Ministère de l'enseignement Supérieur et de la recherche Scientifique
وزارة التعليم العالي والبحث العلمي

Badji Mokhtar Annaba University
Université Badji Mokhtar – Annaba
Faculté des Technologie

Département d'informatique

جامعة باجي مختار– عنابـــــة
كلية تكنولوجيا
قســـــــم الإعلام الآلي

Thèse

Présentée pour obtenir le diplôme de Doctorat en LMD

Spécialité : Gestion et Analyse des Données Massives

Domaine : Informatique

Par :
Sonia Guehria

Thème :

Les Approches Ensemblistes pour la Classification Multi-Label des Données à Grandes Echelles

N°	Nom et prénom	Grade	Etablissement	Qualité
01	Nadir Farah	Prof	Université Badji Mokhtar –Annaba	Président
02	Habiba Belleili	Prof	Université Badji Mokhtar –Annaba	Directrice de thèse
03	Faiz Maazouzi	MCA	Université Mohamed-Chérif Messaadia-Souk Ahras	Examinateur
04	Naouel Zemmal	MCA	Université Mohamed-Chérif Msesaadia- Souk Ahras	Examinateur
05	Abdel Wahid Alalga	MCA	Université Badji Mokhtar –Annaba	Examinateur
06	Akila Djebbar	MCA	Université Badji Mokhtar –Annaba	Examinateur
07	Nabiha Azizi	Prof	Université Badji Mokhtar –Annaba	Invitée

Année universitaire : *2023-2024*

بسم الله الرحمن الرحيم

REMERCIEMENTS

Je tiens à exprimer, avant tout, ma gratitude infinie envers Allah Tout-Puissant,
qui m'a accompagné tout au long de ce chemin, m'accordant la volonté,
la détermination et la persévérance pour surmonter les obstacles
et franchir les seuils de mes destinations.

Mes remerciements les plus sincères à ma directrice de thèse, Pr. Habiba Belleili,
pour ses précieuses orientations et son soutien indéfectible, ainsi qu'à mon invitée
d'honneur, Pr. Nabiha Azizi, pour ses conseils avisés et son suivi rigoureux tout au long
de mes recherches. Leurs disponibilités constantes et la confiance qu'elles m'ont accordée
m'ont permis d'aller au-delà de mes ambitions.

Je remercie profondément Pr. Nadir Farah pour l'honneur
qu'il m'a fait en acceptant de présider ce jury
et pour m'avoir chaleureusement accueillie au sein du laboratoire LABGED,
où j'ai eu l'opportunité de bénéficier d'un environnement de travail stimulant.

Je suis également très honorée et reconnaissante envers
Dr. Faiz Maazouzi et Dr. Naouel Zemmal de l'université de Souk Ahras,
ainsi que
Dr. Abdelwahid Alalga et Dr. Akila Djebbar de l'université d'Annaba,
qui ont accepté d'examiner et d'évaluer mon travail.
Leur expertise et leurs remarques constructives seront d'une grande utilité
pour l'amélioration de ma recherche.

Mes remerciements s'étendent également à tous les enseignants, les responsables
et membres du personnel du département d'Informatique de notre université,
pour leur gentillesse et leur disponibilité.

Enfin, mes remercîments ne peuvent s'achever sans exprimer ma profonde
gratitude à ma famille, qui a fait preuve d'une patience et d'un
soutien inestimables tout au long de ce magnifique voyage.

Merci à vous tous…..

Résumé

La Classification Multi-Label est largement rencontré dans de nombreuses applications du monde réel, suscitant une attention considérable par la communauté du Machine Learning et du Data Mining au cours des dernières décennies. Ce paradigme de classification permet d'associer simultanément plusieurs labels à une instance. Bien que de nombreuses recherches et expérimentations aient été menées pour développer des méthodes d'apprentissage Multi-Label, plusieurs défis scientifiques ont émergé, notamment le déséquilibre des classes, les dépendances entre les labels et la grande dimensionnalité de l'espace de sortie. Pour remédier à ces défis, l'approche Ensemble a été développée, démontrant ainsi sa grande efficacité dans divers domaines d'application.

Malgré les avancées notables réalisées par l'approche Ensemble dans le domaine de la Classification Multi-Label, aucune méthode ensembliste n'a pu prouver sa supériorité par rapport aux autres pour résoudre la majorité des problèmes spécifiques liés au domaine. En effet, la performance de chaque méthode dépend de plusieurs facteurs critiques, notamment les caractéristiques complexes des données multi-label utilisées, les forces et faiblesses des classifieurs testés, ainsi que la mise à l'échelle de certaines données.

L'étude de recherche menée dans cette thèse s'articule autour de deux nouvelles approches ensemblistes: *ConfBoost* et *DisEMLC*. L'objectif visé par ces deux approches est de développer des systèmes robustes et généralisables, capables de relever les défis surmentionnés, tout en assurant la scalabilité des DML.

L'approche *ConfBoost* constitue un méta-modèle qui combine plusieurs Classifieurs Ensemble Multi-Label complémentaires et hétérogènes, tels qu'ECC, EPS, RAKEL, RF-PCT. Cette approche repose sur un paradigme de Stacking pondéré, utilisant une pondération des labels couplée à des seuils ajustés. Des expériences approfondies menées sur des ensembles de données Multi-Label de référence ont mis en évidence l'efficacité et le potentiel de *ConfBoost* en tant que méthode avancée pour les tâches de Classification Multi-Label.

En revanche, *DisEMLC* est une approche distribuée parallèle qui utilise une architecture *MapReduce*. Elle partage avec *ConfBoost* les mêmes classifieurs ensembles, intégrés au niveau des Mappers, et applique un mécanisme de pondération de labels couplé à des seuils ajustés après la phase de Réduction. En tirant parti de l'informatique distribuée parallèle et des méthodologies de classification innovantes, notre système vise à surmonter les problèmes d'évolutivité fréquemment rencontrés dans les tâches de CML.

Ainsi, dans un environnement séquentiel, l'approche *ConfBoost* offre des gains en précision et permet une gestion plus efficace des différents défis posés par le domaine, mais elle est limitée par sa scalabilité et son temps d'exécution. En revanche, dans un environnement distribué, l'approche *DisEMLC* prend toute sa portée. Elle devient plus évolutive, plus efficace, et mieux adaptée aux grands ensembles de données Multi-Label, grâce à la parallélisation des calculs et à la bonne gestion des ressources.

Mots clefs. Approche distribuée ; Apprentissage Ensemble ; Classification Multi-Label ; MapReduce ; Label Pondéré ; Stacking ; Seuillage Ajusté.

Abstract

The Multi-Label Classification paradigm is widely encountered in numerous real-world applications, attracting considerable attention from the Machine Learning and Data Mining communities over the past decades. This classification paradigm allows multiple labels to be assigned to an instance simultaneously. Although extensive research and experimentation have been conducted to develop Multi-Label Learning methods, several scientific challenges have emerged, including imbalance class, label dependencies, and the high dimensionality of the output space. To address these challenges, the Ensemble approach has been developed, demonstrating its great effectiveness in various application domains

Despite the significant advances made by the Ensemble approach in the Multi-Label Classification field, no ensemble method has been conclusively proven to be superior in solving most of the domain-specific problems. Indeed, the performance of each method depends on several critical factors, including the complex characteristics of multi-label data, the strengths and weaknesses of the classifiers used, and the scalability of the tested data.

