الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière
Département d'Informatique

Mémoire de Fin d'études Master

Filière : Informatique

Option : Systèmes Informatiques

Thème :

Automatisation de la sélection des articles scientifiques pour les revues systématiques : Une approche basée sur l'apprentissage automatique

Encadré par :

Dr. HANNOUSSE ABDELHAKIM

Présenté par :

MAHBOUBI ABDELMOKIM

Septembre 2021

Remerciement

Nous remercions en premier lieu ALLAH qui nous a éclairé notre chemin pour achever ce modeste travail.

Je voudrais saisir cette occasion et adresser mes sincères Remerciements et appréciation à Monsieur ***Hannousse Abdelhakim*** *pour son encadrement Et son précieux Soutien et leurs conseils tout au long de mes recherches.*

Je tiens également à remercier les membres du jury pour l'intérêt qu'ils portent à mes re- cherches en acceptant D'examiner mon travail et de l'enrichir de leurs propositions.

Je remercie également tous ceux qui ont, de près ou de loin, aidé à rendre ce travail possible, que ce soit par des informations, des idées ou par des encouragements.

Et bien sûr nous remercions ma famille : mes parents, mes frère et sœur, pour leurs soutiens et aides.

A tous mes collègues et mes amis.

Merci à tous et à toutes.

Dédicaces

Je dédie ce travail :

A ma très chère mère wahida : *Aucune dédicace ne saurait être assez éloquente pour exprimer ce que tu mérites pour tous les sacrifices que tu n'as cessé de me donner depuis ma naissance, durant mon enfance et même l'âge adulte.*

A mon père Abdelghani : *Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours eu pour vous, rein au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bienêtre. Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation.*

A tous les membres de ma famille : *veuillez trouvez dans ce modeste travail l'expression de mon affection, et surtout ma sœur : SELMA et mes frères : TAMER et RAID, les mots ne suffisent guère pour exprimer l'attachement, l'amour et l'affection que je porte pour vous.*

A mes amis : *surtout mes frères : Aymen, Doudou, Moumen, Aimad, Baha, Skander et les autres, sans ton aide, tes conseils et tes encouragements ce travail n'aurait vu le jour.*

Mokim

Résumé

La recherche des articles scientifiques pertinents à un projet de recherche est une tâche indispensable dans le protocole d'élaboration des revues systématiques. Pour éviter le risque d'éliminer des documents pertinents, une recommandation a été adoptée pour l'utilisation de plusieurs moteurs de recherche académiques à la fois : Google Scholar, Scopus, ACM, Springer, IEEE, ScienceDirect et Wiley. Cette recommandation rend la sélection des documents une tâche fastidieuse et coûteuse en termes de temps. Effectivement, la réponse à une requête par chaque moteur de recherche produit un nombre énorme de métadonnées qui doivent être filtrées manuellement. Chaque moteur adopte une stratégie particulière pour trier ses propres résultats, qui ne sont pas fiables pour la sélection rigoureuse des documents pertinents. À travers ce projet, nous contribuons par l'automatisation de la phase de sélection des métadonnées renvoyées par les différents moteurs de recherche en les classifiant selon leurs pertinences « sémantique » à la requête de recherche. Nous proposons donc l'utilisation des techniques de l'intelligence artificielle pour l'automatisation de cette tâche. L'objectif principal visé est la proposition et l'implémentation d'une approche à base d'apprentissage automatique et de l'ontologie du domaine pour la sélection des documents scientifiques depuis l'analyse de leurs métadonnées. L'ontologie du domaine enrichit les termes de recherches et améliore la détection de similarité entre les papiers, alors que l'apprentissage automatique prédit automatiquement la pertinence des papiers. L'expérimentation montre que l'approche proposée réduit jusqu'à 35% de l'effort manuel.

Mots-clés : Apprentissage automatique ; bibliothèques académiques numériques ; pertinence des documents aux revues systématiques.

Abstract

Searching relevant papers to a research project is an essential task in the protocol for developing systematic reviews. To avoid the risk of eliminating relevant documents, a recommendation was adopted for the use of several academic search engines at once: Google Scholar, Scopus, ACM, Springer, IEEE, ScienceDirect and Wiley. This recommendation makes document selection a tedious and time-consuming task. Indeed, the response to a query by each search engine produces a huge amount of metadata that must be filtered manually. Each engine adopts a particular strategy to sort its own results which are unreliable for the careful selection of relevant documents. Through this project, we want to contribute by automating the selection of relevant papers through analyzing the metadata returned by the different search engines and classifying them according to their "semantic" relevance to the search query. We therefore propose the use of artificial intelligence techniques for the automation of this task. The main aim is to propose an approach based on automatic learning and domain ontology for the selection of scientific documents from their metadata. The domain ontology enriches search terms and improves similarity detection between papers while machine learning automatically predicts paper relevance. Experimentation show that the proposed approach reduces up to 35% of manual effort.

Keywords : Machine learning; digital academic libraries; relevance of documents to systematic reviews.

ملخص

عند البحث عن المقالات العلمية ذات الصلة بمشروع بحثي مهمة أساسية في البروتوكول لتطوير المراجعات المنهجية. لتجنب مخاطر حذف المستندات ذات الصلة، تم اعتماد توصية لاستخدام العديد من محركات البحث الأكاديمية في وقت واحد: الباحث تجعل هذه التوصية .Wileyو ScienceDirect، IEEE، Springer، ACM، Scopus، Google العلمي من اختيار المستند مهمة شاقة وتستغرق وقتًا طويلاً. في الواقع، ينتج عن الاستجابة للاستعلام من قبل كل محرك بحث قدرًا هائلاً من البيانات الوصفية التي يجب تصفيتها يدويًا. يتبنى كل محرك إستراتيجية معينة لفرز نتائجه الخاصة التي لا يمكن الاعتماد عليها في الاختيار الدقيق للوثائق ذات الصلة. من خلال هذا المشروع، نريد أن نساهم من خلال أتمتة مرحلة اختيار البيانات الوصفية التي يتم إرجاعها بواسطة محركات البحث المختلفة وتصنيفها حسب صلتها "الدلالية" باستعلام البحث. لذلك نقترح استخدام تقنيات الذكاء الاصطناعي لأتمتة هذه المهمة . الهدف الرئيسي هو اقتراح نهج قائم على التعلم الآلي وأنطولوجيا المجال لاختيار الوثائق العلمية من البيانات الوصفية الخاصة به. يعمل علم الأنطولوجيا على إثراء مصطلحات البحث وتحسين اكتشاف التشابه بين الأوراق بينما يتنبأ التعلم الآلي تلقائيًا بموضوع البحث. تظهر التجربة أن النهج المقترح يقلل حتى 35٪ من الجهد اليدوي.

الكلمات الدالة :

التعلم الالي؛ مكتبات أكاديمية رقمية. أهمية الوثائق للمراجعات المنهجية

TABLE DES MATIERES

LISTE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION GENERALE

INTRODUCTION GENERALE

Les revues systématiques représentent une méthodologie qui permet de produire des synthèses rigoureuses des données probantes disponibles sur des sujets de recherche bien précis. L'élaboration des revues systématiques est connue être une tâche longue et fastidieuse vu le nombre énorme des preuves scientifiques existantes.

L'élaboration des revues systématiques implique différentes étapes ; commençant par la collection des données depuis plusieurs sources, ensuite la sélection des papiers pertinents au sujet de recherche, l'extraction des données aidant à la réponse aux questions de recherche, l'analyse et l'élaboration d'un rapport final. Spécifiquement, la phase de sélection des papiers est l'étape la plus longue vu le nombre énorme de papiers renvoyés par les différentes sources explorées. Les chercheurs doivent consulter tous les papiers, un par un, afin de déterminer la liste exhaustive à inclure dans la revue. Cela entraîne une perte de temps et d'efforts des chercheurs.

De nombreuses recherches ont été effectuées pour améliorer le processus de sélection des documents scientifiques. Certains offrent la possibilité de classer les articles selon leurs ordres de pertinence aux mots-clés de recherche. D'autre utilisent les techniques d'apprentissage automatique pour la classification des documents. Certains sont dédiés à des domaines de recherche spécifiques. Peu d'outils sont consacrés à l'automatisation du processus d'élaboration des revues systématiques dans le domaine informatique. Notamment la phase de sélection des papiers pertinents.

L'objectif principal de ce projet est la proposition d'une approche à base d'apprentissage automatique et de l'ontologie du domaine (Informatique) pour la sélection des documents scientifiques depuis leurs métadonnées. Un outil est mis en œuvre pour l'implémentation de l'approche proposée. Dans ce mémoire, nous décrivons en détails l'approche proposée et la conception de l'outil d'implémentation. Pour cet objectif, le présent mémoire est organisé en quatre chapitres :

Chapitre 1 : dans ce premier chapitre, nous introduisons la notion des revues systématiques, le processus d'élaboration de ces revues et nous décrivons les recherches les plus récentes qui visent à automatiser ce processus.

Chapitre 2 : dans ce chapitre, nous montrons les principes d'apprentissage automatique en se concentrant sur l'apprentissage non-supervisé et semi-supervisé, qui sont les technologies utilisées par notre solution proposée.

Chapitre 3 : nous détaillons, dans ce chapitre, l'architecture globale de notre système proposé. Le système vise à automatiser la sélection des documents scientifiques dans le domaine informatique depuis l'analyse de leurs métadonnées.

Chapitre 4 : dans ce dernier chapitre, nous abordons l'aspect implémentation de notre application, qui consiste à affiner les concepts précédemment développés ainsi que l'environnement de développement, les détails des tests, les résultats obtenus et leurs interprétations.

CHAPITRE I

LES REVUES SYSTEMATIQUES

CHAPITRE I.

LES REVUES SYSTEMATIQUES

Une revue systématique est un rapport scientifique décrivant une synthèse des preuves scientifiques existantes. Il implique une recherche bibliographique large et l'analyse de tous les documents scientifiques pertinents publiés en rapport avec un thème de recherche choisi. Les revues systématiques permettent aux chercheurs et aux professionnels, dans un domaine spécifique, d'accéder aux derniers résultats des preuves scientifiques valides depuis un document unique. Elles permettent ainsi à contribuer à l'orientation de leurs décisions et au choix de leurs modes d'intervention.

L'élaboration des revues systématiques est devenue une méthodologie de recherche largement adoptée dans différents domaines scientifiques, notamment l'informatique. Elle cherche à répondre à des questions de recherche clairement formulées par la synthèse des données et des preuves scientifiques existantes et de valeur. Ces dernières sont repérées suivant un protocole de recherche bien déterminé et approuvé.

Une revue systématique se distingue des autres synthèses scientifiques bien connues comme « *état de l'art* » et « *synthèse de connaissance* » par deux points essentiels [1]:

1. Une recherche et sélection exhaustives des articles scientifiques publiées sur le sujet de recherche en question. Une revue systématique peut aussi inclure des preuves scientifiques de valeur qui n'ont pas été publiés par les canaux scientifiques officiels. Ces ressources sont appelées aussi des *littératures grises*.
2. Adoption des critères explicites approuvés par un réseau d'experts pour la sélection et l'évaluation des publications recueillies.

L'élaboration des revues systématiques est une tâche longue et fastidieuse. Plus d'un expert (au moins deux) doivent intervenir tout au long du processus. Vu le nombre énorme des preuves scientifiques existantes et la diversité des canaux de publications disponibles, différentes questions se posent durant l'élaboration de ce type de

recherche, nous citons, à titre d'exemple : Pourquoi choisir certaines publications et rejeter d'autres ? Comment faire pour regrouper les résultats trouvés ?

Dans le présent chapitre, nous expliquons étape par étape la méthodologie d'élaboration des revues systématiques et les difficultés inhérentes à chaque étape. Nous discutons ainsi les différentes tentatives d'automatisation de ce processus long afin de réduire l'effort et le temps entretenus pour réaliser ce type des synthèses.

1.1. Historique des revues systématiques

Les premières revues systématiques ont été réalisés dans le domaine de médecine. Une attention accrue a été accordée au besoin d'améliorer l'état de la synthèse des preuves entre les années 1970 et 1980. En particulier, *Archie Cochrane* a publié un manuel en 1972 qui a attiré l'attention sur l'importance vitale des essais contrôlés randomisés pour déterminer l'efficacité des traitements de santé. Cela a conduit à une plus grande insistance internationale sur la nécessité d'améliorer la synthèse de la recherche par les décideurs, les chercheurs et les cliniciens [2].

En 2004, Kitchenham et collab. [3, 4] ont suggéré que les chercheurs en génie logiciel empirique devront adopter la pratique des revues systématiques suivant les chercheurs en médecine. Ils ont proposé une approche en génie logiciel fondé sur des preuves appelées « *Ingénierie de logiciels factuelle* » (ou *Evidence-Based Software Engineering* en anglais). L'approche proposée a été inspirée de la méthode d'*Archie Cochrane* adoptée en médecine ; elle repose sur l'agrégation des meilleures preuves disponibles pour répondre aux questions d'ingénierie soulevées par les praticiens et les chercheurs. Les preuves les plus fiables proviennent de l'agrégation de toutes les études empiriques sur un sujet particulier. Kitchenham a adaptée les directives médicales pour les revues systématiques au génie logiciel, et les a ensuite mis à jour pour inclure des informations issues de la recherche en sociologie [5]. Peu à peu, l'élaboration des revues systématiques a été adoptée comme un moyen de résumer de manière complète et systémique les preuves scientifiques existantes. L'approche de Kitchenham a été rapidement adoptée et favorisée par la majorité des journaux scientifiques de

différents domaines, notamment en informatique et a remplacé les synthèses traditionnelles comme « *état de l'art* » et « *synthèse de connaissance* ».

1.2. Avantages des revues systématiques

Les revues systématiques offrent un certain nombre d'avantages. Elles offrent un aperçu clair et complet des recherches disponibles sur un sujet donné. De plus, la revue systématique aide également à identifier les lacunes de la recherche dans la compréhension actuelle d'un domaine. Elles peuvent mettre en évidence des préoccupations méthodologiques dans les études de recherche qui peuvent être utilisées pour améliorer les travaux futurs dans le domaine thématique. Enfin, ils peuvent être utilisés pour identifier des questions pour lesquelles les preuves disponibles fournissent des réponses claires et pour lesquelles des recherches supplémentaires ne sont pas nécessaires. Le processus de réalisation de revues systématiques, en particulier pour les nouveaux chercheurs, est une tâche très bénéfique et utile. Les nouveaux chercheurs affinent leurs connaissances sur le domaine d'intérêt, développent de nouvelles idées de recherche et acquièrent des compétences essentielles dans la synthèse de la littérature existante [2].

1.3. Processus d'élaboration des revues systématiques

Le processus fondamental d'élaboration des revues systématiques est constitué de trois étapes essentielles : *Planification*, *Conduite* et *Rapports* [6].

Dans la phase de planification, les chercheurs doivent justifier le besoin et la portée de la revue, formuler l'ensemble des questions de recherche, développer et valider un protocole spécifiant toutes les décisions pertinentes pour la réalisation de la revue. Le protocole comprend l'identification des termes de recherche, la stratégie de recherche, les sources de littérature qui doivent être utilisées pour récupérer les articles pertinents, comment et dans quelle base les articles trouvés sont sélectionnés et inclus

dans la revue, quelles données doivent être extraites des articles sélectionnés et comment les données extraites sont synthétisées.

Dans la phase de conduite, le protocole initialement validé dans la phase de planification est exécuté. En particulier, les sources identifiées sont utilisées pour récupérer les articles ; les articles trouvés sont examinés pour leurs pertinences au sujet de recherche ; les données utiles sont ensuite extraites des articles sélectionnés puis synthétisées et classées.

Dans la phase rapport, les données extraites des articles sont analysées, les résultats sont interprétés, les questions de recherche ont été répondues et la synthèse est validée et documentée. La figure 1.1 illustre le processus global pour mener des revues systématiques tel que proposé dans [7]. L'étape que l'on veut automatiser par le présent projet est désignée en couleur grise dans la Figure.

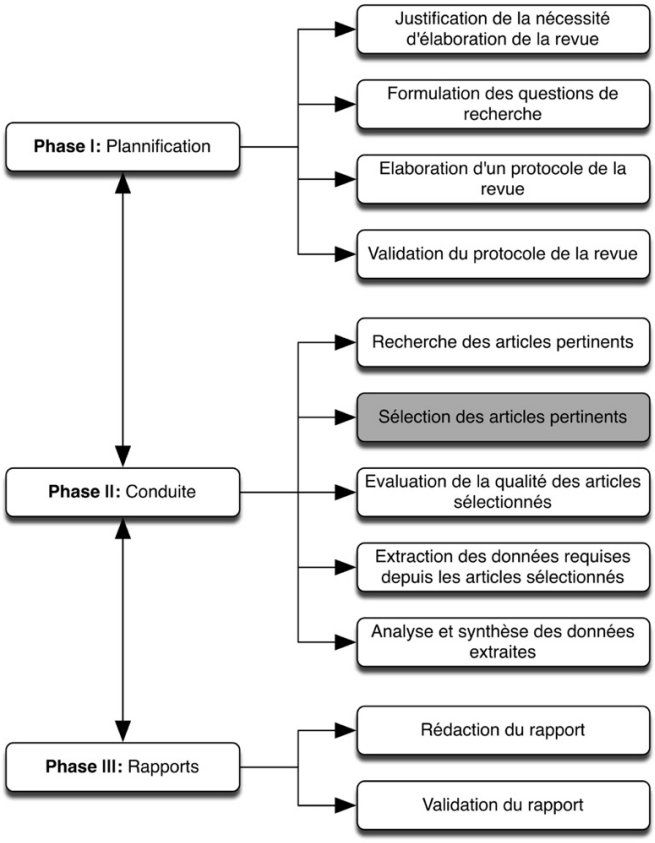

Figure 1.1. Processus général d'élaboration des revues systématiques

1.3.1. Justifier la nécessité de la revue

Avant d'investir le temps et les efforts pour mener une revue systématique, il est important de prendre en compte : (1) s'il existe d'autres revues systématiques qui ont été déjà réalisées sur le même sujet, (2) l'impact de réalisation de la revue systématique dans le sujet abordé. Une revue systématique doit être motivée pour satisfaire des objectifs académiques ou professionnels.

Une étude de Santos & da Silva [8] a révélé que les quatre principaux facteurs qui ont motivé les revues systématiques en informatique sont :

- Acquérir des connaissances sur un domaine d'études particulier,
- Identifier des recommandations pour des recherches ultérieures,
- Etablir le contexte d'un sujet ou d'un problème de recherche,
- Identifier les principales méthodologies et techniques de recherche utilisées dans un sujet ou un domaine de recherche particulier.

1.3.2. Formulation des questions de recherche

Formuler une ou plusieurs questions de recherche est l'étape la plus importante du protocole d'élaboration des revues systématiques, car c'est le début d'un fil conducteur qui relie les étapes du protocole qui se suivent entre elles. Les questions de recherche guident l'ensemble des étapes de revue, fournissant la base pour décider quelles études primaires inclure dans une revue, et donc piloter la stratégie de recherche, et décider quelles données doivent être extraites et comment les données sont synthétisées ou agrégées afin de répondre aux questions [7]. Les questions de recherche doivent être formulées de manière adéquates et précises.

