résoudre les problèmes du secteur éducatif. Les objectifs d'EDM ont été présentés ainsi que les différents types des systèmes éducatifs informatisés. Le chapitre étudie les applications récentes d'EDM en considérant les données d'entrée, les méthodes utilisées ainsi que les objectifs de chaque travail. Les applications d'EDM se divisent en deux grandes catégories, à savoir la modélisation des étudiants et l'aide à la décision. La modélisation des étudiants se charge de plusieurs taches, citons entre autres : la prédiction des performances et des caractéristiques des étudiants. Pour l'aide à la décision, nous avons constaté une tâche importante dans l'amélioration du processus d'apprentissage qui est la recommandation des ressources pédagogiques.

(ii) Le chapitre II a permis de cerner le domaine des systèmes de recommandations. Les différentes techniques de filtrages ont été abordées ainsi que les techniques d'évaluation de ces systèmes. Le chapitre présente une revue des travaux connexes. Il a été constaté que plusieurs chercheurs adoptent le clustering comme une étape de présélection des utilisateurs similaires. Les travaux synthétisés ont été divisés en deux catégories : les modèles de recommandations qui se basent sur le clustering et les modèles qui n'utilise pas le clustering. La performance des modèles basés sur le clustering est toujours meilleure par rapport aux approches de comparaison, cela est dû à la présélection des utilisateurs similaires qui contribue à l'amélioration des différentes techniques de filtrage.

Malgré les efforts fournis par la communauté de recherche, il reste toujours des défis à relever. L'amélioration de la performance des prédictions est l'un des défis qui ont toujours incité l'attention de la communauté de recherche.

La sélection des utilisateurs similaires est considérée comme la première étape du filtrage collaboratif. Une bonne sélection est importante pour bien prédire les entrées manquantes. Comme nous l'avons vu précédemment, plusieurs chercheurs ont rajouté une étape de présélection des utilisateurs similaires dans le processus du filtrage collaboratif, et ce, en utilisant le clustering, après avoir obtenu les clusters d'utilisateur, le processus du filtrage collaboratif est appliqué à chaque cluster.

Dans ce chapitre, on trouve une proposition de solution aux défis présentés.

3.2 Aperçu et motivation de notre proposition

Dans un premier temps, nous avons proposé un modèle de recommandation générique afin d'aider l'utilisateur à obtenir les articles adéquats à son profil, sans qu'il ait à passer un temps énorme à parcourir un large éventail d'article.

Le filtrage collaboratif est la technique de recommandation la plus utilisée dans la littérature et les systèmes de recommandations à base de cette technique sont les plus performants.

Pour rappel, le filtrage collaboratif consiste à prédire les articles (livres, filmes, vêtements, etc.) que des utilisateurs apprécieront dans le futur. Pour prédire les articles susceptibles d'intéresser un utilisateur dans le futur, l'algorithme n'utilise pas seulement l'historique de cet utilisateur, mais toutes les informations existantes dans le système concernant les autres utilisateurs. En d'autres termes, ces algorithmes permettent de détecter les utilisateurs qui ont des goûts similaires pour exploiter ces informations à des fins de recommandations. Par exemple, s'il s'avère qu'Alice, Bob et Martin ont aimé par le passé des livres similaires et qu'Alice achète, ou donne un avis favorable sur le dernier livre de Paul Auster, il est très probable que Bob et Martin l'aiment aussi et il est pertinent de le leur recommander. Prenons un autre exemple plus concret, si vous envisagez de regarder un nouveau film, vous allez généralement demander à vos amis de vous le recommander. Cela part du principe que les utilisateurs font confiance à leurs amis, car ils sont persuadés qu'ils connaissent leurs goûts en matière de films. Par conséquent, nous suivons et regardons généralement ce qui nous est recommandé par un bon ami qui a les mêmes goûts que nous.

La première étape de technique FC est la sélection des utilisateurs similaire à l'utilisateur actif u_i, puis l'algorithme prédit les votes manquants de l'utilisateur u_i en se basant des votes attribués par ses utilisateurs similaires. Par la suite, l'algorithme recommande à l'utilisateur u_i les articles avec le meilleur vote prédit.

Un bon choix des utilisateurs similaires implique une bonne prédiction des votes manquants, par conséquent, le système fournit des recommandations qui satisfassent l'utilisateur actif.

Pour tenter d'optimiser la première étape du processus FC, plusieurs chercheures [10] ont créé une étape de présélections des utilisateurs similaires en utilisant une technique de clustering pour obtenir une première sélection d'utilisateurs similaires. Après l'obtention des clusters, chaque cluster passe par toutes les étapes le processus du filtrage collaboratif, y compris la sélection des utilisateurs similaires.

Le clustering est une technique courante d'analyse statistique des données. Le clustering est le processus de regroupement d'objets similaires dans différents groupes, ou plus précisément, la partition d'un ensemble de données en sous-ensembles, de sorte que les données dans chaque sous-ensemble sont plus similaires les uns aux autres que les objets d'autres sous-ensembles, la partition des sous ensemble se fait selon une certaine mesure de distance définie[130]. Le clustering est un problème très bien étudié dans le domaine de DM, de ML et des disciplines connexes.

En général, les différentes techniques de clustering visent à obtenir des clusters optimaux par rapport à une certaine fonction objective. Par exemple, les algorithmes basés sur les $k-means$ cherchent à minimiser la distance moyenne au carré entre chaque point et son centre de cluster le plus proche [131]. Notre travail vise à optimiser la formation de cluster d'utilisateurs similaires.

Notre modèle comporte deux modules, le premier module"$SimilarUser$ et le deuxième module "$CFProcess$". le module "$SimilarUser$" sert à regrouper les utilisateurs similaires et le module "$CFProcess$" applique le processus du filtrage collaboratif pour chaque groupe obtenu. La figure 3.1 illustre notre modèle.

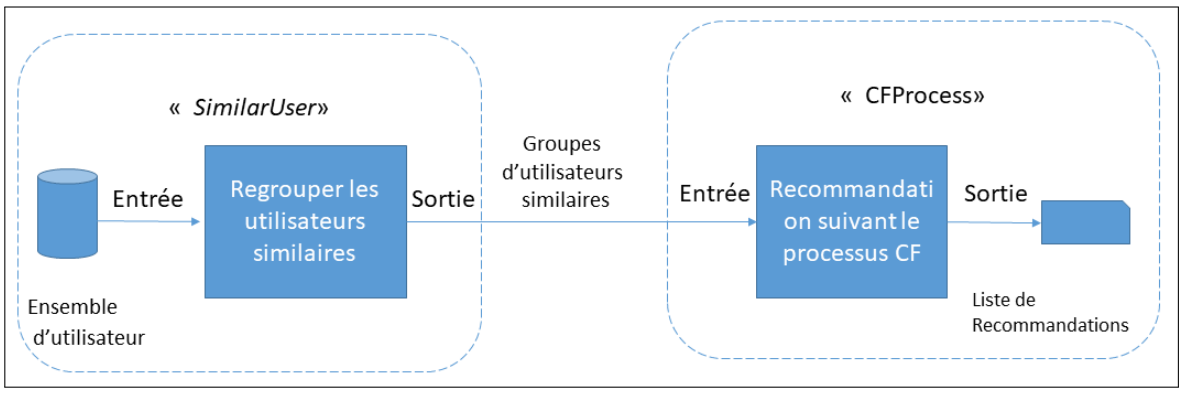

FIGURE 3.1 – Aperçu de notre modèle

Pour obtenir des groupes d'utilisateurs similaires optimaux par le module "$SimilarUser$", nous nous concentrons sur deux objectifs clé : nous cherchons à minimiser 1) la distance moyenne entre chaque utilisateur et son utilisateur le plus proche et 2) la distance moyenne intra-cluster point à point.

Notre travail est fortement motivé par ce qui a été diversement décrit dans la littérature [132] : « le clustering ne devrait pas être fait uniquement sur la base de la distance entre une paire de points, mais aussi sur leur relation avec d'autres points de données ».

Au cours de notre travail, nous nous focalisons sur le module "*SimilarUser*". Ce module a pour fonctionnalité la formation de groupe d'utilisateur similaire suivant une approche prometteuse basée sur la théorie des jeux coopératifs. L'idée est de faire correspondre la formation de clusters à la formation de coalitions en utilisant la valeur de Shapley, dans le cadre d'un jeu convexe défini de manière appropriée. La valeur de Shapley est un concept de solution équitable dans la mesure où elle divise la valeur collective ou totale du jeu entre les joueurs en fonction de leurs contributions marginales à la réalisation de cette valeur collective. Dans le modèle que nous proposons, cette propriété intrinsèque de l'équité basée sur la contribution marginale est utilisée au mieux pour un regroupement efficace.

Dans ce qui suit, un aperçu de la théorie des jeux est présenté.

3.3 Théorie des jeux

La théorie des jeux ou Game Theory (*GT*) est un ensemble d'outils mathématiques utilisés pour analyser les interactions sociales entre les preneurs de décision (appelés désormais joueurs) qui visent à obtenir une distribution équitable et stable de services.
Le but de la *GT* [133] est d'analyser et de modéliser des comportements rationnels de joueurs en situations d'interactions stratégiques. Elle s'intéresse particulièrement aux situations où il y' des joueurs qui prennent des décisions, chacun étant conscient que ses gains dépendent non seulement de sa propre décision, mais aussi des décisions prises par les autres joueurs. Un joueur peut prendre plusieurs décisions et il en choisit une qui lui semble la meilleure, appelée stratégie. En termes mathématiques, « la meilleure décision pour lui » se traduit par la description d'une fonction pour chacun des joueurs qui reflète ses préférences, appelée fonction de gain ou fonction d'utilité. Le type de jeu appliqué pour l'allocation de ressources dépendra des caractéristiques des réseaux, des applications et des objectifs à atteindre.

Pour représenter un jeu en *GT*, trois formes sont distinguées dans la littérature [134] :

— **La forme extensive :** cette forme est la description la plus complète d'un jeu en *GT*. Elle décrit le processus séquentiel de prise de décision et les résultats qui en découlent.

— **La forme normale :** la forme normale est caractérisée par le fait qu'elle ne contient que des informations spécifiques telles que la structure d'interaction stratégique du jeu. Cela permet à la structure mathématique de la forme normale d'être beaucoup plus simple que la forme extensive.

— **La forme de fonction caractéristique :** cette forme nécessite beaucoup moins d'informations par rapport aux formes normale et extensive. Elle ne considère que les gains qui peuvent être alloués aux coalitions de joueurs dans le jeu. Cette forme est la forme de jeu préférée pour représenter un jeu coopératif avec des accords contraignants.

Il existe deux perspectives différentes de haut niveau de la *GT* : les jeux classiques et les jeux évolutifs.

TABLE 3.1 – Dilemme du prisonnier

Matrice des gains		Prisonnier B	
		Avouer	Nier
Prisonnier A	Avouer	5.5	0.20
	Nier	20.0	1.1

Au cours de notre travail, nous nous sommes intéressées aux jeux classiques que nous allons détailler dans ce qui suit.

3.3.1 Jeux classiques

Dans les jeux classiques, tous les joueurs doivent faire des choix rationnels parmi un ensemble de stratégies. Un jeu classique est représenté par (N, S, v) où :

— $N = \{1, ..., N\}$ représente l'ensemble des joueurs. Il convient de noter qu'un joueur peut être un individu ou un groupe d'individus prenant des décisions. De plus, les joueurs sont supposés être rationnels ou rationnels bornés selon le jeu.

— S est l'ensemble de stratégies et est défini comme $S = \{1, ..., S\}$ .. Les stratégies peuvent considérer une action unique, des actions multiples ou une distribution de probabilité sur des actions multiples.

— v est la fonction de valeur ou le gain du joueur i. Il s'agit de la récompense qu'un joueur reçoit en fonction des stratégies adoptées tout au long du jeu. En fait, le gain d'un joueur dépend de sa propre stratégie et des stratégies adoptées par les autres joueurs.

Dilemme du prisonnier

Ce jeu [135] décrit la situation dans laquelle deux prisonniers A et B, soupçonnés d'avoir commis un vol, sont placés en détention. Comme les deux prisonniers seront interrogés dans des pièces séparées, chacun doit décider s'il doit avouer ou mentir.

Comme le montre le tableau 3.1, les deux prisonniers ont les choix suivants : i) s'ils avouent, ils vont tous deux en prison pour cinq ans, ii) s'ils mentent, ils vont tous deux en prison pour un an, et iii) si l'un avoue et l'autre ment, celui qui avoue obtient sa liberté et celui qui ment va en prison pour 20 ans. Afin d'analyser ce jeu, on suppose que les prisonniers ne peuvent pas communiquer et qu'ils prendront leurs décisions de manière simultanée. Par conséquent, chaque prisonnier analysera la meilleure stratégie compte tenu des stratégies possibles de l'autre prisonnier. Si le prisonnier B avoue, il sera condamné à cinq ou zéro an de prison. En revanche, si le prisonnier B ment, il recevra 20 ou un an de prison. Puisqu'en avouant, le prisonnier B obtient moins d'années de prison, il choisira d'avouer. La même procédure s'applique au prisonnier A, donc, avouer est la stratégie dominante. Alors, l'équilibre de *Nash* est $(avouer, avouer)$ puisque cela mène à l'utilité maximale pour chaque prisonnier.

3.3.2 Jeu coopératif

Le jeu coopératif se compose d'outils analytiques qui étudient le comportement de joueurs rationnels lorsqu'ils coopèrent. Dans un jeu coopératif, les joueurs ont la possibilité de se concerter et de s'engager à coopérer avant de définir la stratégie à adopter.
La branche principale de la théorie des jeux coopératifs se concentre sur la formation de groupes de joueurs, également connus sous le nom de coalitions.

De point de vue mathématique, un jeu coopératif peut être défini comme une paire (N, v) où N représente un ensemble de joueurs et $v : 2^N \longrightarrow R$ représente la fonction caractéristique. Étant donné un sous-ensemble S de N, $v(S)$ est souvent appelé la valeur de la coalition S et représente le nombre total d'unités de transfert qui peut être atteint par les joueurs de S, sans l'aide des joueurs de$N \backslash S$ [136]. L'ensemble des joueurs N est appelé la grande coalition et $v(N)$ est appelé la valeur de la grande coalition.

Un jeu coopératif peut être analysé à l'aide d'un concept de solution, qui fournit une méthode de répartition de la valeur totale du jeu entre les joueurs individuels. Dans ce qui suit, nous décrivons deux concepts de solution importants, à savoir le noyau et la valeur de Shapley SV.

Dans cette thèse, nous limiterons notre attention à la théorie des jeux de coalition puisque nous nous concentrons sur la formation de coalitions. Les jeux de coalition sont classés en trois classes [137] : les jeux canoniques, les jeux de formation de coalitions et les jeux de graphes de coalition. Dans le tableau 3.2 nous décrivons les principales caractéristiques de ces classes.

TABLE 3.2 – Classe des jeux coalitionnels

Classe du jeu	**Caractéristiques**
Canonique	La grande coalition, composée de tous les joueurs, est une structure optimale.
Formation de coalitions	La structure du réseau dépend des gains et des coûts de la coopération.
Graphe de coalition	Les interactions des joueurs sont régies par une structure de graphe de communication.

3.3.2.1 Généralité des jeux coalitionnels

Dans les jeux coalitionnels, l'ensemble des joueurs, désigné par $N = \{1, ..., N\}$ cherche à former des groupes coopératifs. Un concept fondamental dans les jeux coalitionnels est la valeur de la coalition C qui représente la valeur d'une coalition dans un jeu et est représentée par v. Par conséquent, un jeu de coalition est défini par (N, v).

La théorie des jeux coopératifs a été largement utilisée pour résoudre le problème d'allocation des ressources dans les réseaux cellulaires sans fil, par exemple [137, 138]. Les joueurs forment des coalitions, au sein de chaque coalition. Les joueurs acquièrent un comportement

coopératif afin de maximiser la valeur de la coalition et par conséquent améliorer leur propre bénéfice.

3.3.2.2 Le noyau

Une allocation de gain $x = \{x_1, x_2..., x_n\}$ désigne un vecteur dans R^n avec x_i représentant l'utilité du joueur i où $i \in N$. L'allocation de gain x est dite rationnelle au niveau de la coalition si $\sum_{i \in C} x_i \geq v(C), \forall C \subseteq N$. Enfin, l'allocation de gain x est dite rationnelle au niveau collectif si $\sum_{i \in N} x_i = v(N)$. Le noyau (core) d'un jeu TU (N, v) est la collection de toutes les allocations de gains qui sont coalitionnellement rationnelles et collectivement rationnelles. On peut montrer que chaque allocation de gain se trouvant dans le noyau d'un jeu (N, v) est stable dans le sens où aucun joueur ne bénéficiera d'une déviation unilatérale d'une allocation de gain donnée dans le noyau. Les éléments du noyau sont donc des allocations de gains potentielles qui pourraient résulter de l'interaction et de la négociation entre des joueurs rationnels.

3.3.2.3 La valeur de Shapley

Dans la théorie des jeux coopératifs, la valeur de Shapley ou Shapley Value (SV) peut être définie comme un concept de solution. Elle définit une approche efficace qui permet de répartir équitablement les gains obtenus par la coopération entre les joueurs associés à un jeu coopératif [8]. Comme certains joueurs peuvent avoir une contribution plus élevée à la valeur totale par rapport à d'autres, une répartition équitable des gains entre les joueurs est importante. La valeur de Shapley concerne l'importance relative de chaque joueur dans le jeu pour décider du gain qui doit être réparti entre les joueurs. La valeur de Shapley est donnée comme suit :

$$\phi_i(N, v) = \sum_{C \subseteq N - i} \frac{|C|!(n - |C| - 1)!}{n!} \{v(C \cup i) - v(C)\} \tag{3.1}$$

où $\phi_i(N, v)$ est le gain attendu du joueur i et $N - i$ désigne $N \backslash \{i\}$.

La valeur de Shapley d'un joueur reflète précisément la valeur marginale que le joueur apporte au jeu et le potentiel de négociation du joueur. La valeur de Shapley est une division de gain unique qui divise la valeur de la coalition et qui satisfait les axiomes suivants [139] :

(**Symétrie**). pour tout jeu (N, v) et permutation π de N on a : $\sum_{i \in N} \phi_i(N, v) = \phi_{\pi(i)}(N, \pi v)$. Cet axiome indique que la valeur ne change pas lors d'une réorganisation arbitraire ou un renommage des joueurs. Ici, nous parlons de symétrie.

(**Propriété du joueur nul**). Pour tout jeu (N, v) il est considéré que : $v(S \cup \{i\}) = v(S) \forall S \subseteq N$, cela affirme que $\phi_i(N, v) = 0$. Cet axiome indique qu'un joueur fictif qui n'a aucun impact sur la valeur globale d'une coalition n'a aucune incitation à y devenir membre.

(**Additivité**). Pour tous deux jeux (N, v) et (N, w) on a : $\phi_i\ (N, v + w) = \phi_i(N, v) + \phi_i(N, w)$ où $(v + w)(S) = v(S) + w(S)$. Si un nombre réel α est utilisé pour mettre à l'échelle la fonction de gain v, alors le même facteur est également utilisé pour mettre à l'échelle la valeur de Shapley.

Autrement dit : $\phi_i(N,\alpha v) = \alpha(N,v)$. Cela indique la propriété de linéarité qui est essentielle pour obtenir l'invariance d'échelle par rapport à la fonction de valeur.

(**Efficacité**). Pour tout jeu (N,v) il est considéré que : $\sum_{i\in N}\phi_i(N,v) = v(N)$. Cet axiome correspond à l'optimalité au sens de Pareto. La totalité de la valeur de la grande coalition qui est le gain maximum possible dans un jeu est intégralement réparti entre les joueurs

3.3.3 Jeu convexe

Un jeu (N,v) est dit convexe si :

$$v(C)+v(D) <= v(C\cup D)+v(C\cap D), \forall C,D \subseteq N \tag{3.2}$$

de manière équivalente, un jeu TU (N,v) est dit convexe si pour tout joueur i , la contribution marginale de i est plus grande dans une coalition plus grande. En d'autres termes :

$$v(C\cup i)-v(C) \leq v(D\cup i)-v(D) \forall C \subseteq D \subseteq N-i, i \in N, \tag{3.3}$$

où la contribution marginale $m(S,j)$ du joueur j dans la coalition S est donnée par :

$$m(S,j) = v(S\cup j)-v(S), S \subseteq N, j \in N, j \notin S. \tag{3.4}$$

3.3.4 La valeur de shapley pour les jeux convexes

Considérons une permutation π des joueurs dans le jeu. Alors, pour une quelconque $|N|!$ des permutations possibles, les segments initiaux de l'ordre sont donnés par :

$$T_{\pi,r} = \{i \in N : \pi(i) \leq r\}, r \in \{1,...,|N|\} \tag{3.5}$$

où $T_{\pi,0} = \{\}$ et $T_{\pi,|N|} = N$. Notons que $\pi(i)$ représente la position du joueur i dans la permutation π. Pour déterminer le noyau d'un ordre particulier π, nous résolvons l'équation ci-dessous :

$$x_i^{\pi}(T_{\pi,r}) = v(T_{\pi,r}), r \in \{1,...,|N|\} \tag{3.6}$$

La solution de ces équations définit le vecteur de gain x^{π} avec des éléments donnés par :

$$x_i^{\pi} = v(T_{\pi,\pi(i)}) - v(T_{\pi,\pi(i)-1}), \forall \{1,...,|N|\}. \tag{3.7}$$

En effet, le vecteur de gain x^{π} représente précisément les points extrêmes du noyau dans les jeux convexes. De plus, il est connu [140, 141] que la valeur de shapley pour un jeu convexe est

le centre de gravité de x^{π}. Ainsi, si Π est l'ensemble de toutes les permutations de N, alors la valeur de shapley du joueur i peut être calculée comme suit :

$$\phi_i = \frac{1}{|N|!} \sum_{\pi \in \Pi} x_i^{\pi} \tag{3.8}$$

3.4 Approche proposée : détail du module "*SimilarUser*"

Comme le montre la figure 3.1 notre approche comporte deux modules. Le premier module a pour rôle la présélection des utilisateurs similaires. Le deuxième module applique le processus du filtrage collaboratif sur chaque groupe d'utilisateurs similaires obtenu.