The research study conducted in this thesis focuses on two new ensemble approaches: *ConfBoost* and *DisEMLC*. The goal of these approaches is to develop robust and generalizable systems capable of addressing the aforementioned challenges while ensuring the scalability of Multi-Label Classification.

The research study conducted in this thesis focuses on two new assembling approaches: *ConfBoost* and *DisEMLC*. The goal of these approaches is to develop robust and generalizable systems capable of addressing the aforementioned challenges while ensuring the scalability of Multi-Label Classification.

ConfBoost approach is a Meta-Model that combines several complementary and heterogeneous Multi-Label Ensemble classifiers, such as ECC, EPS, RAKEL, RF-PCT. This approach relies on a weighted stacking paradigm, using label weighting coupled with adjusted thresholds. In-depth experiments on benchmark datasets have highlighted the effectiveness and potential of ConfBoost as an advanced method for Multi-Label Classification tasks.

DisEMLC, on the other hand, is a distributed approach using the *MapReduce* architecture, which shares the same ensemble classifiers with *ConfBoost*, integrated at the Mapper level. After the Reduction phase, it applies a label weighting mechanism coupled with adjusted thresholds. By leveraging distributed computing and innovative classification methodologies, our system helps overcome the scalability issues commonly encountered in MLC tasks.

Thus, in a sequential environment, *ConfBoost* approach offers gains in accuracy and enables more effective management of the various challenges posed by the domain, but it is limited by its scalability and execution time. In contrast, in a distributed environment, the *DisEMLC* approach reaches its full potential. It becomes more scalable, more efficient, and better suited to large multi-label datasets, thanks to the parallelization of computations and effective resource management.

Keywords: distributed approach ; Ensemble Learning ; Multi-Label Classification ; MapReduce; Weighted Label ; Stacking ; Adjusted Thresholding.

ملخـــــص

لقد أضحت مشكلة التصنيف متعدد الفئات تفرض نفسها بقوة في مختلف المجالات في الوقت الراهن، الأمر الذي استدعى اهتمام رواد التعلم الآلي واستكشاف البيانات بهذه المشكلة خلال السنوات القليلة الماضية، حيث يسمح هذا النموذج بإعطاء عدة تصنيفات لمثال واحد في نفس الوقت.

رغم القيام بالعديد من الأبحاث والتجارب من أجل تطوير طرق التعلم للتحكم في هذا النموذج، فقد واجهتهم تحديات علمية عديدة، لاسيما اختلال توازن فئات التصنيف، الترابط بين هذه الفئات والحجم الكبير للمخرجات. لمعالجة هذه التحديات فقد تم خلق مقاربة تجميعية للتصنيفات، هذه الأخيرة أثبتت فعاليتها في مختلف المجالات والتطبيقات.

بالرغم من التقدم الكبير الذي أحرزته المقاربة التجميعية للتصنيف متعدد الفئات، إلا أن هذا الوضع لم ينجم عنه تفوق طريقة عن أخرى في حل معظم المشكلات المتعلقة بالمجال، في المقابل تعود فعالية كل طريقة إلى العديد من العوامل الحاسمة، لاسيما الخصائص المعقدة الخاصة ببيانات تعدد الفئات، نقاط قوة وضعف المصنفات المستخدمة، بالإضافة إلى البيانات الضخمة المستعملة.

يرتكز موضوع البحث في هذه الأطروحة على مقاربتين جديدتين للتجمّعات *ConfBoost* و *DisEMLC* ، حيث يكون الهدف منهما هو تطوير أنظمة قوية يمكن تعميمها وتكون قادرة على معالجة التحديات المسجلة، مع ضمان قابلية التعامل مع بيانات ضخمة في التصنيف متعدد الفئات.

تعدّ *ConfBoost* نموذجًا مركبا (meta-model) يجمع العديد من المصنّفات التجميعية متعددة الفئات المتنوعة والمكملة لبعضها البعض (ECC، EPS، RAKEL، RFPCT)، تعتمد هذه الطريقة على نموذج التكديس الموزون، الذي يقوم على موازنة الفئات مع إضافة عتبات معدلة، كما أبرزت عدة تجارب على مجموعات بيانات مرجعية فعالة وعلى الإمكانيات التي يتمتع بها *ConfBoost* كطريقة متقدمة تؤدي مهام التصنيف متعدد الفئات.

أما بالنسبة لطريقة *DisEMLC* النّي تستعمل نموذج *MapReduce* فهي تسعمل نفس المصنفات التجميعية التي يستعملها *ConfBoost* والتي تكون مدمجة على مستوى الموزعين (Mappers). بعد عملية التجميع تطبق هذه الطريقة آلية موازنة البيانات المضافة إلى العتبات المعدلة، مع الأخذ بعين الاعتبار الألية الموزعة وطرق التصنيف المبتكرة، كما يساعد هذا النظام على التغلب على مشاكل قابلية التوسع التي تواجهها وظائف التصنيف متعدد الفئات.

في ظل بيئة تسلسلية، توفر طرق التجميع للتصنيف متعدد الفئات تحسينات في الدقة وتساعد على مواجهة مختلف تحديات المجال، لكنها تبقى محدودة من حيث القدرة على التوسع ووقت التنفيذ، من ناحية أخرى، في بيئة موزعة تأخذ هذه الطرق كامل أبعادها، حيث تكون من جهة قابلة للتوسع والكفاءة، ومن جهة أخرى تكون أكثر ملائمة لمجموعات البيانات الكبرى، بفضل التوازي في العمليات وإدارة الموارد بشكل أكثر فعالية.

الكلمات المفتاحية: النهج الموزع ؛ التعلم التجميعي ؛ التصنيف متعدد العلامات ؛ MapReduce ؛ موازنة العلامات ؛ التراكم؛ تعديل العتبة.