Exemples des questions de recherche valides :

- *Dans quelles conditions la technique/modèle/technologie **A** est-elle plus rentable que la technique/modèle/technologie **B** ?*
- *Quels sont les risques/avantages associés au technique/modèle/technologie **A** ?*

En général, il faut déterminer une ou plusieurs questions de recherche pour une revue systématique. Pour chaque question, on fera une hypothèse qu'on veut vérifier. Grâce à une étude, l'hypothèse se confirmera en se basant sur les résultats obtenus par la revue. Une revue systématique dispose généralement d'une question de recherche principale et de plusieurs questions secondaires. La question de recherche principale est celle à laquelle la revue tentera de répondre en priorité. Les questions de recherche secondaires peuvent être inclus afin de raffiner la question principale et mieux identifier son hypothèse.

1.3.3. Stratégie de recherche des articles

L'un des principaux problèmes liés à l'élaboration des revues systématiques est la recherche et l'identification des articles pertinents. La recherche des articles scientifiques est une étape qui consiste à collecter des documents en s'appuyant sur des sources fiables. La surabondance informationnelle et la diversité des sources rend cet exercice complexe et nécessite une attitude rigoureuse et organisée. La recherche des articles appropriées peut être élaborés de différentes manières : recherche automatisée, recherche manuelle, boule de neige et la prise de contact direct avec des chercheurs clés. Une bonne stratégie de recherche utilisera une combinaison de ces méthodes, bien que dans la plupart des cas, une seule méthode est sélectionnée comme méthode de recherche principale, d'autres méthodes peuvent être appliquées par la suite pour une recherche complémentaire [7].

1.3.3.1. Recherche automatisée :

L'adoption d'une recherche automatisée nécessite l'identification des mots-clés et des sources de recherche. En plus, lors de la recherche automatique des articles sur plusieurs sources, il est inévitable d'obtenir des doublons. Ces doublons doivent être éliminés avant de passer à la phase de sélection.

Identification des mots-clés de recherche : Différentes façons peuvent être utilisées pour définir les mots-clés à utiliser pour la rechercher automatique sur les ressources électroniques [7] :

- Adopter des mots-clés en se basant sur la connaissance du domaine et l'expérience passée des chercheurs ou les revues similaires.
- Dériver des mots-clés des questions de recherche traitées dans la revue.
- Passer en revue les termes utilisés dans les résumés, mots-clés et titre de l'ensemble connu des articles. Faire correspondre les termes fréquemment utilisés avec ceux trouvés dans les questions de recherche adoptées pour la revue.

Sélection des sources de recherche : Al-Zubidy et collab. [9] ont remarqué des problèmes liés à l'utilisation de différentes bases de données numériques dans la phase de recherche, tels que : la couverture, l'accès limité et les capacités de filtrage adoptées par chaque moteur de recherche. Pour la réalisation des revues systématiques rigoureuses, il est recommandé d'utiliser plusieurs moteurs de recherche académiques. Dybå et collab. [10] ont suggéré une recherche exclusive dans ACM et IEEE Xplorer. Kuhrmann et collab. [11] ont suggéré l'utilisation de la totalité ou d'un sous-ensemble de six bibliothèques standard (*IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, Wiley Interscience* et *IET*). Haddaway et collab. [12] et Hannousse [6] ont suggéré l'utilisation des moteurs de recherche universels tels que *Google Scholar* et *Semantic Scholar* pour éviter les préjugés en faveur d'un éditeur en particulier. Utiliser une source unique et indépendante (de n'importe quel éditeur) pour rechercher des articles pertinents est une approche pratique pour l'élaboration des revues systématiques. Cela réduit les efforts et facilite la collecte et l'organisation des articles de recherche. Cependant, *Google Scholar* a été largement critiqué pour sa couverture, le nombre limité des résultats de recherche qui peuvent être consultés (1000 max.), la limite de longueur des requêtes et le manque d'un support d'exportation de méta-données directes et complètes des publications (par exemple avec un résumé et liste des mots-clés) [6].

Formulation de(s) requête(s) de recherche : Les résultats de la recherche dans les bases de données numériques dépendent directement des mots-clés de recherche fournis et il convient d'avoir suffisamment de prudence lors de la conception de ces mots-clés. De plus, certaines bases de données numériques, par exemple ACM et *Semantic Scholar* ne prend pas en charge la combinaison des domaines de recherche.

Par conséquent, les utilisateurs doivent être familiarisés avec la structure de la requête dans chaque base de données séparément. De plus, alors que certaines bases de données n'incluent pas automatiquement les synonymes dans leurs chaînes de recherche, d'autres fournissent des variations de tige automatiques assez faibles dans la recherche. En conséquence, les synonymes, les termes apparentés et l'orthographe alternative doivent être inclus manuellement [13]. Il faut aussi noter que chaque base de données numérique possède sa propre syntaxe pour spécifier les termes de recherche. Les chercheurs doivent donc prendre en considération ces différences. Le tableau 1.1 montre un exemple de formulation d'une même requête en utilisant quatre bases de données numériques différentes :

BDD	*Requête Appropriée*
ACM	(**Abstract** : network **AND** security) **AND** (**Title** : challenges **OR** issues)
IEEE	(**"Abstract"**: network security) **AND** ((**"Document Title"**: challenges) **OR** (**"Document Title"**: issues))
ScienceDirect	**Abstract**(social network security) **AND Title**(challenges **OR** issues)
Compendex	((network security) **WN Ab**) **AND** (((challenges) **WN Ti**) **OR** ((issues) **WN Ti**)))

Tableau 1.1. Exemple de formulation d'une requête de recherche

1.3.3.2. Recherche manuelle :

Un processus de recherche manuelle consiste à chercher dans des journaux et des actes de conférences spécifiques au thème de la revue systématique. Cette méthode est pratique pour trouver des articles de recherche de bonne qualité sur des sujets matures. Mais cette approche peut prendre beaucoup de temps et être onéreuse, surtout si le thème de recherche est large ou lorsque le sujet est assez mature [7]. Le problème clé de cette technique est le besoin d'identification des journaux et

conférences les plus appropriés au thème de recherche. Cette liste doit être spécifiée et la sélection doit être justifiée dans la phase de planification.

1.3.3.3. Boule de neige (Snowballing) :

La boule de neige, également appelée analyse des citations [7, 14], peut prendre l'une des deux formes. La *boule de neige en arrière* est une recherche basée sur les listes de références de l'ensemble déjà inclus des articles. Il est généralement utilisé comme méthode secondaire pour améliorer les résultats de la recherche automatisée. La *boule de neige vers l'avant* est le processus de recherche de tous les articles qui citent un article connu ou un ensemble d'articles connus. Cette approche est particulièrement utile lorsqu'il existe un petit nombre d'articles fondateurs susceptibles d'être cités par la plupart des articles ultérieurs sur le thème de la revue.

1.3.4. Sélection des articles pertinents

Une fois la recherche des articles terminée, une sélection des articles pertinents est nécessaire pour filtrer les résultats de la recherche. Le processus de sélection des articles se déroule en plusieurs étapes : une spécification claire des critères d'inclusion et d'exclusion dans la phase de planification, une application réelle de ces critères dans la phase de conduite pour le filtrage et l'élimination des articles non pertinentes. Deux types de critères doivent être spécifiés : *critères d'inclusion* et *d'exclusion*.

1.3.4.1. Critères d'inclusion/exclusion :

Alors que les critères d'inclusion indiquent tout ce qu'une étude doit avoir pour être incluse dans la revue, les critères d'exclusion indiquent les facteurs qui rendraient une étude inéligible comme incluse. Ces critères ont pour but d'identifier les articles ayant une relation directe avec les questions de recherche fixées au paravent.

Certains critères d'inclusion et d'exclusion sont assez génériques et assez faciles à interpréter. Par exemple, les critères relatifs à la date, langage et type de publication

sont fréquemment adoptés. Voici quelques exemples de critères d'inclusion et d'exclusion génériques :

- Inclusion par Date : *n'inclure que les documents publiés entre 2000 et 2021*
- Inclusion par Langage : *n'inclure que les publications écrites en anglais*
- Exclusion par type de publication : *Exclure les thèses et les rapports techniques*

D'autres critères d'inclusion et d'exclusion sont étroitement liés aux questions de recherche et seront donc formulés pour garantir l'inclusion des études susceptibles de contribuer à répondre à ces questions [7]. L'application des critères d'inclusion/exclusions dans la phase de conduite nécessitent différents niveaux de filtrage pour vérifier la pertinence des articles. Cette étape demande aux chercheurs, du temps et des efforts. En plus, lorsque plusieurs articles dupliqués d'une étude existent dans différentes versions qui apparaissent sous forme de livres, de journaux, d'article de conférence et d'atelier, il est recommandé dans [7] de n'inclure que la version la plus complète de l'étude et d'exclure les autres.

1.3.4.2. Filtrage initial :

Un filtrage initial est appliqué dans un premier temps au titre et au résumé des articles (les mots-clés peuvent également être pris en compte). Vu le nombre énorme des articles trouvés depuis la phase de recherche, plusieurs chercheurs doivent intervenir dans cette phase. Le filtrage doit donc être exécuté indépendamment par les différents chercheurs et les conflits doivent être discutés et résolus.

1.3.4.3. Filtrage approfondis :

Si une décision ne peut être prise sur certains articles, le texte intégral de ces articles doit être récupéré et examiné pour décider définitivement si l'étude répond aux critères d'éligibilité de la revue.

1.3.5. Evaluation de la qualité des articles sélectionnés

L'évaluation de la qualité de la littérature examinée est très importante pour les revues systématiques, car la qualité des conclusions dépend entièrement de la qualité des articles sélectionnée [15]. La raison d'évaluer la qualité des articles sélectionnés est d'examiner la confiance dans les résultats de l'analyse élaborés par la revue. L'évaluation de la qualité des articles est généralement effectuée en répondant à un certain nombre de questions liées aux objectifs, à la conception et à la conduite et aux résultats de chaque étude menée dans un article [7]. Un score doit être attribué à chaque réponse et un score final est calculé et attribué à chaque article. Une valeur seuil doit être sélectionné et justifié pour l'inclusion finale des articles.

1.3.6. Extraction des données

Après avoir identifié tous les articles scientifiques et études incluses dans la revue, l'étape suivante consiste à extraire et à analyser les données de ces articles. L'objectif de cette étape est d'extraire, à partir des articles scientifiques inclus, les données nécessaires pour répondre aux questions de recherche. La stratégie d'extraction des données doit être définie et justifiée dans la phase de planification. Il est recommandé d'utiliser des formulaires d'extraction de données pour aider à maintenir la cohérence (entre les études et les différents chercheurs). Bien que les formulaires aient été testés au cours de la phase de planification, il est possible qu'ils doivent être révisés au cours de la phase d'extraction de données pour les adapter au différents d'articles traités [7]. Différents types de données sont généralement extraits selon l'objectif de la revue ; toute revue doit inclure des informations générales qui enregistrent les détails de la publication pour chaque article (type, source, année, etc.). Les autres données extraites dépendent des questions de recherche adoptées dans la revue. Le tableau 1.2 montre un exemple des éléments à inclure dans un formulaire d'extraction de données.

Type de données	*Description*
Informations générales	Inclure les détails de la publication de l'article inclus tels que le journal, le titre, l'auteur, le volume, les numéros de page, etc.

Problème traité	Décrire le but de la recherche indiqué par les auteurs de l'article. Cela inclus l'identification des problèmes et sous-problèmes traités par l'article.
Contribution	Détail de l'approche/technique utilisée pour la résolution du problème adressé.
Evaluation	Décrire les techniques utilisées pour la vérification et/ou validation de la solution proposée.
Résultat	Enregistrer les résultats de l'étude et comment les mesurer
Applicabilité	Indiquer le contexte dans lequel la solution proposée est applicable et efficace.
Commentaire	Dérire les avantages, inconvénients et limites de l'approche proposée

Tableau 1.2. Modèle d'un formulaire d'extraction de données

1.3.7. Synthèse des données

Durant cette étape, les données extraites des articles et études incluses seront analysées et synthétisées pour répondre aux questions de recherche de la revue systématique. Il s'agit de rassembler les résultats des études incluses et de raconter ce que la revue systématique a trouvé. Généralement, la méthode de synthèse varie selon le degré de similitude entre les différents articles inclus. En cas d'homogénéité (les différents articles ont utilisé des méthodes de recherche similaires et adéquates), les articles peuvent être synthétisés en utilisant une *méta-analyse*. La méta-analyse utilise des méthodes statistiques pour combiner les résultats des articles. Dans le cas d'absence d'homogénéité (les articles ont utilisé diverses méthodes de recherche), il est possible d'utiliser des synthèses narratives ou descriptives pour décrire les résultats de chaque étude afin de donner une image globale de la littérature. Des formulaires peuvent aussi être utilisés pour examiner la quantité, la qualité, la cohérence, la généralisabilité et l'applicabilité des preuves présentés dans chaque article.

1.4. Outils d'élaboration des revues systématiques

La nature laborieuse du processus d'élaboration des revues systématiques a conduit au développement et à l'utilisation d'une série d'outils qui fournissent des assistances

automatisées. Un certain nombre d'étapes du processus sont sujettes à des erreurs et prennent du temps lorsqu'elles sont effectuées manuellement. L'automatisation est nécessaire pour soutenir les chercheurs dans l'élaboration des revues systématiques rigoureuse, en moins de temps et avec un minimum d'effort. Différents types d'outils sont disponibles, nous discutons dans cette section un certain nombre de ces outils avec leurs effets bénéfiques et leurs limites.

1.4.1. SLuRp

SLuRp est un outil conçu en Java et SQL pour soutenir de nombreuses étapes du processus d'élaboration des revues systématiques. Il fournit également un mécanisme de recherche automatique des articles depuis certaines bases de données numériques en ligne en utilisant des termes prédéfinis. Il offre aussi la possibilité de travail en équipe avec un mécanisme de gestion des désagréments entre les participants. Il permet aux participants de sauvegarder et consulter les différents critères adoptés dans le protocole, les décisions de chaque participant et la liste des données extraites de chaque article [16]. SLuRp se distingue au niveau de la visualisation des résultats synthétisés. Il permet de générer automatiquement le rapport de la revue en LaTeX. Par contre, il n'assure pas la traçabilité pour identifier la source des données extraites.

1.4.2. StArt

StArt est un outil en constante évolution ; de nouvelles fonctionnalités peuvent souvent être intégrées pour améliorer la prise en charge du processus [17]. Il permet de décrire, sauvegarder, consulter et mettre à jour les différentes étapes du protocole. Il n'offre pas un mécanisme de recherche automatique, mais il permet d'importer les résultats de recherche en format *Bibtex*. Les articles importés peuvent être automatiquement ordonnés selon un score associé en fonction des fréquences des termes de recherche dans le titre et le résumé de chaque article. Comme SLuRp, il permet de visualiser et de synthétiser les données extraites et de générer un rapport final sous format *Excel*.

1.4.3. SLR-Tool

SLR-Tool a la capacité de stocker les données liées à chacune des activités du processus d'élaboration des revues systématiques [18]. Il permet d'affiner les recherches en utilisant des techniques de *Text Mining*. Il permet la définition d'un schéma de classification qui aide les chercheurs à effectuer la synthèse et l'analyse des données. Il utilise des techniques d'exploration de texte pour regrouper les articles par le biais des techniques d'apprentissage automatique (spécifiquement *clustering*) en utilisant le degré de similitudes entre eux. Cela peut aider les chercheurs à l'inclusion ou l'exclusion de certains articles. Toutes les données collectées peuvent être exportées vers des fichiers Excel sous forme de tableaux ou de graphiques. Une des limites remarquables du SLR-Tool est qu'il ne permet pas le travail en équipe alors que c'est l'une des principales exigences du processus d'élaboration des revues systématiques.

1.4.4. SLRTOOL

SLRTOOL est un outil conçu pour prendre en charge le processus d'élaboration des revues systématiques de différentes disciplines. Il considère une revue comme un projet et il offre différentes fenêtres de dialogues pour intégrer les questions de recherche, les critères d'inclusions et d'exclusions etc. Il permet la recherche automatique et l'extraction des méta-données des articles depuis *Google Scholar* [19]. SLRTOOL ne permet pas l'identification et la sélection automatique des articles pertinents. En plus, les articles supplémentaires doivent être insérés un par un, ce qui le rend non adapté aux revues systématiques avec un grand nombre d'articles à traiter.

1.4.5. Sysrev

Sysrev est un outil en ligne qui permet de créer des revues systématiques sous fromes des projets [W1]. Il permet la recherche automatique des articles et l'importation de leurs méta-données depuis PubMed[1]. Le créateur du projet peut inviter des

[1] PubMed : https://pubmed.ncbi.nlm.nih.gov/

collaborateurs, analyser les articles et exporter les résultats de l'analyse. La plateforme gratuite ne prend en charge que les projets en libre accès ou publics. Sysrev utilise l'apprentissage automatique pour avoir une idée sur la possibilité d'inclusion et d'exclusion des articles. Le modèle d'apprentissage intégré se base sur les expériences menées par les autres revues systématiques élaborés précédemment sur la plateforme.

1.4.6. SWIFT-Active Screener

SWIFT-Active Screener est une application Web qui aide à la réalisation des revues systématique de maniérer collaborative. L'outil est spécifiquement conçu pour aider la sélection des articles pertinents à une revue systématique. SWIFT-Active Screener réordonne la liste des articles trouvés selon leurs pertinence à la revue. Pour cela, il utilise l'apprentissage actif, un type d'apprentissage automatique qui analyse les décisions des chercheurs vis-à-vis des articles déjà traités. Pendant la sélection, un modèle binomial négatif est utilisé pour estimer le nombre d'articles pertinents restant dans la liste des articles non traités [20].

1.4.7. DoCTER

DocTER est une application Web gratuite qui utilise les techniques d'apprentissage automatique pour classer les articles scientifiques par ordre de priorité. DocTER utilise une technique de *clustering* basée sur l'analyse des résumés des articles trouvés. DoCTER adopte une approche semi-automatique ; il attribue une référence à chaque cluster, fournissant une sorte de sujet (ensemble de mots-clés). Les participants à la revue ont la possibilité d'examiner les mots-clés et attribuer des niveaux de priorité à chaque cluster [W3].

1.4.8. Buhos

Buhos est une application Web développée en Ruby pour gérer le processus complet d'élaboration des revues systématiques [21]. Il offre des fonctionnalités pour soutenir la description du protocole adopté, la recherche automatique des articles depuis

plusieurs bases de données numériques, la gestion de désagréments de sélection des articles par les différents collaborateurs. Buhos peut être utilisé localement via un serveur Web interne ou en ligne. Le problème majeur de cet outil est que le processus a été conçu pour être réalisé de manière séquentielle ce qui ne reflète pas le processus réel d'élaboration des revues systématiques proposé par Kitchenham et collab. [7].

1.4.9. SRA

SRA est un logiciel gratuit développé par l'université de Bond. SRA a été conçu dans le but de réduire le temps nécessaire à la construction des revues systématiques dans le domaine de la médecine à l'aide des technologies de l'information. SRA permet d'importer les méta-données des articles trouvés en format EndNote, analyser le contenu des articles inclus pour aider à déterminer les mots à utiliser afin de construire une requête de recherche raffinée. Il peut adapter la requête de recherche pour les différentes bases de données numériques. Il permet l'élimination automatiques des articles dupliquées et fournit une interface de lecture rapide pour classer rapidement les articles trouvés. Il permet aussi à plusieurs chercheurs de contribuer dans la sélection des articles. SRA est un logiciel libre construit en utilisant le langage de programmation PHP et CodeIgniter et le stockage de bases de données MySQL [W2].