Dans cette section, nous détaillons la fonctionnalité du module "*SimilarUser*".

3.4.1 Valeur de Shapley pour former des groupes d'utilisateurs similaires

Les performances de l'algorithme *FC* sont très sensibles à sa première étape, qui est la création de groupes d'utilisateurs similaires. En effet, la qualité des résultats de *FC* dépend de la qualité des groupes créés [8]. En fait, le problème de la création de groupes peut être vu comme une tâche de clustering. Par conséquent, le choix d'un algorithme efficace pour créer des clusters d'utilisateurs similaires est l'une des questions les plus importantes de l'algorithme *FC*.

Notre travail consiste en premier lieu à la création de groupe comme une étape de présélection. Puis, l'application de l'algorithme *FC* pour chaque groupe. La première étape (le processus de regroupement des utilisateurs) permet d'améliorer la performance de la recommandation, car le regroupement considéré contient beaucoup moins d'utilisateurs par rapport à la population générale, qui est constituée de tous les utilisateurs.

Au cours de notre travail, nous adoptons le concept de solution *SV* qui permet de créer des groupes d'utilisateurs similaires et de prendre en compte certaines propriétés intrinsèques, qui ne sont pas prises en compte par les algorithmes conventionnels de clustering. L'idée principale ici est de faire correspondre la création de groupes à la formation de coalitions. L'utilisation de la *SV* pour le regroupement d'utilisateurs similaires est justifiée par ses qualités axiomatiques détaillées dans 3.3.2.3, en d'autres termes :

— La symétrie est importante pour garantir l'indépendance de l'ordre (une propriété importante du regroupement). De manière informelle, le modèle produit les mêmes groupes finaux à travers différentes exécutions, indépendamment de la séquence dans laquelle les utilisateurs sont fournis en entrée.

— Propriété du joueur nul, cette propriété implique que si un utilisateur ne contribue pas à la valeur globale d'un groupe, il n'est pas incité à y devenir membre. En d'autres termes, un utilisateur dissimilaire à l'ensemble d'un groupe , n'a pas à faire partie du groupe.

— L'additivité indique la propriété de linéarité qui est primordial pour garantir l'invariance d'échelle par rapport à la fonction de similarité utilisée. Ce qui est parfait pour un regroupement correct.

— Efficacité , en conséquence de cette propriété, la valeur globale qui résulte de la présence de chaque utilisateur de l'ensemble de données est entièrement distribuée entre tous les utilisateurs, ce qui caractérise le comportement collectif des utilisateurs au sein d'un même groupe, ce qui est fondamental pour le clustering.

En fait, la valeur de Shapley est le seul concept de solution qui satisfait ces axiomes simultanément [142], et par conséquent fournit une approche motivée et convaincante pour regrouper les utilisateurs similaires.

3.4.2 Modèle du jeu coopératif

Supposons que $U = \{u_1, u_2, \ldots, u_n\}$ est un ensemble de données de n utilisateurs. Soit $d(u_i, u_j)$la distance entre u_i et u_j $\forall u_i, u_j \in U$, avec $d(u_i, u_i) = 0$; d représente n'importe quelle fonction calculant la distance entre deux points.
Soit $f' : R^+ \cup \{0\} \longrightarrow [0,1)$ une fonction de dissimilarité monotone non décroissante sur d tel que : $f'(0) = 0$. Définissons un mappage de la fonction de similarité correspondante f' : $R^+ \cup \{0\} \longrightarrow [0,1)$, tel que $f(a) = 1 - f'(a)$

Notre problème de regrouper les utilisateurs similaires peut être considéré comme le fait de regrouper les utilisateurs qui sont moins dissimilaires comme indiqué par f' où, de manière équivalente, regrouper les utilisateurs les plus similaires comme indiqué par f.

Nous mettons en place un jeu coopératif (N, v). Dans ce contexte, chacun des n utilisateurs correspond à un joueur dans le jeu, tel que $|N| = n$. Chaque utilisateur interagit avec d'autres utilisateurs et tente de former une coalition (ou un groupe) avec eux, afin de maximiser sa valeur (de similarité). Notons que le grand groupe a la valeur globale maximale de tous les groupes ; cependant, la valeur de shapley dépend de « l'augmentation moyenne de valeur » dans tous les sous-groupes valides plutôt que de la valeur globale. Ceci est important afin de s'assurer qu'un nombre approprié de groupes d'utilisateurs similaires est obtenu au lieu qu'un seul groupe. Nous attribuons $v(u_i) = 0$, pour tout u_i tel que u_i n'est membre d'aucune coalition. Pour une coalition T nous définissons :

$$v(T) = \frac{1}{2} \sum_{\substack{u_i, u_j \in T \\ u_i \neq u_j}} f(d(u_i, u_j)) \tag{3.9}$$

Nous soulignons la pertinence de définir la valeur de la fonction $v(.)$. Notre approche calcule la valeur totale d'une coalition comme la somme des similarités par paire entre les utilisateurs. Notez que cette formulation capture élégamment la notion de regroupement dans sa forme naturelle : les éléments appartenant à un même cluster sont similaires les uns aux autres [143].

3.4.3 Convexité du jeu

Théorème 1. *Définissons la valeur d'un utilisateur individuel u_i, $v(u_i) = 0 \forall i \in \{1,2...,n\}$, et celle d'une coalition T de n utilisateurs du jeu de données,*

$$v(T) = \frac{1}{2} \sum_{\substack{u_i,u_j \in T \\ u_i \neq u_j}} f(d(u_i,u_j)), \tag{3.10}$$

où f est la fonction de similarité. Dans ce contexte, le jeu coopératif (N,v) est un jeu convexe.

Démonstration. Considérons deux coalitions quelconques, C et D , $C \subseteq D \subseteq X \backslash u_p$ où $u_p \in X$. Dans ce cas, on a :

$$\begin{aligned}
v(D \cup \{u_p\}) - v(C \cup \{u_p\}) &= \frac{1}{2} \sum_{\substack{u_i,u_j \in D \\ u_i \neq u_j}} f(d(u_i,u_j)) + \sum_{u_i \in D} f(d(u_i,u_p)) \\
&\quad - \frac{1}{2} \sum_{\substack{u_i,u_j \in C \\ u_i \neq u_j}} f(d(u_i,u_j)) - \sum_{u_i \in C} f(d(u_i,u_p)) \\
&= \frac{1}{2} \sum_{\substack{u_i,u_j \in D \backslash C \\ u_i \neq u_j}} f(d(u_i,u_j)) + \sum_{u_i \in D \backslash C} f(d(u_i,u_p)) + \sum_{u_i \in D \backslash C} f(d(u_i,u_p)) \\
&= v(D) - v(C) + \sum_{u_i \in D \backslash C} f(d(u_i,u_p)) \\
&\geq v(D) - v(C) (\mathrm{f} : \mathrm{R}^+ \cup \{0\} \longrightarrow [0,1))
\end{aligned} \tag{3.11}$$

□

Dans ce qui suit, nous montrons que les utilisateurs les plus similaires ont des *SVs* presque égales.

Théorème 2. *Pour deux utilisateurs quelconques u_i, u_t, tel que $d(u_i,u_t) \leq \varepsilon$, où $\varepsilon \to 0$, dans le cadre du jeu convexe de la section 3.4.2 ont des valeurs de SV presque égales.*

Démonstration. Comme nous l'avons mentionné dans la section 3.3.4,la *SV* de l'utilisateur u est calculée comme suit :

$$\begin{aligned}
\phi_i &= \frac{1}{n!} \sum_{\pi \in \Pi} \left[v(T_{\pi,\pi(i)}) - v(T_{\pi,\pi(i)-1}) \right] \\
&= \frac{1}{n!} \sum_{\pi \in \Pi} \Big[\sum_{\substack{\pi(p) \leq \pi(i) \\ \pi(q) < \pi(p)}} f(d(u_p,u_q)) - \sum_{\substack{\pi(p) \leq \pi(i)-1 \\ \pi(q) < \pi(p)}} f(d(u_p,u_q)) \Big] \\
&= \frac{1}{n!} \sum_{\pi \in \Pi} \sum_{\pi(p) < \pi(i)} f(d(u_i;u_p))
\end{aligned} \tag{3.12}$$

□

Comme nous l'avons indiqué dans la section 3.4.2 f représente la fonction de similarité. Pour un utilisateur u_t, $t \in \{1,2,...,n\}, t \neq i$ tel que $d(u_i,u_t) \leq \varepsilon$,on a : $f(d(u_i,u_t) \to 1$ (plus petite distance implique une grande similarité) et $f(d(u_i,u_p) \to f(d(u_t,u_p)$ ce qui implique que $\phi_t \to \phi_i$.

Ce théorème prouve que les utilisateurs similaires ont des SVs presque égales. Par conséquent, les utilisateurs ayants des SVs presque égales devraient faire partie du même groupe.

Il faut noter que le théorème ne révèle rien sur les utilisateurs les moins similaires. En d'autres termes, le théorème n'empêche pas que les utilisateurs les moins similaires les uns des autres aient des SVs similaires.

3.4.4 Valeur de Shapley de notre jeu défini

Le calcul exact de la valeur de Shapley se fait suivant la formule 3.1. Dans ce qui suit, nous présentons la formule finale du calcul de SV pour chaque utilisateur dans le cadre de notre modèle du jeu coopératif.

Pour notre choix de fonction de valeur, $v(T)$, nous pouvons calculer efficacement la valeur de Shapley en utilisant le résultat de convexité du théorème 1, comme indiqué ci-dessous.

Puisque notre jeu est convexe, la valeur de Shapley d'un utilisateur u_i peut être calculée en utilisant les formules 3.8 et 3.7 comme suit :

$$\phi_i = \frac{1}{|N|!} \sum_{\pi \in \Pi} \left[v(T_{\pi,\pi(i)}) - v(T_{\pi,\pi(i)-1}) \right], \tag{3.13}$$

ce qui peut être reformuler en utilisant 3.9 comme suit :

$$\begin{aligned} \phi_i &= \frac{1}{|n|!} \sum_{\pi \in \Pi} \sum_{\pi(j)<\pi(i)} f(d(u_i,u_j)) \\ &= \frac{1}{|n|!} \Bigg[\sum_{\substack{\pi(i)=1 \\ \pi(j)<\pi(i) \\ \pi \in \Pi}} f(d(u_i,u_j)) \\ &+ \sum_{\substack{\pi(i)=2 \\ \pi(j)<\pi(i) \\ \pi \in \Pi}} f(d(u_i,u_j)) + ... + \sum_{\substack{\pi(i)=n \\ \pi(j)<\pi(i) \\ \pi \in \Pi}} f(d(u_i,u_j)) \Bigg] \end{aligned} \tag{3.14}$$

Avec la formule obtenue, chaque u_i a $(n-1)!$ permutations possibles. Par conséquent, si l'on fait la somme de toutes ces permutations, chaque utilisateur autre que u_i apparait exactement $\frac{(n-1)!}{(n-1)}$ fois dans chacune des positions précédentes $(1,2,...,(i-1))$. De ce fait, on obtient :

$$\begin{aligned}\phi_i &= \frac{(n-1)!}{(n-1)n!}[1+2+...+(n-1)] \sum_{\substack{u_i \in U \\ i \neq j}} f(d(u_i,u_j)) \\ &= \frac{(n-1)!}{(n-1)n!}[(n-1)\frac{n}{2}] \sum_{\substack{u_i \in U \\ i \neq j}} f(d(u_i,u_j)) \\ &= \frac{1}{2} \sum_{\substack{u_i \in U \\ i \neq j}} f(d(u_i,u_j))\end{aligned} \tag{3.15}$$

La formule finale du calcul de SV pour un utilisateur donné est donnée par :

$$\phi_i = \frac{1}{2} \sum_{\substack{u_i \in U \\ i \neq j}} f(d(u_i,u_j)) \tag{3.16}$$

Cette dernière est issue de la fonction caractéristique définie 3.9 et de la convexité de notre jeu coopératif.

3.4.5 La fonction de similarité f utilisée

Pour calculer la similarité entre les utilisateurs d'un système de recommandation à base du filtrage collaboratif, le coefficient de corrélation de Pearson 2.1 est le plus utilisé dans la littérature [89, 144]. Ce dernier mesure la dépendance entre deux vecteurs d'évaluations d'items de deux utilisateurs différents. Nous avons adopté ce coefficient pour le calcul de la fonction de similarité f.

3.4.6 Algorithme $CF-GT$

Notre modèle de recommandation [8] comprend deux phases (modules), la première sert à regrouper les utilisateurs similaires suivant un modèle de jeu coopératif, en utilisant SV . Quant à la seconde, elle consiste à l'application de l'algorithme de filtrage collaboratif sur chaque groupe d'utilisateurs pour un utilisateur actif u_i. La figure 3.2 schématise la fonctionnalité des deux modules ainsi que le lien entre ces derniers.

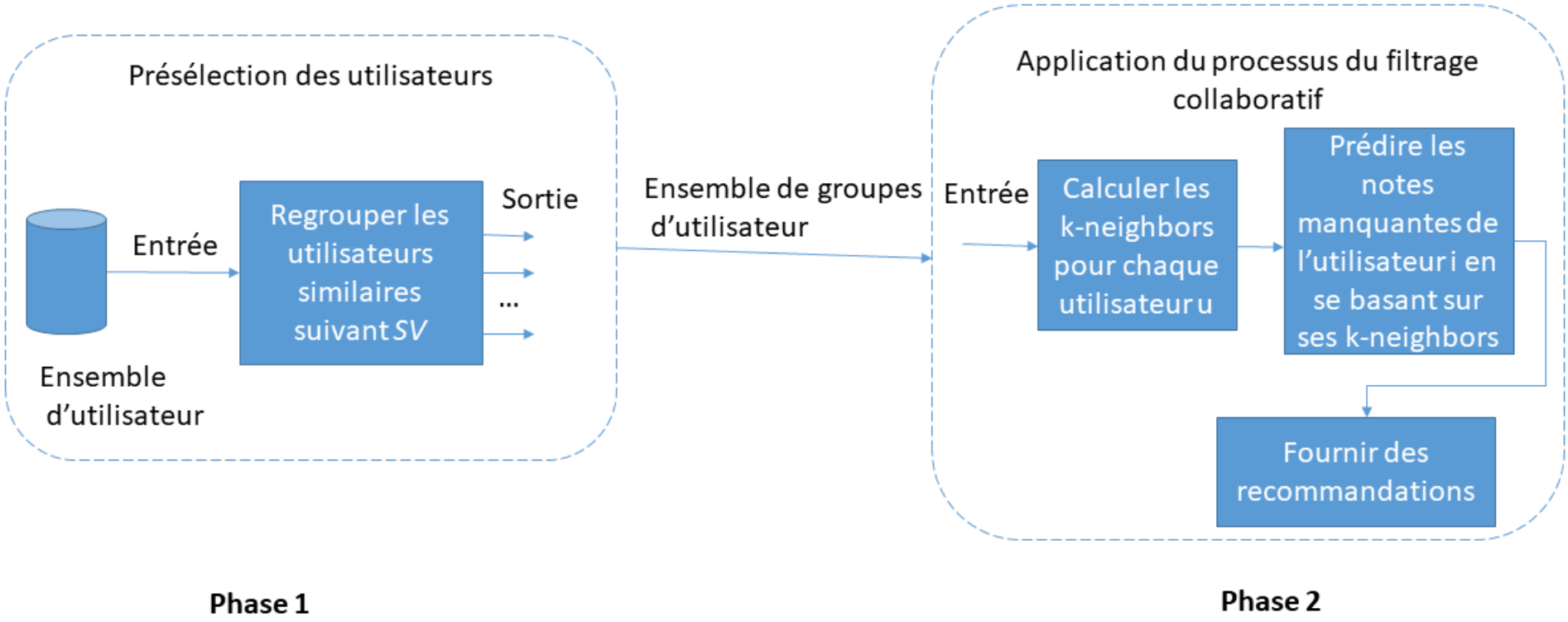

FIGURE 3.2 – Approche $CF - GT$

Phase 1 : (Création de groupe d'utilisateur en utilisant *SV*)

Dans cette étape, $CF-GT$ prend en entrée un ensemble de données d'utilisateurs à regrouper ainsi qu'un seuil de similarité σ. Tout d'abord, la valeur de Shaply de chaque utilisateur est calculée suivant 3.16. Ensuite, de manière itérative, l'algorithme choisit un utilisateur non attribué (u_m) dont la valeur de Shapley est la plus grande et l'assigne comme centre du groupe g_m, puis attribue tous les utilisateurs dont la similarité avec u_m est supérieure ou égale à σ au groupe g_m.

L'utilisation du seuil σ garantit que :
1) les utilisateurs proches, qui ont tendance à avoir une *SV* presque égale, sont affectés au même groupe et que 2) le point de départ de chaque groupe est raisonnablement éloigné. En sortie, il produit un ensemble *G* contenant tous les groupes d'utilisateurs similaires g_i. L'étape de création de groupes d'utilisateurs similaires est résumée dans l'Algorithme 1 :

Algorithm 1: "*SimilarUser*" :création de groupes d'utilisateurs similaires

Input: $U = \{u_1, u_2, \ldots, u_n\}$ ensemble d'utilisateurs
σ seuil de similarité.
Output: G=$\{g_1, g_2, \ldots, g_k\}$ ensemble de groupes d'utilisateurs similaires.

Calculer *SV* pour chaque utilisateur de *U*
for $i = 1$ *to* n **do**
    $\phi_i = \frac{1}{2}\sum_{\substack{u_i \in U \\ i \neq j}} f(d(u_i, u_j))$
Former des groupes d'utilisateurs similaires en suivant σ
$C = U$;
while $C \neq \emptyset$ **do**
    m=argmax(ϕ_i) ;
    G_m=$\{u_i \in C : f(d(u_m, u_i)) \geq \sigma\}$;
    C=C \G_m ;

3.5 Approche proposée : détail module "*CFProcess*"

Comme illustré dans la figure 3.1, le module "*CFProcess*" de notre approche prend en entrée la sortie du module "*SimilarUser*". Pour rappel, le premier module se charge de la présélection d'utilisateurs similaires. Il prend en entrée un ensemble de donnée (utilisateur / item) et en sortie un ensemble de groupe d'utilisateurs similaires *G*.

Une fois l'ensemble *G* obtenu, il est directement envoyé comme entrée du module "*CFProcess*". Ce dernier traite chaque ensemble g_i de *G* séparément en y appliquant le processus du filtrage collaboratif.

Le filtrage collaboratif vise à recommander des items susceptibles d'intéresser l'utilisateur. L'algorithme passe par les étapes suivantes :

- Rechercher les utilisateurs similaires à l'utilisateur actif u_i.
- Prédire les notes manquantes de l'utilisateur u_i en fonction des notes attribuées par les utilisateurs similaires obtenus par l'étape précédente.
- Générer le $top-N$ recommandations.

Dans ce qui suit, nous détaillons chaque étape du FC.

3.5.1 Rechercher les utilisateurs similaires

Cette étape prend en entrée l'ensemble G d'utilisateurs similaires obtenu par le module "*SimilarUser*". Pour chaque groupe $g_i \in G$ cette étape calcule les utilisateurs les plus similaires à l'utilisateur actif u_i et génère ses $k-neighbors$ (les k utilisateurs les plus similaires à u_i).

La similarité entre les utilisateurs est calculée suivant le coefficient de corrélation de Pearson 2.1

Une fois les utilisateurs similaires à l'utilisateur actif u_i obtenus, l'algorithme prédit les notes manquantes de u_i en se basant sur les notes attribuées par ses utilisateurs similaires.

3.5.2 Prédiction des entrées manquantes

Cette étape reçoit les utilisateurs similaires à l'utilisateur actif u_i et prédit les notes manquantes suivant la formule 2.5.

Après la prédiction des notes, l'utilisateur u_i n'aura aucune entrée manquante.

3.5.3 $top-N$ recommandations

La génération des $top-N$ recommandations est la dernière étape du processus du filtrage collaboratif. Les notes prédites par l'étape précédente sont classées par ordre décroissant. Ensuite, le système propose à l'utilisateur les N premiers items qu'il n'a pas consultés auparavant.

Phase 2 : (Génération des $top-N$ recommandations)

Dans cette étape, $CF-GT$ prend en entrée l'ensemble G. Cette étape a pour but de fournir les $top-N$ recommandations pour chaque utilisateur sur la base du calcul de la matrice utilisateurs/éléments et de la prédiction des évaluations. L'étape de génération des $top-N$ recommandations est résumée dans l'Algorithme 2 :

3.6 Validation du modèle $CF-GT$

Dans cette partie, nous présentons le protocole expérimental utilisé pour évaluer le modèle proposé.

Algorithm 2: Génération des $top-N$ recommandations

Input: G=$\{g_1, g_2, \ldots, g_k\}$ ensemble de groupes d'utilisateurs similaires.
Output: top-N recommendations

1 **foreach** $g_i \in G$ **do**
    Construire la matrice utilisateur \ item M
    Calculer la similarité entre les utilisateurs suivant M
2 **foreach** *utilisateur* u_i **do**
    Donner les k-neighbors de u_i `/* les k utilisateur les plus similaires */`
    Predire la note qui l'utilisateur u_i donnerait à chaque item selon les k-neighbors
    Fournir les top-N recommandations

3.6.1 Dataset

Afin d'évaluer l'algorithme $CF-GT$ nous avons choisi le jeu de données $MovieLens100k$ [145] qui couvre le domaine de recommandations de films.