TABLE DES MATIERES

Liste des Figures

Liste des Tables

Liste des Algorithmes et Equations

Liste des Abréviations

AA	Adaptation d'Algorithme
AvA	All versus All
BR	Binary Relevance
BP-MLL	Back Propagation for Multi-Label Learning
BVCL	Base des Vecteurs des Canfiance des Labels
CC	Classifier Chain
CML	Classification Multi-Label
CEML	Classifieurs Ensemble Multi-Label
DisEMLC	Distributed approach for Ensemble MLC
DML	Données Multi-Label
DT	Decision Trees
EBR	Ensemble of Binary Relevance classifiers
ECC	Ensemble of Classifier Chains
EPS	Ensemble of Pruned Sets
HOMER	Hierarchy Of Multi-label classifiER
k-NN	k-Nearest Neighbors
LP	Label Powerset
ML C4.5	Multi-Label C4.5
ML-RBF	Multi-Label Radial Base Function
MLS	Multi-label Stacking
PCT	Predictive Clustering Trees
PS	Power Set
RAKEL	RAndom k-labELsets
RF-PCT	Random Forest of Predictive Clustering Trees
RFML-C4.5	Random Forest of ML-C4.5
SVM	Support Vector Machine
TP	Transformation de Problème
TREMLC	Triple Random Ensemble for Multi-label Classification

1

INTRODUCTION GENERALE

Sommaire

1.1 Présentation et motivation de la thèse

L'intelligence artificielle (IA) constitue une science multidisciplinaire qui représente un domaine de recherche en plein essor et prometteur dans divers domaines d'application. La diversité de ses techniques et approches a permis de développer des systèmes intelligents favorisant une prise de décision rapides et efficaces. L'apprentissage automatique, l'une des branches centrales de l'IA, s'est révélé être un outil puissant pour résoudre une multitude de défis scientifiques. Grâce à des algorithmes sophistiqués et à l'exploitation de grands ensembles de données, l'utilisation de l'apprentissage automatique a donné aux machines la possibilité de construire des modèles à partir de différentes données pour prendre des décisions. Sa capacité qui a largement évolué en fonction des données disponibles, a conduit à des avancées significatives dans différents domaines tels que la santé, la finance, l'industrie, et bien d'autres encore.

Au cours des dernières décennies, la taille et la difficulté de l'extraction de connaissances variées, combinées à l'augmentation exponentielle des données provenant de différentes sources, ont profondément transformé le destin de l'information. Cette croissance spectaculaire des données, souvent désignée sous le terme de Big Data, a suscité un besoin impératif de développer de nouvelles techniques d'apprentissage capables d'analyser et de traiter efficacement des données complexes et diversifiées. En conséquence, de nombreuses données ont été référencées en Multi-Label, favorisant ainsi l'émergence d'un nouveau paradigme d'apprentissage connu sous la désignation de l'Apprentissage Multi-Label. Ce nouveau paradigme a ouvert des perspectives considérables pour la résolution de problèmes complexes et une prise de décision plus éclairée dans une multitude de domaines.

L'apprentissage Multi-Label, qui vise à former un modèle prédictif à partir des données Multi-Label, repose sur deux axes principaux : la Classification Multi-Label et le Classement Multi-Label. L'objectif de la Classification est de construire un modèle capable de générer une liste de labels pertinents pour un exemple jamais rencontré. En revanche, le Classement cherche à établir un modèle attribuant à chaque nouvel exemple une liste ordonnée de préférences parmi un ensemble prédéfini de labels. Il convient de mentionner que nous nous concentrons dans cette thèse sur la tâche de la Classification Multi-Label (CML) dans le but de développer des modèles et des algorithmes capables de générer avec précision et pertinence une prédiction Multi-Label associés à chaque exemple.

Contrairement aux tâches de la classification conventionnelles, où chaque instance est associée à une seule étiquette de classe (qu'elle soit binaire ou multi-classe), le problème de CML permet d'attribuer à chaque instance une liste prédéfinie de labels. Par exemple, en génomique fonctionnelle, un gène peut être lié à plusieurs fonctions, telles que le métabolisme, la synthèse des protéines et la transcription. En outre, la CML a émergé dans le but de surmonter les limitations de la classification mono-label qui ne permet pas la modélisation des relations multiples entre les labels lors du traitement de données complexes. De plus, ce paradigme de classification est plus adaptable et plus cohérent pour mieux représenter la complexité inhérente de certaines natures de données, offrant ainsi de nouvelles perspectives pour résoudre des problèmes réels dans divers domaines.

Le problème de CML a suscité un vif intérêt au cours de la dernière décennie au sein de la communauté du Machine Learning et du Data Mining. Initialement adaptée à des applications de catégorisation de texte [1], [2], [3], il s'est étendu à divers domaines, tels que la classification fonctionnelle des gènes [4], [5], l'annotation sémantique des images [6], de vidéos [7], [8], [9], et de l'audio [10]. De plus, il a été appliqué à la recommandation de tags [3], à la fouille de données web et de règles [11], [12], et même à l'étiquetage de cartes [13]. Récemment, la CML a été largement utilisée dans la recommandation de camions de restauration [14] et dans la reconnaissance des activités humaines dans les maisons intelligentes [15]. Par ailleurs, de nombreuses recherches et expérimentations ont conduit au développement d'une large variété de méthodes pour CML.

Une typologie initiale proposée par Tsoumakas et Katakis [16] a permis de catégoriser les méthodes de CML en deux grandes familles à savoir : *l'approche de Transformation de Problème (TP)* et *l'approche d'Adaptation d'Algorithme (AA)*. La première approche représente des algorithmes indépendants qui transforment le problème de CML en plusieurs problèmes de classification mono-label. Tandis que, la deuxième approche adapte des algorithmes de classification mono-label afin de traiter des Données Multi- Label (DML).

Malgré des recherches importantes consacrées au développement des méthodes de CML visant à obtenir des systèmes robustes et performants, plusieurs défis scientifiques ont émergé dans le domaine. Il s'agit notamment, de la difficulté de modéliser les dépendances entre labels, le déséquilibre des classes ainsi que le fléau de la dimensionnalité causé par l'augmentation constante du nombre des labels.

Pour répondre à ces défis, une troisième famille d'approche pour CML a été développée à partir des deux approches précédentes, désignée par *l'approche Ensemble* [17], [18], [19], [20]. Cette approche repose sur le principe de combiner plusieurs modèles de CML à l'aide de techniques d'agrégation judicieuses, dans le but d'améliorer la robustesse et les performances globales du système.

L'approche Ensemble a marqué une avancée significative dans le domaine de la CML, ouvrant de nouvelles perspectives pour améliorer la robustesse des systèmes d'apprentissage qui traitent des données complexes et variées. Cette approche a non seulement démontré ses capacités à accroître l'efficacité prédictive des modèles individuels [19], mais elle a également permis de résoudre des problèmes spécifiques liés aux DML.

1.2 Problématique et contribution de l'étude

Malgré les avancées notables réalisées par l'approche Ensemble dans le domaine de la CML, aucune méthode ensembliste n'a pu démontrer sa supériorité par rapport aux autres pour résoudre la majorité des problèmes spécifiques liés au domaine. En effet, l'efficacité de chaque méthode dépend de plusieurs facteurs critiques :

- ✓ Caractéristiques des données : Les caractéristiques complexes des ensembles DML présentent des défis particuliers tels que les dépendances entre labels, le déséquilibre des classes, et la dimensionnalité élevée de l'espace de sortie. Ces caractéristiques ont un impact significatif sur les performances prédictives du modèle global.
- ✓ Forces et Faiblesses des Classifieurs : Les points forts et les faiblesses des classifieurs ensemble, ainsi que leur capacité à gérer les complexités des DML, jouent un rôle crucial dans l'efficacité globale du modèle final.
- ✓ Mise à l'échelle des données : Les ensembles de données à grande échelle, caractérisés par un grand nombre d'instances avec un nombre modéré de labels, constitue un autre défi majeur pour les modèles ensemblistes. Cette dimension supplémentaire a tendance de compliquer le traitement et l'analyse des données, augmentant ainsi les exigences en termes de calcul et de ressources.