Outil	*Recherche automatisée*	*Sélection des articles*	*Évaluation de qualité*	*Extraction de données*	*Synthèse des données*
SLuRp	⊙	O	O	O	●
StArt	O	⊙	O	O	●
SLR-Tool	O	⊙	O	O	●
SLRTOOL	⊙	O	O	O	●
Sysrev	⊙	⊙	O	⊙	●
SWIFT-Active Screener	O	⊙	O	O	O

DoCTER	○	◉	○	○	○
Buhos	●	○	◉	○	○
SRA	○	○	○	○	●

Tableau 1.3. Outils disponibles pour l'automatisation du processus d'élaboration des revues systématiques

Le tableau 1.3 montre la liste des outils discutés précédemment avec leurs degrés d'automatisation de chaque étape du processus d'élaboration des revues systématiques. Le cas d'une automatisation parfaite est désigné, dans le tableau, par une cercle rempli (●), une automatisation partielle est désignée par une cercle avec un point au centre (◉), une étape non automatisée est désignée par une cercle vide (○). D'après l'analyse du tableau, nous constatons que la sélection des articles n'a jamais été automatisé parfaitement. Certains outils offrent la possibilité de classer les articles selon leurs ordres de pertinence aux mots-clés de recherche (cas de StArt), d'autre utilisent les techniques d'apprentissage automatique pour la classification des documents (cas de SLR-Tool et DocTER). DoCTER par exemple permet de choisir un des algorithmes de classification connus (e.g. ; K-Means ou LDA, voir Chapitre 2 pour plus de détails). Certains outils sont dédiés à des domaines de recherche spécifiques. DoCTER, Sysrev, SWIFT-Active Screener et SRA sont uniquement testés et appliqués au domaine de la médecine. Peu d'outils sont consacrés à l'automatisation du processus d'élaboration des revues systématiques dans le domaine informatique.

1.5. Conclusion

En conclusion, les revues systématiques représentent une méthodologie qui permet de produire une synthèse rigoureuse des données probantes disponibles sur un sujet bien précis, malgré son importance, la réalisation des revues systématiques demande un investissement conséquent en termes de temps et effort de travail en particulier lorsque c'est la première fois que la question de recherche est formulée. Différentes tentatives ont été réalisées pour l'automatisation du processus d'élaboration des revues

systématique. Aucun des outils existants n'offre une automatisation complète et efficace du processus, notamment de la phase cruciale de sélection des articles pertinents, ce qui motive l'objectif du projet réalisé dans ce mémoire.

CHAPITRE II

L'APPRENTISSAGE AUTOMATIQUE

CHAPITRE II.
L'APPRENTISSAGE AUTOMATIQUE

L'apprentissage automatique est une branche de l'intelligence artificielle (IA) qui se concentre sur le développement et l'utilisation des algorithmes qui apprennent à partir des expériences passées et des données enregistrées. Ces algorithmes améliorent leurs précisions au fil du temps sans être programmées pour le faire [W4]. Les algorithmes d'apprentissage automatique sont *entraînés* sur des données disponibles afin de faire des prédictions sur de nouvelles données. Plus l'algorithme est bien entrainé (i.e. ; quantité et qualité des données utilisées en phase d'apprentissage), plus les prédictions deviendront précises. L'objectif principal est de permettre aux machines d'apprendre automatiquement sans l'intervention humaine et d'ajuster ses actions en conséquence. Dans ce chapitre, nous présentons le principe de l'apprentissage automatique et ses différents types en se focalisant sur l'apprentissage non-supervisé et semi-supervisé.

2.1. Historique de l'apprentissage automatique

L'apprentissage automatique est initialement fondé sur le principe des réseaux de neurones artificiels. L'histoire des réseaux de neurones remonte à 1943, lorsque le neurophysiologiste Warren McCulloch et le mathématicien Walter Pitts ont publié un article décrivant une analogie entre les relais téléphoniques et la connexion des cellules neuronales du cerveau. Warren McCulloch et Walter Pitts ont décidé alors de créer un circuit électrique qui mime le fonctionnement des neurones biologique, et c'est ainsi que le réseau de neurones artificiel est né [W5].

En 1958, Frank Rosenblatt a conçu le premier réseau de neurones (appelé *Perceptron*) pour la reconnaissance des formes par les ordinateurs. Un autre exemple extrêmement précoce d'un réseau de neurones est venu en 1959, lorsque Bernard Widrow et Marcian Hoff ont créé deux modèles à l'Université de Stanford. Le premier s'appelait ADELINE

et pouvait détecter des modèles binaires (e.g. ; dans un flux de bits, il pourrait prédire ce que serait le bit suivant). Le second modèle s'appelait MADELINE et pouvait éliminer l'écho sur les lignes téléphoniques, il avait donc une application utile dans le monde réel. Malgré le succès de MADELINE, il n'y a pas eu beaucoup de progrès dans le domaine jusqu'à la fin des années 1970 pour de nombreuses raisons, principalement la popularité de l'architecture Von Neumann [W5, W6].

Gerald DeJonge en 1981 a introduit le concept d'apprentissage dans lequel un ordinateur analyse les données et crée des règles pour éliminer les informations inutiles [W6]. L'année qui suit a marqué l'intérêt aux développements des réseaux de neurones, lorsque John Hopfield a suggéré de créer un réseau doté de lignes bidirectionnelles, similaires au fonctionnement réel des neurones [W5].

En 1998, les recherches des laboratoires AT&T Bell ont abouti à une bonne précision dans la détection des codes postaux manuscrits du service postal américain. Depuis 2000, de nombreuses entreprises ont réalisé que l'apprentissage automatique augmenterait le potentiel de calcul. C'est pourquoi plus de recherches ont été faites, afin de garder une longueur d'avance sur la concurrence [W5].

2.2. Applications de l'apprentissage automatique

L'apprentissage automatique est particulièrement présente et appliquée dans différents domaines. Les applications classiques incluent la reconnaissance des écritures, de visages et de paroles et la traduction automatiques des textes. Les applications récentes incluent la cyber-sécurité, la recommandation de produits et les véhicules autonomes. Les applications de l'apprentissage automatique sont multiples et peuvent perfectionner sensiblement différents domaines. En fonction de la nature des données, de la masse à traiter et de l'utilisation des informations obtenues, le choix d'appliquer un tel type d'apprentissage automatique va pouvoir varier. Quoi qu'il en soit, l'apprentissage automatique dispose donc d'un véritable potentiel et peut permettre à de nombreux domaines de s'améliorer. Pour une vision plus large sur l'applicabilité de l'apprentissage automatique sur les différents domaines, la figure 2.1. montre l'ensemble des applications où l'apprentissage automatique a marqué son succès depuis son apparence en 1981.

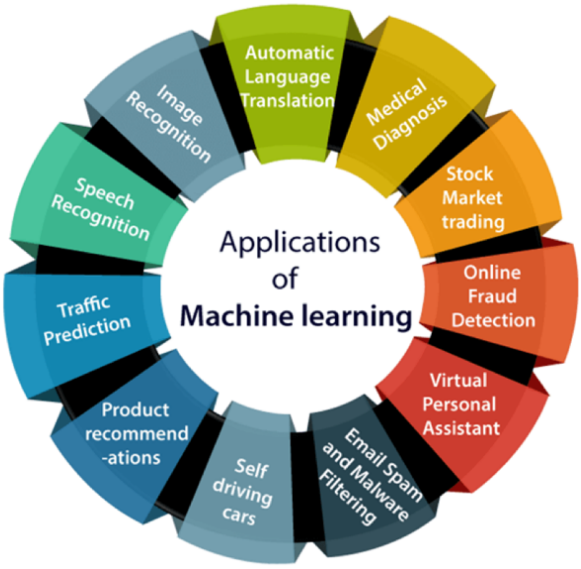

Figure 2.1. Applications concrètes de l'apprentissage automatique

2.3. Le processus d'apprentissage automatique

La conception et le développement d'un modèle d'apprentissage automatique pour résoudre un problème donné est un processus de plusieurs étapes. Le processus implique deux phases fondamentales (voir la figure 2.2.): la *préparation des données* et la *conduite et l'évaluation du modèle d'apprentissage*. Chaque phase comporte un ensemble d'étapes qui peuvent être exécutées de manière séquentielle ou en parallèle. Ces étapes sont expliquées en détails dans les sous sections qui suivent.

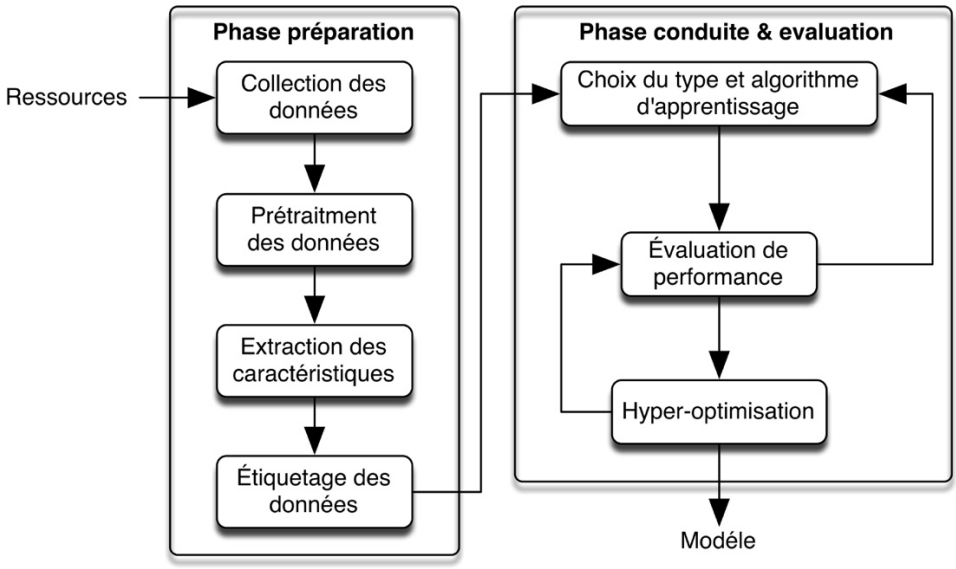

Figure 2.2. Processus général de l'apprentissage automatique

2.3.1. Collection des données

La collecte des données est le fondement du processus d'apprentissage automatique. L'apprentissage sur des données similaires ou des données de types particuliers peuvent rendre le modèle complètement inefficace pour d'autres types de données. C'est pourquoi il est impératif que les considérations nécessaires soient prises lors de la collecte des données, car les erreurs commises à cette étape ne feront que s'amplifier au fur et à mesure pendant la progression vers les dernières étapes [W5].

2.3.2. Prétraitement des données

Le prétraitement des données consiste à nettoyer et à préparer les données pour la phase d'apprentissage. Cela comprend le filtrage (e.g.; suppression des bruits), le formatage (e.g.; utiliser le même codage), la normalisation (e.g.; garder la même taille pour tous les individus) et le traitement des données manquantes (e.g. ; récupération des informations manquantes ou élimination des individus avec des informations insuffisantes). Cette étape est la plus délicate, 80% des efforts pour la réalisation des modèles d'apprentissage efficaces est généralement consacrée au prétraitement des données [W7]. Le prétraitement des données est un moyen d'assurer que les données utilisées pour l'apprentissage d'un algorithme sont exactes, complètes et pertinentes. D'un côté, l'utilisation des données incomplètes ou brutes pour l'apprentissage peut entraîner plusieurs erreurs, ce qui entraînera en fin de compte une précision globale beaucoup plus faible. D'un autre côté, des données bien conçues peuvent améliorer l'efficacité du modèle. Il est donc judicieux d'examiner l'ensemble des données utilisées afin qu'elles puissent produire des résultats meilleurs et significatifs.

2.3.3. Extraction des caractéristiques

L'extraction des caractéristiques consiste à transformer des données brutes en un ensemble de caractéristiques qui représentent mieux le problème sous-jacent des modèles prédictifs, ce qui améliore la précision du modèle sur des nouvelles données. En d'autres termes, l'extraction des caractéristiques permet de transférer les données collectées en des éléments que l'algorithme d'apprentissage peut comprendre. Les connaissances des domaines sont fréquemment utilisées pour en tirer le meilleur parti.

Si cette étape est effectuée correctement, la puissance prédictive des algorithmes d'apprentissage automatique en sera forcément améliorée.

2.3.4. Étiquetage des données

L'étiquetage des données est un élément clé de la préparation des données pour l'apprentissage automatique, car il spécifie les décisions à partir desquelles le modèle apprendra. Une étiquette est le résultat (i.e. ; réponse de modèle) souhaité pour chaque individu de l'ensemble de données utilisé dans la phase d'apprentissage. Bien que certains types d'apprentissage ne nécessitent pas de données étiquetées (voir la section 2.3.5), de nombreux systèmes d'apprentissage automatique s'appuient toujours sur des données étiquetées pour apprendre et exécuter les tâches qui leur sont données.

2.3.5. Choix du type et algorithme d'apprentissage

Les algorithmes utilisés dans l'apprentissage automatique se divisent en trois types de bases (voir Figure 2.3.) : *apprentissage supervisé*, *non-supervisé* et *semi-supervisé*. Cette classification est basée sur le degré de disponibilité des données étiquetées.

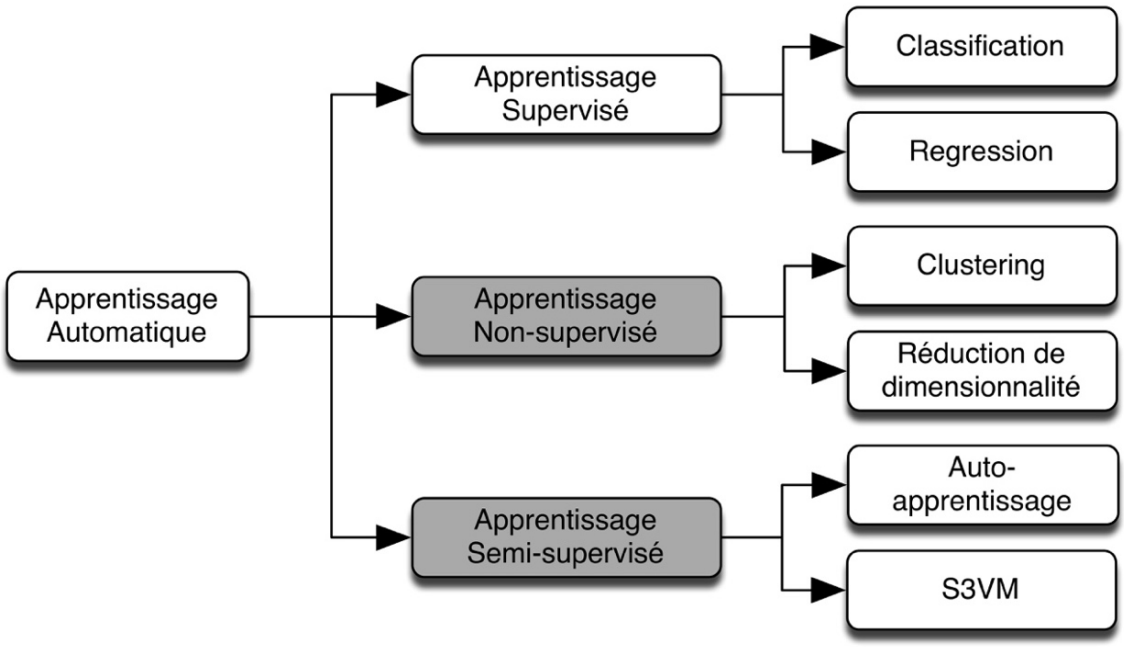

Figure 2.3. Taxonomie des techniques d'apprentissage automatique

2.3.5.1. Apprentissage supervisé :

L'apprentissage supervisé consiste à entrainer les algorithmes exclusivement sur des données étiquetées (i.e.; avec des résultats connues pour chaque individu de l'ensemble de données utilisée en apprentissage) pour réaliser des prédictions sur des nouveaux individus non-étiquetés. Pendant la phase d'apprentissage, l'algorithme supervisé cherche à apprendre les relations entre les données en entrées et les étiquettes qui leur correspondent, afin de déduire des règles d'apprentissage. Ces règles sont par la suite utilisées pour prédire les étiquettes pour un nouveau jeu de données. Le principal avantage de l'apprentissage supervisé est qu'il nous permet de recueillir des informations ou de produire un rendement d'informations à partir des expériences passées. Deux techniques de prédictions sont utilisées dans l'apprentissage supervisé : la *classification* et la *régression*.

A. La Classification :

Les algorithmes de classification prévoient des valeurs discrètes [W8]. Ces algorithmes visent à classer les données en entrée en deux ou plusieurs catégories. Les algorithmes de classification couramment utilisés sont : les machines à vecteurs de support (SVM), naïve bayésienne (NB) et les k plus proches voisins (KNN).

B. La Régression :

Les algorithmes de régression prévoient des valeurs continues [W8]. Les algorithmes à base de régression sont donc utilisés si les réponses à prédire sont des valeurs réelles. Les algorithmes de régression couramment utilisés sont : arbre de décisions (DT), forêt d'arbres décisionnels (RF) et réseaux de neurones (ANN).

2.3.5.2. Apprentissage Non-supervisé :

L'étiquetage des données nécessite généralement l'intervention des experts humains. Dans certains domaines, cette opération peut devenir difficile voire fastidieuse ou impossible lorsque le nombre des données est important. L'apprentissage non supervisé prend en charge le problème en apprenant des données non-étiquetées ce qui réduit le risque d'erreur humaine et la disponibilité des experts.

Le principe de l'apprentissage automatique non-supervisé est de regrouper les individus de l'ensemble de données sans aucune connaissance préalable de ses

étiquettes. L'apprentissage non-supervisé alors vise à découvrir les structures sous-jacentes et intrinsèques dans les données d'entrées pour savoir prédire sur des nouvelles données [W9]. L'apprentissage non-supervisé est relativement simple et rapide à faire. Les techniques les plus répondus de ce type d'apprentissage sont : la technique de *clustering* et la *réduction de dimensionnalités*.

A. Le Clustering :

Le clustering est une technique qui divise l'ensemble de données en entrées, en un certain nombre de groupes appelés *clusters* afin que les individus de l'ensemble de données appartenant à un seul cluster aient des caractéristiques similaires. Un cluster n'est rien d'autre qu'un regroupement des individus, la distance entre les individus au sein d'un même cluster est donc minimale. Les algorithmes couramment utilisés pour le clustering sont: les K-moyennes (K-Means) et le clustering hiérarchique (Hierarchical Clustering). Dans ce qui suit nous détaillons l'algorithme K-moyennes, utilisé pour la réalisation du présent projet.

K-moyennes (K-Means) :

Le but de l'algorithme K-moyennes est de trouver K groupes où les individus de l'ensemble de données peuvent être classés. L'algorithme attribue de manière itérative chaque individu à l'un des K groupes en appliquant une mesure de distance fournie. Les individus de l'ensemble de données sont donc regroupés selon la similitude de leurs caractéristiques. La mesure de distance détermine la similitude entre deux individus et affecte la forme du cluster. La mesures de distance la plus adoptée est la *distance euclidienne* qui est définie selon la formule suivante :

$$d_2(x_i, x_j) = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2}$$

Où $x_i = (x_{i,1}, \dots, x_{i,d})$ et $x_j = (x_{j,1}, \dots, x_{j,d})$ sont deux vecteurs caractéristiques de dimension d représentant deux individus : x_i et x_j de l'ensemble de données respectivement. Les résultats de l'algorithme K-moyennes sont :

1. K valeurs désignant les centres de gravité de chaque cluster ; ces valeurs sont utilisées par la suite pour étiqueter de nouveaux individus.
2. Étiquettes pour les données d'apprentissage où chaque élément de l'ensemble de données en entrées est attribué à un seul groupe (cluster).