$MovieLens100k$ dataset

Ce jeu de données se compose de 100000 évaluations de 943 utilisateurs sur 1682 éléments. Les évaluations ont été faites sur une échelle de 1 à 5 étoiles par des utilisateurs qui ont rejoint MovieLens en 2000 et ont saisi au moins 20 évaluations. Cet ensemble de données est le plus petit ensemble de MovieLens, Avec le plus petit $MovieLens100k$, c'est l'un des plus utilisés pour tester les algorithmes de recommandation dans la littérature. Il a l'avantage d'être suffisamment petit pour que les calculs puissent être effectués en peu de temps sur du matériel de base et fournir des résultats significatifs grâce à la densité et à la qualité des données. Cela facilite la reproductibilité de nos expériences et la comparaison avec les résultats d'autres auteurs. Pour ces raisons, nous l'avons choisi comme le jeu de données dans nos expériences.

3.6.2 Modèles de comparaison

Afin de valider notre approche, nous l'avons comparé aux approches existantes. Nous nous sommes concentrés sur les algorithmes de filtrage collaboratif, car non seulement notre modèle se base sur cet algorithme, mais aussi, ils se sont avérés très efficaces avec l'ensemble de données *MovieLens* présenté dans la section précédente. En effet, nous avons choisi le filtrage collaboratif conventionnel et le filtrage collaboratif basé sur $K-means$ ($K-means-CF$). Le choix de ces deux algorithmes se justifie par le fait que le filtrage collaboratif est un algorithme de base qui reste une référence importante. Quant au $K-means-CF$, c'est un modèle qui est basé sur le clustering et ça démarche et très proche de notre modèle $CF-GT$. Ces deux algorithmes nous permettent d'évaluer les performances des recommandations fournies par notre approche.

3.6.3 Protocole d'évaluation

Dans cette section, nous détaillons la manière dont nous avons réalisé l'évaluation des recommandations fournies par les algorithmes de recommandation de référence précédents et notre modèle proposé sur le jeu de données $MovieLens100k$ et $EduTest$.

Nous avons testé notre modèle par rapport aux évaluations prédites et à la pertinence des recommandations fournies. Pour de telles évaluations, deux étapes sont requises : l'ensemble de données de recommandation sera d'abord divisé en ensembles d'apprentissage et de test. Les modèles de recommandation sont appris sur l'ensemble d'apprentissage et évalués sur l'ensemble de test. Par exemple, nous pouvons utiliser 80 % de l'ensemble de données pour l'apprentissage, tandis que les 20 % restants servent à l'évaluation. Les protocoles d'évaluation les plus couramment utilisés sont la validation croisée 5 fois [146]. Nous avons adopté ce protocole pour l'évaluation de notre modèle où 4 jeux sont utilisés pour l'apprentissage et le cinquième est utilisé pour le test.

Dans cette thèse, toutes les évaluations sont basées sur le protocole hors ligne [147] (c'est-à-dire sur des jeux de données inchangés). Comme dernière phase, les prédictions des notes fournies par le système sont évaluées en termes de prédiction et de pertinence. **La plupart des modèles de recommandation, nous trions les évaluations prédites des éléments candidats et recommandons les $top-N$ éléments à l'utilisateur cible (les meilleurs N éléments.**

3.6.4 Résultats

Dans cette section, les résultats de l'évaluation de notre approche $CF-GT$ sont présentés en effectuant diverses expériences. Pour toute expérience réalisée, nous considérons trois valeurs du seuil de similarité $\sigma = 0.79, 0.86$ et 0.87 qui donne respectivement deux, trois et quatre groupes d'utilisateurs similaires.

1. **Expérience 1** Pour les métriques d'évaluation de l'exactitude 2.6.2.1, nous mesurons la valeur MAE entre les évaluations prédites et réelles et nous le comparons avec les approches citées précédemment. Comme dans les études précédentes [8, 148, 149], nous considérons différent nombre de voisins k pour cette évaluation $k = \{5;10;15;20;25;30;35;40;45$ et 50.

   Les figures 3.3, 3.4 et 3.5 montrent la précision de la prédiction pour différentes tailles de voisinage k sur le jeu de données $MovieLens$.

   On constate que pour les tailles de voisinage considérées, $CF-GT$ permet d'améliorer remarquablement la précision de prédiction par rapport à l'algorithme FC et $K-means-CF$. Toutes les méthodes donnent les prédictions les plus précises lorsque le nombre de voisins est d'environ 40.

   Sans exception, $CF-GT$ surpasse les algorithmes de test quel que soit le nombre de voisins.

   La supériorité de la méthode proposée peut s'expliquer par le fait que $CF-GT$ exploite

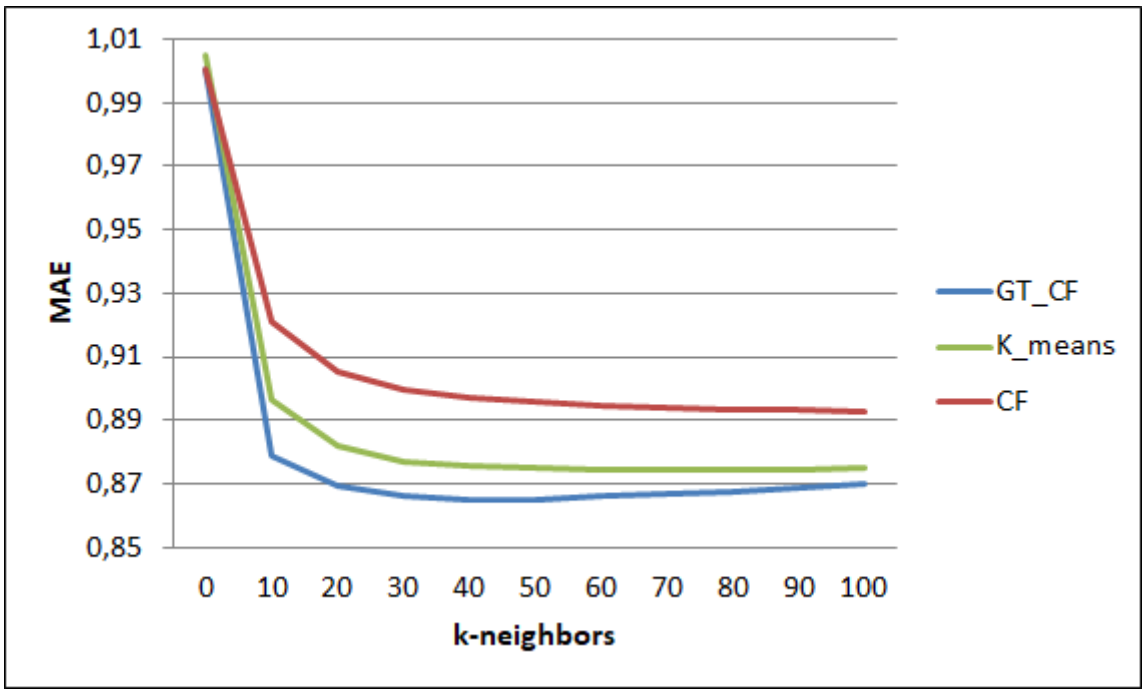

FIGURE 3.3 – *MAE* pour σ =0.79 / MovieLens

le concept de solution *SV* pour former un groupe d'utilisateurs similaires comme présélection pour l'algorithme *FC*.

Nous rappelons que le concept *SV* prend en compte la relation entre une paire d'utilisateurs et les autres utilisateurs du même groupe.

2. **Expérience 2**

   Pour déterminer l'efficacité du système, nous avons calculé la précision (pour déterminer la probabilité qu'un élément recommandé soit pertinent) et le rappel (pour déterminer la probabilité qu'un élément pertinent soit recommandé).

   En ce qui concerne la pertinence des recommandations fournie 2.6.2.2l, la précision et le rappel ont été calculés sur différents nombres de $top-N$. Dans notre étude, nous considérons ($N = \{5;10;15;20;25;30;35;40;45$ et 50) ce qui signifie que nous évaluons la méthode lors de la recommandation des éléments du $top-N$ par le système de recommandation proposé. Les tableaux 3.3, 3.4 et 3.5 montrent les valeurs de précision et de rappel pour les différents $top-N$ avec la base *MovieLens*.

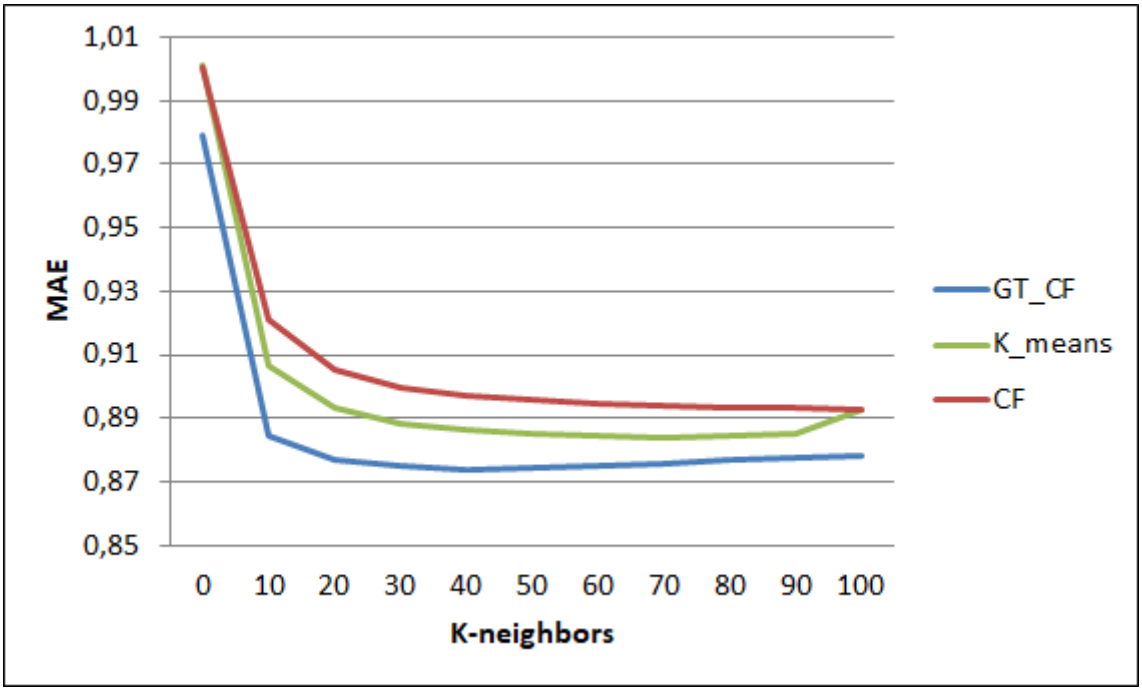

FIGURE 3.4 – *MAE* pour $\sigma = 0.86$ / MovieLens

TABLE 3.3 – Précision et rappel pour $\sigma = 0.79$ *MovieLens*

	CF-GT		FC		K_means	
Top	**Précision**	**Rappel**	**Précision**	**Rappel**	**Précision**	**Rappel**
5	25%	25%	20%	20%	21%	21%
10	37%	38%	33%	33%	33%	33%
15	45%	45%	40%	41%	41%	42%
20	50%	51%	46%	48%	47%	49%
25	54%	55%	50%	52%	51%	52%
30	57%	58%	53%	55%	54%	55%
35	59%	61%	56%	58%	56%	59%
40	61%	63%	58%	60%	59%	61%
45	62%	65%	59%	62%	59%	63%
50	63%	66%	61%	64%	61%	65%

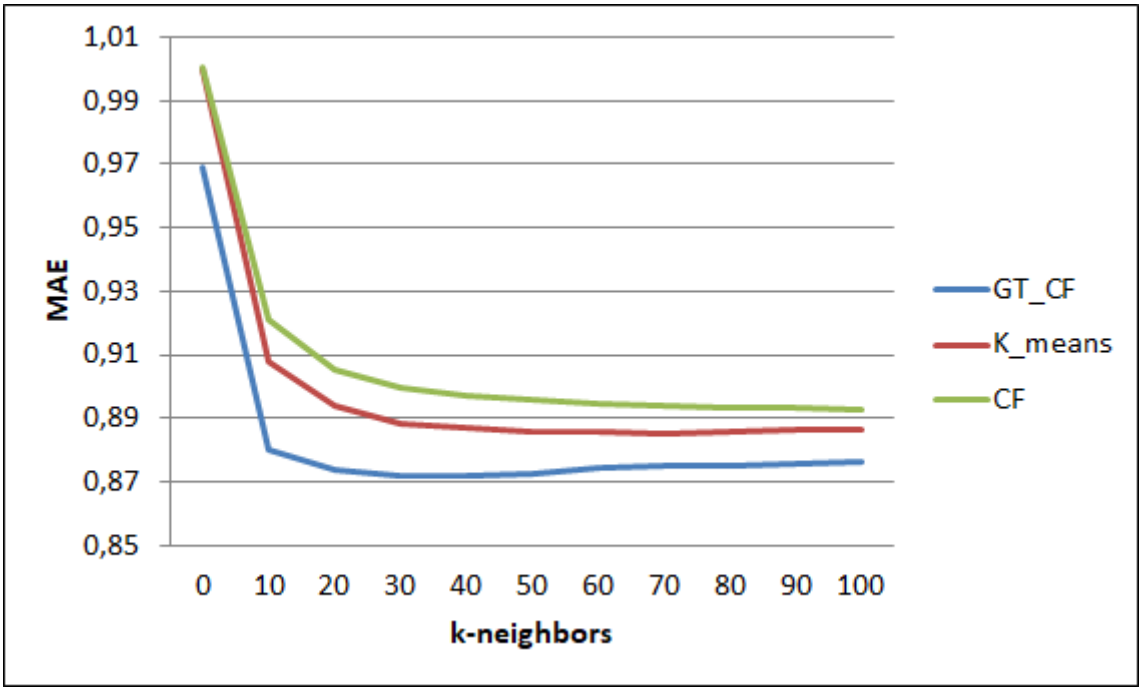

FIGURE 3.5 – *MAE* pour $\sigma = 0.87$ / MovieLens

TABLE 3.4 – Précision et rappel pour $\sigma = 0.86$ *MovieLens*

	CF-GT		FC		K_means	
Top	**Précision**	**Rappel**	**Précision**	**Rappel**	**Précision**	**Rappel**
5	25%	25%	20%	20%	22%	22%
10	37%	37%	33%	33%	34%	34%
15	45%	46%	40%	41%	42%	43%
20	51%	51%	46%	48%	47%	50%
25	54%	55%	50%	52%	51%	53%
30	57%	59%	53%	55%	54%	56%
35	60%	61%	56%	58%	57%	59%
40	61%	63%	58%	60%	59%	61%
45	63%	65%	59%	62%	60%	63%
50	64%	67%	61%	64%	62%	65%

TABLE 3.5 – Précision et rappel pour $\sigma = 0.87$ *MovieLens*

	CF-GT		*FC*		*K_means*	
Top	**Précision**	**Rappel**	**Précision**	**Rappel**	**Précision**	**Rappel**
5	26%	26%	20%	20%	21%	21%
10	38%	39%	33%	33%	35%	34%
15	46%	47%	40%	41%	42%	42%
20	53%	52%	46%	48%	47%	49%
25	56%	56%	50%	52%	51%	53%
30	58%	60%	53%	55%	54%	56%
35	61%	62%	56%	58%	57%	59%
40	62%	64%	58%	60%	59%	62%
45	64%	66%	59%	62%	61%	63%
50	65%	68%	61%	64%	62%	65%

À partir de ces tableaux, nous pouvons observer que pour différents $top-N$, la précision et le rappel obtenus par notre méthode $CF-GT$ sont meilleurs par rapport aux méthodes d'évaluation considérées.

3.7 Discussion

Pour optimiser le processus du filtrage collaboratif, le clustering a été choisi comme étape de présélection des utilisateurs similaires. Les techniques de clustering conventionnelles, par exemple les $k-means$, tentent de minimiser la distance moyenne au carré entre chaque point et son centre de cluster le plus proche. Pour regrouper les utilisateurs les plus similaires, ces algorithmes utilisent la distance entre l'utilisateur et les différents centroïdes. La distance utilisateur-utilisateur n'est pas prise en compte.

Notre modèle $CF-GT$ présente une nouvelle méthode de filtrage collaboratif basée sur le clustering. $CF-GT$ est constitué de deux modules : «*SimilarUser* » et «*CFProcess*. Le premier module est le cœur de notre contribution. Ce dernier présélectionne les utilisateurs similaires grâce à une méthode de clustering basée sur le concept de solution "valeur de Shapley". Ce dernier donne une solution optimale en maintenant au minimum les distances utilisateur à centre et utilisateur à utilisateur, au sein d'un cluster.

En observant les résultats en termes de *MAE* et de (précision et rappel) nous pouvons remarquer que l'erreur résultante de FC est supérieure par rapport à l'erreur de notre modèle $CF-GT$ et le modèle de comparaison $K-means-CF$. De même, nous remarquons que la précision des recommandations fournies par $CF-GT$ et $K-means-CF$ surpasse la précision de FC. ces résultats concordent avec notre point de départ : les modèles qui adoptent le clustering comme étape de présélection des utilisateurs similaires donnent de meilleurs résultats par rapport à ceux qui n'utilise pas une présélection.

En ce qui concerne la comparaison de notre modèle $CF-GT$ avec le $K-means-CF$ nous pouvons constater que les prédictions des entrées manquantes données par notre modèle

$CF-GT$ sont plus précises par rapport à celles données par $K-means-CF$. Nous constatons également que les recommandations fournies par $K-means-CF$ sont moins exactes que celles fournies par $CF-GT$. Ce constat peut être expliqué par le fait que le module « *SimilarUser*» produit la présélection des utilisateurs similaires avec la prise en considération de l'importance de chaque utilisateur par rapport à son représentant le plus proche et aux autres utilisateurs du même cluster, contrairement à la présélection avec $k-means$ qui prend en compte que l'importance de chaque utilisateur par rapport à son représentant le plus proche (centre du cluster).

3.8 Conclusion

Dans ce chapitre, nous avons présenté une contribution afin d'optimiser les recommandations fournies par le filtrage collaboratif. L'analyse de la littérature nous a permis de conclure que plusieurs chercheurs se servent du clustering comme une étape de présélection des utilisateurs similaires, les résultats expérimentaux de ces études montrent une amélioration significative des résultats obtenus. Dans cette perspective, nous avons développé un nouveau modèle de recommandation $CF-GT$ basé sur le filtrage collaboratif et le concept de solution SV de la théorie des jeux.

$CF-GT$ est un modèle générique constitué de deux modules : « *SimarUser* » et « *CFProcess* ». «*SimilarUser* » se charge de la présélection des utilisateurs similaire en utilisant SV. Quant à «*CFProcess* » il se charge de l'application de FC pour chaque groupe obtenu par le premier module.

Après avoir donné les détails de notre approche, nous l'avons validée avec une série d'expériences. Les résultats expérimentaux, ainsi que le protocole expérimental suivi, ont été détaillés.

Afin d'aller jusqu'au bout de l'objectif de cette thèse, l'application de notre approche dans le domaine éducatif s'impose. Dans le chapitre suivant, nous présentons les détails de l'adaptation de $CF-GT$ au domaine éducatif. La validation dans le domaine éducatif est également présentée.

Chapitre 4

Application du modèle $CF-GT$ dans le domaine éducatif

4.1 Introduction

Dans le chapitre précédent, nous avons présenté une nouvelle approche de filtrage collaboratif qui se base sur la présélection des utilisateurs similaires en utilisant le concept de solution *SV* issue de la théorie des jeux. Le modèle proposé $CF-GT$ est un modèle de recommandations générique.

Pour répondre à l'objectif de notre thèse, nous avons adapté le modèle $CF-GT$ au domaine éducatif pour recommander des ressources pédagogiques aux apprenants. Nous avons effectué une adaptation au niveau de la fonction de similarité et nous avons construit notre base de données éducatives nommée $EduTest$.

Dans ce chapitre, nous allons détailler notre base de données et expliquer l'adaptation du modèle $CF-GT$ au domaine éducatif. L'adaptation a donné un nouveau modèle nommé $Edu-CF-GT$. Après les détails du modèle $Edu-CF-GT$, nous allons valider la proposition en utilisant $Edutest$ dataset en suivant le même protocole expérimental suivi dans le chapitre 3.

4.2 Implémentation du système

L'adaptation du modèle $CF-GT$ au domaine éducatif a donné naissance à un nouveau modèle nommé $Edu-CF-GT$. Les utilisateurs cibles sont des étudiants dont le profil prend en compte plusieurs critères : (nom, niveau, spécialité, ect.)

Le modèle $Edu-CF-GT$ comprend deux modules : "$InfoCollect$" et "$CF-GTProcess$". Le module "$InfoCollect$" consiste à la collecte des données relatives aux étudiants. Le module "$CF-GTProcess$" se charge de la recommandation suivant le modèle $CF-GT$ et les données d'entrée.

4.2.1 Schéma global du modèle $Edu - CF - GT$

La figure 4.1 illustre le modèle $Edu - CF - GT$.