De plus, la littérature a rarement exploré les techniques d'agrégation pour construire les ensembles.

Leur construction été souvent réalisées par des stratégies de combinaison simples telles que, le vote majoritaire, la moyenne et la moyenne pondérée, ou de techniques avancées comme, le Bagging, le Boosting et le Stacking. Néanmoins, ces techniques n'ont pas suffisamment pris en compte les caractéristiques complexes des DML, ce qui a limité leur impact sur les performances prédictives des systèmes d'apprentissage.

En tenant compte de ces considérations, l'étude de recherche menée dans cette thèse s'articule autour de deux contributions principales :

- Premièrement, nous proposons une nouvelle approche ensembliste pour la CML, nommée *ConfBoost*, qui représente un méta-modèle basé sur la collaboration de classifieurs ensemble multi-label (CEML) ECC, EPS, RAKEL et RFPCT, chacun étant capable de résoudre un problème spécifique posé par le domaine. L'approche proposée, repose sur un paradigme de Stacking pondéré, utilisant la pondération des labels couplée à des seuils ajustés. L'intégration de la pondération de labels en fonction de leurs scores de confiance vise, d'une part, à générer des prédictions plus pertinentes et à améliorer la précision en atténuant l'impact des labels non pertinents pendant le processus de Stacking. D'autre part, l'attribution de poids plus élevés à certains labels permet une meilleure discrimination et adaptabilité pour capturer les relations complexes entre labels. Par ailleurs, l'application de seuils ajustés permet au modèle de générer des prédictions plus précises en se concentrant sur les labels les plus pertinents et informatifs.

- Deuxièmement, pour tester la portée des modèles ensemble sur des DML à grande échelle, nous avons proposé une approche distribuée et parallèle, appelée *DisEMLC*, intégrant les mêmes CEML utilisés par *ConfBoost*, avec une intégration d'un mécanisme de pondération des labels. La diversité des CEML intégrés au niveau des mappers permet à *DisEMLC*, d'une part, de tirer parti des forces spécifiques de chaque classifieur pour améliorer la précision globale. D'autre part, de traiter les données en parallèle, accélérant ainsi le processus de classification. Par ailleurs, l'adaptation de la pondération des labels après la phase de réduction permet d'ajuster l'importance relative de chaque label en fonction de sa fréquence dans les données. Enfin, l'application d'une fonction de seuillage permet de filtrer les labels moins pertinents ou peu fiables, améliorant ainsi la précision des résultats finaux en se concentrant sur les labels les plus importants.

1.3 Structure de la thèse

Complété par cette introduction, le manuscrit est organisé en cinq chapitres, auxquels s'ajoute une conclusion ainsi que des perspectives futures, tel que illustré dans la Figure 1.2.

- Chapitre 2, intitulé ***"Classification Multi-Label"***, dresse un état de l'art des méthodes existantes de la CML telles qu'elles sont présentées dans la littérature. Nous nous appuyons principalement sur une typologie initiale élaborée par Tsoumakas et Katakis (2007) pour classer ces méthodes en deux grandes familles : Transformation de Problème (TP) et Adaptation d'Algorithme (AA). Ensuite, nous discutons des défis scientifiques introduits par la CML, suivis d'une exploration de ses divers domaines d'application. De plus, nous examinons les caractéristiques des DML, les mesures d'évaluation sur lesquelles nous nous appuyons pour évaluer nos différents travaux. A la fin de ce chapitre, nous aborderons le concept de la CML à grande échelle.

- Chapitre 3, intitulé ***" Classification en Ensemble"***, offre un aperçu des méthodes d'ensemble de pointe pour la CML, après avoir examiné les méthodes ensemblistes conventionnelles. Ensuite, nous introduisons les différentes taxonomies de ces méthodes, suivies d'une proposition d'une nouvelle catégorisation. Enfin, nous clôturons ce chapitre par une évaluation expérimentale des performances des différentes méthodes ensemble populaires sur divers ensembles de DML en Bioinformatique. Nous détaillant les résultats de ces expériences selon diverses mesures d'évaluation, afin de déterminer l'efficacité et la robustesse de chacune.

- Chapitre 4, intitulé ***"Optimisation des Performances de la Classification Multi-Label par un Méta-Modèle "***, présente une approche avancée d'ensemble pour les tâches de CML, nommée *ConfBoost*. Après avoir détaillé l'architecture de l'approche proposée ainsi que ses différentes composantes, nous présentons une étude expérimentale comprenant la description des ensembles DML testés et les paramètres expérimentaux sélectionnés. Ainsi, nous analysons de prêt les résultats expérimentaux selon trois scénarios différents: i) Une comparaison des performances des méthodes ensemblistes individuelles sur des ensembles de DML de domaines variés, ii) Une comparaison des performances des différentes approches combinées, et iii) Une comparaison des performances de *ConfBoost* avec d'autres méthodes connexes. A la fin de ce chapitre, nous soulignons les implications de l'approche *ConfBoost* dans le contexte de la CML.

- Chapitre 5, intitulé ***" Approche Distribuée Parallèle basée sur des Classifieurs Ensemble Multi- Label "***, présente une nouvelle approche distribuée et parallèle appelée, *DisEMLC*, fondée sur les mêmes CEML utilisés par *ConfBoost*, pour traiter des ensembles DML à grande échelle. Après avoir donné un aperçu sur le concept de Big Data et décrire le fonctionnement du framework *MapReduce*, nous présentons l'architecture de l'approche proposée *DisEMLC*. Ensuite, dans la section de l'étude expérimentale, nous présentons les ensembles de DML testés et de mesures d'evaluation utilisées. Les résultats expérimentaux obtenus sont analysés et discutés à travers deux expériences distinctes : i) Une comparaison des performances entre l'approche distribuée *DisEMLC* et l'approche séquentielle *ConfBoost*, et ii) Une évaluation du temps d'exécution d'une tâche de CML réalisée par *DisEMLC* dans différentes configurations de cluster. Enfin, nous clôturons ce chapitre par un résumé général, mettant en évidence les performances et les implications de *DisEMLC* dans un environnement de mise à l'échelle.

La Figure 1.1, résume l'organisation générale de la thèse et met en lumière ses différentes contributions. Cette représentation offre une vue d'ensemble claire de sa structure et de son contenu, permettant ainsi de comprendre les apports spécifiques de la recherche.