Les étapes suivantes illustrent le fonctionnement de l'algorithme K-moyennes:

E1 : Sélectionner le nombre K pour déterminer le nombre souhaité de clusters.

E2 : *Initialisation* : Sélectionnez K centres de gravité (points moyens) de manière aléatoire.

E3 : *Assignation* : Associer chaque élément de l'ensemble de données à un centre de gravité le plus proche en appliquant une mesure de distance.

E4 : *Calcul de points moyens* : Mettre à jour le centre de gravité de chaque cluster.

E5 : Répéter **E3** et **E4** jusqu'à ce qu'une nouvelle réaffectation ne se produit.

La figure 2.4. montre un exemple démonstratif expliquant les différents étapes de l'algorithme K-moyennes.

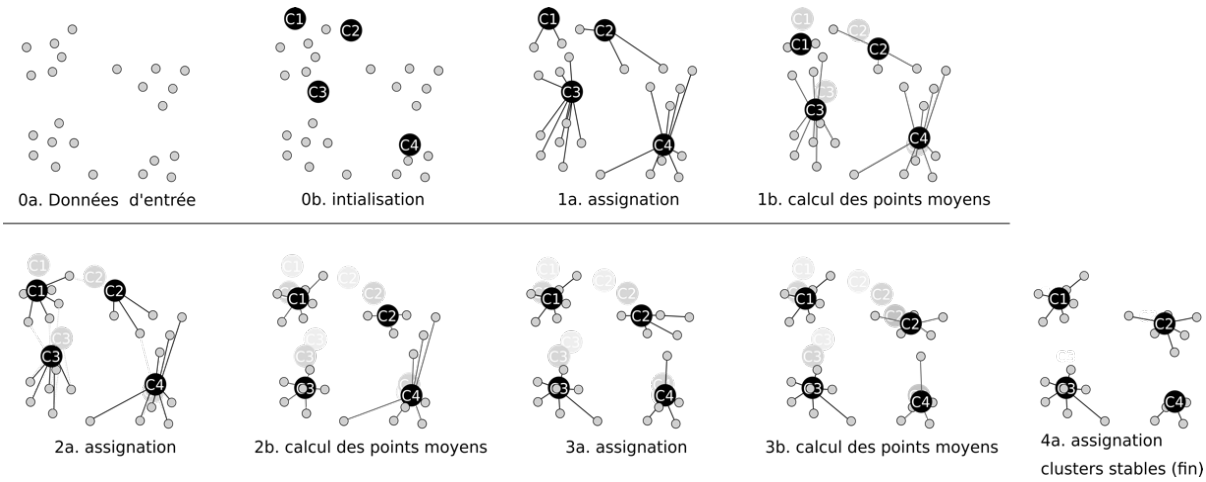

Figure 2.4. Exemple sur le clustering K-moyennes [W10]

L'algorithme K-moyennes a plusieurs avantages et inconvénients [W11]:

+ Il est facile d'implémenter.
+ Il s'adapte mieux aux divers changements de données.
+ Permet le regroupement des grandes masses de données.

+ Les résultats sont très faciles à interpréter.
+ Il est rapide et efficace en termes de coûts de calcul.
– Il donne des résultats variables sur différentes exécutions avec des partitionnements initiaux différents, ce qui entraîne une incohérence.
– Sensible au bruit et aux anomalies.

B. Réduction de dimensionnalité :

La réduction de dimensionnalité est une technique d'apprentissage non-supervisé qui vise à réduire le nombre de caractéristiques utilisées pour la catégorisation des individus de l'ensemble de données. En effet, cela permet d'une part, de réduire l'impact des caractéristiques non pertinentes qui peuvent induire l'algorithme d'apprentissage en erreur. D'autre part, cela permet de limiter le nombre de possibilités à tester, ce qui permet de doubler la vitesse de l'algorithme d'apprentissage. Pour cet effet, les caractéristiques sont généralement combinées afin d'obtenir un plus petit nombre de nouvelles caractéristiques symboliques plus expressives et/ou moins redondantes.

Parmi les algorithmes couramment utilisés pour la réduction des dimensionnalités, on peut citer : l'analyse en composants principales (PCA), l'analyse sémantique latente (LSA) et l'allocation de dirichlet latente (LDA). Dans ce qui suit, nous présentons la technique LDA qui est fréquemment utilisée pour la classification des documents et qui est adoptée pour la réalisation de notre projet.

Allocation de Dirichlet latente (LDA) :

L'algorithme d'allocation de dirichlet latente (LDA) est un algorithme d'apprentissage non-supervisé. Il s'agit probablement de la technique de modélisation thématique la plus connue. En LDA, chaque document est vu comme un vecteur de probabilité d'appartenance à une thématique (i.e. ; cluster) où une thématique est définie comme un ensemble de mots sous-jacents. Le but de LDA est de découvrir les thématiques auxquelles appartient un document, à la lumière des mots qu'il contient. Les caractéristiques sont la présence (ou le nombre d'occurrences) de chaque mot dans un document et les catégories sont les thématiques découvertes dans la phase d'apprentissage. Le résultat du processus d'apprentissage de l'algorithme LDA sont :

1. Un vecteur de probabilité d'association de chaque document à une thématique.
2. Ensemble des termes associés à chaque thématique.

Les étapes suivantes illustrent le fonctionnement de l'algorithme LDA [21]:

E1 : Extraire l'ensemble des mots appartenant à chaque document dans l'ensemble de données.

E2 : Déterminer les mots qui appartiennent à une thématique ou la probabilité que des mots appartiennent à une thématique par:

E2.1. : Parcourir chaque document et attribuer au hasard chaque mot du document à l'une des k thématiques (k est choisi au préalable).

E2.2. : Pour chaque document *d*, parcourir chaque mot *w* et calculer :

1. *p(t | d) :* La proportion de mots dans le document *d* qui sont affectés à la thématique *t*.
2. *p(w | t) :* La proportion d'affectations à la thématique *t* sur tous les documents issus de ce mot *w*.

E3. : Mettre à jour la probabilité que le mot *w* appartienne à la thématique *t* par la valeur : *p(t | d)* x *p(w | t)*

E4. : Répéter **E2** et **E3** un certain nombre d'itérations suffisant pour que les assignations se stabilisent.

L'algorithme LDA a plusieurs avantages et inconvénients [22]:

+ Sa complexité n'augmente pas avec le nombre de documents ; autrement dit le nombre de paramètres n'augmente pas quand on ajoute des documents au corpus ce qui rend le modèle moins sensible au problème d'*overfitting*, récurrent en Fouille de Données
+ Il est plus complet, au sens où tous les paramètres ont une loi générative.
− Le principal inconvénient du LDA est la difficulté d'estimation des paramètres.

2 .3.5.3. Apprentissage semi-supervisé :

L'apprentissage semi-supervisé est le type d'apprentissage automatique qui utilise une quantité limitée de données étiquetées et d'un grand nombre de données non étiquetées pour l'apprentissage des algorithmes. L'apprentissage semi-supervisé offre un bon compromis entre les deux apprentissages : supervisé et non-supervisé [23].

Deux techniques très utilisées pour l'apprentissage semi-supervisé sont : *auto-apprentissage* (self-training) et *Séparateur semi-supervisé à vaste marge* (S3VM).

A. Auto-apprentissage (Self-training):

Suivant la technique de l'auto-apprentissage, l'algorithme est d'abord entraîné avec l'ensemble des individus étiquetés disponibles. L'algorithme est, ensuite, utilisé pour étiqueter les autres individus. Les nouveaux individus étiquetés avec un haut degré de confiance sont ajoutés aux données d'apprentissage. L'algorithme est ré-entrainé sur le nouvel ensemble de données obtenu et la procédure est répétée jusqu'à satisfaire un critère d'arrêt. Différentes implémentations existent dans la littérature.

Les avantages et inconvénients de l'auto-apprentissage sont [24]:

+ Une méthode d'apprentissage très simple.
+ Une méthode qui s'applique aux algorithmes d'apprentissages connues.
+ Souvent utilisé dans des tâches réelles comme le traitement du langage naturel.
– Les premières erreurs pourraient se renforcer.

B. Séparateur semi-supervisé à vaste marge (S3VM) :

S3VM est une technique largement adoptée pour entraîner de façon semi-supervisée les machines à vecteurs de support (SVMs). Suivant la technique S3VM, tout étiquetage possible pour l'ensemble des individus non-étiquetés est d'abord énuméré. Par la suite, un ensemble de SVMs standards est entrainé chacun sur un étiquetage possible avec l'ensemble des individus originalement étiquetés. Finalement, l'étiquetage qui produit le SVM le plus performant est adopté [25].

2.3.6. Évaluation de performance

Après la phase d'apprentissage, un modèle est créé. Il est nécessaire de vérifier le bon fonctionnement et la généralisation de ce modèle. L'évaluation de la prédiction d'un modèle avec les mêmes données qui ont été utilisées pour l'apprentissage n'est pas utile. Pour évaluer correctement un modèle, il doit être testé sur des données qui ne faisaient pas partie des données d'apprentissage. Les résultats de prédiction doivent être comparés aux valeurs des résultats connues.

2.3.6.1. Méthodes de validation

Pour valider correctement les modèles d'apprentissage, deux méthodes de validation sont utilisées : *échantillonnage* (Sampling) et la *validation croisée* (cross-validation).

A. Échantillonnage (Sampling) :

L'échantillonnage consiste à diviser l'ensemble collecté de données en deux parties : une partie pour l'apprentissage et l'autre pour le test. Différentes techniques d'échantillonnages sont utilisées selon la nature et la taille de l'ensemble de données : *aléatoire*, *rejet* et *préférentielle*. La figure 2.5. montre un exemple simple d'un échantillonnage où l'ensemble de données de 16 individus est divisé en deux parties égales ; une pour l'apprentissage (8 individus) et une autre pour le test (8 individus).

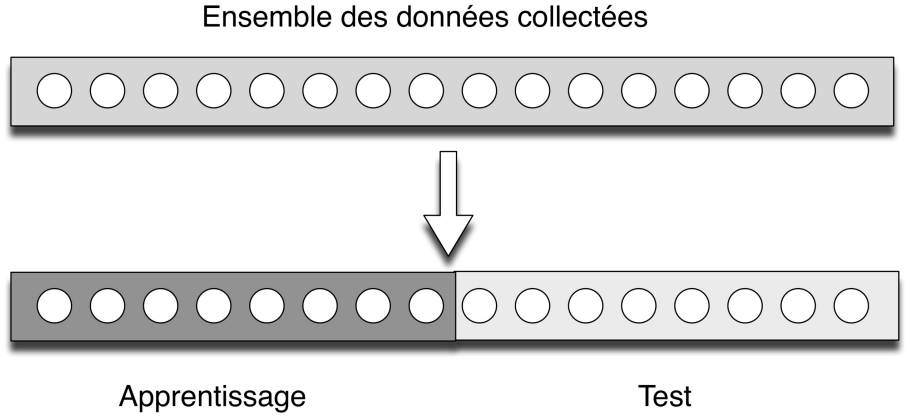

Figure 2.5. Exemple sur l'échantillonnage

B. Validation croisée (cross-validation) :

La cross-validation consiste simplement à diviser l'ensemble de données collectée en k échantillons. Un des k échantillons est sélectionné pour validation alors que les k-1 autres échantillons sont utilisés pour l'apprentissage. Le processus de validation se répète k fois, en sélectionnant à chaque fois un échantillon différent pour la validation. En effet, à chaque fois, un modèle différent est généré et sa performance est mesurée est sauvegardée. La moyenne et l'écart type des k scores de performances peuvent être calculés pour estimer le biais et la variance de la performance de validation [26]. La figure 2.6. montre un exemple sur la validation croisée à 5 parties. L'ensemble de données est divisé en 5 parties égales. Les individus pour la partie du test sont désignés

par des jetons blancs alors que les individus des parties d'apprentissage sont désignés par des jetons noirs. Dans le cas où le nombre des individus n'est pas divisible par le nombre des parties souhaité, la dernière partie doit inclure le reste de la division.

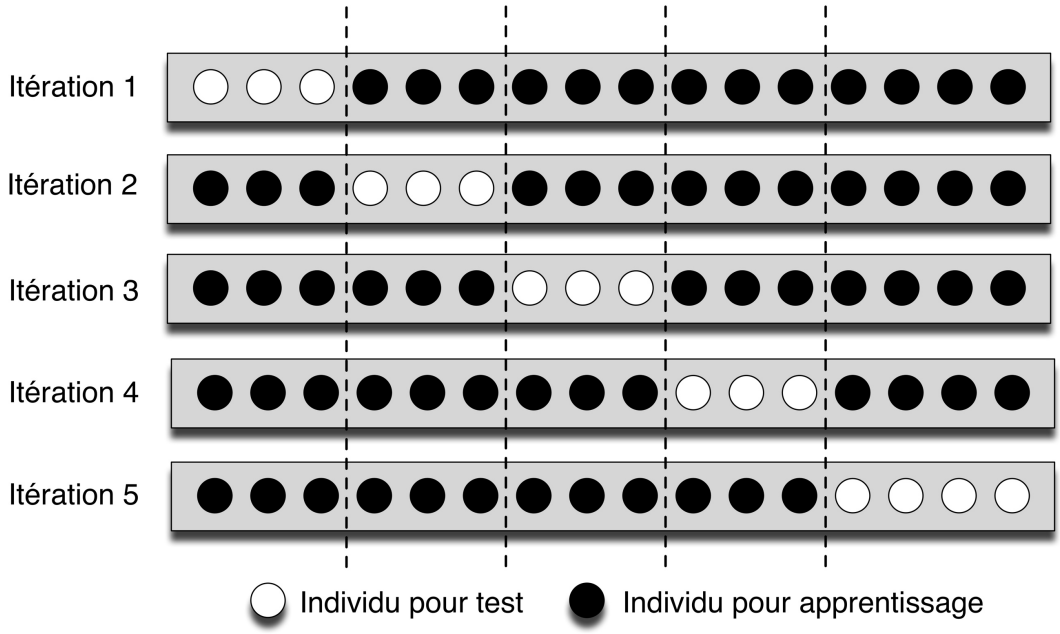

Figure 2.6. Exemple sur la validation croisée

2.3.6.2. Mesures de performance

Les mesures de performance sont des métriques récapitulatives indiquant la qualité de correspondance entre les valeurs prévues et les valeurs obtenues par le modèle. Toutes ces mesures sont basées sur la matrice de confusion décrites dans le tableau 2.1 pour une classification binaire :

		Prédiction	
		C1	**C2**
Valeurs réelles	**C1**	Vrai positif (VP)	Faux négatif (FN)
	C2	Faux positif (FP)	Vrai négatif (VN)

Tableau 2.1. Matrice de confusion.

D'après la matrice de confusion, le vrai positif (VP) indique le nombre des individus de l'ensemble de validation qui sont correctement classés en C1, contrairement au faux

négatif (FN) qui indique le nombre des individus de C1 qui sont mal classés en C2. Un vrai négatif (VN) montre le nombre des individus qui sont correctement classés en C2 alors que le faux positif (FP) indique le nombre des individus de C2 qui sont mal classés en C1. Voici la liste des mesures généralement adoptées pour la comparaison et validation des modèles :

1. **Rappel :** C'est la proportion des individus de la classe C_i qui ont été effectivement identifiées par le modèle.

$$Rappel_{C_i} = \frac{VP_{C_i}}{VP_{C_i} + FN_{C_i}}$$

2. **Précision :** C'est la proportion des individus de C_i qui ont était effectivement identifiées correctement par le modèle.

$$Précision_{C_i} = \frac{VP_{C_i}}{VP_{C_i} + FP_{C_i}}$$

4. **F1-score :** C'est la moyenne harmonique entre la précision et le rappel. Par conséquent, ce score prend en compte à la fois les faux positifs et les faux négatifs. F1-Score est essentiellement utile surtout si la répartition des classes est inégale.

$$F1 - Score_{C_i} = 2 * \frac{Precision_{C_i} * Rappel_{C_i}}{precision_{C_i} + Rappel_{C_i}}$$

2.3.7. Hyper-optimisation

Chaque algorithme d'apprentissage automatique comporte un ou plusieurs paramètres. Ces paramètres contrôlent la précision du modèle. Par conséquent, les valeurs de ces hyper-paramètres sont particulièrement importants pour améliorer la performance des modèles. L'hyper-optimisation est le processus de sélection d'hyper-paramètres optimaux pour le modèle conçu. Il est souvent recommandé d'ajuster ces hyper-paramètres selon la nature du problème étudié et l'ensemble de données utilisé

pour l'apprentissage. Deux techniques de bases sont utilisées pour ce faire : la *recherche de grille* et la *recherche aléatoire*.

2.3.7.1. La recherche de grille :

La recherche de grille fonctionne en essayant chaque combinaison possible de valeurs de paramètres que nous voulons essayer pour un modèle. La recherche de grille est effectuée de manière automatique mais peut devenir coûteuse en termes de calcul si le nombre des valeurs à explorer est important.

2.3.7.2. Recherche aléatoire :

La méthode de recherche aléatoire consiste à utiliser des valeurs des hyper-paramètres sélectionnées de manière aléatoires pour obtenir la meilleure solution pour un modèle. L'inconvénient de la recherche aléatoire est qu'elle peut dans certains cas manquer de valeurs significatives dans l'espace de recherche.

2.4. Conclusion :

Pour conclure, l'apprentissage automatique se développe rapidement dans le domaine de l'intelligence artificielle. L'apprentissage automatique a monté son efficacité pour résoudre des problèmes trop complexes pratiquement dans tous les domaines évolutifs. Pour les systèmes complexes, différentes techniques d'apprentissage automatique peuvent être combinés pour avoir une meilleure performance. Dans le prochain chapitre, nous explorons la capacité de l'apprentissage automatique pour l'automatisation d'une tâche très difficile dans le processus de réalisation des revues systématiques : la sélection des articles pertinents à un sujet de recherche.

CHAPITRE III

UN SYSTEME SEMI-SUPERVISE POUR LA SELECTION DES ARTICLES POUR DES REVUES SYSTEMATIQUES

CHAPITRE III.

UN SYSTEME SEMI-SUPERVISE POUR LA SELECTION DES ARTICLES POUR DES REVUES SYSTEMATIQUES

Dans ce chapitre, nous décrivons notre système proposé pour l'automatisation de la phase de sélection des articles pour l'élaboration des revues systématiques. Le système combine des modèles d'apprentissage non-supervisés et semi-supervisés. Seules les métadonnées sont utilisées pour l'entraînement et le test des modèles, au lieu des textes intégraux. Les textes intégraux ne sont pas toujours gratuitement accessibles et nécessitent des prétraitements avancés pour faire face à des contenus supplémentaires tels que les images et les tableaux [28]. Les métadonnées utilisées dans cette étude sont obtenues par le moteur de recherche Semantic Scholar via une API fournie. Le système proposé est constitué alors de trois étapes essentielles : *collection de données*, *extraction des caractéristiques* et *classification* via un apprentissage automatique comme le montre la figure 3.1. Dans ce qui suit, nous détaillons chacune de ces étapes.