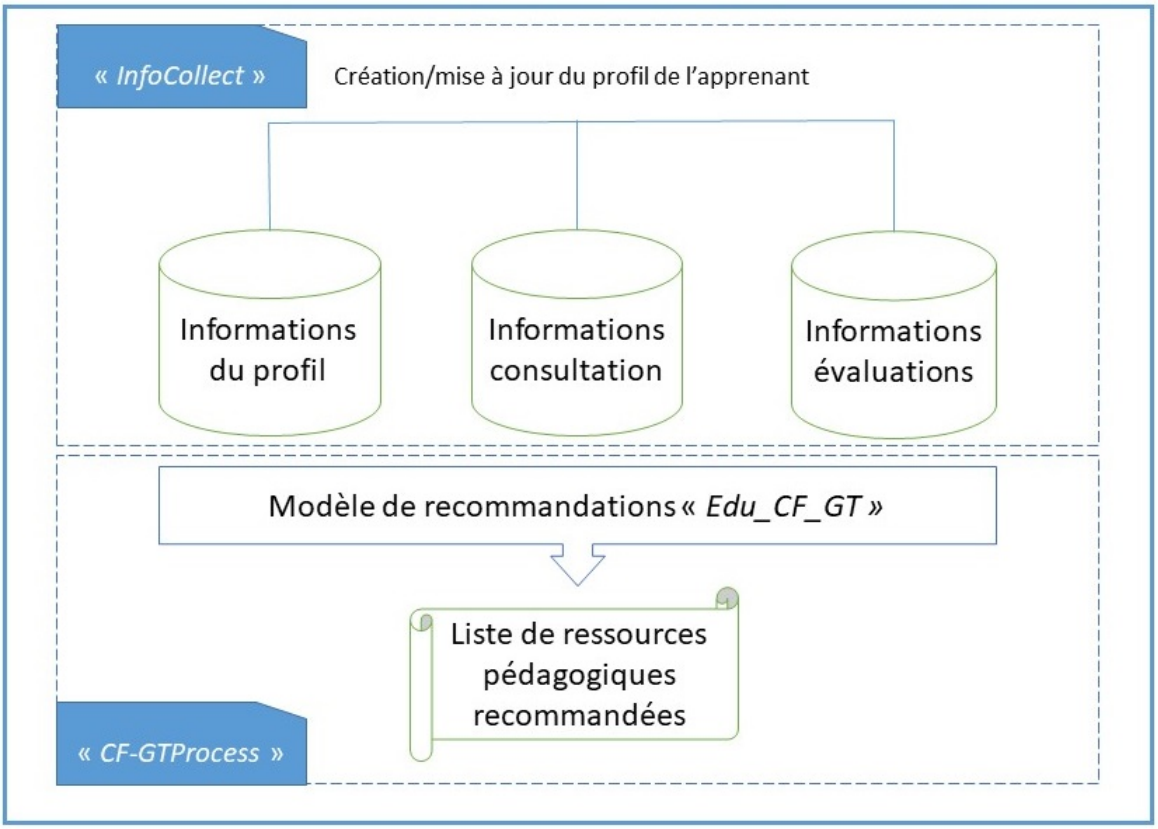

FIGURE 4.1 – Schéma global du système de recommandation de ressources pédagogiques

4.2.1.1 Module "*InfoCollect*"

Le premier module sert de collecte d'informations qui caractérisent les apprenants.

Chaque utilisateur du système d'apprentissage informatisé est décrit par des informations qui caractérisent son profil.

Les utilisateurs peuvent consulter les ressources pédagogiques et attribuent une note à chaque ressource évaluant sa qualité.

Le système de recommandation utilise les informations citées pour générer des recommandations personnalisées pour chaque apprenant de l'environnement d'apprentissage.

Une fois connectés dans l'environnement d'apprentissage, les apprenants préfèrent obtenir des recommandations appropriées pour leurs besoins pédagogiques sans avoir à passer des heures à consulter les milliers de ressources chargées dans le système.

L'environnement d'apprentissage informatisé récupère ces informations à chaque fois qu'un apprenant effectue une nouvelle action (consulte une nouvelle ressource ou attribue une note à une ressource) et met à jour son profil.

4.2.1.2 Module "$CF-GTProcess$"

Le deuxième module se charge de la recommandation allant de la sélection des utilisateurs similaires aux générations de ressources pédagogiques. Le premier module prépare les données nécessaires au bon déroulement de la tâche du deuxième module qui s'appuie sur le modèle $CF-GT$. Ce module passe par les étapes ci-dessous :

— Récupération des données de sorties du module "$InfoCollect$".

— Calcule de la distance entre l'utilisateur courant u_a et les autres utilisateurs suivant SV.

— Application du processus FC sur le groupe G_i obtenu :

- C alcule de similarité entre pair d'utilisateur (u_a, u_i) tel que $u_i \in G_i$
- P rédiction des évaluations manquantes de l'utilisateur courant.
- G énération des top_N recommandation.

4.3 Adaptation de l'approche $CF-GT$ au domaine éducatif

Afin de répondre à notre objectif de base, nous avons construit un modèle de recommandations de ressources pédagogiques en adaptant le modèle générique proposé au domaine éducatif. Un principal changement a été fait qui consiste à adapter la fonction de similarité utilisée.

Pour le modèle $CF-GT$ nous nous sommes basés uniquement sur les votes attribués par les utilisateurs aux items. Pour sélectionner les utilisateurs similaires, le modèle $Edu-CF-GT$ se base sur plusieurs critères, à savoir : le profil de l'apprenant, informations sur la consultation d'une ressource pédagogique et l'évaluation des ressources pédagogiques.

Dans ce qui suit, nous allons détailler la fonction de similarité utilisée ainsi que le jeu de données utilisé.

4.3.1 Base formelle du modèle $Edu-CF-GT$

Le but de cette section est la définition des concepts de base permettant le calcul des évaluations manquantes pour la recommandation de ressources pédagogiques. Ces concepts, présentés dans le tableau 4.1, sont liés au profil de l'utilisateur, aux informations de consultation et d'évaluation (visite et évaluation des ressources pédagogiques).

4.3.2 Adaptation de la fonction de similarité

Dans la littérature, la plupart des modèles de recommandation, qui se basent sur le filtrage collaboratif, utilisent le coefficient de corrélation de Pearson 2.1 pour calculer la similarité entre les utilisateurs. Ce coefficient mesure la corrélation linéaire entre deux variables. Dans les systèmes de recommandation, ce coefficient sert à mesurer la dépendance entre deux vecteurs d'évaluations de ressources appartenant à deux utilisateurs. Ces travaux s'intéressent principalement à l'évaluation des ressources pour calculer la similarité entre les utilisateurs. C'était le cas pour

TABLE 4.1 – Base formelle du modèle

Élément	Description
U	L'ensemble de tous les utilisateurs de l'environnement d'apprentissage informatisé.
$RC[u]$	L'ensemble des ressources consultées par u.
$RE[u]$	L'ensemble des ressources évaluées par u.
D	Ensemble des domaines pédagogiques présents dans le système.
$Vis(u,r)$	Utilisée pour connaitre les ressources consultées par l'utilisateur u. Cette fonction est égale à 1 si l'utilisateur u_a consulté la ressource r et 0 dans le cas contraire.
$Eval(u,r)$	Représente l'évaluation attribuée par l'utilisateur u à la ressource r.
$SEval[u,v]$	$Seval[u,r] = RE[u] \cap RE[v]$: représente l'ensemble des ressources co-évaluées par les utilisateurs u et v.
P	Le profil de l'utilisateur est composé de deux parties : une partie statique et une autre dynamique. La partie statique contient des données qui ne changent pas, telles que le nom, date de naissance, etc. La partie dynamique quant à elle contient des informations qui peuvent évoluer dans le temps telles que le niveau de connaissance de l'utilisateur, ses préférences, etc. Toutes ces informations statiques et dynamiques, sont représentées par l'ensemble P.
$P[u]$	$P[u] = \{(p,val) \mid p \in P :$ val est la valeur de la caractéristique c de l'utilisateur u. Il représente l'ensemble formé de couples caractéristique/valeur qui définissent le profil de l'utilisateur u.

notre modèle générique, nous nous sommes basés uniquement sur les évaluations des items pour calculer la similarité entre les utilisateurs.

Le modèle $Edu-CF-GT$, se base sur trois critères pour mesurer la similarité entre les utilisateurs : profile de l'utilisateur, information sur la consultation et évaluation des ressources pédagogiques. La similarité entre les utilisateurs comprend :

- $SimProf(u,v)$ la similarité entre le profil de l'apprenant u et v.
- $SimRessCon(u,v)$ la similarité entre l'apprenant u et v en termes de ressources consultées.
- $SimEval(u,v)$ la similarité entre l'apprenant u et v en termes de ressources pédagogiques évaluées.

À partir des trois similarités, nous calculons la similarité entre les apprenants u et v, notée $Sim(u,v)$. $Sim(u,v)$ est donnée par :

$$Sim(u,v) = (SimProf(u,v) + SimRessCon(u,v) + SimEval(u,v))/3 \tag{4.1}$$

Pour chaque similarité calculée, nous nous sommes basés sur :

- Jaccard 2.4 pour mesurer $SimProf(u,v)$, :

$$SimProf(u,v) = \frac{Card(P[u] \cap P[v])}{Card(P[u] \cup P[v])} \tag{4.2}$$

- Pour mesurer la similarité en termes de ressources consultées par deux utilisateurs u et v, nous nous sommes basés sur la similarité de Jaccard 2.4. L'équation 4.3 formalise $SimRessCon(u,v)$. Cette mesure est relative au nombre de ressources co-consultées par les deux utilisateurs et au nombre total de ressources visitées par les deux utilisateurs.

$$SimRessCon(u,v) = \frac{Card(RC[u] \cap RC[v])}{Card(RC[u] \cup RC[v])} \tag{4.3}$$

  Lorsque les deux utilisateurs u et v n'ont pas encore consulté des ressources, cette formule n'est pas appliquée, car l'union est nulle. Dans ce cas, $SimRessCon(u,v)$ est égal à zéro.

- Pour le calcul de la similarité en termes d'évaluation $SimEval(u,v)$, définie par l'équation 4.3.2, nous nous sommes basés sur le coefficient de corrélation de Pearson
  2.1.

$$SimEval(u,v) = \frac{\sum_{r \in SEval[u,v]} (Eval(u,r) - \overline{Eval}(u,.))(Eval(v,r) - \overline{Eval}(v,.))}{\sqrt{\sum_{r \in SEval[u,v]} (Eval(u,r) - \overline{Eval}(u,.))^2 \sum_{r \in SEval[u,v]} (Eval(v,r) - \overline{Eval}(v,.))^2}}$$

(4.4)

Pour l'exécution du modèle $Edu-CF-GT$, la fonction f de similarité du modèle $CF-GT$ est comme suit : $f(u_i,u_j) = SimEval(u_i,u_j)$.

4.4 Validation du modèle $Edu-CF-GT$

Pour valider le modèle $Edu-CF-GT$, nous avons réalisé une analyse hors ligne à l'aide d'un jeu de données *EduTest* que nous avons créé. Cette simulation nous a aidé à tester notre proposition pour évaluer son efficacité.

4.4.1 Données de simulation

En raison du manque de bases de données publiques dédiées au domaine de l'éducation et de l'absence des systèmes d'apprentissages informatisés dans nos universités, nous nous sommes retrouvés face à une situation d'absence de données de test. Pour pallier ce problème, nous avons généré une base de données synthétiques nommée «*EduTest*».

Cet ensemble fournit des informations principalement sur 1) les caractéristiques de l'utilisateur telles que l'âge, préférences, etc. 2) l'information sur la consultation d'une ressource ou pas et 3) les valeurs d'évaluation que les utilisateurs attribuent aux ressources pédagogiques.

Le profil de l'apprenant comporte les informations suivantes :

- Nom
- Niveau (licence ou master)
- Spécialité (ISIL, SIQ, IL, SIR, SSI, TAL)
- Langage préféré (Java, C++, C,Python)
- Expérience en programmation (1-5)
- Style d'apprentissage (livre, vidéo)

Le tableau 4.2 présente un exemple du profil de 13 apprenants de la base *EduTest*.

TABLE 4.2 – Exemple du profil de 13 apprenants de la base *EduTest*

Nom	Niveau	Spécialité	Langage préféré	Expérience	Style d'apprentissage
A1	Master	TAL	Python	3	Vidéo
A2	Licence	ISIL	Java	2	Texte
A3	Licence	SIQ	C++	2	Texte
A4	Licence	SIQ	Python	2	Vidéo
A5	Master	SIR	Python	4	Texte
A6	Licence	ISIL	C++	3	Texte
A7	Master	IL	Java	3	Texte
A8	Master	SIR	Java	3	Texte
A9	Master	SSI	C++	4	Vidéo
A10	Licence	ISIL	Python	2	Vidéo
A11	Licence	SIQ	C++	3	Texte
A12	Licence	SIQ	Python	2	Texte
A13	Master	SIR	Java	4	Vidéo

L'évaluation des ressources pédagogiques varie sur une échelle de 1 à 5.

Ressource consultée : la valeur 1 signifie que l'apprenant a déjà consulté la ressource, la valeur 0 signifie que la ressource n'a pas été consultée par l'apprenant.

Le tableau 4.3 présente des informations sur la consultation et l'évaluation de 10 ressources par 13 apprenants de la base *EduTest*.

Pour chaque ressource et chaque apprenant, on distingue un couple sous la forme : (consultée ou non consultée $\{1/0\}$; Évaluation de la ressource $\{0..5\}$). 0 indique que la ressource n'a pas été évaluée, et les valeurs de 1 à 5 indiquent de « ressource non appréciée » à « appréciée beaucoup ».

TABLE 4.3 – Informations sur la consultation et évaluation de 8 ressources par 13 apprenants de la base *EduTest*

Nom	Ress1	Ress2	Ress3	Ress4	Ress5	Ress6	Ress7	Ress8
A1	(0 ;NULL)	(0 ;NULL)	(0 ;NULL)	(0 ;NULL)	(0 ;NULL)	(1 ;0)	(0 ;NULL)	(1 ;1)
A2	(1 ;5)	(1 ;0)	(0 ;NULL)	(1 ;0)	(1 ;0)	(0 ;NULL)	(0 ;NULL)	(1 ;4)
A3	(0 ;NULL)	(1 ;0)	(1 ;3)	(0 ;NULL)	(1 ;1)	(0 ;NULL)	(0 ;NULL)	(1 ;0)
A4	(1 ;4)	(1 ;5)	(0 ;NULL)	(0 ;NULL)	(0 ;NULL)	(1 ;1)	(1 ;3)	(0 ;NULL)
A5	(1 ;1)	(0 ;NULL)	(0 ;NULL)	(1 ;2)	(0 ;NULL)	(1 ;5)	(0 ;NULL)	(1 ;4)
A6	(0 ;NULL)	(1 ;0)	(1 ;2)	(0 ;NULL)	(1 ;1)	(0 ;NULL)	(1 ;2)	(1 ;2)
A7	(0 ;NULL)	(1 ;0)	(1 ;3)	(1 ;5)	(1 ;4)	(0 ;NULL)	(1 ;0)	(1 ;0)
A8	(0 ;NULL)	(0 ;NULL)	(1 ;1)	(1 ;1)	(1 ;4)	(1 ;0)	(1 ;0)	(1 ;2)
A9	(1 ;2)	(0 ;NULL)	(0 ;NULL)	(0 ;NULL)	(0 ;NULL)	(0 ;NULL)	(1 ;1)	(1 ;2)
A10	(0 ;NULL)	(0 ;NULL)	(1 ;2)	(1 ;2)	(1 ;1)	(1 ;2)	(0 ;NULL)	(1 ;3)
A11	(1 ;1)	(0 ;NULL)	(1 ;2)	(0 ;NULL)	(0 ;NULL)	(0 ;NULL)	(1 ;5)	(0 ;NULL)
A12	(0 ;NULL)	(0 ;NULL)	(0 ;NULL)	(0 ;NULL)	(1 ;3)	(0 ;NULL)	(1 ;4)	(0 ;NULL)
A13	(0 ;NULL)	(0 ;NULL)	(0 ;NULL)	(0 ;NULL)	(0 ;NULL)	(0 ;NULL)	(1 ;1)	(0 ;NULL)

EduTest se compose de 800 apprenants et de 500 ressources pédagogiques. Les ressources pédagogiques ont été consultées 120000 fois. La base de données comporte 2800 évaluations.

4.4.2 Expérimentations et résultats

Afin de valider l'approche de recommandation de ressources pédagogique, nous avons adopté le même protocole expérimental suivi pour la validation de l'approche générique $CF-GT$. Les algorithmes FC et $k-means-CF$ ont été maintenus pour la comparaison.

Expérience 1 :

Les figures 4.2, 4.3 et 4.4 montrent la précision de la prédiction pour différentes tailles de voisinage k sur le jeu de données $EduTest$.

Expérience 2 :

Les tableaux 4.4, 4.5 et 4.6 montrent les valeurs de précision et de rappel pour les différents $top-N$ avec la base $EduTest$.

TABLE 4.4 – Précision et rappel pour $\sigma = 0.79$ *EduTest*

	Edu-CF-GT		FC		K_means	
Top	**Précision**	**Rappel**	**Précision**	**Rappel**	**Précision**	**Rappel**
5	26%	26%	21%	21%	24%	23%
10	39%	40%	35%	35%	37%	36%
15	47%	47%	42%	43%	43%	44%
20	52%	53%	48%	50%	49%	52%
25	56%	56%	52%	54%	53%	54%
30	59%	60%	55%	57%	56%	58%
35	59%	60%	54%	55%	55%	56%
40	58%	59%	52%	52%	55%	53%
45	55%	56%	50%	50%	54%	52%
50	55%	55%	49%	49%	52%	51%

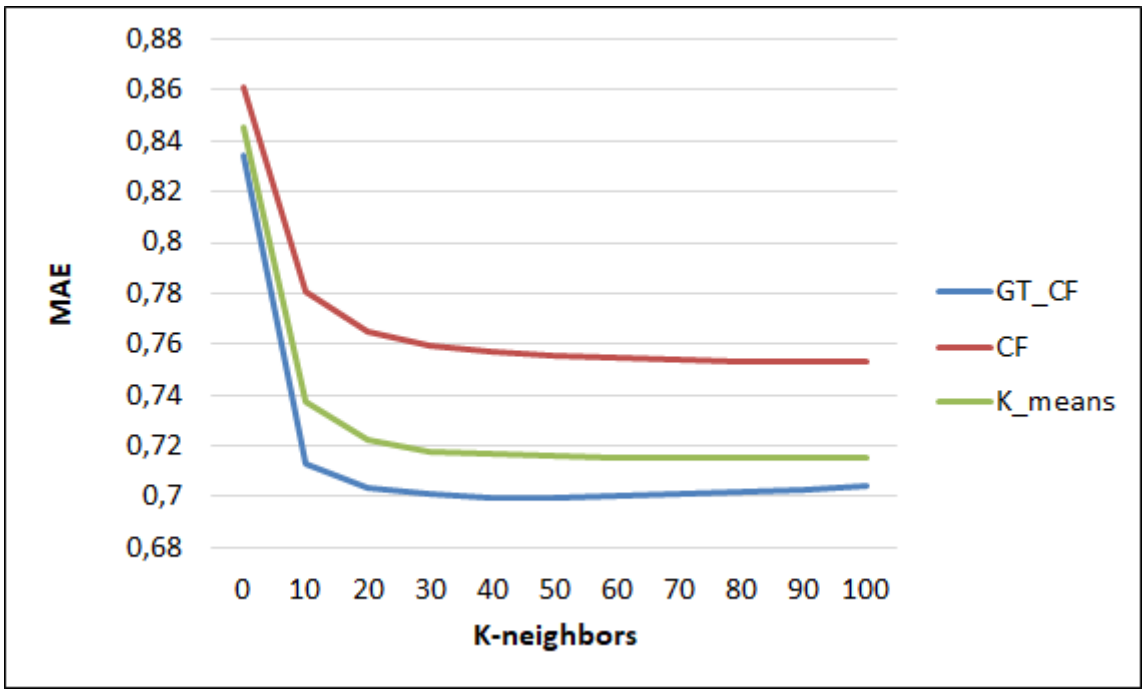

FIGURE 4.2 – *MAE* pour σ =0.79 / EduTest

TABLE 4.5 – Précision et rappel pour $\sigma = 0.86$ *EduTest*

	Edu-CF-GT		FC		K_means	
Top	**Précision**	**Rappel**	**Précision**	**Rappel**	**Précision**	**Rappel**
5	27%	28%	21%	21%	23%	22%
10	39%	39%	35%	35%	36%	36%
15	47%	47%	42%	43%	44%	44%
20	53%	54%	48%	50%	49%	51%
25	56%	57%	52%	54%	53%	55%
30	59%	61%	55%	57%	57%	59%
35	58%	60%	54%	55%	56%	56%
40	57%	59%	52%	52%	56%	53%
45	55%	57%	50%	50%	53%	52%
50	53%	55%	49%	49%	50%	50%

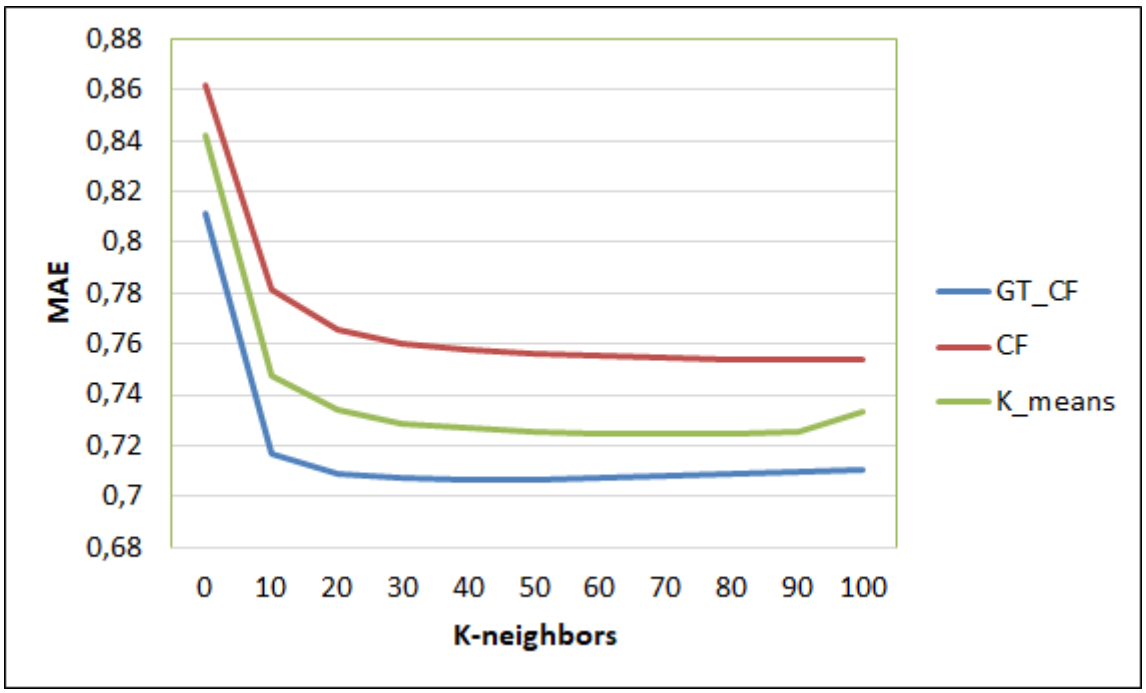

FIGURE 4.3 – *MAE* pour $\sigma = 0.86$ / EduTest

TABLE 4.6 – Précision and rappel pour $\sigma = 0.87$ *EduTest*

	Edu-CF-GT		FC		K_means	
Top	**Précision**	**Rappel**	**Précision**	**Rappel**	**Précision**	**Rappel**
5	28%	28%	21%	21%	25%	23%
10	40%	41%	35%	35%	36%	36%
15	48%	49%	42%	43%	43%	45%
20	55%	54%	48%	50%	51%	51%
25	58%	58%	52%	54%	55%	55%
30	60%	63%	55%	57%	53%	59%
35	60%	64%	54%	55%	53%	57%
40	58%	60%	52%	52%	52%	53%
45	56%	58%	50%	50%	51%	52%
50	55%	57%	49%	49%	50%	52%

4.5 Discussion

Le modèle $Edu-CF-GT$ est une adaptation du modèle $CF-GT$ au domaine éducatif. L'évaluation par rapport à la précision des prédictions et à l'exactitude des recommandations fournies en utilisant le jeu de données *EduTest* affirme que $Edu-CF-GT$ surpasse les deux modèles de test FC et $k-means-CF$. Ces résultats s'expliquent par le fait $Edu-CF-GT$ produit une bonne présélection des apprenants similaires. De la même manière comme interprétée pour le modèle générique $CF-GT$.