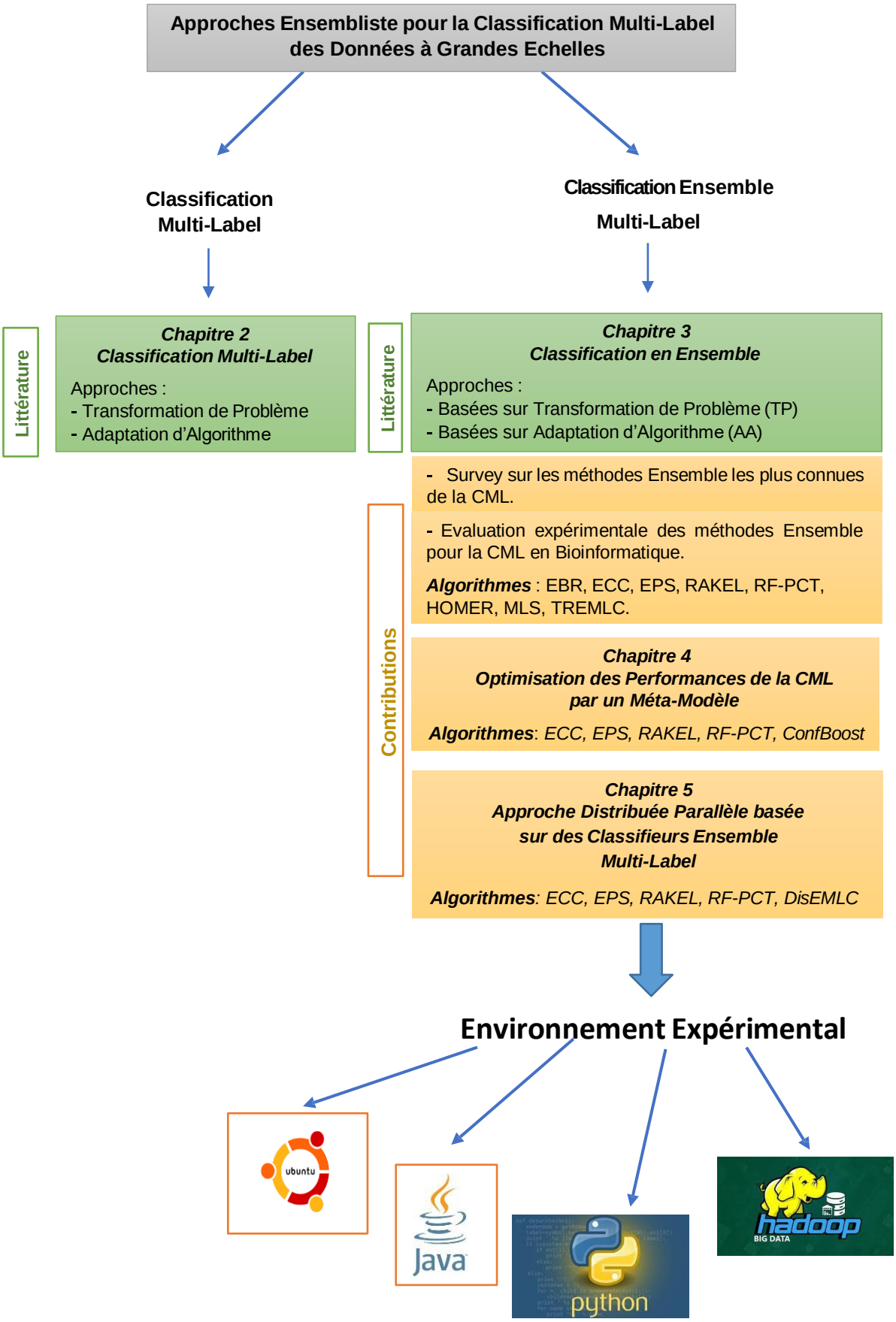

Figure.1.2 Organisation et contributions de la thèse

2

CLASSIFICATION MULTI-LABEL

Sommaire

2.1 Introduction

Le problème de la Classification Multi-Label (CML) a reçu une attention considérable au cours de ces dernières années, principalement en raison de la démultiplication de grandes quantités de données popularisées par le phénomène du Bigdata. Ce problème consiste à attribuer simultanément plusieurs labels à une instance, ce qui va au-delà des scénarios traditionnels de classification. La CML est motivée par la nécessité de modéliser la complexité inhérente des données, dont chacune peut être référencée par plusieurs catégories ou concepts simultanément.

Par exemple, dans le domaine de l'annotation sémantique des images, une seule image peut contenir plusieurs objets ou scènes différents, et il est donc nécessaire d'attribuer plusieurs labels pour décrire correctement son contenu. Comme illustré dans la Figure 2.1, une image d'une scène naturelle peut être associée à plusieurs labels simultanément, tels que l'eau, les montagnes, les fleurs, et le ciel.

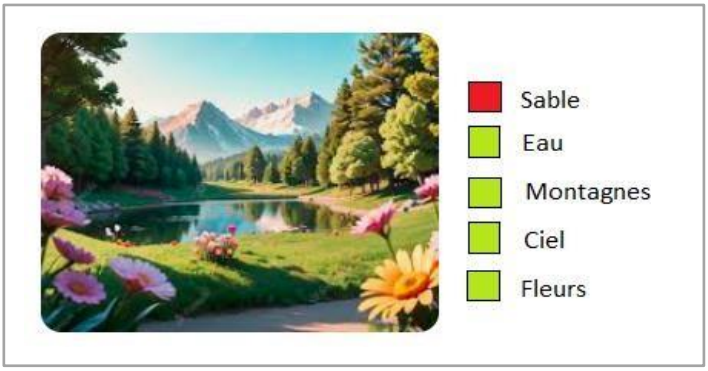

Figure.2.1 Principe de la Classification Multi-Label

Après avoir introduire une définition formelle du problème de CML, ce chapitre examine plusieurs méthodes bien connues des deux approches initiales de CML. Il comprend quatre méthodes de Transformation de Problème (TP) et quatre méthodes d'Adaptation d'Algorithme (AA), ainsi que treize mesures d'évaluation, notamment six mesures basées sur des exemples, six mesures basées sur les labels et la courbe AUC. Cette large gamme des méthodes de CML, recueillie de la littérature [21], [22], [23], [24], [16] offre une vue globale du fonctionnement de ces méthodes, de leurs tendances et de leurs applications. Elle a permet également de situer nos travaux futurs axés sur les méthodes Ensemble construits à partir sur ces approches. En outre, le chapitre discute les défis scientifiques introduits par le domaine, suivi d'une exploration de ses divers domaines d'application.

De plus, nous examinons de près les caractéristiques des DML, ainsi que les mesures d'évaluation que nous utilisons pour évaluer nos différents travaux. Enfin, nous clôturons ce chapitre par un aperçu de l'intégration des DML à grande échelle dans le domaine de la CML.

2.2 Définition formelle

La CML est une tâche d'apprentissage automatique où chaque instance peut être associée simultanément à un ensemble prédéfini de labels. Sa description formelle, telle que présentée par Tsoumakas et al. [16], définit un ensemble de labels L, tel que L= $\{y_j | 1 \leq i \leq n\}$, et un ensemble d'apprentissage multi-label S, tel que S = $\{(xi, Yi) \mid 1 \leq i \leq m\}$, où $xi \in X$ est une instance unique, et $Yi \subseteq L$ représente l'ensemble de labels pertinents associés à xi. L'objectif de cette tâche consiste à concevoir un classifieur multi-label H qui prédit un ensemble de labels à partir d'une nouvelle instance.

La Table 2.1 illustre une représentation d'un ensemble de DML S composé de n instances, tel que S = $\{(x_1, y_1), \cdots, (x_n, y_n)\}$, et k sorties binaires. Cet ensemble de données peut être transformé en un ensemble de jeux de données binaires ou multi-classe adaptée aux méthodes de classification déjà disponibles.