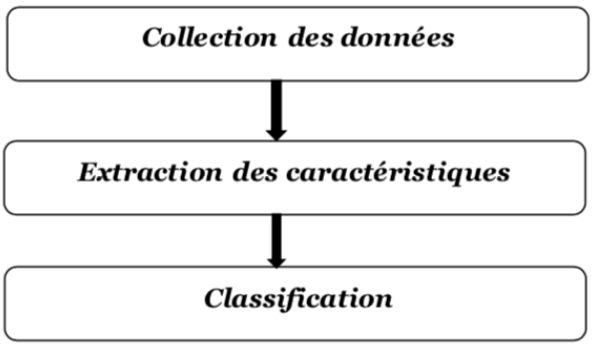

Figure 3.1. Les étapes de construction de notre système

3.1. Collection des données

Le système proposé utilise les métadonnées des articles scientifiques publiés dans plusieurs sources de données. Ces métadonnées incluent des informations pertinentes sur les articles publiés. Malheureusement, certaines données ne peuvent pas être directement obtenues depuis les sources des publications, nous citons par exemple la liste complète des citations et des références. Pour cela, nous utilisons un moteur de recherche académique et générique appelé Semantic Scholar (S2) pour la récupération des données manquantes.

3.1.1. S2 - Semantic Scholar

Très attendus par les chercheurs depuis assez longtemps, des exemples substantiels de moteurs de recherche dépendant de l'intelligence artificielle commencent à apparaître. Établi par l'institut AI2 (Allen Institute for Artificial Intelligence), Semantic Scholar (S2) a commencé comme un moteur de recherche pour le génie logiciel, la géoscience et les neurosciences en 2015. Vue l'incapacité des chercheurs à examiner toutes les publications dans leurs disciplines, l'objectif du projet été d'utiliser l'apprentissage automatique du texte pour aider les chercheurs à trouver rapidement les meilleures publications dans leurs domines et surmonter la surcharge informationnelle [29]. La figure 3.2 montre la page d'accueil de S2.

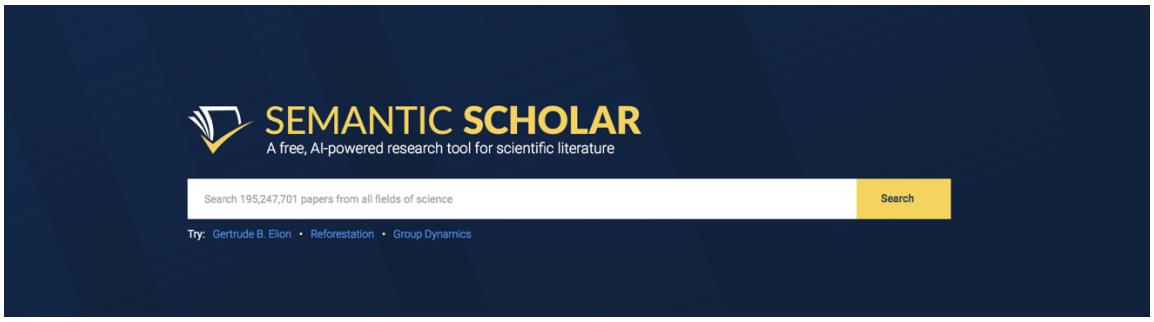

Figure 3.2. Page d'accueil du Semantic scholar (https://www.semanticscholar.org)

Actuellement, S2 couvre plus de 195 millions d'articles de recherche et se développe rapidement. S2 intègre un ensemble de fonctionnalités précieuses pour effectuer des revues systématiques [6] :

1. Moteur de recherche gratuit basé sur les techniques de l'intelligence artificielle.
2. Utilise un langage de requête simple avec une taille limite raisonnable.
3. Fournit une API pour extraire des informations sur des enregistrements individuels à la demande.
4. Couvre plusieurs disciplines inclut l'informatique.
5. Fournit des résumés complets des articles.
6. Fournit une liste des sujets abordés extraits automatiquement de chaque article.
7. Offre des mécanismes de filtrage utiles : par domaine de recherche, par période de publication, par type de publication, par auteur et par source.
8. Permet l'exportation des métadonnées dans des formats utiles tels que BibTex.
9. Permet de télécharger les corpus des métadonnées de tous les articles indexés par le moteur de recherche. Cela permet d'effectuer la recherche des articles hors ligne et d'examiner des algorithmes de recherche personnalisés.
10. Les auteurs et les articles sont distingués à l'aide d'identificateurs uniques.
11. Fournit des listes de citations et de références pour chaque article. Ces listes sont également incluses dans des corpus de métadonnées.
12. Permet l'exploration des citations de papier par différents types de citations : contexte, résultats, méthodes ou tout autre.
13. Fournit un support pour la collaboration et la rétroaction pour corriger les métadonnées et inclure plus de ressources.

Le moteur de recherche S2 fournit une API pour se connecter et extraire des données depuis ses enregistrements. Avec l'API fournie, il est possible de récupérer un enregistrement contenant toutes les informations concernant un article depuis son DOI (Digital Object Identifier) ou un auteur depuis son identificateur associé par S2. L'API renvoie une structure JSON (JavaScript Objet Notation) décrivant l'article ou l'auteur. Le tableau 3.1 montre la liste des champs inclus dans les métadonnées renvoyées par

S2 API pour un article. Chaque champ est décrit par un identificateur, type de données est une brève description.

Identificateur	Type	Description
id	Chaîne	Identificateur unique associé par S2 à chaque article
title	Chaîne	Titre de l'article
paperAbstract	Chaîne	Résumé de l'article
entities	Liste	Liste des sujets extraits de l'article
S2Url	Chaîne	URL vers la page de l'article dans S2
S2PdfUrl	Chaîne	URL du fichier PDF sur S2
pdfUrls	Liste	Liste des URL externes vers le fichier PDF
authors	Liste	Liste des auteurs de l'article avec un identifiant unique et nom
InCitations	Liste	Liste des identificateurs des papiers S2 citant l'article
outCitations	Liste	Liste des identificateurs S2 des papiers cités par l'article
year	Entier	Année de publication de l'article
venue	Chaîne	Source de publication
journalName	Chaîne	Nom de la revue publiant l'article
journalVolume	Chaîne	Volume du journal de l'article
journalPages	Chaîne	Pages de l'article sous la forme « *startPage – endPage* »
doi	Chaîne	Identifiant d'objet numérique du papier (DOI)
doiURL	Chaîne	Lien DOI
pmid	Chaîne	Identifiant du papier sur PubMed
fieldsOfStudy	Liste	Domaines de recherche abordés par l'article
magID	Chaîne	Identifiant unique utilisé par Microsoft Academic Graph.

Tableau 3.1. Structure des métadonnées renvoyées par Semantic Scholar API

3.1.2. Processus de collection des données

La collection des articles scientifiques pour l'élaboration d'une revue systématique se fait en examinons plusieurs bases de données académiques en ligne comme IEEE, ACM, Elsevier, Springer et autres. Dans ce projet, nous visons la réplication des revues systématiques existantes pour montrer l'efficacité du système proposé. Malheureusement, les revues systématiques existantes n'offrent que les listes des articles inclus. Pour obtenir des ensembles de données utiles pour réplication (avec des articles inclus et d'autres exclus), nous adoptons un processus à deux étapes. Premièrement, une re-

cherche est effectuée sur IEEE, ACM et Scopus en utilisant la même requête de recherche adopté par la revue systématique. Les métadonnées collectées sont assemblées et les dupliques sont éliminées. Toutes les articles trouvés et non inclus dans la revue systématique sont considérés comme exclus. Ensuite, les métadonnées de tous les articles (inclus et exclus) sont complétées en interrogeant S2 à travers son API, en utilisant le DOI de chaque article. À la fin, nous obtenons des ensembles de données utiles. La figure 3.3 montre le processus de collection des données adopté dans ce projet.

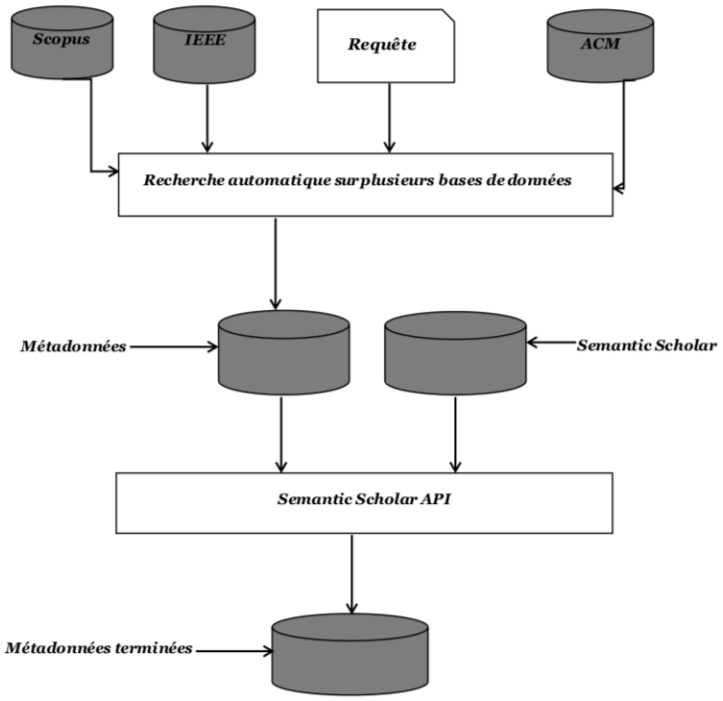

Figure 3.3. Processus de collection de données

3.2. Extraction des caractéristiques

Le choix et l'extraction des caractéristiques est une étape fondamentale pour l'élaboration de l'apprentissage automatique (Voir Chapitre 2). Avant de commencer la présentation des caractéristiques adoptées dans notre projet, nous introduisant le processus de raffinement de la requête de recherche. Ce processus est utilisé pour déterminer la

relation sémantique de chaque article scientifique avec la requête utilisée pour le trouver. Pour cela, nous utilisons les termes de l'ontologie CSO (Computer Science Ontology) pour enrichir les requêtes de recherche par des termes similaires et équivalents.

3.2.1. CSO – Computer Science Ontology :

L'ontologie informatique (CSO) est une taxonomie générée automatiquement des sujets et des termes scientifiques dans le domaine informatique. Elle a été produite par l'université ouverte (The *Open University*) du Royaume-Uni en collaboration avec *Springer Nature* en utilisant l'algorithme Klink-2 sur un large nombre d'articles scientifiques [30]. L'algorithme Klink-2 impliques plusieurs techniques d'analyse sémantique, d'apprentissage automatique et des connaissances provenant de sources externes. Plusieurs connexions entre les termes de l'ontologie ont été révisées manuellement par des experts du domaine. La dernière version de l'ontologie CSO (version 3.2) comprend environ 14.000 termes de recherche et plus de 160.000 relations sémantiques. L'ontologie CSO est disponible pour téléchargement en formats : OWL, Turtle, N-Triples et CSV. De nouvelles versions sont régulièrement publiées sur le portail CSO [W12]. La figure 3.4 montre la page d'accueil de CSO :

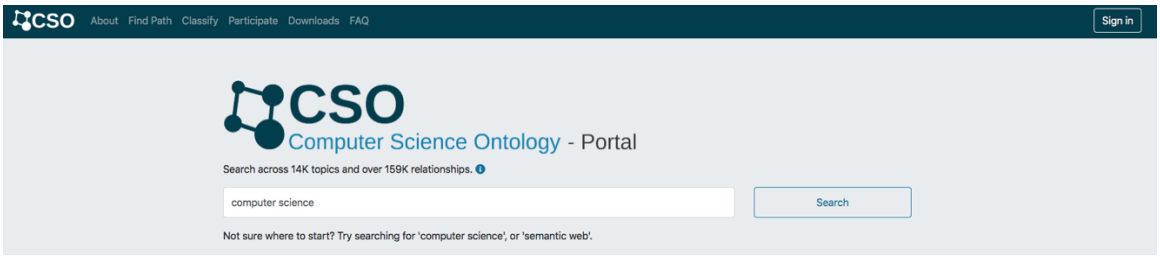

Figure 3.4. Page d'accueil du CSO (https://cso.kmi.open.ac.uk/home)

Pour un sujet de recherche donné, avec l'ontologie CSO, on peut déterminer ses relations sémantiques avec d'autres sujets dans le domaine informatique. Le modèle de données CSO comprend plusieurs relations sémantiques dont :

- *RelatedEquivalent* : montre que deux termes ou sujets sont équivalents et peuvent être utilisés de manière interchangeable.

- *SuperTopicOf /SubTopicOf*: indique qu'un sujet est un super-domaine ou sous-domaine d'un autre.
- *ContributesTo* : montre que les résultats de recherche d'un sujet contribuent à un autre.

La figure 3.5 montre un graphe indiquant la relation du terme '*machine learning*' avec d'autres termes en informatique inclus dans l'ontologie CSO.

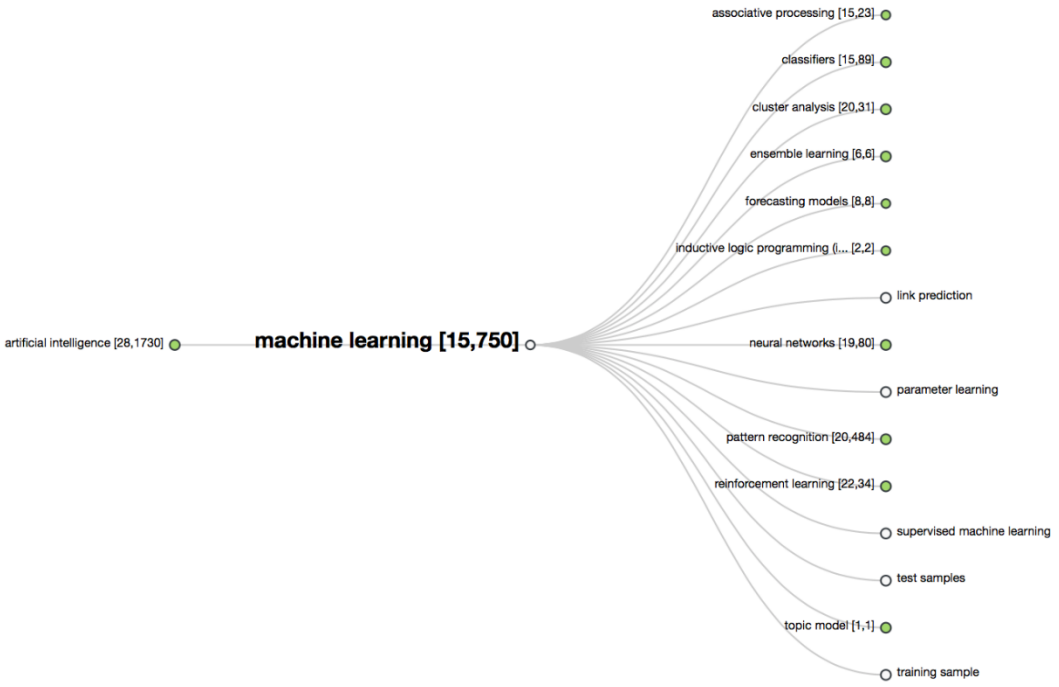

Figure 3.5. Un exemple sur l'utilisation de l'otologie CSO

Pour le raffinement d'une requête de recherche, nous ajoutons pour chaque terme de la requêtes l'ensemble des termes en connexion directes avec les deux relations : *SuperTopicOf* et *SubTopicOf*. Spécifiquement, on remplace chaque terme *t* d'une requête de recherche *Q* par une conjonction du terme *t* lui-même avec tous les termes de ses sous-domaines et super-domaines trouvés dans CSO.

$$\forall t \in Q: t \longleftarrow t \vee \left(\bigvee_{t_0 \in CSO_{subtopics}(t)} t_0 \right) \vee \left(\bigvee_{t_1 \in CSO_{supertopics}(t)} t_1 \right)$$

3.2.2. Les caractéristiques adoptées :

Après avoir effectué la collection des données et le processus du raffinement de la requête de recherche, nous commençons l'extraction des caractéristiques. Ces caractéristiques peuvent être classées en : *caractéristiques générales*, *parentés par paires et modélisation thématiques*. Dans ce qui suit, nous décrivant chacune de ces caractéristiques avec la raison pour lequel chaque caractéristique est choisie.

3.2.2.1. Les caractéristiques générales :

Ces caractéristiques sont liées uniquement à chaque article individuellement.

A. Nombre de pages

Il est recommandé, dans le processus d'élaboration des revues systématiques, d'inclure des articles complets (long) en raison de leur clarté, de leur contenu propre et détaillé [7]. Les études détaillées sont importantes pour la phase d'extraction des données dans le processus d'élaboration des revues systématiques. Par conséquent, nous considérons le nombre de pages de chaque papier p. Dans les métadonnées récupérées par S2, le champ *journalPages* est données sous forme d'une chaîne de caractères de la forme: « *startPage – endPage* » (Voir Tableau 3.1). Le nombre de pages de chaque article p est calculé comme suit :

$$nbPage(p) = endPage(p) - starPage(p)$$

Où :

$$startPage(p) = int(journalPages(p).split('-')[0])$$

Et :

$$endPage(p) = int(journalPages(p).split('-')[1])$$

B. Nombre de citations

Les articles les plus cités devraient avoir un impact scientifique substantiel et une contribution significative à l'ensemble des connaissances [31]. Par conséquent, ils sont susceptibles d'être inclus dans des revues systématiques. Le nombre de citations est mesuré en considérant la liste des citations donnée par S2 dans le champ *inCiations*.

$$nbCitations(p) = |inCitations(p)|$$

C. Relation avec la requête

La relation d'un article avec la requête de recherche est un facteur important pour la sélection des articles. Dans cette étude, nous mesurons la pertinence de chaque article par rapport à la requête élargie utilisée pour la récupérer. Les requêtes originales sont complétées par des termes de l'ontologie (CSO) comme indiqué dans la section 3.2.1. La relation de chaque article p et une requête de recherche élargie Q est définie par :

$$relQuery(p, Q) = \frac{\sum_{c \in Q}(w_c \times \sum_{t \in c} occurance(t, p))}{\sum_{c \in Q} w_c}$$

Notons que chaque requête est d'abord transformée en forme normale conjonctive formant une conjonction des clauses $c \in C$. Chaque clause c est formée par une disjonction des termes t. Nous cherchons le nombre d'occurrence de chaque terme t d'une clause c dans le titre et le résumé d'un article p. Dans cette étude, les poids w_c associés à chaque clause c dans Q sont simplement calculés par :

$$w_{c_1} = 1$$
$$w_{c_{i+1}} = w_{c_i} + 1$$

Ainsi, les termes de la clause suivante sont privilégiés car ils décrivent des sujets plus détaillés par rapport aux termes de sa clause prédécesseur. Ceci est inspiré de la méthode PICO (Population, Intervention, Comparison, Outcome) qui est largement adoptée pour définir les chaînes de recherche dans les revues systématiques [7].