Si on compare les résultats de l'évaluation du modèle $CF-GT$ et $Edu-CF-GT$, nous pouvons clairement remarquer, que l'approche $Edu-CF-GT$ donne une valeur de MAE inférieur

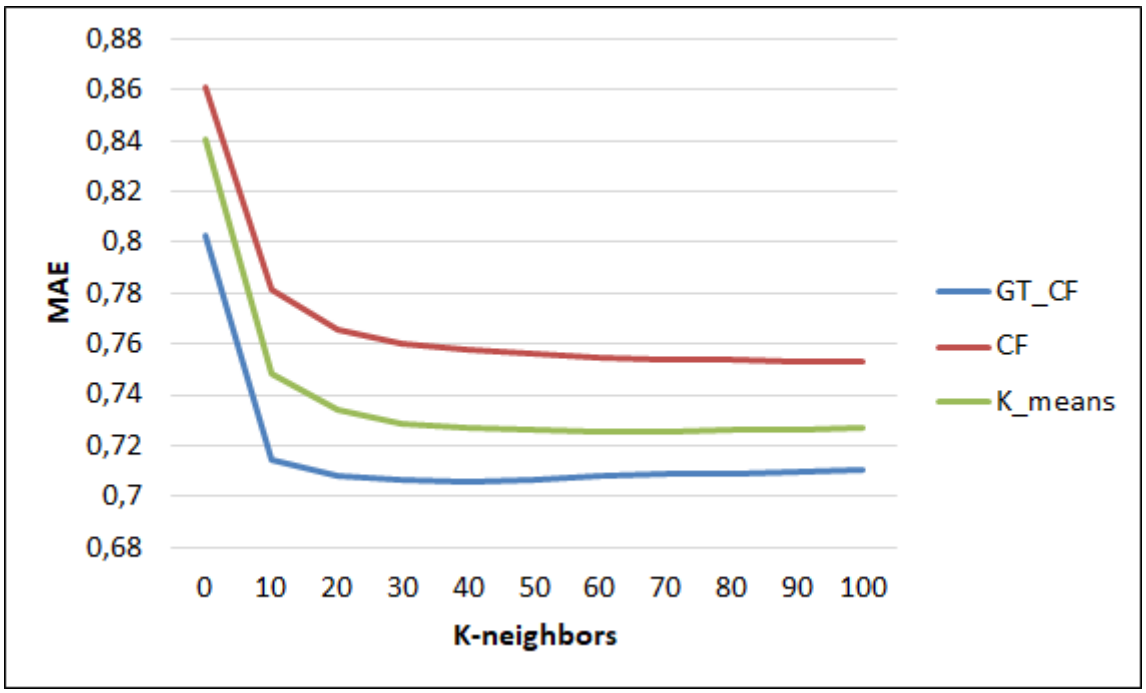

FIGURE 4.4 – *MAE* pour $\sigma = 0.87$ / EduTest

à celle produite par $CG-GT$. Nous remarquons aussi que la *prcision* et le *rappel* du modèle $Edu-CF-GT$ surpasse les valeurs de $CF-GT$. Cela est expliqué par le fait que l'intégration de plusieurs caractéristiques de l'apprenant améliore la sélection d'utilisateurs similaires et par conséquent la prédiction d'évaluations devient de plus en plus précises et les recommandations plus pertinentes.

4.6 Conclusion

Dans ce chapitre, nous avons présenté un modèle de recommandation de ressources pédagogiques $Edu-CF-GT$. Ce modèle est une adaptation du modèle $CF-GT$ au modèle éducatif. Pour rappel, le modèle $CF-GT$ est constitué de deux modules : « *SimilarUser*» et « *CFProcess* ». Le premier module sert d'une présélection des utilisateurs similaires. Le deuxième applique le processus du filtrage collaboratif pour chaque groupe obtenu. La particularité de l'approche $CF-GT$ par rapport aux approches qui adoptent le clustering comme étape de présélection est que le module « *SimilarUser* » se base sur le concept de solution *SV* pour former les groupes d'utilisateurs similaire. Ce concept prend en compte non seulement la relation utilisateur centre de cluster, mais aussi entre pair utilisateur-utilisateur.

L'adaptation du modèle $CF-GT$ au domaine éducatif réside dans la fonction de similarité utilisée pour choisir les utilisateurs similaires. L'intégration de plusieurs caractéristiques des apprenants a induit à l'adaptation de la fonction de similarité. Les données d'entrée ont également changé, nous avons construit notre propre jeu de données *EduTest* qui comporte plusieurs caractéristiques de l'apprenant.

Le modèle $Edu-CF-GT$ a été testé et validé. Les résultats révèlent l'efficacité de l'approche et que le modèle $CF-GT$ est vraiment générique.

Avec la croissance exponentielle des données et informations générées par les systèmes d'enseignement à distance et les plateformes de e-learning, les techniques classiques de *ML* s'avèrent limitées pour modéliser les interactions utilisateur/items et générer de meilleures recommandations. C'est pour cette raison que le deep learning a imposé son intégration au sein des systèmes de recommandation.

Le chapitre suivant présente une nouvelle approche $CNN-CF-GT$ qui consiste à l'intégration du *CNN* pour améliorer le modèle $CF-GT$.

Chapitre 5

Système de recommandation fondé sur la théorie des jeux et le deep learning

5.1 Introduction

Au cours des dernières années, l'apprentissage profond a révolutionné plusieurs domaines, notamment l'analyse d'images, la reconnaissance vocale et le traitement du langage. L'apprentissage profond est également devenu omniprésent et a démontré son efficacité dans le domaine des systèmes de recommandation et de la recherche d'informations [16]. Contrairement aux systèmes de recommandation conventionnels, l'apprentissage profond a la capacité unique de capturer avec succès les interactions non triviales et non linéaires entre l'utilisateur et l'objet, permettant ainsi la codification d'abstractions plus complexes.

Le deep learning a la capacité de coder automatiquement la représentation apprise à partir des données. Ces représentations apprises à partir des données sont généralement plus performantes que les représentations manuelles. En outre, les réseaux neuronaux peuvent approximer n'importe quelle fonction avec une précision arbitraire, moyennant une capacité suffisante [150]. Un réseau neuronal se compose de plusieurs couches, chacune effectuant une transformation non linéaire simple. Chaque couche apprend plusieurs niveaux d'abstraction, en commençant par une structure grossière dans les couches les plus basses et en poursuivant le raffinement dans les couches suivantes. Collectivement, le réseau entier apprend une hiérarchie complète de concepts abstraits de complexité croissante. La terminologie «deep learning » ou "apprentissage profond" fait référence à la profondeur ou au nombre de couches non linéaires empilées ensemble qui sont apprises de bout en bout.

Les SRs sont constitués de données de haute dimension en raison du volume croissant d'utilisateurs et d'articles. Les articles sont souvent accompagnés d'une forme de métadonnées riches telles que du texte non structuré provenant d'un résumé ou d'une description de produit et des données catégorielles telles que le domaine d'une ressource pédagogique. Par conséquent, le deep learning peut être utilisé pour extraire des représentations de caractéristiques riches du contenu des articles de manière automatisée. En conjonction avec les interactions non linéaires

entre les utilisateurs et les articles, une structure plus complexe des préférences des utilisateurs peut être extraite des données à haute dimension. En outre, de nombreux algorithmes de recommandation traditionnels peuvent être exprimés sous la forme d'une architecture neuronale peu profonde constituée d'une seule couche linéaire [151, 152]. Parmi les techniques de filtrage collaboratif, la méthode de factorisation matricielle [153] est la plus populaire. Comme nous l'avons vu dans la section 2.4.2 la méthode fonctionne en représentant d'abord chaque utilisateur et chaque article avec un vecteur de caractéristiques latentes, puis en projetant les caractéristiques latentes de l'utilisateur et de l'article dans des vecteurs latents. Par conséquent, l'interaction entre l'utilisateur et l'article peut être modélisée par les produits internes des vecteurs latents. Divers efforts ont été fournis par la communauté de recherche pour améliorer la méthode de filtrage collaboratif utilisant la factorisation matricielle, comme la fusion du modèle de voisinage avec les méthodes de factorisation matricielle [154], l'utilisation de machines de factorisation pour une modélisation générique des caractéristiques [155] et la combinaison de la factorisation matricielle avec des modèles thématiques du contenu des articles [156]. Bien que la factorisation matricielle soit la technique la plus populaire pour le filtrage collaboratif, ses performances dépendent fortement du choix de la fonction interactive - produit interne, où les performances peuvent être améliorées en introduisant des termes de biais d'utilisateur et d'élément dans le produit interne [157]. Cependant, la simple multiplication linéaire des caractéristiques latentes peut ne pas être suffisante pour décrire l'interaction complexe de l'utilisateur et de l'article. Un autre problème commun associé à un système de recommandation qui conduit à une performance moins efficace des méthodes existantes dans la littérature est la grande rareté de l'interaction complexe de l'utilisateur et du produit, de nombreux éléments nuls dans la matrice d'interaction utilisateur-item.

Récemment, les réseaux neuronaux profonds ont été utilisés pour remédier à certaines des lacunes évoquées afin d'améliorer le filtrage collaboratif. Par exemple, dans [157] les auteurs ont intégré des perceptrons multicouches (MLP) et la factorisation matricielle dans un cadre de filtrage collaboratif basé sur un réseau de neurones (NCF) pour modéliser les vecteurs latents des utilisateurs et des articles qui pourraient apprendre une fonction arbitraire à partir des données pour surmonter l'interaction complexe de l'utilisateur et de l'article ; le cadre proposé a montré des améliorations significatives par rapport à d'autres méthodes.

Le réseau neuronal convolutif (CNN) est un réseau neuronal à anticipation développé à l'origine pour la vision par ordinateur. Bien qu'il ne soit pas couramment appliqué au domaine des systèmes de recommandation, des études ont été menées sur l'utilisation du CNN pour extraire des informations auxiliaires afin de créer des recommandations. Le CNN a été utilisé pour la recommandation de musique en analysant l'acoustique des chansons et en faisant des prédictions basées sur le modèle latent[158], et pour les recommandations de vidéos en modélisant les facteurs latents sur les critiques de vidéos et en les intégrant avec la factorisation matricielle probabiliste [159].

Pour tenter d'améliorer notre modèle $Edu-CF-GT$, le *CNN* a été utilisé pour modéliser directement l'interaction utilisateur-item à partir des données dans le deuxième module de $Edu-CF-GT$. Le *CNN* a été exploité, car il comporte une couche de convolution qui extrait les caractéristiques locales en convoluant les signaux d'entrée des neurones adjacents, ce qui lui permet d'être plus flexible lors de l'apprentissage des caractéristiques.

Nous commençons le chapitre par donner un aperçu sur le deep learning et en particulier le deep learning exploité pour la recommandation. Ensuite, nous détaillons l'amélioration du deuxième module de notre modèle "C$F-GT$"

5.2 Préliminaires

Dans cette section, nous donnons un aperçu du deep learning et ses différentes architectures. Nous nous focalisons en particulier sur l'exploitation du deep learning dans l'intérêt des modèles de recommandations.

5.2.1 Deep learning : définition

Le Deep Learning (DL) est un sous-domaine de l'apprentissage automatique (machine learning) qui traite des réseaux de neurones artificiels (RNA), qui sont des algorithmes inspirés de la structure et du fonctionnement du cerveau. Le deep learning a été appliqué dans une grande variété de domaines, notamment : la vision par ordinateur [160] , le traitement automatique des langues [161] , le traitement d'images [162] et les SRs [163], donnant des résultats très prometteurs. Le DL aide les machines à apprendre des modèles mathématiques extrêmement compliqués pour représentation des données, qui peuvent ensuite être utilisés pour effectuer une analyse précise des données. Ces modèles mesurent de manière hiérarchique des fonctions de données d'entrée non linéaires/ linéaires qui sont pondérées par des paramètres du modèle. Le fait de traiter ces fonctions comme des "couches" de traitement de données encourage souvent le terme "deep learning" par l'utilisation hiérarchique d'un grand nombre de ces couches. L'objectif commun des techniques du DL est d'utiliser un ensemble de données d'apprentissage pour apprendre de manière itérative les paramètres du modèle de calcul, de sorte que le modèle s'améliore lentement dans l'exécution d'une fonction souhaitée, telle que la classification. En général, le modèle de calcul lui-même prend la forme d'un réseau neuronal artificiel (RNA) [164] composé de plusieurs couches de neurones/perceptrons [165], tandis que ses paramètres (c'est-à-dire les poids du réseau) déterminent la force des interactions entre les neurones de différents niveaux. La figure 5.1 illustre cette définition.

Lorsqu'ils sont formés pour une tâche spécifique, les modèles DL peuvent également effectuer la même tâche de manière efficace en utilisant une série de données inédites (c'est-à-dire des données de test). Les modèles DL ont connu un succès considérable dans les tâches d'apprentissage supervisé et non supervisé [166]. En général, les architectures DL peuvent être divisées

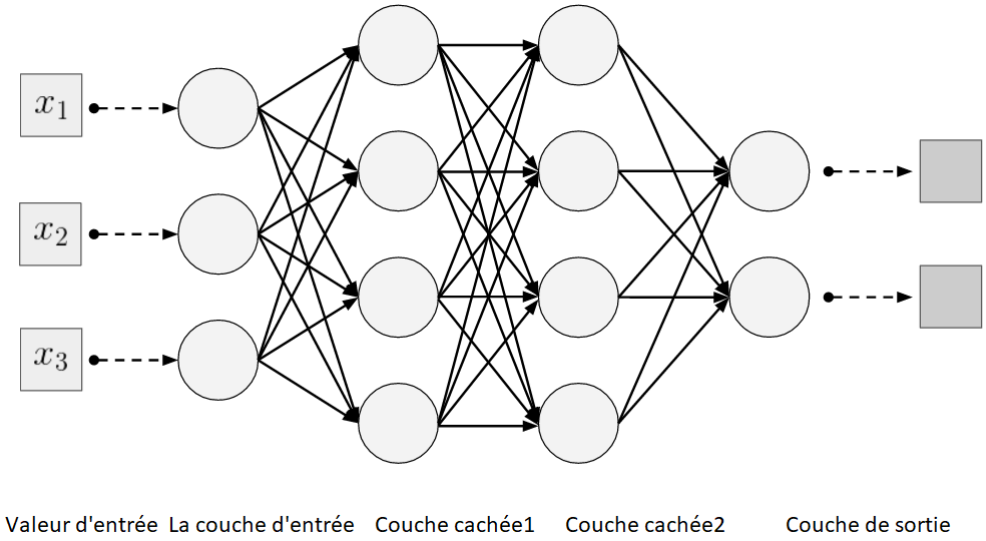

FIGURE 5.1 – Architecture générale du deep learning

en trois grandes catégories, à savoir les architectures profondes génératives, discriminatives et hybrides [166, 167].

5.2.2 deep learning : architecture

Le deep learning présente plusieurs formes d'architecture. L'architecture du DL est généralement divisée en deux grandes classes : générative et discriminative [168]. L'architecture de réseau générative génère des candidats tandis que le réseau discriminatif les évalue [169]. Dans ce qui suit, nous décrivons différents paradigmes architecturaux de l'apprentissage profond.

Multilayer Perception (*MLP*)

Le MLP est un réseau neuronal de type feed-forward. Le *MLP* est considéré comme l'architecture *DL* la plus simple [166] avec une ou plusieurs couches cachées entre les couches d'entrée et de sortie. Chaque nœud utilise une fonction d'activation non linéaire. Tous les nœuds sont entièrement connectés ; dans une couche d'activation, tous les nœuds ont un poids par rapport aux nœuds de la couche précédente.

Convolutional Neural Network (*CNN*)

Le *CNN* utilise des perceptrons pour traiter des données de grande dimension [170]. Il contient plusieurs couches. Dans ces réseaux, on distingue trois types de couches : les couches convolutionnelles (pour l'extraction de caractéristiques), les couches de mise en commun (pour réduire la dimensionnalité des données) et les couches entièrement connectées. La figure 5.2 illustre l'architecture générale du CNN.

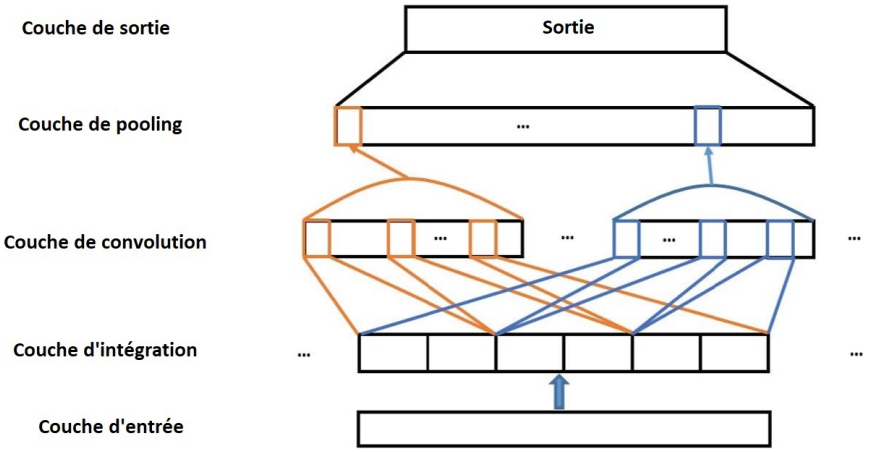

FIGURE 5.2 – Architecture générale du CNN

Autoencoder (*AE*)

Les autoencodeurs sont des réseaux neuronaux à anticipation (feedforward) qui construisent une représentation compressée en formant un goulot d'étranglement dans l'architecture avant de tenter de récupérer les entrées initiales du modèle. L'*AE* vise à construire une donnée dans la couche de sortie similaire à la donnée d'entrée. En général, la couche la plus centrale est utilisée pour représenter les caractéristiques de la donnée d'entrée. Plusieurs variantes d'autoencodeurs existent dans la littérature : denoising autoencoder, marginalized denoising autoencoder, stacked denoising autoencoders (SADE), sparse autoencoder, contractive autoencoder and variational autoencoder[16, 150].

Recurrent Neural Network (*RNN*)

Le *RNN* est spécifiquement utilisé pour la modélisation de données séquentielles en intégrant simplement des couches temporelles pour capturer des informations séquentielles [171]. Les *RNNs* possèdent des boucles et des mémoires pour se souvenir des traitements précédents, contrairement aux réseaux à anticipation (réseau feedforward)[172]. Dans les *RNNs*, les unités cachées de la cellule récurrente sont utilisées pour apprendre le changement complexe. L'activation des états cachés suivants de l'état caché précédent est utilisée pour traiter l'état caché actuel.

Plusieurs architectures du DL existent dans la littérature, nous avons cité les plus importantes. Pour des modèles plus avancés, nous suggérons [16, 150] . Le tableau 5.1quatres architectures citées.

5.2.3 Deep learning pour les systèmes de recommandations

Récemment, l'application du *DL* aux systèmes de recommandation a suscité un grand intérêt. En particulier, le *DL* permet d'apprendre des représentations non linéaires robustes à partir de données.

TABLE 5.1 – Résumé des différentes architectures du deep learning

Modèle	Classe	Apprentissage	Points forts	Points faibles
MLP	Discriminative	Supervisé	Transformation non linéaire	Complexité élevée et convergence lente
CNN	Discriminative	Supervisé	Puissant pour l'extraction de caractéristiques avec des informations contextuelles	Nécessitent un paramétrage important
AE	Générative	Non supervisé	Il peut utiliser une stratégie d'apprentissage non supervisée pour apprendre des représentations de caractéristiques plus complexes.	La scalabilité aux données hautement dimensionnelles est limitée.
RNN	Discriminative	Supervisé	Assez puissant et flexible	Nécéssite beaucoup de données

Les autoencodeurs ont été un choix populaire d'architecture d'apprentissage profond pour les systèmes de recommandation [156, 173, 174, 175, 176]. Essentiellement, l'autoencodeur agit comme une décomposition non linéaire de la matrice d'évaluation en remplaçant le produit interne linéaire traditionnel dans la factorisation de la matrice. Par exemple, AutoRec [174] décompose la matrice d'évaluation avec un autoencodeur suivi d'une reconstruction pour prédire directement les évaluations et obtenir des résultats compétitifs sur de nombreux ensembles de données de référence. Un autre exemple est celui des denoising AEs collaboratifs (CDAE) [173] qui traitent la recommandation $top-n$ en intégrant un biais spécifique à l'utilisateur dans un auto-encodeur. La démonstration du CDAE peut être considérée comme une généralisation de nombreuses méthodes de filtrage collaboratif existantes examinant les fonctions de perte par points et par paires. Dans [175] les auteurs adoptent un "marginalized denoising AE" pour diminuer les coûts de calcul associés au DL. Ils utilisent deux AEs, l'un pour le contenu de l'article et l'autre pour le contenu de l'utilisateur, reliés à des facteurs latents de l'utilisateur et de l'article.