Table.2.1 Exemple d'une base de données Multi-Label

S	X					Y			
S_1	x_1	x_2	---	x_{i-1}	x_i	y_1	y_2	---	y_n
S_2	x_{11}	x_{12}	---	x_{1i-1}	x_{1i}	1	0	---	1
S_3	x_{21}	x_{22}	---	x_{2i-2}	x_{2i}	1	1	---	0
---	---	---	---	---	---	1	1	---	1
S_n	x_{m1}	x_{m2}	---	x_{mi-1}	x_{mi}	0	0	---	1

2.3 Approches de la Classification Multi-Label

Une première catégorisation des méthodes de la CML a permis de distinguer deux grandes familles d'approches: *Transformation de Problème (TP)* et *Adaptation d'Algorithme (AA)* [16]. La première famille comprend des algorithmes indépendants qui transforment le problème de CML en un ou plusieurs problèmes de classification mono- label.

En revanche, la deuxième famille vise à étendre la portée des algorithmes d'apprentissage conventionnels pour résoudre un problème de CML. La Figure 2.2 illustre les méthodes de CML les plus populaires des deux premières approches.

Il est important de noter que les méthodes de la première catégorie se distinguent par leur simplicité et leur polyvalence, ce qui facilite leur application dans divers domaines. De plus, elles visent à réduire la complexité du problème tout en améliorant la précision et la stabilité du processus global. Cependant, les méthodes de la deuxième catégorie sont réputées pour leur efficacité de calcul, mais elles sont conçues pour des domaines d'application très spécifiques, ce qui limite leur généralité.

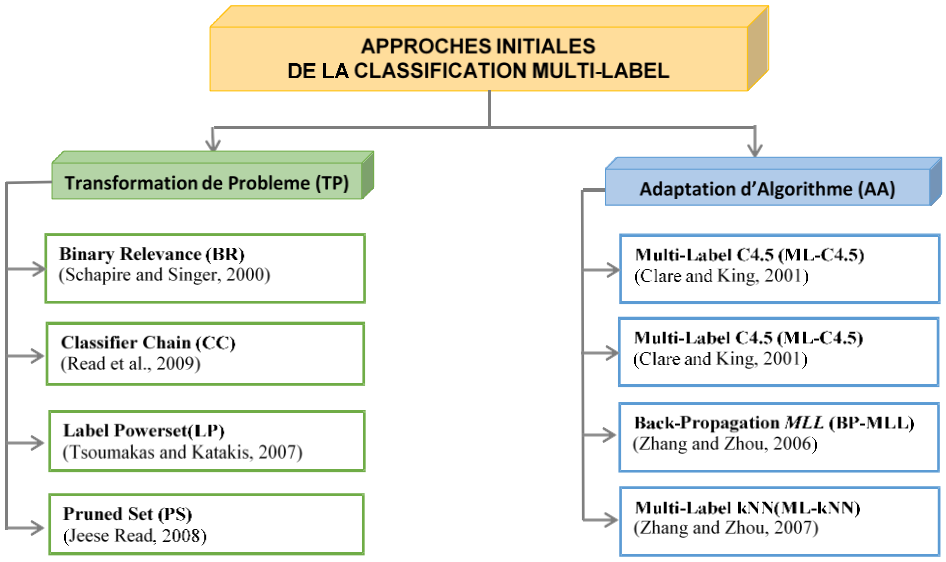

Figure.2.2 Approches initiales de la Classification Multi-Label

2.3.1 Approche de Transformation de Problème (TP)

Cette catégorie d'approche transforme un problème de CML en un ensemble de problèmes de classification mono-label, et applique par la suite un algorithme indépendant pour chaque sous-problème. Les méthodes de cette approche reposent soit sur des techniques de binarization en utilisant les stratégies OvA ou AvA, telles que les méthodes Binary Relevance (BR) [25] et Classifer Chains (CC) [26], soit sur le principe des combinaisons des labels, telles que les méthodes Label Powerset (LP) [18] et Pruned sets (PS) [27].

2.3.1.1 Méthode Binary Relevance (BR)

BR [25] est la méthode la plus simple et la plus populaire qui décompose un problème de CML en plusieurs sous-problèmes de classification binaire en utilisant la stratégie OvA. Cette méthode permet de former un classifieur binaire indépendant par label, ou chaque classifieur C_i est chargé de prédire la présence ou l'absence du label correspondant L_j. Durant la classification d'une nouvelle instance x_i, les predictions de tous les classifieurs binaires sont agrégées pour générer l'ensemble de labels pertinents S_i de cet exemple. La Figure 2.3 illustre le fonctionnement de la méthode BR pour un exemple d'une scène naturelle, sachant que, l'ensemble des labels comporte : Eau, Ciel, et Montagnes.

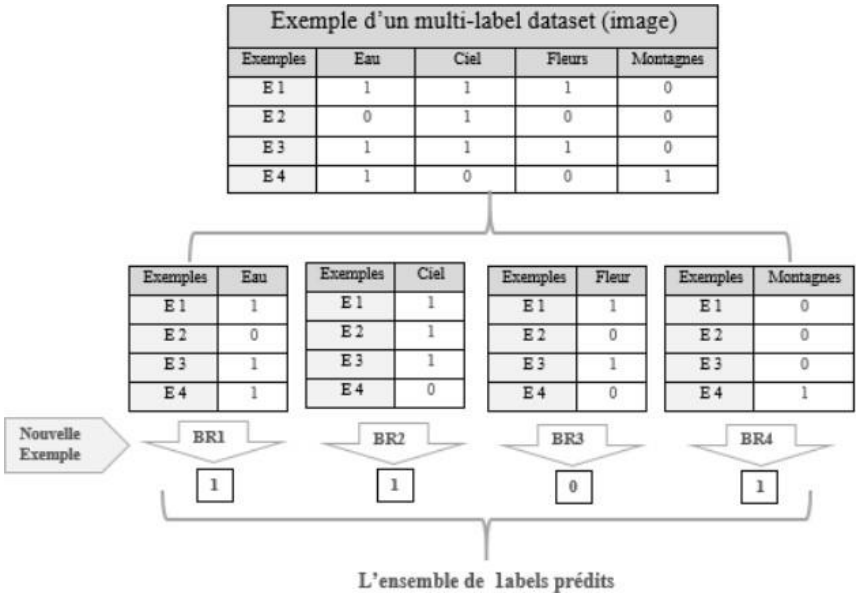

Figure.2.3 Illustration de la méthode BR

2.3.1.2 Méthode Classifier Chains (CC)

CC [26] est une méthode basée sur BR, où une chaine de classifieurs CC_1 , CC_2 ,.., CC_n est créée pendant l'apprentissage. Chaque classifieur binaire CC_i apprend un label L_j et inclut tous les labels associés aux classifieurs précédents dans la chaîne dans son espace d'attributs. Lors de la classification d'une nouvelle instance $\boldsymbol{x_i}$, le processus commence avec le premier classifieur et se déroule le long de la chaîne. Chaque classificateur détermine la probabilité que x_i soit classé dans L_1, L_2, L_3,..., L_q e et CC renvoie l'ensemble des prédictions générées par tous les classifieurs. La Figure 2.4 représente le fonctionnement de la méthode CC, le sous ensemble de labels prédits sera (Eau, Ciel, Montagnes).