3.2.2.2. Les parentés par paires :

Chaque caractéristique incluse dans cette catégorie identifie une parenté par paires des articles récupérés. Cinq mesures de parenté par paires sont prises en compte :

A. Parenté par citations

Les articles avec des citations communes sont susceptibles d'avoir des sujets communs. En utilisant les métadonnées S2, la valeur de cette cractéristique est donnée par :

$$citeRelatedness(p_1, p_2) = \begin{cases} 1, & \textbf{\textit{Si}}\ |inCitations(p_1) \cap inCitations(p_2)| > 0 \\ 0, & \textbf{\textit{Sinon}} \end{cases}$$

B. Parenté par références

De même, les articles avec des références communes sont susceptibles d'aborder des sujets similaires. Cette caractéristique est mesurée comme suit :

$$refRelatedness(p_1, p_2) = \begin{cases} 1, & \textbf{\textit{Si}}\ |outCitations(p_1) \cap outCitations(p_2)| > 0 \\ 0, & \textbf{\textit{Sinon}} \end{cases}$$

C. Parenté par auteurs

Les articles publiés avec des auteurs en communs sont susceptibles d'être liés au même sujet. Cette caractéristique est mesurée comme suit :

$$authRelatedness(p_1, p_2) = \begin{cases} 1, & \textbf{\textit{Si}}\ |authors(p_1) \cap authors(p_2)| > 0 \\ 0, & \textbf{\textit{Sinon}} \end{cases}$$

D. Parenté par snowballing

Un article p_1 inclus dans la liste de références (backward snowballing) ou dans la liste de citations (forward snowballing) d'un autre article p_2 est susceptible d'avoir des sujets similaires à p_2. Cela peut âtre mesurée comme suit :

$$snbRelatedness(p_1, p_2) = \begin{cases} 1, & \textbf{\textit{Si}}\ id(p_1) \in inCitations(p_2) \cup outCitations(p_2) \\ 0, & \textbf{\textit{Sinon}} \end{cases}$$

E. Similitude PDSM

Cette mesure de similarité est adoptée en raison de son efficacité dans le classement des documents par rapport aux autres mesures de similarité traditionnelles [32]. La

mesure de similarité PDSM est basée sur la fréquence des termes et le nombre de termes apparaissant dans au moins un des deux documents. En conséquence, étant donné deux articles p_1 et p_2 et une liste de termes de vocabulaire *vocab*, le PDSM est définit par :

$$pdsm(p_1, p_2) = \frac{\sum_{i=1}^{|vocab|} Min(tf_{1i}, tf_{2i})}{\sum_{i=1}^{|vocab|} Max(tf_{1i}, tf_{2i})} \times \frac{in(p_1, p_2) + 1}{|vocab| - out(p_1, p_2) + 1}$$

Où tf_{ji} représente la fréquence du i-ème terme du vocabulaire dans l'article j ; $in(p_1, p_2)$ décrit le nombre de termes présents de p_1 dans p_2 et $out(p_1, p_2)$ représente le nombre de termes p_1 absents dans p_2.

3.2.2.3. Modélisation thématique :

Les caractéristiques de cette catégorie décrivent la similarité des articles vis-à-vis des thématiques ou sujets abstraits qui peuvent être découverts en analysant les titres et les résumés des articles récupérés. C'est une technique utile pour révéler les structures sémantiques cachées dans les textes. Nous considérons la probabilité qu'un article fasse partie de chaque thématique résultant du modèle LDA (Voir Chapitre 2).

3.3. Processus de classification

Pour notre projet, deux types de classifieurs non-supervisés sont utilisés : KMeans [33] et LDA [34]. Le premier est adopté pour classer les articles sur la base des parentés par paires. Par conséquent, cinq modèles KMeans indépendants sont adoptés, chacun étant utilisé pour classer les articles en fonction d'une parenté par paires spécifique. Le modèle LDA est utilisé pour obtenir une classification par des thématiques abstraites. Afin d'augmenter les performances des classifieurs utilisés, nous utilisons la méthode de recherche par grille [35]. Les modèles KMeans sont ajustés en faisant varier le nombre de clusters de 2 à 100 par pas de 1. Le nombre optimal de clusters est celui fournissant la valeur maximale du score Silhouette. Nous adoptons l'utilisation d'une version LDA parallélisée (LDAMulticore) qui permet d'utiliser tous les cœurs CPU pour

accélérer la phase d'apprentissage. Les hyperparamètres du modèle LDA sont ajustés comme suit et les valeurs fournissant le meilleur score de Cohérence sont adoptées :

- On fait varier la valeur du paramètre *alpha* de LDA de 0,01 à 1 par pas de 0,3. Nous considérons également les deux valeurs prédéfinies *symétriques* et *asymétriques*.
- De même, on fait varier la valeur du paramètre bêta de 0,01 à 1 par pas de 0,3 en plus de la valeur prédéfinie *symétrique*.
- Le nombre de sujets est varié de 2 à 11 par pas de 1.

Enfin, les modèles sont combinés de manière empilable. Par conséquent, les sorties des cinq modèles KMeans et du modèle LDA sont combinées avec les trois caractéristiques générales. La matrice de valeurs résultante est transmise à un modèle semi-supervisé pour la prédiction finale. Dans cette étude, nous utilisons une méthode d'auto-apprentissage [24]. En utilisant une telle technique, un classifieur de base est d'abord entrainé sur un petit ensemble de articles étiquetés disponibles. Le classifieur est ensuite utilisé pour étiqueter les articles restants. De nouveaux articles étiquetés avec un degré de confiance élevé sont automatiquement ajoutés aux données d'apprentissage. Le classifieur est ré-entrainé sur les nouveaux articles et la procédure est répétée jusqu'à l'exploration de tous les articles. Cette méthode d'autoformation permet d'encapsuler des algorithmes d'apprentissage supervisés traditionnels pour devenir des modèles semi-supervisés. Dans ce projet nous examinons différentes classifieurs. La figure 3.6 montre le processus général de classification utilisé dans ce projet.

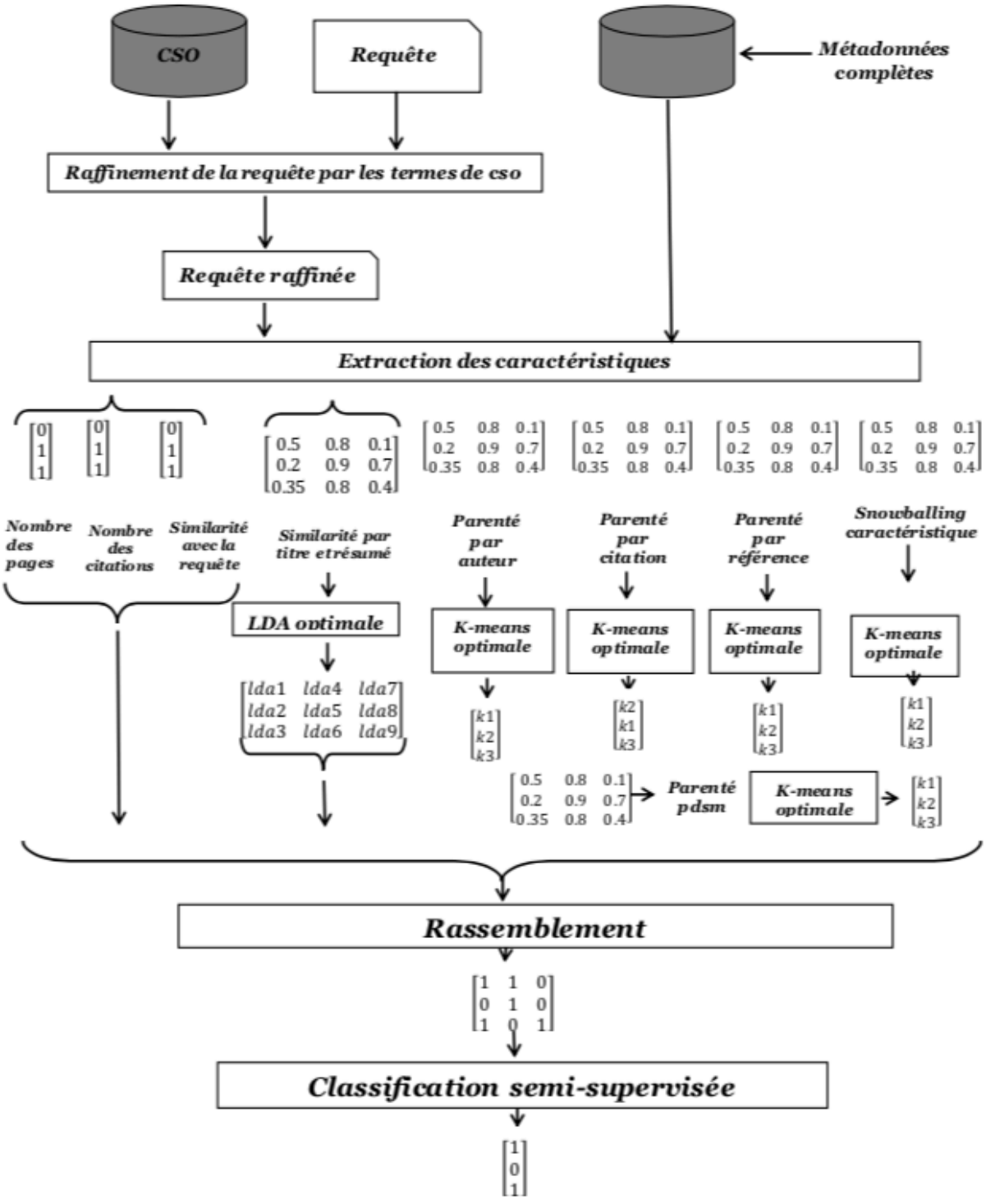

Figure 3.6. Processus de classification

3.4. Conclusion

L'automatisation du processus d'élaboration des revues systématiques devient une nécessité pour réduire le temps et les efforts nécessaires à la réalisation des études de recherche aussi précieuses. Dans ce chapitre, nous avons proposé un système de sélection automatique des articles pertinents en analysant simplement leurs métadonnées obtenues à partir de Semantic Scholar (S2). Le système proposé utilise une combinaison des modèles d'apprentissage automatique non-supervisé et semi-supervisé pour prédire l'inclusion ou l'exclusion des articles dans une revue systématique spécifique. Nous avons présenté les différentes techniques et outils utilisés pour la construction du système proposé. Nous avons également présenté l'ensemble de caractéristiques choisies ainsi que le processus de classification adopté. Le prochain chapitre sera consacré à l'implémentation et la validation du système proposé.

CHAPITRE IV

VALIDATION & IMPLEMENTATION

CHAPITRE IV.

VALIDATION & IMPLEMENTATION

Nous avons présenté dans le chapitre précédent l'architecture détaillée de notre système. Il est impératif de valider et expérimenter le système proposé avec des données réelles. Dans ce chapitre, nous introduisons les essais élaborés pour la validation de notre système. Nous décrivons aussi la phase d'implémentation d'un outil implémentant l'architecture proposée. Nous décrivons par conséquent l'environnement de développement, les différentes bibliothèques utilisées pour mettre en œuvre l'outil proposé.

4.1. Collection des données

Pour valider l'architecture proposée en chapitre III, nous avons exploité des ensembles de données élaborées dans [6]. Ces ensembles incluent des métadonnées de revues systématiques réelles élaborées manuellement. Les métadonnées dont collectés suivant le processus décrit dans la figure 3.3 (Voir chapitre III). Chaque ensemble de données comporte les métadonnées des papiers inclus ainsi que des métadonnées des papiers exclus. Les métadonnées initiales sont complétées par les métadonnées données de Semantic Scholar. Au total, trois revues systématiques sont expérimentées avec le système proposé. Le tableau 4.1 montre le nombre des papiers inclus et exclus de chaque revue. Le tableau 4.2 décrit les requêtes de recherche utilisées pour la collection des papiers de chaque revue :

Revue systématique	# papiers inclus	# papiers exclus	# total
R1 [36]	69	173	242
R2 [37]	64	547	611
R3 [38]	22	337	359

Tableau 4.1. Ensemble de données utilisées pour la validation

Revue	Requête de recherche
R1 [36]	("continuous integration" OR "rapid integration" OR "fast integration" OR "quick integration" OR "frequent integration" OR "continuous delivery" OR "rapid delivery" OR "fast delivery" OR "quick delivery" OR "frequent delivery" OR "continuous deployment" OR "rapid deployment" OR "fast deployment" OR "quick deployment" OR "frequent deployment" OR "continuous re-lease" OR "rapid release" OR "fast re-lease" OR "quick release" OR "frequent release" OR "deployability" OR "continuous build" OR "rapid build" OR "fast build" OR "frequent build" OR "quick build") AND ("software" OR "information system" OR "information technology" OR "cloud*" OR "service engineering")
R2 [37]	("smell" OR "design flaw" OR "disharmony" OR "code anomaly" OR "design anomaly" OR "anti-pattern") AND ("experiment" OR "empirical" OR "survey" OR "ethnography" OR "action research" OR "exploratory analysis" OR "study" OR "controlled")
R3 [38]	("model driven" OR "MDE" OR "MDD") AND ("systematic review" OR "literature review" OR "literature survey" OR "survey" OR "overview of research" OR "mapping study" OR "review")

Tableau 4.2. Requête de recherche utilisée pour chaque revue

Les ensembles de données utilisés dans ce projet sont structurés sous la forme d'un dictionnaire de données sauvegardé dans un fichier JSON. Ce dictionnaire englobe la liste des papiers inclus et exclus de chaque revue systématique. Chaque papier est décrit selon la structure des métadonnées utilisée par Semantic Scholar (voir chapitre III). Nous rappelons ici la liste des informations fournis pour chaque papier :

- *ss_id* : identificateur Semantic Scholar de papier.
- *year* : année de publication.
- *title* : titre du papier.
- *abstract* : résumé du papier.
- *pages* : nombre de pages du papier.
- *author* : liste des identifiants Semantic Scholar des auteurs du papier.
- *incitations* : liste des identificateurs des papiers référençant le papier.

- *outcitations* : liste des identificateurs des papiers référencés dans le papier.

4.2. Extraction des caractéristiques

Après la collection de données, les requêtes de recherche sont raffinées en utilisant l'ontologie CSO. L'exemple suivant montre la requête de la revue R3 raffinées où les nouveaux termes sont désignés en couleur vert.

("model driven" OR "MDE" OR "MDD" OR "model-driven engineering" OR "model to model transformation" OR "domain specific modeling" OR "concrete syntax" OR "architecting" OR "code generation" OR "model transformation") AND ("systematic review" OR "literature review" OR "literature survey" OR "survey" OR "overview of research" OR "mapping study" OR "review")

Les requêtes raffinées sont utilisées pour l'extraction des caractéristiques discutées dans le chapitre précédent. La figure 4.1 montre un aperçu sur la matrice de caractéristiques de R3.

	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	review	mapping study	overview of research	survey	literature sur	literature re	systematic n	mdd	mde	model driver	lda_topic_2	lda_topic_1	lda_topic_0	rel_	sim_	ref_	cit_	pdsm_	auth_	sn_	query_relate	citations	#pages
2	1	0	0	0	0	0	0	0	1	1	0.95976555	0.04013243	0	59	25	2	35	8	12	42	4	0.2519685	12
3	1	0	0	0	0	0	1	0	0	1	0.9973339	0	0	21	25	61	49	34	12	51	4	0.34782609	24
4	1	0	0	0	0	1	1	0	0	0	0.99982905	0	0	91	21	66	10	13	23	26	5.33333333	0.3	15
5	1	0	0	0	0	0	1	1	1	1	0.99980956	0	0	97	73	88	38	34	1	37	6.33333333	0.41304348	15
6	1	0	0	0	0	0	1	1	1	1	0.99980199	0	0	1	80	2	1	3	52	68	3.66666667	0	21
7	1	0	0	0	0	0	1	0	1	1	0.99979383	0	0	15	59	54	2	37	26	7	6.33333333	0.1	25
8	1	0	0	0	0	1	1	0	0	0	0.99984604	0	0	21	40	20	62	18	52	12	5.33333333	0.90909091	25
9	0	0	0	1	0	0	0	1	0	1	0.99963641	0	0	17	95	47	47	7	1	21	3	0.3	6
10	1	0	0	0	0	1	0	0	0	0	0.99778271	0	0	6	40	2	41	6	1	23	1.33333333	0.27272727	30
11	1	0	0	0	0	0	1	1	0	1	0.9976474	0	0	0	73	51	77	18	68	11	2.66666667	0.36	17
12	1	0	0	0	0	0	1	1	0	1	0.99981308	0	0	73	24	45	83	2	1	0	2.66666667	1	10
13	1	0	0	0	0	1	1	1	1	1	0.99989414	0	0	4	96	14	1	13	89	44	7.66666667	0	22
14	1	0	0	0	0	1	1	0	1	1	0.99981135	0	0	87	96	13	76	26	1	49	5.33333333	0.26086957	39
15	0	1	0	1	0	0	0	0	1	1	0.99984252	0	0	7	41	57	82	19	36	8	1.33333333	0.57894737	17
16	1	0	0	0	0	0	1	0	0	1	0.99982905	0	0	38	96	49	18	18	1	12	2.66666667	0.16666667	46
17	1	0	0	0	0	0	1	0	0	1	0.99984956	0	0	17	44	31	50	34	14	45	5.33333333	0.75	10
18	1	0	0	0	0	1	1	0	1	1	0.99989891	0	0	75	96	21	4	3	1	26	3.66666667	1.30769231	18
19	1	0	0	0	0	0	1	1	1	1	0.9974024	0	0	85	73	5	75	18	1	0	4.66666667	0.22222222	10
20	1	0	0	0	0	0	1	0	1	1	0.99974024	0	0	21	25	60	4	3	62	59	3	2.8	12
21	1	1	0	0	0	0	0	0	1	1	0.9997701	0	0	32	25	33	81	45	1	0	3	0.07142857	9
22	1	0	0	0	0	1	1	1	0	1	0.99980581	0	0	38	95	16	25	2	1	29	4.33333333	0.09090909	8
23	1	0	0	0	0	1	0	0	1	1	0.99984956	0	0	21	25	89	4	22	62	4	2	14	6
24	0	0	0	1	0	0	0	0	1	0	0	0.83653128	0.16340768	74	34	93	37	61	1	0	3	0.03333333	7
25	0	0	0	0	0	0	0	0	0	0	0.99978495	0	0	65	27	67	1	2	10	0	0	0	8
26	0	0	0	0	0	0	0	0	1	0	0	0.99977779	0	71	32	94	1	1	35	72	3.66666667	0	8
27	1	0	0	0	0	0	1	1	0	0	0	0.999861	0	54	49	2	1	15	1	0	4.33333333	0.14285714	11
28	0	0	0	1	0	0	0	0	1	0	0	0.99984372	0	22	4	15	1	5	50	39	4	0	8
29	1	0	0	0	0	0	1	0	1	0	0	0.05254861	0.94737887	1	51	2	1	15	1	0	2	0	8
30	0	0	0	0	0	0	0	0	0	1	0.99974352	0	0	1	52	2	1	21	1	79	1	0	4
31	0	0	0	0	0	0	0	0	0	1	0.99968743	0	0	1	21	2	1	39	1	0	0.33333333	0	4
32	1	0	0	0	0	0	1	1	0	0	0.24817625	0.75174683	0	16	84	2	51	23	60	0	4	0.11111111	25
33	1	0	0	0	0	0	1	1	0	0	0	0.90862566	0.09120487	1	0	2	1	11	1	0	3	0	7
34	0	0	0	1	0	0	0	0	1	0	0	0.32838243	0.67155975	78	14	2	1	86	16	0	1.66666667	0	10
35	0	0	0	1	0	0	0	0	1	0	0	0.35115698	0.64875293	48	65	34	34	1	4	15	1	0.11764706	6
36	1	0	0	0	0	0	1	1	0	0	0.05175907	0.94817382	0	1	49	2	1	11	1	64	5	0	11
37	0	0	0	1	0	0	0	0	0	1	0.99983609	0	0	27	53	1	1	5	1	0	2	0	14
38	0	0	0	1	0	0	0	1	0	0	0	0.99987876	0	1	61	2	1	1	1	60	2.66666667	0	8
39	0	0	0	0	0	0	0	1	0	0	0	0.3282181	0.67172629	39	92	98	12	1	1	0	2	0.66666667	5
40	0	0	0	1	0	0	0	0	1	0	0	0.99987257	0	3	34	8	59	48	24	33	3	0.07692308	9
41	0	0	0	1	0	0	0	0	1	0	0	0.25168434	0.74827099	1	29	2	1	0	1	71	1.33333333	0	8
42	0	0	0	1	0	0	0	1	0	0	0	0.99985176	0	3	38	8	42	1	30	0	2.33333333	0.24242424	10
43	0	0	0	1	0	0	0	0	1	0	0	0.99988157	0	71	36	97	1	10	35	72	5.33333333	0	8
44	0	1	0	1	0	0	0	0	1	1	0.99978948	0	0	7	41	42	1	19	36	93	1.33333333	0	6
45	0	0	0	1	0	0	0	0	0	0	0.99962968	0	0	58	88	2	10	1	49	0	0.66666667	0.04918033	31
46	1	0	0	0	0	0	0	0	1	1	0.99981648	0	0	1	93	2	1	2	1	35	1	0	10
47	0	0	0	1	0	0	0	0	1	0	0	0.29037562	0.70958304	51	31	90	71	1	1	0	4.66666667	0.125	11
48	0	0	0	1	0	0	0	1	0	0	0	0.99987799	0	1	38	2	30	1	1	50	2.33333333	0.25	7
49	0	0	0	1	0	0	0	0	1	0	0.10159162	0.89835429	0	24	57	96	1	27	1	27	2.33333333	1	15
50	0	0	0	1	0	0	0	0	0	1	0	0.99983191	0	1	31	2	45	81	1	0	1.33333333	0.25	7

Goulao-2016_with_lda

Figure 4.1. Aperçu sur la matrice des caractéristiques de R3

4.3. Résultats de classification

Dans cette section, nous présentons les résultats obtenus pour chaque revue. Rappelons que nous avons adopté une solution semi-supervisée à base d'auto-apprentissage (voir Chapitre II). Nous avons donc expérimenté différents classifieurs de base d'apprentissage supervisé, qui sont transformés par la méthode auto-apprentissage en modèles d'apprentissage semi-supervisés. Dans cette expérimentation, nous cherchons le classifieur qui donne la meilleure valeur de F1-score. Six classifieurs sont examinés : Arbre de décision (DT), Naïve Bayes (NB), Machine à Vecteurs de Support (SVM), Régression Logistique (LR), Gradient Stochastique (SGD) et Forêt d'Arbres Décisionnels (RF). L'application de ces classifieurs sur les trois ensembles de données R1-R3 décrits dans la section 4.1 est donnée dans le tableau 4.3. Nous avons testé notre système avec différents seuils de notation manuels ; les résultats du tableau 4.3 sont obtenus avec un seuil de 65%, ce qui montre la capacité de notre système à réduire 35% de l'effort des chercheurs dans la sélection des articles pertinents à la requête de recherche.