Nous allons maintenant nous intéresser aux modèles utilisant des MLPs. La factorisation matricielle par réseau neuronal (NNMF) [177] adopte une approche qui remplace le produit interne traditionnel de la factorisation matricielle par une fonction apprise à partir d'un réseau neuronal à action directe. De même, le filtrage collaboratif neuronal (NCF) [178] associe la sortie d'un MLP concaténé avec les facteurs latents de la factorisation matricielle en appliquant une

transformation non linéaire pour produire une interaction locale avant d'effectuer la recommandation finale. Le *MLP* et la factorisation matricielle conservent chacun des espaces d'intégration distincts pour les facteurs latents de l'utilisateur et de l'article, ce qui permet d'obtenir la complexité requise pour la tâche à accomplir. Les auteurs [179] couplent étroitement un réseau neuronal profond à l'apprentissage d'une correspondance entre le contenu et les facteurs latents appris existants et les facteurs latents de l'utilisateur et de l'article, afin de résoudre le problème du démarrage à froid de l'utilisateur ou de l'article. Les auteurs de [180] s'attaquent à la recommandation d'applications mobiles dans la boutique Google Play en formant conjointement un modèle linéaire généralisé et un réseau neuronal profond sur les données démographiques de l'utilisateur et les installations implicites d'applications.

La nature séquentielle des RNN offre des propriétés souhaitables pour les systèmes de recommandation basés sur le temps [181] et sur la session [182]. Par exemple, dans [181] les auteurs représentent les facteurs latents de l'utilisateur et de l'élément avec deux *RNN* pour capturer l'aspect temporel de la recommandation de films. Collectivement, les états cachés des RNN représentent les préférences et les évaluations de l'utilisateur à chaque intervalle de temps, tandis que des facteurs stationnaires supplémentaires sont maintenus pour gérer les préférences sur le long terme d'un utilisateur. Alors que d'autres méthodes définissent de manière heuristique les changements temporels relatifs [183, 184], d'autres proposent des cellules récurrentes complexes et spécialisées basées sur la cellule de mémoire à long et court terme (LSTM) et la cellule d'unité récurrente à déclenchement (GRU) [184, 185, 186]. Jannach et Ludewig [187] interpolent le *KNN* avec le *RNN* basé sur la session proposé par Hidasi et al. [182], ce qui permet de réaliser des gains de performance supplémentaires. Jing et Smola [188] dotent un *RNN* d'une analyse de survie pour prédire le retour futur d'un utilisateur donné. Le *RNN* aborde l'aspect temporel en consultant les états cachés précédents avec le taux de survie pour aborder le temps de retour de l'utilisateur.

Dans [189], les auteurs ont abordé la recommandation musicale par une approche en deux étapes : une étape qui utilise la factorisation matricielle, puis la deuxième étape qui utilise le *CNN*. La factorisation matricielle est utilisée pour obtenir les facteurs latents des éléments, puis le *CNN* conventionnel est appliqué pour apprendre la représentation des caractéristiques des informations de contenu en traitant ces facteurs latents comme sortie.

Les modèles *CNN* ont été également utilisés pour capturer des représentations de caractéristiques d'articles localisées - texte [190, 191] et images [163]. Dans [192] les auteurs exploitent des bases de connaissances textuelles, structurelles et visuelles avec des encodeurs automatiques convolutionnels et de débruitage (denoising) pour améliorer le modèle de facteur latent.

5.3 Notre proposition

Afin d'optimiser notre modèle $Edu-CF-GT$, nous avons intégré le *CNN* au processus du filtrage collaboratif. Pour rappel, le modèle $Edu-CF-GT$ est constitué de deux principaux

modules. Le premier module sert de présélection d'apprenants similaires en utilisant le concept de solution « Shapley Value » de la théorie des jeux. Le deuxième module applique le processus du filtrage collaboratif pour chaque groupe d'utilisateurs similaires obtenu par le premier module.

Pour tenter d'améliorer notre modèle, le *CNN* a été utilisé pour modéliser directement l'interaction utilisateur-article à partir des données pour fournir des recommandations. Le *CNN* a été exploré, car il comporte une couche de convolution qui extrait les caractéristiques locales en convoluant les signaux d'entrée des neurones adjacents, ce qui lui permet d'être plus flexible lors de l'apprentissage des caractéristiques.

Les systèmes de recommandations se basent sur l'avis des apprenants concernant les ressources pédagogiques. Les utilisateurs expriment leurs avis par une note attribuée à la ressource, c'est ce qu'on appelle le feedback explicite qui permet de construire la matrice de notation utilisateur/item. Un autre type de feedback est distingué : le feedback implicite. Ce type de feedback est collecté en se basant sur le comportement de l'utilisateur. Pour optimiser le deuxième module de $Edu-CF-GT$, nous nous sommes basées sur le feedback implicite.

Dans ce qui suit, nous discutons la collecte du feedback implicite. Après, nous présentons la description des différents composants du modèle proposé et enfin, la fonction objective utilisée pour optimiser les paramètres du modèle.

5.3.1 Feedback implicite

Les systèmes de recommandation s'appuient sur différents types d'entrées pour générer une recommandation. L'entrée la plus utile est le feedback explicite fourni par les utilisateurs concernant leur intérêt pour les items, comme les étoiles ou les boutons pouce en haut/bas. Cependant, ce type d'information n'est pas toujours disponible, ce qui conduit à la nécessité d'utiliser les feedbacks implicites tels que l'historique des achats, l'historique de navigation, les schémas de recherche et les mouvements de la souris qui peuvent indirectement refléter la préférence de l'utilisateur. Le modèle que nous allons proposer se base sur les feedbacks implicites. Bien que les feedbacks implicites soient plus faciles à collecter, il est plus difficile de les utiliser, car ils ne reflètent pas directement la préférence de l'utilisateur et il n'y a généralement pratiquement pas de commentaires négatifs (il est difficile de déterminer si les utilisateurs n'aiment pas les articles).

Supposons que Uet I représentent le nombre d'utilisateurs et d'items respectivement, la matrice d'interaction utilisateur-item, $P = \left[p_{ui}\right]^{U*I}$, dont les éléments p_{ui} représentent l'interaction entre l'utilisateur u et l'article i où,

$$p_{ui} = \begin{cases} 1, & \text{si une intéraction est observée} \\ 0, & \text{si non} \end{cases} \tag{5.1}$$

Ici, une valeur de 1 pour p_{ui} indique qu'il y a une interaction entre l'utilisateur u et l'élément i ; cependant, cela ne signifie pas que u aime réellement i. De même, une valeur de 0 ne signifie

pas nécessairement que u n'aime pas i, il se peut que l'utilisateur ne connaisse pas l'élément, ce qui entraîne une entrée non observée dans la matrice d'interaction utilisateur-article. Cela reflète les défis que pose l'utilisation de données implicites. Les entrées observées (1) indiquent que les utilisateurs connaissent les articles, mais il y a de nombreuses entrées non observées (0), qui peuvent être simplement des données manquantes ou la liste d'articles est trop longue, les utilisateurs ne peuvent pas parcourir tous les articles.

L'objectif de la recommandation avec le feedback implicite est de générer une liste classée d'articles qui reflète la préférence des utilisateurs en estimant les scores des entrées non observées dans la matrice d'interaction utilisateur-item, P. En général, les approches basées sur un modèle supposent que l'interaction utilisateur-item peut être décrite comme $\widehat{p}_{ui} = f(u,i|\theta)$, où $\widehat{p}_{ui}$ représente le score estimé de l'interaction p_{ui}, f est le modèle/fonction utilisé pour générer le score estimé et θ désigne les paramètres du modèle de f.

Pour estimer les paramètres θ les approches existantes suivent généralement le paradigme du machine learning qui optimise une fonction objective. Deux types de fonctions objectives sont le plus souvent utilisés dans la littérature : la perte pointwise [193, 194] et la perte pairwise [195, 196]. La première tente généralement de minimiser la perte au carré entre $\widehat{p}_{ui}$ et p_{ui}, et traite toutes les entrées non observées comme un feedback négatif ou choisit sélectivement certaines entrées non observées comme négatives [193]. Pour la méthode de perte par paire [195], les entrées observées sont classées plus haut que les entrées non observées, ce qui maximise la marge entre l'entrée observée et l'entrée non observée au lieu de minimiser la perte entre $\widehat{p}_{ui}$ et p_{ui}.

Pour optimiser notre modèle, un réseau de neurones a été utilisé pour estimer $\widehat{p}_{ui}$ permettant la prise en charge de l'apprentissage ponctuel et par paires pointwise et pairwise, plus de détail sont disponibles dans la section suivante.

5.3.2 Modèle de *FC* basé sur *GT* et le *CNN* pour les ressources pédagogiques

Comme nous l'avons indiqué au début de ce chapitre, nous tentons d'optimiser notre modèle $Edu-CF-GT$ en intégrant le *CNN* au deuxième module pour implémenter le processus du filtrage collaboratif, ce qui donne naissance à un nouveau modèle nommé $CNN-CF-GT$.

Le modèle $CNN-CF-GT$ est constitué de deux modules. Le premier module c'est le module "$SimilarUser$" du modèle $Edu-CF-GT$, sa fonctionnalité est préservée. Le deuxième module nommé "$CNN-CF$" prend en entrée la sortie du module "$SimilarUser$". "$CNN-CF$" est l'implémentation du modèle FC en utilisant *CNN*.

La figure 5.3 illustre le cadre général du module $CNN-CF$.

Comme dit précédemment, pour ce module, nous utilisons le feedback implicite pour fournir les recommandations.

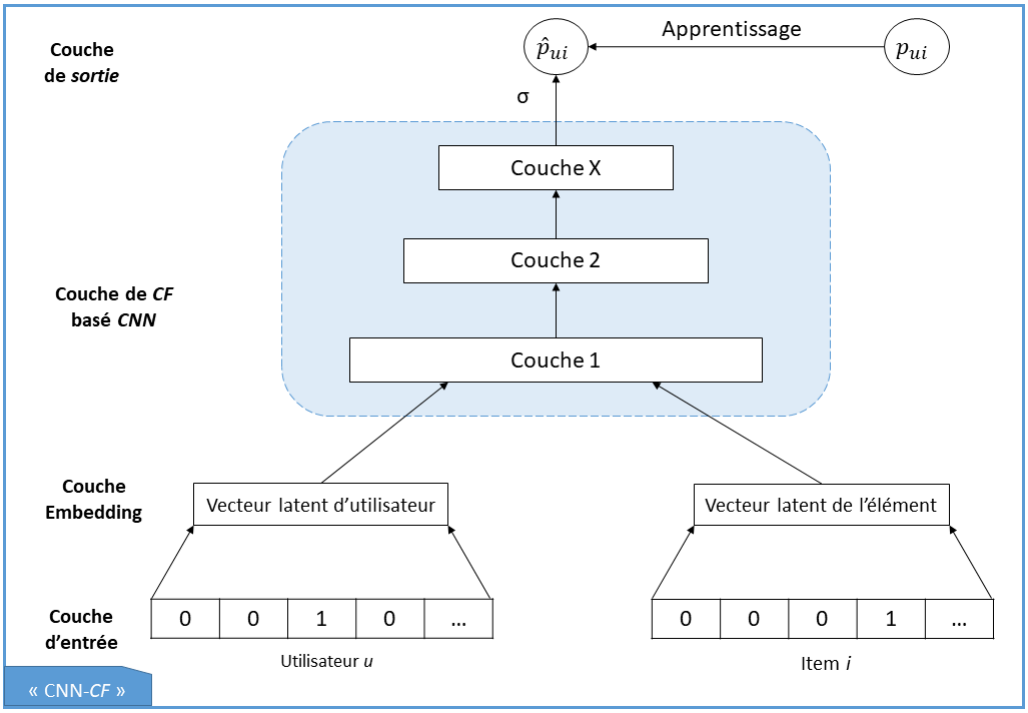

FIGURE 5.3 – Architecture générale du module "$CNN-CF$"

5.3.2.1 Détail du module "$CNN-CF$"

Dans ce travail, le *CNN* a été utilisé pour apprendre l'interaction entre les caractéristiques de l'utilisateur et de l'item avec un niveau élevé de flexibilité et de non-linéarité.

Entrée La première couche est la couche d'entrée, où deux vecteurs de caractéristiques décrivant l'utilisateur u et l'élément i ont été utilisés comme entrée. Nous n'avons utilisé que les identifiants de l'utilisateur u et de l'élément i, ces identifiants ont été convertis en un vecteur binarisé.

Embedding Après la couche d'entrée, il y a une couche d'intégration. Dans cette couche, l'identifiant de l'utilisateur et de l'item codé dans la première couche ont été utilisés pour obtenir leurs intégrations respectives à k-dimensions. Ces intégrations servent de vecteurs latents x_u et y_i. ensuite, la concaténation des deux vecteurs forme le vecteur d'interaction z^{cnn},

$$z^{cnn} = \varphi_1(x_u, y_i) = \begin{bmatrix} x_u \\ y_i \end{bmatrix}.$$

Convolution La couche de convolution a été utilisée pour extraire les caractéristiques contextuelles. L'architecture de convolution décrite dans [197] a été mise en œuvre pour analyser les caractéristiques contextuelles entre les vecteurs latents de l'utilisateur et de l'élément en traitant le vecteur d'interaction, z^{cnn}, de la couche d'intégration. Par exemple, une caractéristique contextuelle, c_n^m, a été extraite par le m^{ime} poids partagé, $W_c^m \in R_w^{'2k}$, dont la taille de la fenêtre w détermine le nombre de facteurs environnants,

$$c_n^m = f\left(W_c^m * z_{(:,n:(n+w-1))}^{cnn} + b_c^m\right), \tag{5.2}$$

où $*$ est un opérateur de convolution, $b_c^m \in R$ est un biais pour W_c^m et f est une fonction d'activation. Pour notre modèle, Rectifier Linear Unit [150] a été utilisée comme fonction d'activation car elle permet d'éviter le problème de la disparition du gradient (vanishing gradient problem) qui conduit normalement à une optimisation lente et à un mauvais minimum local [198]. Le vecteur de caractéristiques contextuelles, c^m, d'une interaction avec W_c^m pourrait être construit comme suit :

$$c^m = [c_1^m, c_2^m, ..., c_n^m, c_{k-w+1}^m] \tag{5.3}$$

il faut savoir qu'un poids partagé ne capture qu'un seul type de caractéristiques contextuelles. Dans notre modèle, plusieurs poids partagés ont été déployés pour capturer plusieurs types de caractéristiques et générer autant de nombres j de vecteurs de caractéristiques contextuelles, donc par exemple, si nous avons choisi d'avoir 10 poids partagés différents W_c^m (où$m = 1,2,3,...,10$), nous obtiendrons 10 vecteurs de caractéristiques contextuelles différents.

Pooling Cette couche extrait les caractéristiques représentatives de la couche convolutionnelle, où une interaction peut être représentée par un nombre j de vecteurs de caractéristiques contextuelles. D'après l'équation 5.3, il pourrait y avoir trop de caractéristiques contextuelles et la plupart d'entre elles pourraient ne pas être utiles.

Par conséquent, à l'aide du max-pooling, seule la caractéristique contextuelle maximale de chaque vecteur de caractéristiques contextuelles a été extraite.

$$z_f^{cnn} = [max(c^1), max(c^2), ..., max(c^i)] \tag{5.4}$$

où c^m est un vecteur de caractéristique contextuelle extrait par le m^{ime} $poidspartag$,W$_c^m$.

Dans le cas où plusieurs couches de *CNN* seraient utilisées, le vecteur de l'équation 5.4 servirait d'entrée pour les couches suivantes, donc :

$$z_1^{cnn} = \varphi_1(x_u, y_i),$$

$$z_2^{cnn} = \varphi_2(z_1^{cnn}),$$

$$. \tag{5.5}$$

$$.$$

$$z_f^{cnn} = \varphi_f(z_f^{cnn}) = cnn\left(W, \begin{bmatrix} x_u \\ y_i \end{bmatrix}.\right)$$

où W désigne l'ensemble des variables de poids et de biais pour prévenir le bruit et $\begin{bmatrix} x_u \\ y_i \end{bmatrix}$. représente le vecteur d'interaction original.

Sortie Au niveau de la couche de sortie, les caractéristiques obtenues à partir de la couche précédente sont ensuite projetées dans la couche de sortie pour produire le score estimé :

$$\widehat{p}_{ui} = \sigma\left[h^T z_f^{cnn}\right] \tag{5.6}$$

où σ est la fonction sigmoïde donnée par : $\sigma(a) = 1/(1+\exp^{-a})$ et h est le poids de la couche de sortie.

5.3.2.2 La fonction objective

La recommandation de feedback implicite a été convertie en un problème de classification binaire, où 1 indique que l'utilisateur a interagi avec l'article et 0 dans le cas contraire. Cependant, n'avoir que la sortie binaire, soit 1 ou 0, était un problème trivial. Par conséquent, une approche probabiliste a été utilisée dans cette étude, où le score de prédiction, $\widehat{p}_{ui}$, représentant la probabilité que i soit pertinent pour u, serait généré à la place. Pour exprimer le score d'une manière probabiliste, le score a dû être contraint dans la gamme entre 0 et 1, ce qui a été fait en appliquant une fonction logistique sur la fonction d'activation de la couche de sortie. Avec cela, la fonction de vraisemblance est définie comme :

$$p(O, O^-|X, Y, \theta_f) = \sum_{(u,i)\in O} \widehat{p}_{ui} \sum_{(u,i)\in O^-} (1 - \widehat{p}_{ui}), \tag{5.7}$$

où X et Y représentent la matrice des vecteurs latents des utilisateurs et items respectivement : O et O^- représentent l'ensemble des interactions observées et l'ensemble des instances négatives

respectivement ; et θ_f est le paramètre du modèle de la fonction de vraisemblance.

En appliquant le logarithme négatif de la fonction de vraisemblance, nous obtenons :

$$\begin{aligned} L &= -\sum_{(u,i)\in O} log\widehat{p}_{ui} - \sum_{(u,j)\in O^-} log\widehat{(1-\widehat{p}_{uj})} \\ &= -\sum_{(u,i)\in O\cup O^-} p_{ui} log\widehat{p}_{ui} + (1-p_{ui}) log(1-\widehat{p}_{ui}) \end{aligned}$$

(5.8)

L est la fonction objective connue sous le nom de perte d'entropie croisée binaire et devait être minimisée pendant la formation en utilisant la descente de gradient stochastique [199] pour produire les meilleures prédictions.

5.4 Validation du modèle $CNN-CF-GT$

Pour valider le modèle $CNN-CF-GT$, nous avons réalisé une analyse hors ligne à l'aide du jeu de données *EduTest* détaillé dans le chapitre précédent. Cette simulation nous a aidé à tester notre proposition pour évaluer son efficacité.

5.4.1 Données de simulation

Afin d'évaluer le modèle $CNN-CF-GT$ pour la recommandation de ressources pédagogiques, nous avons utilisé le jeu de données *EduTest* (section 4.4.1).

Comme expliqué précédemment, le module «$CNN-CF$» se base sur le feedback implicite. Pour cela, nous avons apporté un changement au niveau du jeu de données *EduTest*. En ce qui concerne la relation entre l'apprenant et une ressource dans *EduTest* on distingue un couple (information de consultation, évaluation de la ressource), le premier élément indique si la ressource a été consulté et prend la valeur 1 sinon 0. Le deuxième élément varie de 1 à 5 indiquant la note attribuée à la ressource par l'apprenant cible. Pour l'évaluation du modèle $CNN-CF-GT$, nous ignorons la note attribuée à la ressource et utiliser l'information de consultation comme feedback implicite.

Le tableau 5.2 présente un aperçu du feedback implicite de 13 apprenants à 10 ressources pédagogiques.

TABLE 5.2 – Représentation du feedback implicite dans *EduTest*

Nom	Ress1	Ress2	Ress3	Ress4	Ress5	Ress6	Ress7	Ress8
A1	0	0	0	0	0	1	0	1
A2	1	1	0	1	1	0	0	1
A3	0	1	1	0	1	0	0	1
A4	1	1	0	0	0	1	1	0
A5	1	0	0	0	0	1	0	1
A6	0	1	1	0	1	0	1	1
A7	0	1	1	1	1	0	1	0
A8	0	0	1	1	1	1	1	1
A9	1	0	0	0	0	0	1	1
A10	0	0	1	1	1	1	0	1
A11	1	0	1	0	0	0	1	0
A12	0	0	0	0	1	0	1	0
A13	0	0	0	0	0	0	1	0

Nous soulignons que pour l'exécution du modèle $CNN-CF-GT$ la fonctionnalité du module *SimilarUser* est entièrement identique à son exécution pour le modèle $Edu-CF-GT$. Le module prend en compte toutes les caractéristiques de la base de test *EduTest* et la fonction de similarité utilisée pour deux utilisateurs u_i et u_j est $SimEval(u_i,u_j)$ 4.3.2.

5.4.2 Modèles de comparaison

Afin de valider le modèle $CNN-CF-GT$, nous l'avons comparé à deux autres approches. Nous nous sommes concentrés sur les algorithmes de filtrage collaboratif puisque $CNN-CF-GT$ se base sur cet algorithme. En effet, nous avons comparé notre méthode avec le modèle du module « $CNN-CF$ », en d'autres termes, nous avons ignoré la fonctionnalité du module « *SimilarUser*» pour voir l'impact de la présélection des utilisateurs similaires. Pour le deuxième modèle de test, nous avons préservé l'architecture générale de $CNN-CF-GT$ et avons remplacé la fonctionnalité du module *SimilarUser* par le processus $k-means$, et ce pour voir l'impacte de la présélection en utilisant SV. Le choix de ces deux algorithmes se justifie par le fait que le filtrage collaboratif est un algorithme de base qui reste une référence importante.