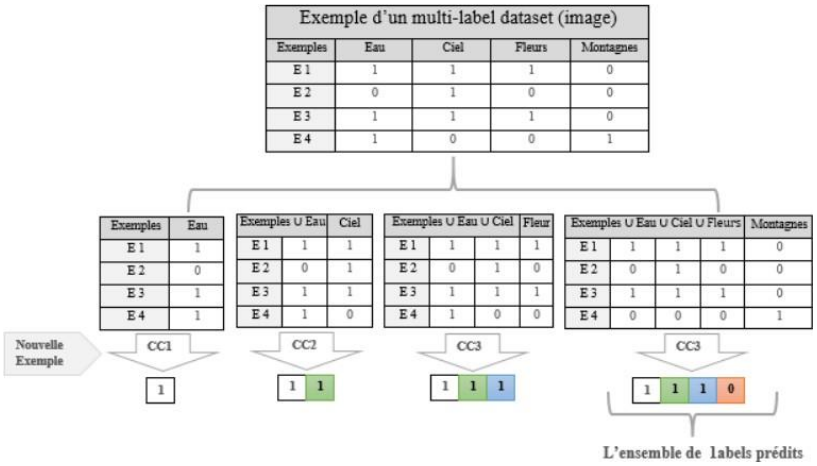

Figure.2.4 Illustration de la méthode CC

2.3.1.3 Méthode Label Powerset (LP)

Dans la méthode LP [18] chaque combinaison distincte de labels dans l'ensemble d'apprentissage constitue une nouvelle classe. Pour chaque sous problème créé, un classifieur binaire standard, tel que la régression logistique ou un SVM, est formé pour prédire la présence ou l'absence des labels de cette classe spécifique. Lors de la classification d'une nouvelle instance x_i, LP attribue la classe la plus probable à cette instance, puis convertit cette classe à la combinaison initiale de labels. Dans l'exemple illustré dans la Figure 2.5, LP fera une prédiction de classe pour une instance donnée qui est composée d'un ensemble de labels pertinents.

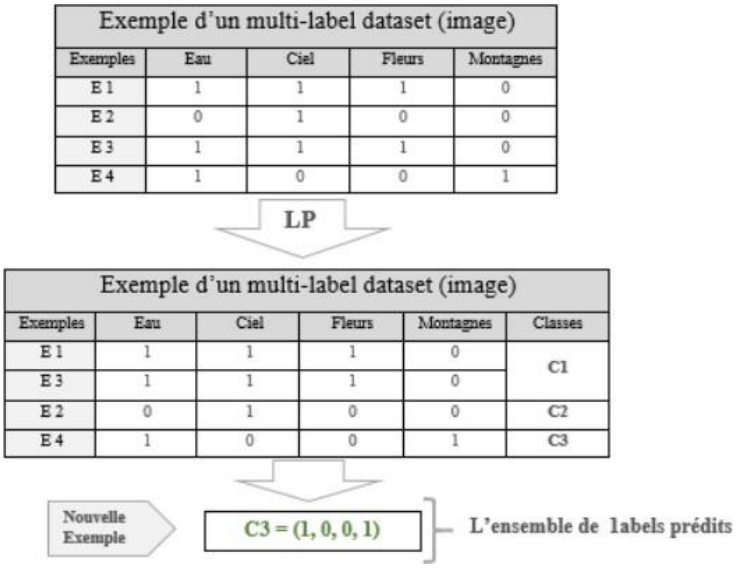

Figure.2.5 Illustration de la méthode LP

2.3.1.4 Méthode Pruned Set (PS)

PS [27] est une méthode élaguée qui permet de réduire la complexité du traitement en éliminant les ensembles de label les moins fréquents par apport à un seuil prédéfini. Ainsi, seuls les labels ayant des probabilités supérieures ou égales à ce seuil seront conservés. Dans l'exemple illustré par la Figure 2.6, PS ne conserve que la classe C1, qui possède un ensemble de label plus fréquent par rapport aux autres classes. Les classes qui se produisent rarement, telles que 2, 3, et 4 seront éliminées. Cependant, il convient de mentionner que la méthode PS est similaire à LP si l'on ne prend en compte que les ensembles de labels qui sont distinctement présents dans les données d'apprentissage.

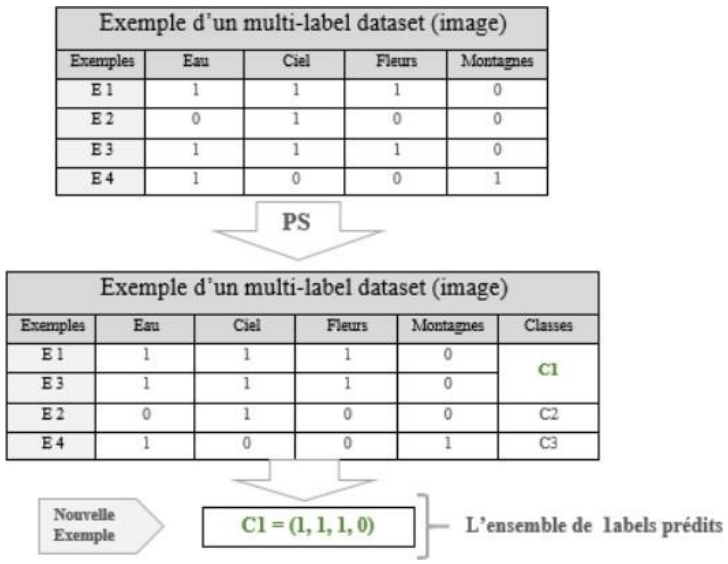

Figure.2.6 Illustration de la méthode PS

2.3.2 Approche d'Adaptation d'Algorithme (AA)

Cette catégorie d'approche consiste à utiliser des méthodes liées à des algorithmes de classification spécifiques pour traiter des problèmes multi-label. Les méthodes de cette approche sont principalement construites autour des algorithmes d'arbres de décision (DT), de réseaux de neurones (NN), et des k-plus proches voisins (k-NN). Parmi les méthodes les plus répandues de cette approche, on trouve Multi-Label C4.5 (ML-C4.5) [4], Predictive Clustering Trees (PCT) [28], Back Propagation for Multi-Label Learning (BP-MLL) [29], ainsi que la méthode de Multi-Label k-Nearest Neighbors (ML-kNN) [30].

2.3.2.1 Méthodes basées sur les arbres de décision

La méthode Multi-Label C4.5 [4] est une adaptation de l'algorithme C4.5 [31] à la méthode Multi-Label Decision Tree (ML-DT) pour gérer des DML. Cette adaptation est réalisée en utilisant un critère de gain d'information basé sur une version modifiée de l'entropie originale de l'algorithme C4.5, comme définie par l'équation (2.1).