Model	R1			R2			R3		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
DT	90,50	89,67	89,88	90,3	82,02	84,47	94,62	73,82	80,6
NB	86,69	83,47	84,11	89,81	69,76	43,96	94,03	69,64	77,53
SVM	86,13	83,06	83,7	89,38	78,66	81,82	94,21	80,78	85,45
LR	85,56	82,64	83,28	89,53	82,81	84,94	93,97	86,07	88,95
SGD	57,02	57,58	57,18	61,83	73,81	61,81	97,7	75,67	85,28
RF	92,59	91,74	91,92	92,32	84,58	86,68	96,57	92,2	93,55

Tableau 4.3. Résultats de classification

Le tableau 4.3 montre que l'algorithme de forêt d'arbres décisionnels donne le meilleur score pour les 3 ensembles de données expérimentés allons jusqu'à 93.55% en termes de F1-score.

4.4. Conception de l'outil

Pour la conception de notre outil, nous avons utilisé la méthodologie UML (Unified Modeling Language), qui est un langage de modélisation visuel. Cette méthodologie nous a permis de concevoir et de mettre en œuvre le logiciel de manière consistante et facile. Cette modélisation est basée sur les descriptions de différents diagrammes en adoptant certains ensembles de règles. Chaque diagramme décrit une vue différente détaillée du logiciel. La méthodologie UML est considérée comme l'une des méthodes les plus utilisées pour la conception des systèmes complexes [39]. Dans ce qui suit, nous décrivons deux types de diagrammes utilisés pour la conception de notre outil.

4.4.1. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est le premier schéma du modèle UML, c'est un moyen qui nous a permis d'identifier les caractéristiques fonctionnelles de notre outil. En d'autres termes, le diagramme de cas d'utilisation est une description de ce qui se passe lorsque l'utilisateur se connecte à l'application.

Selon le diagramme de cas d'utilisation décrit dans la figure 4.2, l'utilisateur peut interagir avec l'application en spécifiant le chemin d'un fichier JSON comportant les métadonnées des papiers trouvés par la requête de recherche. Avant de commencer la phase d'apprentissage automatique, l'utilisateur doit sélectionner manuellement quelques papiers à inclure et à exclure. Cela est nécessaire pour un apprentissage semi-supervisé. Le système commence à analyser les métadonnées, faire l'extraction des caractéristiques et lance le processus d'apprentissage semi-supervisé tout en utilisant la liste des papiers notée par l'utilisateur. À la fin, l'utilisateur peut visualiser et récupérer au choix, la liste des papiers inclus et exclus suivant les prédictions de notre modèle.

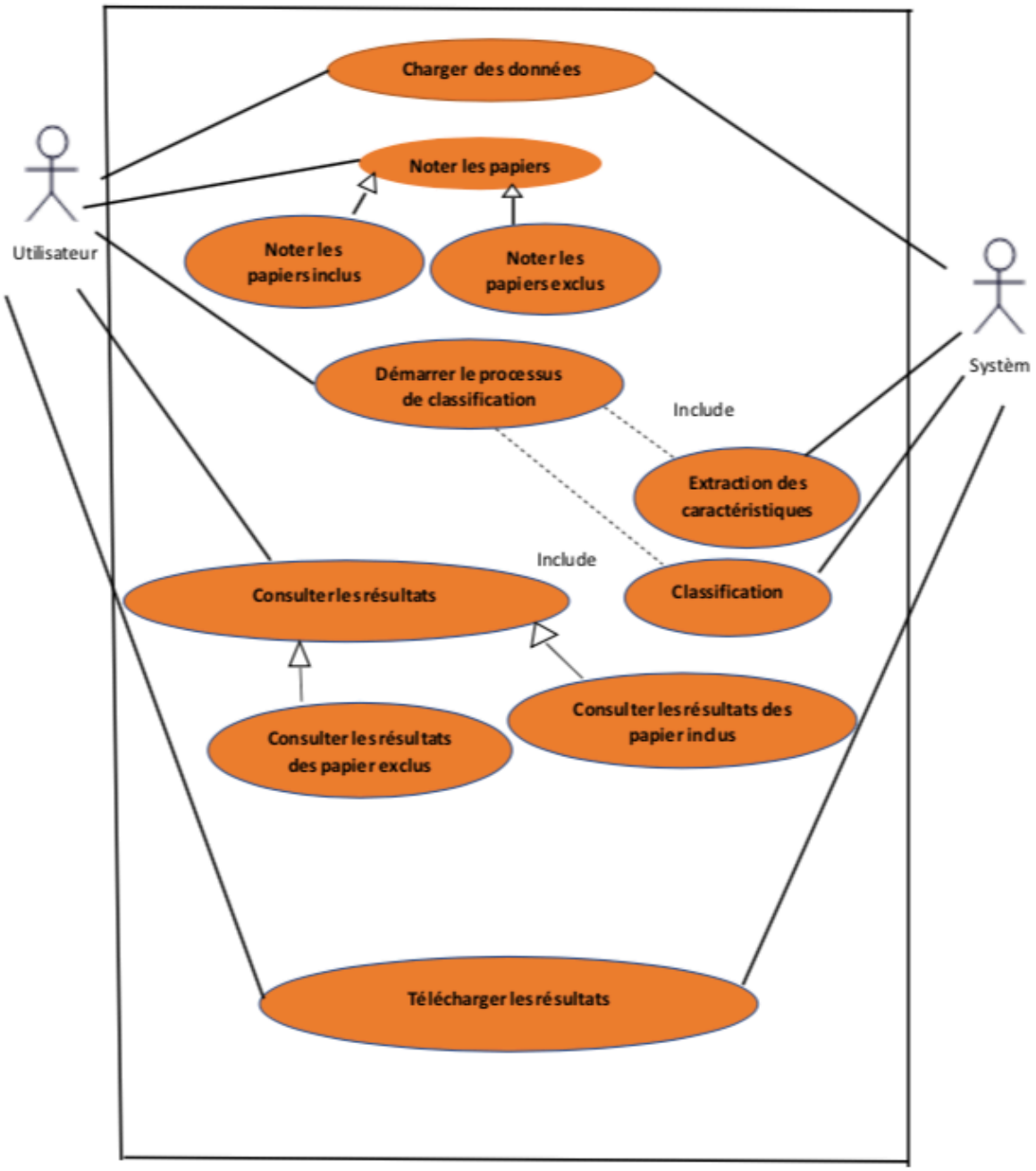

Figure 4.2. Diagramme de cas d'utilisation

4.4.2. Diagramme de séquences

Le diagramme de séquence est le diagramme d'interaction le plus fréquent. Ce diagramme est conçu en fonction du temps, indiquant les opérations à réaliser, les messages à envoyer et les tâches à accomplir. La figure 4.3 montre le diagramme de séquences adopté dans notre application.

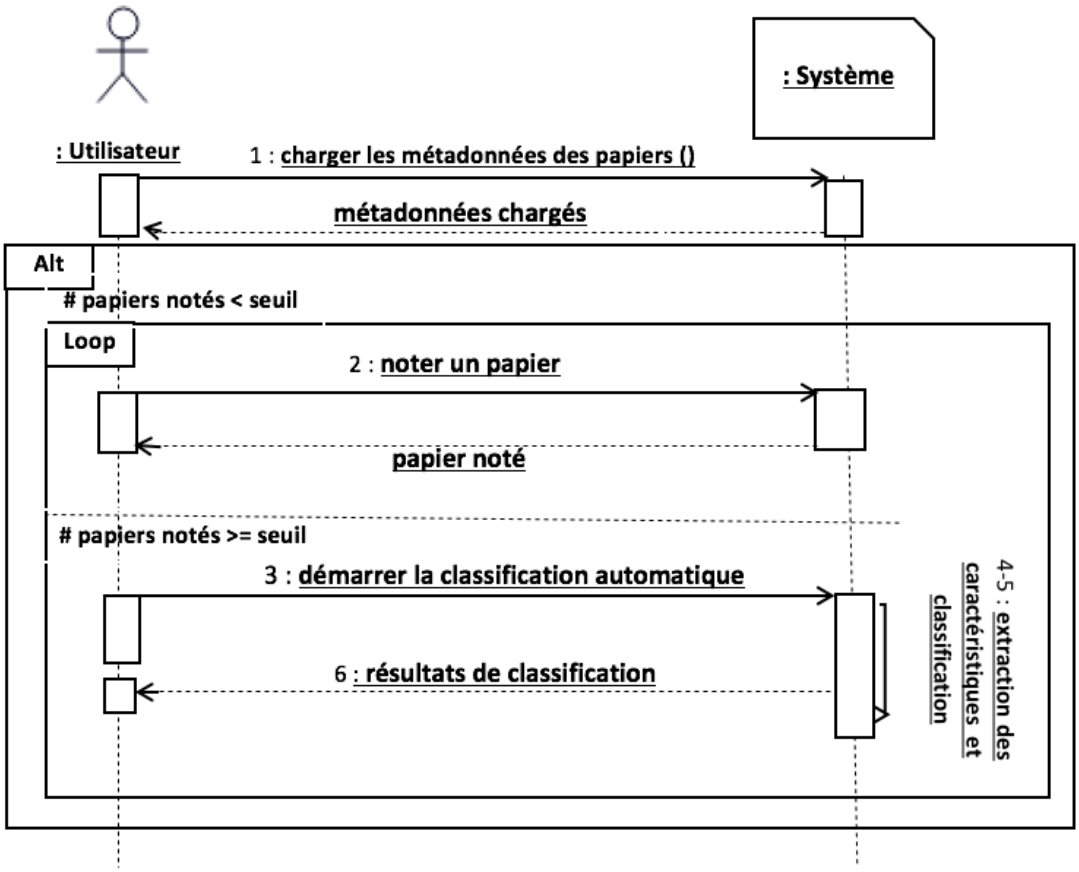

Figure 4.3. Diagramme de séquences

L'utilisateur commence à indiquer au système le chemin vers l'ensemble de données au format JSON. L'utilisateur sera alors invité à noter quelques papiers en sélectionnant l'un des états : inclus ou exclus. Le système compte le nombre des papiers notés. Après un certain seuil, l'utilisateur est autorisé à lancer la classification automatique. À la fin, l'utilisateur peut consulter la liste de tous les papiers inclus et exclus de l'ensemble de données préalablement chargé.

4.5. Environnement de développement :

4.5.1. Partie matérielle

Ce projet a été développé sous une machine ayant les caractéristiques suivantes :

- **Processeur :** Intel®Core™i3-4005U CPU@1.70GHz 1.70GHz
- **Mémoire installée (RAM) :**4,00Go
- **Disque Dur :** 500Go

4.5.2. Partie logicielle :

4.5.2.1. Plateforme PyCharm

Pycharm est un éditeur de code très pratique qui permet de créer des projets et d'éditer des scripts python. Cet éditeur est développé par l'entreprise JetBrains. Ce logiciel est multiplateforme, c'est-à-dire qu'il fonctionne sous les différents systèmes d'exploitation, en plus d'être compatible avec plusieurs versions de python. La plateforme PyCharm est disponible en deux éditions, une édition professionnelle diffusée sous licence propriétaire et une édition communautaire diffusée sous licence Apache. Dans notre projet, nous avons utilisé l'édition communautaire version 2021.1.

4.5.2.2. Langage de programmation Python

Python est un langage de programmation libre de haut niveau, créé par Guido van Rossum, sa première publication a été en 1991. Python est un langage interprété, c'est-à-dire qu'il peut être exécuté sans compilation. Il est caractérisé par un système de typage dynamique, une gestion de mémoire automatique et une bibliothèque multifonctionnelle complète. Ce langage prend en charge plusieurs paradigmes de programmation, et il est adaptable pour tous les systèmes d'exploitation. Python est un langage très approprié pour les apprenants débutants, mais il est aussi très motivant pour les utilisateurs expérimentés [40]. Pendant la réalisation de notre projet, nous avons utilisé la version 3.6 ainsi qu'un ensemble de bibliothèques utiles. Nous classons ces bibliothèques selon leurs rôles dans la réalisation de notre projet en 4 catégories : *gestion de*

l'ensemble de données, *extraction des caractéristiques*, *gestion des modèles d'apprentissage* et *développement de l'interface graphique.*

4.5.2.2.1 Gestion de l'ensemble de données

Pandas (version 1.1.5) :

Est un package Python qui fournit des structures de données rapides, adaptables et expressives destinées à rendre le travail avec des données « relationnelles » ou « étiquetées » à la fois simple et intuitif. Il vise à être le bloc de construction fondamental de haut niveau pour effectuer une enquête de données fonctionnelle et authentique en Python. En outre, il a pour objectif plus vaste de devenir l'outil d'analyse/manipulation de données open source le plus puissant et le plus flexible. Dans note projet, les ensembles de données après l'extraction des caractéristiques sont chargés sous forme de bases de données Pandas.

NumPy (version 1.19.5) :

Numpy est la bibliothèque responsable du calcul mathématiques en python, elle se base sur l'utilisation des tableaux de N dimensions. Ces tableaux sont très rapides, et facilitent les opérations mathématiques sur un grand nombre de données [33]. Nous avons utilisé cette bibliothèque pour appliquer des opérations mathématiques dans la phase de traitement de l'ensemble de données.

JSON (JavaScript Object Notation) :

Est un langage d'échange textuel imprimé léger. Pour la machine, ce format est facilement généré et analysé. Pour les programmeurs, il est pratique à écrire à lire en raison de la syntaxe simple et de la structure arborescente. JSON est utilisé pour traiter des données organisées (comme XML par exemple).

CSV (en anglais, comma separated values) :

Est le fichier de base de données recueillies - sans formatage particulier. Chaque champ est séparé par une virgule. Puisque plusieurs applications utilisent des formats de fichier différents, les fichiers CSV servent de format universel permettant de voir les données dans une variété d'applications, comme Microsoft Excel, Numbers, le tableur Google ou autre.

4.5.2.2.2 Extraction des caractéristiques

SpaCy (version 3.0.5) :

Est une bibliothèque open source gratuite pour le traitement avancé du langage naturel (NLP) en Python. Elle est conçue spécifiquement pour une utilisation en production et nous aide à créer des applications qui traitent et « comprennent » de gros volumes de texte. Elle est utilisée pour les prétraitements des métadonnées collectés notamment l'identification des mots les plus importants dans les titres et résumés de chaque papier.

Natural Langage Toolkit (NLTK) (version 3.6.1) :

Bibliothèques de gestion de texte pour la tokenisation, l'analyse, la caractérisation, le radicalisme, le balisage et la réflexion sémantique.

4.5.2.2.3 Gestion des modèles d'apprentissage

Scikit-learn (version 0.24.1) :
Est l'une des bibliothèques Python les plus utilisées pour la science des données et l'apprentissage automatique. Elle permet d'effectuer de nombreuses opérations et fournit une variété d'algorithmes. Scikit-learn propose également une excellente documentation sur ses classes, méthodes et fonctions, ainsi que des explications sur le contexte des algorithmes utilisés. Scikit-learn a été utilisée pour les modèles d'apprentissage non-supervisé comme LDA et KMeans.

4.5.2.2.4 Développement de l'interface graphique

Tkinter (version 1.0):
Tkinter est la bibliothèque GUI standard pour Python. Lorsqu'il est combiné avec Tkinter, Python fournit un moyen rapide et facile de créer des applications GUI. Tkinter fournit une puissante interface orientée objet.

4.6. Mode d'utilisation de l'outil

L'outil développé dans le cadre de ce projet est constitué d'une seule fenêtre de base avec laquelle l'utilisateur peut effectuer les opérations suivantes :

1. Charger l'ensemble des papiers à analyser

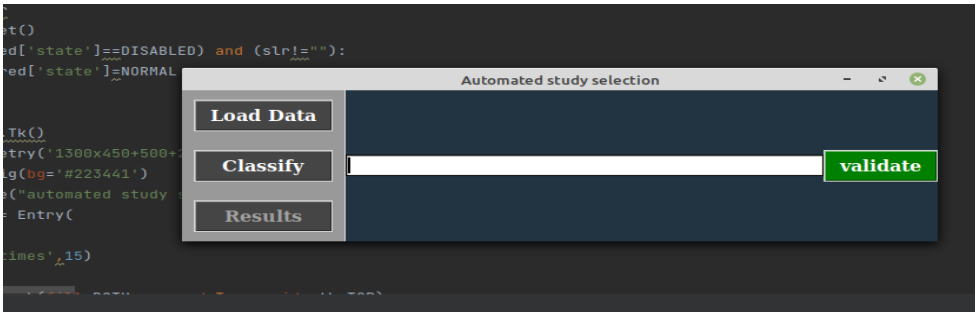

Figure 4.4. Chargement de l'ensemble des métadonnées

2. Noter un certain nombre de papiers : Choisir d'inclure ou d'exclue un papier en sélectionnant l'un des boutons « include » ou « exclude » respectivement.