5.4.3 Protocole d'évaluation

Comme pour nos deux premières propositions $CF-GT$ et $Edu-CF-GT$, nous avons testé le modèle par rapport aux recommandations fournies. Nous avons utilisé l'ensemble de données *EduTest*, comme dans [156, 200] nous sélectionnons aléatoirement P ressources pédagogiques associés à chaque apprenant pour former l'ensemble d'apprentissage et nous utilisons tout le

reste comme ensemble de test. Afin d'évaluer et de comparer les modèles dans les contextes clairsemés et denses, nous avons fixé P à 1 et 10, respectivement, dans nos expériences. Pour chaque valeur de P, nous répétons l'évaluation cinq fois avec différents ensembles d'apprentissage choisis au hasard et la performance moyenne est présentée.

Nous utilisons le rappel comme mesure de performance parce que les informations de notation sont sous la forme de feedback implicite (comme détaillé dans la section 5.3.1). Plus précisément, une entrée nulle peut être due au fait que l'apprenant n'est pas intéressé par la ressource pédagogique, ou qu'il n'est pas au courant de son existence. En tant que telle, la précision n'est pas une mesure de performance appropriée [156]. Comme la plupart des systèmes de recommandation, nous trions les évaluations prédites des éléments candidats et recommandons-les $top-N$ ressources à l'apprenant cible. Le résultat final indiqué est le rappel moyen pour tous les utilisateurs.

5.4.4 Résultats

Les tableaux 5.3, 5.4 et 5.5 montrent les valeurs de rappel pour les différents $top-N$ avec la base *EduTest* pour le modèle $CNN-CF-GT$.

TABLE 5.3 – Rappel pour $\sigma = 0.79$ *EduTest* pour $CNN-CF-GT$

Top	**CNN-CF-GT**	CNN-CF	K_means
5	38%	28%	32%
10	44%	39%	40%
15	56%	48%	50%
20	64%	58%	61%
25	68%	61%	63%
30	73%	64%	70%
35	73%	65%	71%
40	72%	65%	69%
45	70%	63%	67%
50	68%	63%	65%

TABLE 5.4 – Rappel pour $\sigma = 0.86$ *EduTest* pour $CNN-CF-GT$

Top	**CNN-CF-GT**	CNN-CF	K_means
5	39%	28%	34%
10	44%	39%	41%
15	56%	48%	51%
20	66%	58%	61%
25	68%	61%	64%
30	72%	64%	68%
35	72%	65%	67%
40	69%	65%	66%
45	67%	63%	65%
50	64%	63%	64%

TABLE 5.5 – Rappel pour $\sigma = 0.87$ *EduTest* pour $CNN - CF - GT$

Top	**CNN-CF-GT**	CNN-CF	K_means
5	39%	28%	32%
10	46%	39%	42%
15	58%	48%	51%
20	65%	58%	63%
25	69%	61%	65%
30	74%	64%	70%
35	73%	65%	70%
40	71%	65%	68%
45	71%	63%	66%
50	69%	63%	65%

5.4.5 Discussion

Le modèle $CNN - CF - GT$ est une amélioration du modèle $Edu - CF - GT$. Nous avons intégré le *CNN* au deuxième module pour implémenter le processus du filtrage collaboratif. Pour ce modèle, le feedback implicite a été utilisé. Le *CNN* est utilisé pour modéliser l'interaction entre apprenant/ressource pédagogique et prédire l'avis de chaque apprenant sur les ressources. Pour le deuxième module « $CNN - CF$ » il y a que le feedback des apprenants par rapport aux ressources pédagogiques qui a été exploité (le profil des apprenants n'a pas été pris en charge).

L'évaluation par rapport à l'exactitude des recommandations fournies en utilisant le jeu de données *EduTest* révèle que $CNN - CF - GT$ a pu améliorer les deux modèles de test $CNN - CF$ et $k - means - CF$. Ces résultats s'expliquent par le fait $Edu - CF - GT$ produit une bonne présélection des apprenants similaires. De la même manière comme interprété pour le modèle générique $CF - GT$.

Si on compare les résultats de l'évaluation du modèle $CNN - CF - GT$ et $Edu - CF - GT$, nous pouvons clairement remarquer, que les valeurs de *rappel* obtenu par le modèle $CNN - CF - GT$ surpasse les valeurs de $Edu - CF - GT$. Cela indique la pertinence du *CNN* à modéliser l'interaction utilisateur/item et prédire les notes.

5.5 Conclusion

Dans ce chapitre, nous avons présenté un modèle de recommandation de ressources pédagogiques $CNN - CF - GT$. Ce modèle est une amélioration du modèle $Edu - CF - GT$.

Le modèle $CNN - CF - GT$ utilise le feedback implicite, pour cela, nous avons apporté des modifications à notre base de test *EduTest* pour l'adapter à l'entrée du modèle. Dans ce travail, la fonctionnalité du premier module de $Edu - CF - GT$ a été maintenue. L'amélioration touche particulièrement le deuxième module du modèle. Le *CNN* a été utilisé pour modéliser directement l'interaction apprenant-ressource d'apprentissage à partir des données et l'a intégré à la

factorisation matricielle pour fournir des recommandations. Le *CNN* a été exploré, car il comporte une couche de convolution qui extrait les caractéristiques locales en convoluant les signaux d'entrée des neurones adjacents, ce qui lui permet d'être plus flexible lors de l'apprentissage des caractéristiques.

Le modèle $CNN - CF - GT$ a été testé et validé par rapport au rappel. Il a été démontré que notre modèle proposé a pu améliorer les méthodes existantes, à savoir notre modèle $CNN - CF$.

Le modèle peut être étendu, une couche supplémentaire sera ajoutée pour prendre en compte toutes les caractéristiques des apprenants. $CNN - CF - GT$ peut également être amélioré en cherchant de meilleurs paramètres pour le modèle et de la fonction objective.

Conclusion Générale

Conclusion

Dans cette thèse, nous nous sommes intéressées au domaine de la fouille de données éducatives. Ce domaine a pour but l'amélioration du processus d'apprentissage des apprenants pour obtenir de meilleurs résultats. Nous distinguons plusieurs domaines d'application en EDM, au cours de notre travail, nous nous sommes intéressées aux systèmes de recommandations.

Les systèmes de recommandations sont une solution à la surcharge d'information, ils ont été conçus dans un premier temps pour faire face à l'énorme quantité de données générée dans le web et aider l'utilisateur à obtenir la ressource adéquate à ses besoins dans un temps optimal.

Nous avons commencé notre recherche par étudier l'état de l'art de l'EDM. L'étude de l'état de l'art est résumée dans le chapitre 1. Après cette étude, nous avons choisi un domaine d'application qui est les systèmes de recommandations.

Comme deuxième étape de notre recherche, nous avons effectué un état de l'art des systèmes de recommandations, ce qui est résumé dans le deuxième chapitre. Au cours de cette étude, nous avons constaté que la sélection d'utilisateur similaire est une étape sensible dans le processus de recommandations. Il a été remarqué également que parmi tous les challenges du domaine, la précision des recommandations reste toujours un défi à relever. Plusieurs chercheurs ont adopté le clustering comme une étape de présélection des utilisateurs similaires. L'étape de présélection prend en compte la relation entre une paire d'utilisateur pour former les groupes de profils similaires. Après cette étape, un processus de filtrage conventionnel est appliqué sur chaque groupe obtenu. Les résultats expérimentaux de leurs travaux s'avèrent prometteurs.

L'objectif de notre travail est d'améliorer l'étape de la présélection des utilisateurs similaires dans le cadre du filtrage collaboratif. Le but de ce travail est d'améliorer la précision des ressources pédagogiques recommandées aux apprenants.

Notre travail est fortement motivé par ce qui a été diversement décrit dans la littérature [132] : le regroupement doit être effectué non seulement sur la base de la distance entre une paire de points, mais aussi sur leur relation avec d'autres points. Dans cette perspective, nous avons proposé un système nommé $CF-GT$ dans un cadre générique, notre but et d'optimiser le processus du filtrage collaboratif en présélectionnant des groupes d'utilisateurs similaire.

Notre modèle $CF-GT$ a comme première étape la présélection d'utilisateurs similaire. La présélection se base sur la théorie des jeux coopératifs. L'idée est de faire correspondre la

formation de clusters à la formation de coalitions, en utilisant la valeur de Shapley. La valeur de Shapley est un concept de solution équitable dans la mesure où elle divise la valeur collective ou totale du jeu entre les joueurs en fonction de leurs contributions marginales à la réalisation de cette valeur collective. La valeur de Shapley prend en considération les propriétés intrinsèques d'un cluster. Après l'obtention de groupes d'utilisateurs similaire, on applique le processus du filtrage collaboratif sur chaque groupe. $CF-GT$ a été testé par rapport à deux indicateurs de performance : le MAE et le précision/rappel. Les résultats expérimentaux montrent que notre modèle minimise l'erreur absolue moyenne et améliore nettement la précision et rappel de notre modèle. Les résultats ont été comparés avec d'autres modèles, à savoir le filtrage collaboratif FC et le filtrage collaboratif basé sur le $k-means$ ($k-means-CF$). La comparaison montre clairement que notre modèle améliore ces travaux.

La pertinence de notre approche demeure dans le fait que la présélection d'utilisateurs similaires soutient le processus du filtrage collaboratif. En plus, la présélection adoptée n'est pas conventionnelle, mais elle prend en considération la propriété intrinsèque d'un cluster.

Comme susmentionné, le modèle $CF-GT$ est un cadre de recommandation générique. Afin de prouver son utilité à l'intérêt de l'aide à la décision dans le domaine éducatif, nous avons testé le modèle dans un contexte éducatif. Le travail mené dans cette perspective a touché deux volets : la conception d'une base de test éducatif qui comprend plusieurs caractéristiques et l'adaptation du modèle $CF-GT$ aux entrées de la base conçue. Ce qui a donné existence à un nouveau modèle $Edu-CF-GT$. Le modèle a été testé et les résultats affirment que le modèle de recommandation $Edu-CF-GT$ améliore d'une manière significative les travaux conventionnels. Une nette amélioration a été constatée également par rapport au modèle $CF-GT$. L'amélioration est due au fait que $Edu-CF-GT$ prend en compte plusieurs caractéristiques contrairement au modèle $CF-GT$ qui ne prend en compte qu'une seule caractéristique.

Avec le développement rapide des plateformes d'enseignement en ligne public et privée, les données générées sont d'une croissance exponentielle. Nous avons pensé à améliorer le modèle $Edu-CF-GT$ afin qu'il préserve ses performances même dans un environnement de big data. Pour cela, nous avons proposé un nouveau modèle $CNN-CF-GT$, dans ce modèle, nous avons intégré le CNN pour améliorer le deuxième module du modèle $Edu-CF-GT$. L'intégration du deep learning dans un processus de recommandation permet de traiter les problèmes d'interaction utilisateur/item complexes et de refléter précisément les préférences de l'utilisateur. Les résultats expérimentaux de notre modèle révèlent son efficacité par rapport au modèle $Edu-CF-GT$.

Toutefois, nos travaux n'ont traité que le problème de la précision des recommandations fournies. La diversité et est un volet important à prendre en compte pour satisfaire les apprenants et améliorer leurs processus d'apprentissage.

Perspectives

Les travaux que nous avons réalisés pourraient être complétés en abordant différents aspects. Il serait pertinent de considérer le problème de la diversité des recommandations de ressources pédagogiques. Cela permet d'une part de réduire la redondance dans la liste de recommandations et d'autre part tenir compte des divers intérêts des apprenants, souvent peu ou mal spécifiés.

Après l'optimisation de nos travaux, nous envisageons de développer une plateforme d'enseignement en ligne pour l'université et y intégrer notre modèle de recommandations optimisé. Cela permet aux étudiants de bénéficier de cours adéquats à leurs profils et intérêts. Après la collecte de données suffisantes, le système sera testé et amélioré si nécessaire. Notre perspective est motivée par la situation sanitaire actuelle. La pandémie actuelle de la « COVID-19 » a chamboulé tous les secteurs, à savoir le secteur éducatif. Avec l'absence des systèmes d'enseignement en ligne, les enseignants trouvent des difficultés à garder un bon rythme d'enseignement. La plateforme sera bénéfique pour les étudiants, elle facilitera l'accès aux cours et supports d'apprentissage adéquats à leurs besoins et intérêts. D'une autre part, la plateforme facilitera la tâche de l'éducation à nos enseignants.

Bibliographie

[1] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems : State of the art and trends. *Recommender systems handbook*, pages 73–105, 2011.

[2] Behdad Bakhshinategh, Osmar R Zaiane, Samira ElAtia, and Donald Ipperciel. Educational data mining applications and tasks : A survey of the last 10 years. *Education and Information Technologies*, 23(1) :537–553, 2018.

[3] Geeta Kashyap and E. Chauhan. Review on educational data mining techniques. 2015.

[4] María Cora Urdaneta-Ponte, Amaia Mendez-Zorrilla, and Ibon Oleagordia-Ruiz. Recommendation systems for education : systematic review. *Electronics*, 10(14) :1611, 2021.

[5] Tim J Berners-Lee. The world-wide web. *Computer networks and ISDN systems*, 25(4-5) :454–459, 1992.

[6] Folasade Olubusola Isinkaye, YO Folajimi, and Bolande Adefowoke Ojokoh. Recommendation systems : Principles, methods and evaluation. *Egyptian Informatics Journal*, 16(3) :261–273, 2015.

[7] Selma Benkessirat, Narhimène Boustia, and Nachida Rezoug. Overview of recommendation systems. In *Smart Education and e-Learning 2019*, pages 357–372. Springer, 2019.

[8] Selma Benkessirat, Narhimene Boustia, and Rezoug Nachida. A new collaborative filtering approach based on game theory for recommendation systems. *Journal of Web Engineering*, pages 303–326, 2021.

[9] Lipi Shah, Hetal Gaudani, and Prem Balani. Survey on recommendation system. *International Journal of Computer Applications*, 137(7) :43–49, 2016.

[10] LR Divyaa and Nargis Pervin. Towards generating scalable personalized recommendations : Integrating social trust, social bias, and geo-spatial clustering. *Decision Support Systems*, 2019.

[11] Gui-Rong Xue, Chenxi Lin, Qiang Yang, WenSi Xi, Hua-Jun Zeng, Yong Yu, and Zheng Chen. Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of*

the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pages 114–121. ACM, 2005.

[12] Linqi Song, Cem Tekin, and Mihaela Van Der Schaar. Clustering based online learning in recommender systems : a bandit approach. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4528–4532. IEEE, 2014.

[13] Ngo Tung Son, Dao Huy Dat, Nguyen Quang Trung, and Bui Ngoc Anh. Combination of dimensionality reduction and user clustering for collaborative-filtering. In *Proceedings of the 2017 International Conference on Computer Science and Artificial Intelligence*, pages 125–130. ACM, 2017.

[14] Hafed Zarzour, Ziad Al-Sharif, Mahmoud Al-Ayyoub, and Yaser Jararweh. A new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques. In *2018 9th International Conference on Information and Communication Systems (ICICS)*, pages 102–106. IEEE, 2018.

[15] Utkarsh Gupta and Nagamma Patil. Recommender system based on hierarchical clustering algorithm chameleon. In *2015 IEEE International Advance Computing Conference (IACC)*, pages 1006–1010. IEEE, 2015.

[16] Benkessirat Selma, Boustia Narhimène, and Rezoug Nachida. Deep learning for recommender systems : Literature review and perspectives. In *2021 International Conference on Recent Advances in Mathematics and Informatics (ICRAMI)*, pages 1–7. IEEE, 2021.

[17] Cristóbal Romero and Sebastián Ventura. Educational data mining : a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(6) :601–618, 2010.

[18] Cristobal Romero and Sebastian Ventura. Educational data mining and learning analytics : An updated survey. *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery*, 10(3) :e1355, 2020.

[19] Cristobal Romero and Sebastian Ventura. Data mining in education. *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery*, 3(1) :12–27, 2013.

[20] Ryan SJD Baker, Kalina Yacef, et al. The state of educational data mining in 2009 : A review and future visions. *Journal of educational data mining*, 1(1) :3–17, 2009.

[21] Cristóbal Romero and Sebastián Ventura. Educational data mining : a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(6) :601–618, 2010.

[22] Alejandro Peña-Ayala. Educational data mining : A survey and a data mining-based analysis of recent works. *Expert systems with applications*, 41(4) :1432–1462, 2014.

[23] Leah P Macfadyen and Shane Dawson. Numbers are not enough. why e-learning analytics failed to inform an institutional strategic plan. *J. Educ. Technol. Soc.*, 15(3) :149–163, 2012.

[24] Ángel Del Blanco, Ángel Serrano, Manuel Freire, Iván Martínez-Ortiz, and Baltasar Fernández-Manjón. E-learning standards and learning analytics. can data collection be improved by using standard data models ? In *2013 IEEE Global Engineering Education Conference (EDUCON)*, pages 1255–1261. IEEE, 2013.

[25] Stephen Downes. Places to go : Connectivism & connective knowledge. *Innovate : Journal of Online Education*, 5(1) :6, 2008.

[26] Julika Siemer and Marios C Angelides. A comprehensive method for the evaluation of complete intelligent tutoring systems. *Decision support systems*, 22(1) :85–102, 1998.

[27] Santosh Ray and Mohammed Saeed. Applications of educational data mining and learning analytics tools in handling big data in higher education. In *Applications of big data analytics*, pages 135–160. Springer, 2018.

[28] Hanan Aldowah, Hosam Al-Samarraie, and Wan Mohamad Fauzy. Educational data mining and learning analytics for 21st century higher education : A review and synthesis. *Telematics and Informatics*, 37 :13–49, 2019.

[29] Yu Wang, Tong Li, Congkai Geng, and Yihan Wang. Recognizing patterns of student's modeling behaviour patterns via process mining. *Smart Learning Environments*, 6(1) :1–16, 2019.

[30] Dario Delgado-Quintero, Olmer Garcia-Bedoya, Diego Aranda-Lozano, Pablo Munevar-Garcia, and Cesar O Diaz. Academic behavior analysis in virtual courses using a data mining approach. In *International Conference on Applied Informatics*, pages 17–31. Springer, 2019.

[31] Snježana Križanić. Educational data mining using cluster analysis and decision tree technique : A case study. *International Journal of Engineering Business Management*, 12 :1847979020908675, 2020.

[32] Liyan Tu. Analysis and prediction method of student behavior mining based on campus big data. In *International Conference on Advanced Hybrid Information Processing*, pages 363–371. Springer, 2019.

[33] Kunyanuth Kularbphettong. Analysis of students' behavior based on educational data mining. In *Proceedings of the Computational Methods in Systems and Software*, pages 167–172. Springer, 2018.

[34] Galina Deeva and Jochen De Weerdt. Understanding automated feedback in learning processes by mining local patterns. In *International Conference on Business Process Management*, pages 56–68. Springer, 2019.

[35] Junyi Zheng and Wenhui Peng. Establishment of problem e-learning behavior scale. In *International Conference of Artificial Intelligence, Medical Engineering, Education*, pages 394–403. Springer, 2020.

[36] Ateya Iram. Sentiment analysis of student's facebook posts. In *International Conference on Intelligent Technologies and Applications*, pages 86–97. Springer, 2019.

[37] Lamiaa Mostafa. Student sentiment analysis using gamification for education context. In *International Conference on Advanced Intelligent Systems and Informatics*, pages 329–339. Springer, 2020.

[38] Shuhan Liu and Genfu Yang. Negative sentiment analysis of mooc comments based on machine learning. In *Advanced Graphic Communication, Printing and Packaging Technology*, pages 562–567. Springer, 2020.

[39] María Lucía Barrón Estrada, Ramón Zatarain Cabada, Raúl Oramas Bustillos, and Mario Graff. Opinion mining and emotion recognition applied to learning environments. *Expert Systems with Applications*, 150 :113265, 2020.

[40] Nguyen Thi Phuong Giang, Tran Thanh Dien, and Tran Thi Minh Khoa. Sentiment analysis for university students' feedback. In *Future of Information and Communication Conference*, pages 55–66. Springer, 2020.

[41] Ebru Özpolat and Gözde B Akar. Automatic detection of learning styles for an e-learning system. *Computers & Education*, 53(2) :355–367, 2009.

[42] Samuel Amponsah. Exploring the dominant learning styles of adult learners in higher education. *International Review of Education*, 66(4) :531–550, 2020.

[43] Roberto D Costa, Gustavo F Souza, Ricardo AM Valentim, and Thales B Castro. The theory of learning styles applied to distance learning. *Cognitive Systems Research*, 64 :134–145, 2020.

[44] Wafaa S Sayed, Mostafa Gamal, Moemen Abdelrazek, and Samah El-Tantawy. Towards a learning style and knowledge level-based adaptive personalized platform for an effective and advanced learning for school students. *Recent Advances in Engineering Mathematics and Physics*, pages 261–273, 2020.

[45] Gabriela Czibula, Andrei Mihai, and Liana Maria Crivei. S prar : a novel relational association rule mining classification model applied for academic performance prediction. *Procedia Computer Science*, 159 :20–29, 2019.

[46] Kadri Umbleja and Manabu Ichino. Predicting students' behavior during an e-learning course using data mining. In *International Conference on Interactive Collaborative Learning*, pages 175–189. Springer, 2018.

[47] MohammadNoor Injadat, Abdallah Moubayed, Ali Bou Nassif, and Abdallah Shami. Systematic ensemble model selection approach for educational data mining. *Knowledge-Based Systems*, 200 :105992, 2020.

[48] Alberto Rivas, Alfonso Gonzalez-Briones, Guillermo Hernandez, Javier Prieto, and Pablo Chamoso. Artificial neural network analysis of the academic performance of students in virtual learning environments. *Neurocomputing*, 423 :713–720, 2021.

[49] Amjad Abu Saa, Mostafa Al-Emran, and Khaled Shaalan. Mining student information system records to predict students' academic performance. In *International conference on advanced machine learning technologies and applications*, pages 229–239. Springer, 2020.

[50] Samuel-Soma M Ajibade, Nor Bahiah Ahmad, and Siti Mariyam Shamsuddin. A data mining approach to predict academic performance of students using ensemble techniques. In *International Conference on Intelligent Systems Design and Applications*, pages 749–760. Springer, 2020.

[51] Amal Alhadabi and Aryn C Karpinski. Grit, self-efficacy, achievement orientation goals, and academic performance in university students. *International Journal of Adolescence and Youth*, 25(1) :519–535, 2020.

[52] Viet Anh Nguyen, Hoa-Huy Nguyen, Duc-Loc Nguyen, and Minh-Duc Le. A course recommendation model for students based on learning outcome. *Education and Information Technologies*, pages 1–27, 2021.

[53] J Naren, M Zarina Banu, and S Lohavani. Recommendation system for students' course selection. In *Smart Systems and IoT : Innovations in Computing*, pages 825–834. Springer, 2020.

[54] Ivana Ognjanovic, Dragan Gasevic, and Shane Dawson. Using institutional data to predict student course selections in higher education. *The Internet and Higher Education*, 29 :49–62, 2016.

[55] Padmaja Appalla, Venu Madhav Kuthadi, and Tshilidzi Marwala. An efficient educational data mining approach to support e-learning. *Wireless Networks*, 23(4) :1011–1024, 2017.

[56] Roopam Sadh and Rajeev Kumar. Edm framework for knowledge discovery in educational domain. In *Recent Trends in Communication, Computing, and Electronics*, pages 409–417. Springer, 2019.

[57] Kristof Coussement, Minh Phan, Arno De Caigny, Dries F Benoit, and Annelies Raes. Predicting student dropout in subscription-based online learning environments : The beneficial impact of the logit leaf model. *Decision Support Systems*, 135 :113325, 2020.

[58] Zne-Jung Lee and Chou-Yuan Lee. A parallel intelligent algorithm applied to predict students dropping out of university. *The Journal of Supercomputing*, 76(2) :1049–1062, 2020.

[59] Boris Perez, Camilo Castellanos, and Dario Correal. Applying data mining techniques to predict student dropout : a case study. In *2018 IEEE 1st colombian conference on applications in computational intelligence (colcaci)*, pages 1–6. IEEE, 2018.

[60] Maria Prudência Martins, Vera L Migueis, DSB Fonseca, and Paulo DF Gouveia. Previsão do abandono académico numa instituição de ensino superior com recurso a data mining. 2020.

[61] X Palacios-Pacheco, W Villegas-Ch, and Sergio Luján-Mora. Application of data mining for the detection of variables that cause university desertion. In *International Conference on Technology Trends*, pages 510–520. Springer, 2019.

[62] Nindhia Hutagaol and Suharjito Suharjito. Predictive modelling of student dropout using ensemble classifier method in higher education. *Advances in Science, Technology and Engineering Systems Journal*, 4(4) :206–211, 2019.

[63] Galina Deeva, Johannes De Smedt, Pieter De Koninck, and Jochen De Weerdt. Dropout prediction in moocs : A comparison between process and sequence mining. In *International Conference on Business Process Management*, pages 243–255. Springer, 2017.

[64] Ghazala Bilquise, Sherief Abdallah, and Thaeer Kobbaey. Predicting student retention among a homogeneous population using data mining. In *Machine Learning and Big Data Analytics Paradigms : Analysis, Applications and Challenges*, pages 243–260. Springer, 2021.

[65] Anupam Khan and Soumya K Ghosh. Student performance analysis and prediction in classroom learning : A review of educational data mining studies. *Education and information technologies*, 26(1) :205–240, 2021.

[66] Clare Baek and Tenzin Doleck. Educational data mining versus learning analytics : A review of publications from 2015 to 2019. *Interactive Learning Environments*, pages 1–23, 2021.

[67] Avita Katal, Mohammad Wazid, and Rayan H Goudar. Big data : issues, challenges, tools and good practices. In *2013 Sixth international conference on contemporary computing (IC3)*, pages 404–409. IEEE, 2013.

[68] Jui-Long Hung, Morgan C Wang, Shuyan Wang, Maha Abdelrasoul, Yaohang Li, and Wu He. Identifying at-risk students for early interventions—a time-series clustering approach. *IEEE Transactions on Emerging Topics in Computing*, 5(1) :45–55, 2019.

[69] Scott Crossley, Ran Liu, and Danielle McNamara. Predicting math performance using natural language processing tools. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, pages 339–347, 2020.

[70] Gale M Sinatra, Benjamin C Heddy, and Doug Lombardi. The challenges of defining and measuring student engagement in science, 2015.

[71] Riccardo Pecori. A virtual learning architecture enhanced by fog computing and big data streams. *Future Internet*, 10(1) :4, 2018.

[72] Beijie Xu and Mimi Recker. Teaching analytics : A clustering and triangulation study of digital library user data. *Journal of Educational Technology & Society*, 15(3) :103–115, 2012.

[73] Beijie Xu. *Clustering educational digital library usage data : Comparisons of latent class analysis and K-Means algorithms*. Utah State University, 2013.

[74] Jacqueline Feild. Improving student performance using nudge analytics. *International Educational Data Mining Society*, 2015.

[75] Tal Soffer, Tali Kahan, and Eynat Livne. E-assessment of online academic courses via students' activities and perceptions. *Studies in Educational Evaluation*, 54 :83–93, 2017.

[76] Kimberly E Arnold and Matthew D Pistilli. Course signals at purdue : Using learning analytics to increase student success. In *Proceedings of the 2nd international conference on learning analytics and knowledge*, pages 267–270, 2012.

[77] Benjamin D Nye, Donald M Morrison, and Borhan Samei. Automated session-quality assessment for human tutoring based on expert ratings of tutoring success. *International Educational Data Mining Society*, 2017.

[78] Barbara Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004) :1–26, 2004.

[79] Marco De Gemmis, Leo Iaquinta, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro. Preference learning in recommender systems. *Preference Learning*, 41 :41–55, 2009.

[80] Charu C Aggarwal et al. *Recommender systems*, volume 1. Springer, 2016.

[81] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.

[82] Krupa Patel and Hiren B Patel. A state-of-the-art survey on recommendation system and prospective extensions. *Computers and Electronics in Agriculture*, 178 :105779, 2020.

[83] Alexander Felfernig, Ludovico Boratto, Martin Stettinger, and Marko Tkalčič. Decision tasks and basic algorithms. In *Group Recommender Systems*, pages 3–26. Springer, 2018.

[84] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[85] Poonam B Thorat, RM Goudar, and Sunita Barve. Survey on collaborative filtering, content-based filtering and hybrid recommendation system. *International Journal of Computer Applications*, 110(4) :31–36, 2015.

[86] Dunja Mladenic. Text-learning and related intelligent agents : a survey. *IEEE intelligent systems and their applications*, 14(4) :44–54, 1999.

[87] Tom Mitchell and Machine Learning McGraw-Hill. Edition, 1997.

[88] Dunja Mladenic and Browsing Assistant Personal Webwatcher. Machine learning used by personal webwatcher. 1999.

[89] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1) :5–53, 2004.

[90] Michael D Ekstrand, John T Riedl, and Joseph A Konstan. *Collaborative filtering recommender systems*. Now Publishers Inc, 2011.

[91] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6) :734–749, 2005.

[92] Cong Yu, Laks Lakshmanan, and Sihem Amer-Yahia. It takes variety to make a world : diversification in recommender systems. In *Proceedings of the 12th international conference on extending database technology : Advances in database technology*, pages 368–378, 2009.

[93] Najdt Mustafa, Ashraf Osman Ibrahim, Ali Ahmed, and Afnizanfaizal Abdullah. Collaborative filtering : Techniques and applications. In *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, pages 1–6. IEEE, 2017.

[94] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009, 2009.

[95] Mohammad Reza Zarei and Mohammad Reza Moosavi. A memory-based collaborative filtering recommender system using social ties. In *2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA)*, pages 263–267. IEEE, 2019.

[96] SongJie Gong, HongWu Ye, and HengSong Tan. Combining memory-based and model-based collaborative filtering in recommender system. In *2009 Pacific-Asia Conference on Circuits, Communications and Systems*, pages 690–693. IEEE, 2009.

[97] Miha Grčar, Dunja Mladenič, Blaž Fortuna, and Marko Grobelnik. Data sparsity issues in the collaborative filtering framework. In *International workshop on knowledge discovery on the web*, pages 58–76. Springer, 2005.

[98] Shalini Christabel Stephen, Hong Xie, and Shri Rai. Measures of similarity in memory-based collaborative filtering recommender system : A comparison. In *Proceedings of the 4th Multidisciplinary International Social Networks Conference*, pages 1–8, 2017.

[99] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.

[100] PH Aditya, Indra Budi, and Q Munajat. A comparative analysis of memory-based and model-based collaborative filtering on the implementation of recommender system for e-commerce in indonesia : A case study pt x. In *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pages 303–308. IEEE, 2016.

[101] Steffen Rendle and Christoph Freudenthaler. Improving pairwise learning for item recommendation from implicit feedback. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 273–282, 2014.

[102] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Nuria Oliver, and Alan Hanjalic. Climf : learning to maximize reciprocal rank with collaborative less-is-more filtering. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 139–146, 2012.

[103] Gábor Takács and Domonkos Tikk. Alternating least squares for personalized ranking. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 83–90, 2012.

[104] Markus Weimer, Alexandros Karatzoglou, Quoc Le, and Alex Smola. Cofi rank-maximum margin matrix factorization for collaborative ranking. *Advances in neural information processing systems*, 20, 2007.

[105] Rachana Mehta and Keyur Rana. A review on matrix factorization techniques in recommender systems. In *2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA)*, pages 269–274. IEEE, 2017.

[106] A Iskold. Rethinking recommendation engines [electronic version]. readwriteweb weblog. retrieved 2009, nov 13, 2008.

[107] Manos Papagelis, Dimitris Plexousakis, and Themistoklis Kutsuras. Alleviating the sparsity problem of collaborative filtering using trust inferences. In *International conference on trust management*, pages 224–239. Springer, 2005.

[108] R Burke. Hybrid web recommender systems, the adaptive web : Methods and strategies of web personalization, volume 4321 of lecture notes in computer science, 2007.

[109] Verus Pronk, Wim Verhaegh, Adolf Proidl, and Marco Tiemann. Incorporating user control into recommender systems based on naive bayesian classification. In *Proceedings of the 2007 ACM conference on Recommender systems*, pages 73–80, 2007.

[110] Chumki Basu, Haym Hirsh, William Cohen, et al. Recommendation as classification : Using social and content-based information in recommendation. In *Aaai/iaai*, pages 714–720, 1998.

[111] Yi Zhang and Jamie Callan. Maximum likelihood estimation for filtering thresholds. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 294–302, 2001.

[112] Robin Burke. Hybrid recommender systems : Survey and experiments. *User modeling and user-adapted interaction*, 12(4) :331–370, 2002.

[113] Daniel Billsus and Michael J Pazzani. A hybrid user model for news story classification. In *Um99 user modeling*, pages 99–108. Springer, 1999.

[114] Tim Miranda, Mark Claypool, Anuja Gokhale, Tim Mir, Pavel Murnikov, Dmitry Netes, and Matthew Sartin. Combining content-based and collaborative filters in an online newspaper. In *In Proceedings of ACM SIGIR Workshop on Recommender Systems*. Citeseer, 1999.

[115] Ingo Schwab, Alfred Kobsa, and Ivan Koychev. Learning user interests through positive examples using content analysis and collaborative filtering. *Internal Memo, GMD, St. Augustin, Germany*, 2001.

[116] Gebrekirstos G Gebremeskel and Arjen P de Vries. Recommender systems evaluations : Offline, online, time and a/a test. 2016.

[117] Qi He, Daniel Kifer, Jian Pei, Prasenjit Mitra, and C Lee Giles. Citation recommendation without author supervision. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 755–764, 2011.

[118] Qi He, Jian Pei, Daniel Kifer, Prasenjit Mitra, and Lee Giles. Context-aware citation recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 421–430, 2010.

[119] Cornelia Caragea, Adrian Silvescu, Prasenjit Mitra, and C Lee Giles. Can't see the forest for the trees ? a citation recommendation system. In *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*, pages 111–114, 2013.

[120] Ding Zhou, Shenghuo Zhu, Kai Yu, Xiaodan Song, Belle L Tseng, Hongyuan Zha, and C Lee Giles. Learning multiple graphs for document recommendations. In *Proceedings of the 17th international conference on World Wide Web*, pages 141–150, 2008.

[121] David M Pennock, Eric J Horvitz, Steve Lawrence, and C Lee Giles. Collaborative filtering by personality diagnosis : A hybrid memory-and model-based approach. *arXiv preprint arXiv :1301.3885*, 2013.

[122] Wenyi Huang, Saurabh Kataria, Cornelia Caragea, Prasenjit Mitra, C Lee Giles, and Lior Rokach. Recommending citations : translating papers into references. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1910–1914, 2012.

[123] Wenyi Huang, Saurabh Kataria, Cornelia Caragea, Prasenjit Mitra, C Lee Giles, and Lior Rokach. Recommending citations : translating papers into references. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1910–1914, 2012.

[124] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1) :5–53, 2004.

[125] Tianfeng Chai and Roland R Draxler. Root mean square error (rmse) or mean absolute error (mae) ?–arguments against avoiding rmse in the literature. *Geoscientific model development*, 7(3) :1247–1250, 2014.

[126] Joeran Beel, Bela Gipp, Stefan Langer, and Marcel Genzmehr. Docear : An academic literature suite for searching, organizing and creating academic literature. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*, pages 465–466. ACM, 2011.

[127] Joel P Lucas, Nuno Luz, MaríA N Moreno, Ricardo Anacleto, Ana Almeida Figueiredo, and Constantino Martins. A hybrid recommendation approach for a tourism system. *Expert Systems with Applications*, 40(9) :3532–3550, 2013.

[128] Mohammad Tahmasebi, Faranak Fotouhi Ghazvini, and Mahdi Esmaeili. Implementation and evaluation of a resource-based learning recommender based on learning style and web page features. *Journal of Web Engineering*, 17(3-4) :284–304, 2018.

[129] Mojtaba Salehi, Mohammad Pourzaferani, and Seyed Amir Razavi. Hybrid attribute-based recommender system for learning material using genetic algorithm and a multidimensional information model. *Egyptian Informatics Journal*, 14(1) :67–78, 2013.

[130] T Soni Madhulatha. An overview on clustering methods. *arXiv preprint arXiv :1205.1117*, 2012.

[131] Syed Fawad Hussain and Muhammad Haris. A k-means based co-clustering (kcc) algorithm for sparse, high dimensional data. *Expert Systems with Applications*, 118 :20–34, 2019.

[132] Samuel Bulò and Marcello Pelillo. A game-theoretic approach to hypergraph clustering. *Advances in neural information processing systems*, 22, 2009.

[133] Oskar Morgenstern and John Von Neumann. *Theory of games and economic behavior*. Princeton university press, 1953.

[134] Robert P Gilles. *The cooperative game theory of networks and hierarchies*, volume 44. Springer Science & Business Media, 2010.

[135] Steven Kuhn. Prisoner's dilemma. 1997.

[136] Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge. Computational aspects of cooperative game theory. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(6) :1–168, 2011.

[137] Walid Saad, Zhu Han, Mérouane Debbah, Are Hjorungnes, and Tamer Basar. Coalitional game theory for communication networks. *Ieee signal processing magazine*, 26(5) :77–97, 2009.

[138] Dejun Yang, Xi Fang, and Guoliang Xue. Game theory in cooperative communications. *IEEE Wireless Communications*, 19(2) :44–49, 2012.

[139] Manfred Besner. Axiomatizations of the proportional shapley value. *Theory and Decision*, 86(2) :161–183, 2019.

[140] Lloyd S Shapley. Cores of convex games. *International journal of game theory*, 1(1) :11–26, 1971.

[141] Lecture Notes By and Y Narahari. The shapley value. 2012.

[142] Roger B Myerson. Game theory : Analysis of conflict. 2013.

[143] Itziar Frades and Rune Matthiesen. Overview on techniques in cluster analysis. *Bioinformatics methods in clinical research*, pages 81–107, 2010.

[144] Mohammed Tadlaoui, Karim Sehaba, Sébastien George, Azeddine Chikh, and Karim Bouamrane. Social recommender approach for technology-enhanced learning. *International Journal of Learning Technology*, 13(1) :61–89, 2018.

[145] GroupLens. Movielens 100k. .http://files.grouplens.org/datasets/movielens/ml-100k.zip, 2000 (accessed September 1, 2019).

[146] Fajie Yuan, Guibing Guo, Joemon M Jose, Long Chen, Haitao Yu, and Weinan Zhang. Lambdafm : learning optimal ranking with factorization machines using lambda surrogates. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 227–236, 2016.

[147] Mark Levy. Offline evaluation of recommender systems : all pain and no gain ? In *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation*, pages 1–1, 2013.

[148] Mehrbakhsh Nilashi, Othman Ibrahim, and Karamollah Bagherifard. A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. *Expert Systems with Applications*, 92 :507–520, 2018.

[149] Haifeng Liu, Zheng Hu, Ahmad Mian, Hui Tian, and Xuzhen Zhu. A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-based systems*, 56 :156–166, 2014.

[150] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[151] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the ninth ACM international conference on web search and data mining*, pages 153–162, 2016.

[152] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. Attentional factorization machines : Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv :1708.04617*, 2017.

[153] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8) :30–37, 2009.

[154] Yehuda Koren. Factorization meets the neighborhood : a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, 2008.

[155] Steffen Rendle. Factorization machines. In *2010 IEEE International conference on data mining*, pages 995–1000. IEEE, 2010.

[156] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1235–1244, 2015.

[157] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[158] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. *Advances in neural information processing systems*, 26, 2013.

[159] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM conference on recommender systems*, pages 233–240, 2016.

[160] Tara N Sainath, Brian Kingsbury, Abdel-rahman Mohamed, George E Dahl, George Saon, Hagen Soltau, Tomas Beran, Aleksandr Y Aravkin, and Bhuvana Ramabhadran. Improvements to deep convolutional neural networks for lvcsr. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 315–320. IEEE, 2013.

[161] Soujanya Poria, Erik Cambria, and Alexander Gelbukh. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108 :42–49, 2016.

[162] Evgin Goceri and Numan Goceri. Deep learning in medical image analysis : recent advances and future trends. 2017.

[163] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system : A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1) :1–38, 2019.

[164] J SCHALKOFF Robert and RJ Schalkoff. Artificial intelligence : An engineering approach, 1990.

[165] Frank Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, Cornell Aeronautical Lab Inc Buffalo NY, 1961.

[166] Li Deng and Dong Yu. Deep learning : methods and applications. *Foundations and trends in signal processing*, 7(3–4) :197–387, 2014.

[167] Aminu Da'u and Naomie Salim. Recommendation system based on deep learning methods : a systematic review and new directions. *Artificial Intelligence Review*, 53(4) :2709–2748, 2020.

[168] Li Deng and Dong Yu. Deep learning : methods and applications. *Foundations and trends in signal processing*, 7(3–4) :197–387, 2014.

[169] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[170] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 77 :354–377, 2018.

[171] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234 :11–26, 2017.

[172] Henry Friday Nweke, Ying Wah Teh, Mohammed Ali Al-Garadi, and Uzoma Rita Alo. Deep learning algorithms for human activity recognition using mobile and wearable sensor networks : State of the art and research challenges. *Expert Systems with Applications*, 105 :233–261, 2018.

[173] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the ninth ACM international conference on web search and data mining*, pages 153–162, 2016.

[174] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. Autorec : Autoencoders meet collaborative filtering. In *Proceedings of the 24th international conference on World Wide Web*, pages 111–112, 2015.

[175] Sheng Li, Jaya Kawale, and Yun Fu. Deep collaborative filtering via marginalized denoising auto-encoder. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 811–820, 2015.

[176] Shuai Zhang, Lina Yao, and Xiwei Xu. Autosvd++ an efficient hybrid collaborative filtering model via contractive auto-encoders. In *Proceedings of the 40th International ACM*

SIGIR conference on Research and Development in Information Retrieval, pages 957–960, 2017.

[177] Gintare Karolina Dziugaite and Daniel M Roy. Neural network matrix factorization. *arXiv preprint arXiv :1511.06443*, 2015.

[178] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[179] Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. Dropoutnet : Addressing cold start in recommender systems. *Advances in neural information processing systems*, 30, 2017.

[180] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10, 2016.

[181] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 495–503, 2017.

[182] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv :1511.06939*, 2015.

[183] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H Chi. Latent cross : Making use of context in recurrent recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 46–54, 2018.

[184] Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. What to do next : Modeling user behaviors by time-lstm. In *IJCAI*, volume 17, pages 3602–3608, 2017.

[185] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. Sequential user-based recurrent neural network recommendations. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 152–160, 2017.

[186] Wenjie Pei, Jie Yang, Zhu Sun, Jie Zhang, Alessandro Bozzon, and David MJ Tax. Interacting attention-gated recurrent networks for recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1459–1468, 2017.

[187] Dietmar Jannach and Malte Ludewig. When recurrent neural networks meet the neighborhood for session-based recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 306–310, 2017.

[188] How Jing and Alexander J Smola. Neural survival recommender. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 515–524, 2017.

[189] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. *Advances in neural information processing systems*, 26, 2013.

[190] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 297–305, 2017.

[191] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM conference on recommender systems*, pages 233–240, 2016.

[192] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 353–362, 2016.

[193] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 549–558, 2016.

[194] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE international conference on data mining*, pages 263–272. Ieee, 2008.

[195] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr : Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv :1205.2618*, 2012.

[196] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. *Advances in neural information processing systems*, 26, 2013.

[197] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar*, page 1746–1751, 2014.

[198] X Glorot, A Bordes, and Y Bengio. Deep sparse rectifier neural networks. inproceedings of the fourteenth international conference on artificial intelligence and statistics (pp. 315–323). In *JMLR Workshop and Conference Proceedings*, 2011.

[199] Jack Kiefer and Jacob Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, pages 462–466, 1952.

[200] Hao Wang and Wu-Jun Li. Relational collaborative topic regression for recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 27(5) :1343–1355, 2014.