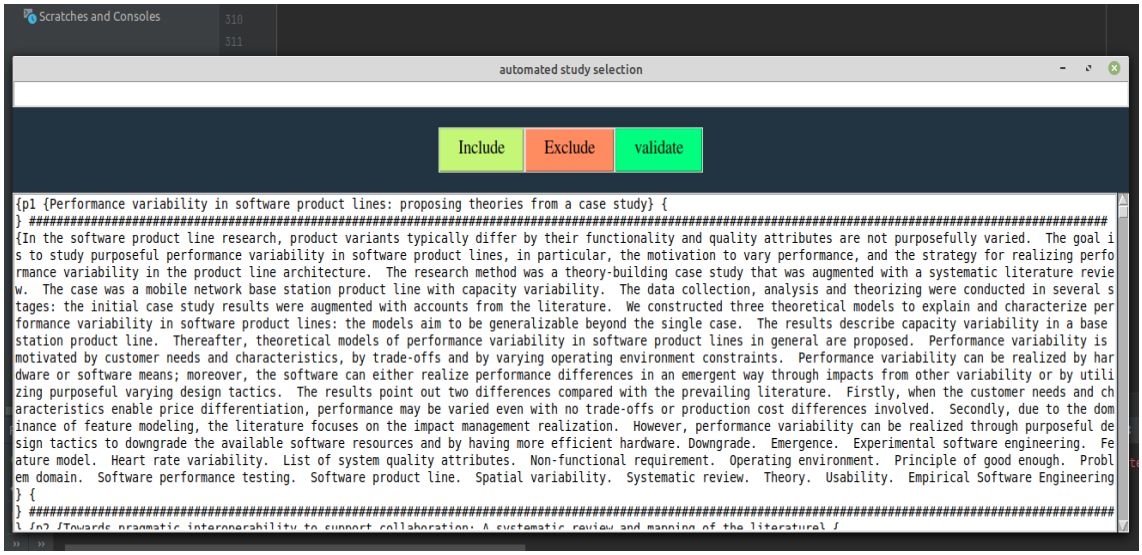

Figure 4.5. Notation des papiers

3. Lancer l'apprentissage automatique en appuyant sur le bouton « Validate » et la fenêtre suivante s'affiche :

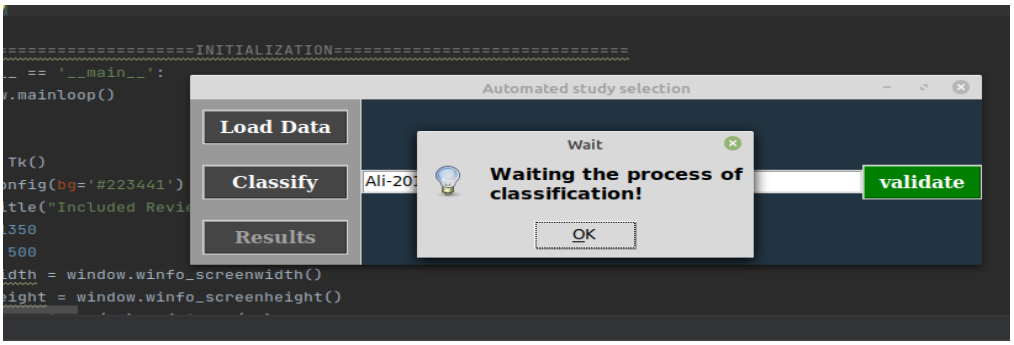

Figure 4.6. Lancement du processus d'apprentissage automatique

4. Consulter les résultats de classification : l'utilisateur peut sélectionner les types des papier inclus ou exclus en choisissant les boutons « Included » ou « Excluded » respectivement. L'utilisateur a aussi la possibilité de sauvegarder les résultats dans un fichier CSV afin de compléter les autres étapes d'élaboration de la revue.

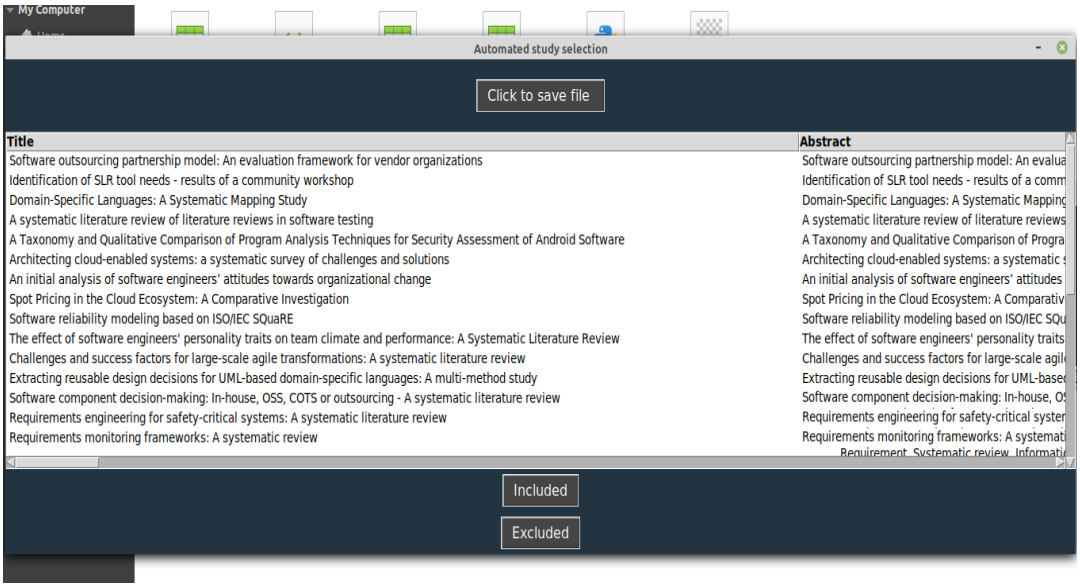

Figure 4.7. Consultation des résultats de classification

4.7. Conclusion

Dans ce chapitre, nous avons présenté l'ensemble des outils utilisés pour la conception et la mise en œuvre de notre application. Nous également validé notre système avec des ensembles de données réelles ; et nous avons présenté un aperçu sur la conception et l'implémentation de notre application.

CONCLUSION GENERALE

CONCLUSION GENERALE

Les revues systématiques représentent une méthodologie qui vise à élaborer des synthèses des résultats de recherche publiés sur des sujets bien précis. L'intérêt remarquable des revues systématiques dans le développement de la recherche est ralenti par la complexité de leurs processus d'élaboration. Cela nous a motivé à contribuer à l'automatisation du processus d'élaboration des revues systématiques dans le domaine informatique. Dans notre recherche nous avons étudié les travaux les plus récents sur l'automatisation du processus d'élaboration des revues systématiques dans les différents domaines. D'après l'analyse des papiers étudiés, nous constatons que la sélection des articles n'a jamais été automatisé parfaitement. Spécifiquement, peu d'outils sont consacrés à l'automatisation du processus de sélection des papiers dans le domaine informatique ; aucun des outils existants n'offre une automatisation complète et efficace de cette phase cruciale.

Par ce projet, vu la nature du problème traité, nous avons exploité deux technologies de l'intelligence artificielle pour l'automatisation de la phase de sélection des papiers pertinents depuis l'analyse de leurs métadonnées : (1) l'apprentissage automatique semi-supervisé et non-supervisé et (2) l'ontologie du domaine. La première est utilisée pour la prédiction automatique de la pertinence des papiers ; la deuxième est exploitée pour enrichir les termes de recherche utilisés pour mesurer la similarité des papiers. L'expérimentation montre que la capacité du système proposé réduit jusqu'à 35% de l'effort des chercheurs dans la sélection des articles pertinents. Notre futur objectif est de travailler sur l'amélioration de cette capacité afin de réduire au mieux l'effort des chercheurs.

BIBLIOGRAPHIE & WEBOGRAPHIE

BIBLIOGRAPHIE

[1] Romain Sordello, Anne Villemey, Arzhvael Jeusset, Marianne Vargac, Yves Bertheau, Aurélie Coulon, Nadine Deniaud, Frédérique Flamerie de Lachapelle, Eric Guinard, Hervé Jactel, Emmanuel Jaslier, Eric Le Mitouard, Vanessa Rauel, Véronique Roy, Sylvie Vanpeene, Isabelle Witté, Julien Touroult, "*Conseils méthodologiques pour la réalisation d'une revue systématique à travers l'expérience de COHNECS-IT",* Rapport technique N.83*,* Unité Mixte de Service Patrimoine Naturel (PatriNat), CNRS, France, juin 2017.

[2] Tina Poklepović Peričić et Sarah Tanveer, "*Why systematic reviews matter: A brief history, overview and practical guide for authors*", Elsevier connect, authors'update, juillet 2019.

[3] Kitchenham B.A., Dybå T., Jørgensen M., Evidence-based software engineering, in: Proceedings of the 26th International Conference on Software Engineering, (ICSE '04), IEEE Computer Society, Washington DC, USA, pp. 273–281, 2004.

[4] Dybå T., Kitchenham B.A., Jørgensen M., Evidence-based software engineering for practitioners, IEEE Software 22 (1), 58–65, 2005.

[5] Barbara Kitchenham, Rialette Pretorius, David Budgen, O. Pearl Brereton, Mark Turner, Mahmood Niazi, Stephen Linkman, "*Systematic literature reviews in software engineering – A tertiary study*", Information and Software Technology, 52 (8), pp. 792-805, 2010.

[6] Abdelhakim Hannousse, "*Searching relevant papers for software engineering secondary studies: Semantic Scholar coverage and identification role*", IET Software, 15 (1), pp. 126-146, 2021.

[7] Kitchenham, B., Budgen, D., Brereton, P., "*Evidence-Based Software Engineering and Systematic Reviews*", CRC Press, Boca Raton, 2015.

[8] Santos, R. E. S. and da Silva, F. Q. "*Motivation to perform systematic reviews and their impact on software engineering practice*", in Proceedings of ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), IEEE Computer Society Press, pp. 292– 295, 2013

[9] Al-Zubidy, A., Carver, J.C., Hale, D.P., Hassler, E.E. "*Vision for SLR tooling infrastructure: Prioritizing value-added requirements"*. Information and Software Technology. 91, pp. 72–81, 2017.

[10] Dybå, T., Dingsøyr, T., Hanssen, G.K. "*Applying systematic reviews to diverse study types: an experience report*". In proceedings of the First International Symposium on Empirical Software Engineering and Measurement, Madrid, Spain, pp. 225–234, 2007.

[11] Kuhrmann, M., Fernández, D.M., Daneva, M. *"On the pragmatic design of literature studies in software engineering: an experience-based guideline"*. Empirical Software Engineering, 22(6), pp. 2852–2891, 2017.

[12] Haddaway, N.R., Collins, A.M., Coughlin, D., Kirk, S. *"The role of Google Scholar in evidence reviews and its applicability to grey literature searching*". PLoS One. 10(9), pp. 1–17, 2015.

[13] Mohammad Ghafari, Mortaza Saleh, Touraj Ebrahimi. "*A Federated Search Approach to Facilitate Systematic Literature Review in Software Engineering*", International Journal of Software Engineering & Applications, 3(2), pp. 13-24, 2012.

[14] Wohlin, C. *"Second-generation systematic literature studies using snowballing*". In proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering, EASE'16, ACM, Limerick, Ireland, pp. 1–6, 2016.

[15] Lanxin Yang, He Zhang, Haifeng Shen, Xin Huang, Xin Zhou, Guoping Rong, Dong Shao.
"*Quality Assessment in Systematic Literature Reviews: A Software Engineering Perspective*", Information and Software Technology, Vol. 130, pp. 1-24, 2021.

[16] David Bowes, Tracy Hall, and Sarah Beecham. "*SLuRp: a tool to help large complex systematic literature reviews deliver valid and rigorous results*". In Proceedings of the 2nd international workshop on Evidential assessment of software technologies (EAST'12), Lund Sweden, ACM, pp. 33–36, 2012.

[17] Fabbri, S., Octaviano, F., Silva, C., Di Thommazo, A., Hernandes, E., and Belgamo, A. "*Improvements in the Start tool to better support the systematic review process"*. In Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering (EASE'16), Limerick, Ireland, ACM, pp. 1-5, 2016.

[18] Ana M. Fernández-Sáez, Marcela Genero, Francisco P. Romero. "*SLR-Tool: A Tool for Performing Systematic Literature Reviews*". In Ernest Teniente, Silvia Abrahão, editors, XV Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2010), Valencia, Spain, September 7-10, 2010. Actas. pp. 329-332, IBERGARCETA Pub. S.L., 2010.

[19] Barn Balbir, Raimondi Franco, Athiappan Lalith and Clark. "*Slrtool: a tool to support collaborative systematic literature reviews*". In Proceedings of the 16th International Conference on Enterprise Information Systems (ICEIS'2014) - 16th International, Lisbon, Portugal, SCITEPRESS - Science and Technology Publications, pp. 440, 447, 2014.

[20] Brian E. Howard, Jason Phillips, Arpit Tandon, Adyasha Maharana, Rebecca Elmore, Deepak Mav, Alex Sedykh, Kristina Thayer, B. Alex Merrick, Vickie Walker, Andrew Rooney, Ruchir R. Shah, "*SWIFT-Active Screener: Accelerated document screening through active learning and integrated recall estimation*", Environment International, Vol. 138, 2020,

[21] Claudio Bustos Navarrete, María Gabriela Morales Malverde, Pedro Salcedo Lagos, Alejandro Díaz Mujica, "*Buhos: A web-based systematic literature review management software*", SoftwareX, Vol. 7, pp. 360-372, 2018.

[21] István Bíró, Jácint Szabó, *Latent Dirichlet Allocation for Automatic Document Categorization*. In: Buntine W., Grobelnik M., Mladenić D., Shawe-Taylor J. (eds) *Machine Learning and Knowledge Discovery in Databases*. ECML PKDD 2009. Lecture Notes in Computer Science, vol 5782. Springer, Berlin, Heidelberg, 2009

[22] Théo Francesiaz, Raphaël Graille, and Brahim Metahri, *Introduction aux modèles probabilistes utilisés en fouille de données*, Rapport IMAG, 2015.

[23] Lakhdari Salsabil, Saidi Amaria. *Étude des techniques d'apprentissage semi-supervisé par regroupement*, mémoire master en génie biomédical, Université Abou Bekr Belkaid, Tlemcen, Algérie, 2017.

[24] Xiaojin Zhu, *Semi-Supervised Learning Literature Survey*, technical report TR 1530, University of Wisconsin, July 2008

[25] Kristin P. Bennett and Ayhan Demiriz. *Semi-supervised support vector machines*. In Proceedings of the 1998 conference on Advances in neural information processing systems II. MIT Press, Cambridge, MA, USA, pp. 368–374, 1999.

[26] Michael W Browne, *Cross-Validation Methods*, Journal of Mathematical Psychology, 44(1), pp. 108-132, 2000.

[27] Shai Shalev-Shwartz and Shai Ben-David, *Understanding Machine Learning: From Theory to Algorithms,* Cambridge University Press, USA, 2014.

[28] Dieste Oscar and Padua Anna Griman, *Developing search strategies for detecting relevant experiments for systematic reviews.* In the 2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement. pp. 215–224. IEEE Computer Society, Los Alamitos, CA, USA, 2007

[29] Fricke Suzanne, *Semantic scholar*. Journal of the Medical Library Association, 106(1), pp. 145-147, 2018

[30] Salatino Angelo A., Thanapalasingam Thiviyan, Mannocci Andrea, Osborne Francesco and Motta Enrico, *The computer science ontology: A large-scale taxonomy of research areas*. In Vrandecic, D., Bontcheva, K., Suàrez-Figueroa, M.C., Presutti, V., Celino, I., Sabou, M., Kaffee, L.A., Simperl, E. (eds.) The Semantic Web – ISWC 2018. Springer International Publishing, Cham, pp. 187–205, 2018

[31] Molléri Jefferson Seide, Petersen Kai and Mendes Emilia, Tow*ards understanding the relation between citations and research quality in software engineering studies*. Scientometrics, 117(3), pp. 1453–1478, 2018

[32] Oghbaie Marzieh and Mohammadi Zanjireh Morteza, *Pairwise document similarity measure based on present term set.* Journal of Big Data, 5(52), pp. 1-23, 2018

[33] Vassilvitskii Sergei and Arthur David, *k-means++: The advantages of careful seeding*. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, New Orleans Louisiana, USA, ACM, pp. 1027–1035, 2006

[34] Blei David. M., Ng Andrew Y. and Jordan Michael I., *Latent dirichlet allocation*. The Journal of machine Learning research, 3, pp. 993–1022, 2003

[35] Ghawi Raji and Pfeffer Jürgen, *Efficient hyperparameter tuning with grid search for text categorization using knn approach with bm25 similarity*. Open Computer Science, 9(1), pp. 160–180, 2019

[36] Shahin, M., Babar, M.A., Zhu, L. *Continuous integration, delivery and deployment: a systematic review on approaches, tools, challenges and practices*. IEEE Access. 5, pp. 3909–3943, 2017.

[37] Santos, J.A.M., Rocha-Junior, J.B., Prates, L.C.L., do Nascimento, R.S., Freitas, M.F., de Mendonça, M.G.: *A systematic review on the code smell effect*. J. Syst. Software. 144, pp. 450–477, 2018.

[38] Goulão, M., Amaral, V., Mernik, M. *Quality in model-driven engineering: a tertiary study*. Software Qual. J. 24(3), pp. 601–633, 2016

[39] Alan Dennis, Barbara Haley Wixom and David Tegarden. *Systems Analysis and Design with UML*. 4th Edition, Wiley Publishing, 2012.

[40] Younes Derfoufi. *Programmation en langage Python.* Support de cours de CRMEF OUJDA, Maroc, 2019.

WEBOGRAPHIE

[W1] Sysrev: *The first fair review platform*, https://blog.sysrev.com/ [consulté Mai 2021]

[W2] SRA *: Systematic Review Accelerator*, https://sr-accelerator.com/ [consulté Mai 2021]

[W3] DoCTER*: Leverage a standalone text mining software application to prioritize documents for expert review,* https://www.icf.com/technology/docter [consulté Mai 2021]

[W4] IBM Cloud Education, *Machine learning*, https://www.ibm.com/cloud/learn/machine-learning, [consulté Mai 2021].

[W5] Universalis encyclopedie, *Apprentissage profond ou deep learning, Réseaux de neurones formels*, https://www.universalis.fr/encyclopedie/apprentissage-profond-deep-learning/2-reseaux-de-neurones-formels/, [consulté Mai 2021].

[W6] Ebergementwebs, *Qu'est-ce que l'apprentissage automatique et pourquoi Important ?,* https://www.hebergementwebs.com/nouvelles/qu-est-ce-que-l-apprentissage-automatique-et-pourquoi-est-il-important, [consulté Mai 2021].

[W7] Lucas Scott, *Data Preparation for Machine Learning: The Ultimate Resource Guide*, 2020, https://lionbridge.ai/articles/data-preparation-for-machine-learning-the-ultimate-resource-guide/, [consulté Mai 2021].

[W8] MathWorks.com, *Machine Learning: 3 choses à savoir*, https://fr.mathworks.com/discovery/machine-learning.html, [consulté Mai 2021].

[W9] DataRobot.com, *Unsupervised Machine Learning*, https://www.datarobot.com/wiki/unsupervised-machine-learning/, [consulté Mai 2021].

[W10] Mquantin, *Illustration du déroulement de l'algorithme des k-means*, 2017, https://upload.wikimedia.org/wikipedia/commons/f/fb/K-means.png?uselang=fr, [consulté Mai 2021].

[W11] Analyticsinsights.io, *K-means: l'analyse par k-means améliore la precision des clusters*, https://analyticsinsights.io/k-means/#:~:text=L'analyse%20par%20K%2Dmeans,am%C3%A9liore%20la%20pr%C3%A9cision%20des%20clusters, [consulté Mai 2021].

[W12] CSO : Computer Science Ontology: https://cso.kmi.open.ac.uk/home, [consulté Juillet 2021].