Cette méthode autorise plusieurs labels dans les feuilles de l'arbre, dont l'entropie quantifie les informations nécessaires à la description des labels associés aux instances. Il est évident de noter que la construction de l'arbre est effectué en sélectionnant récursivement des attributs pour partitionner les données en sous-ensembles plus petits D_1,…, D_n.

$$Entropy\ (D) = -\sum_{i=1}^{q} (p(\lambda_i) \log p(\lambda_i) + q(\lambda_i) \log q(\lambda_i)) \tag{2.1}$$

Où D est l'ensemble de données, (λ_i) représente la fréquence relative du label λ_i et $(\lambda_i) = 1 - (\lambda_i)$.

En outre, la méthode Predictive Clustering Trees (PCT) [28] est un arbre de décision basé sur une hiérarchie de grappes, où le nœud supérieur correspond à un cluster contenant toutes les données, divisé de manière récursive en sous-grappes dans les nœuds enfants. Cette technique est similaire à celle de l'algorithme C4.5, où l'attribut de division est choisi en fonction d'une mesure d'entropie. Toutefois, PCT adopte une fonction de variation pour décrire les nœuds, et attribue ensuite des probabilités aux labels à chaque nœud de l'arbre, de manière à ce que chaque label puisse être associé à l'instance correspondante. Ainsi, l'arbre prédit plusieurs labels à la fois en utilisant un processus d'induction dans le PCT, qui calcule la somme des indices de Gini de tous les labels pour déterminer la meilleure sélection à chaque nœud. La fonction de variance de PCT est formulée par l'équation (2.2).

$$Var(D) = \sum_{i=1}^{q} Gini\big(D, Y^i\big), Gini\big(D, Y^i\big) = 1 - \big(p^2(\lambda_i) + q^2(\lambda_i)\big) \tag{2.2}$$

Où D est l'ensemble de données, (λ_i) représente la fréquence relative du label λ_i et $(\lambda_i) = 1 - (\lambda_i)$.

2.3.2.2 Méthodes basées sur les réseaux de neurones

La méthode BP-MLL (Back-Propagation for Multi-Label Learning) [29] repose sur ANN spécifiquement conçu pour la CML. Cette méthode utilise des fonctions d'activation pour apprendre des relations complexes entre labels. Lorsque les données sont propagées à travers le réseau, une comparaison est effectuée entre les labels prédites et ceux réels, avec un ajustement itératif des poids du réseau pour minimiser l'erreur. Lors de la prédiction, un seuil de décision est appliqué pour attribuer les labels finaux à la nouvelle instance, en fonction de la confiance du modèle.

2.3.2.3 Méthodes basées sur les k-plus proches voisins

La méthode ML-kNN [30], basée sur l'adaptation de l'algorithme k-NN, utilise un k-NN pour traiter chaque label indépendamment. Pour classer une nouvelle instance x_i, ML-kNN recherche les k plus proches voisins de cette instance en utilisant une mesure de distance. Ensuite, il examine les labels associés aux voisins de x_i en comptant leurs occurrences pour estimer la probabilité a priori (probabilité de l'apparition de L_i dans D) et la probabilité a posteriori (probabilité de l'association de L_i à x_i) de chaque label appartenant à l'ensemble d'apprentissage. Pour prédire l'ensemble de labels associés à l'instance x_i, ML-kNN utilise les probabilités de chaque label de telle sorte que ceux ayant les probabilités les plus élevées constituent les labels prédits de la nouvelle instance, comme défini dans l'équation (2.3).

$$y^i = \begin{cases} 1 & if \ (c^j|y^j = 1)\ p\ (y^j = 1) \geq (c^j|y^j = 0)(y^j = 0) \\ 0 & otherwise \end{cases} \qquad (2.3)$$

Nous notons que :

- ✓ $(c^j|y^j = 1)$ est la probabilité conditionnelle du j-ème label étant égale à 1.
- ✓ $(y^j = 1)$, $(y^j = 0)$ sont les probabilités que le j-ème label soit égal à 1 ou 0 respectivement, dans l'ensemble d'apprentissage.
- ✓ $(c^j|y^j = 0)$ est la probabilité conditionnelle du j-ème label étant égale à 1 sachant que le j-ème label est égal à 0.

2.3.3 Analyse comparative entre les approches TP et AA

Les deux approches TP et AA de CML présentent chacune leurs spécificités selon plusieurs critères [22], [32], [33], [34]. Ces critères incluent notamment la stratégie de classification adoptée, les processus d'apprentissage / prédiction mis en œuvre, la préservation de la corrélation entre labels, ainsi que la sensibilité au déséquilibre, comme mentionné dans la Table 2.2.

Ainsi, chaque approche présente des avantages et des inconvénients selon les critères spécifiques énumérés. Par exemple, l'approche TP semble plus simple à implémenter mais Incapable de modéliser explicitement les dépendances entre labels, tandis que l'approche AA peut mieux traiter ces dépendances mais nécessite une adaptation plus complexe des algorithmes de base. Cependant, Le choix entre les deux approches dépend des caractéristiques des ensembles de DML expérimentés et des exigences spécifiques du problème traité.

Table.2.2 Principales distinctions entre les approches TP et AA.

Critères	**Approches de la Classification Multi-Label**	
	TP	**AA**
Stratégie	Transformer un problème de Classification Multi-Label en un ensemble de problèmes de CML.	Etendre les algorithmes d'apprentissage conventionnels pour résoudre un problème de CML.
Processus d'Apprentissage	Un classifieur binaire est entrainé pour chaque label indépendamment des autres classifieurs.	Algorithme de base modifié pour prendre en compte les caractéristiques spécifiques des DML.
Processus de Prédiction	Chaque classifieur binaire doit déterminer la présence ou l'absence de son label correspondant. Les prédictions pour chaque label sont combinées pour générer la prédiction Multi-Label finale.	Dépend de la manière dont l'algorithme de base a été adapté.
Dépendances entre Labels	Incapable de modéliser explicitement les dépendances entre labels.	Faible capture des dépendances entre labels.

Critères	**Approches de la Classification Multi-Label (Suite)**	
	TP	**AA**
Sensibilité au déséquilibre	Sensible au déséquilibre des classes ayant un grand nombre de labels.	La sensibilité au déséquilibre dépend de la technique d'adaptation utilisée.
Dimensionnalité de la sortie	N'est pas prise en charge	N'est pas prise en charge

2.4 Défis de la Classification Multi-Label

Bien que les méthodes des approches TP et AA pour CML aient prouvé leurs efficacité dans la résolution des problèmes complexes issus de la classification mono-label, plusieurs défis scientifiques ont émergé nécessitant une attention particulière. Ces problèmes, susceptibles de conduire à une dégradation des performances des modelés, comprennent notamment :

2.4.1 Difficulté de modéliser les dépendances entre labels

L'un des défis majeur dans la modélisation de données réside dans la capture des dépendances complexes entre labels. Par exemple, lorsqu'une image est étiquetée comme "Sable" et "Coquillage", la probabilité d'attribuer également l'étiquette "Plage" à cette image augmente de manière significative. Cette interdépendance entre les labels s'étend également aux données textuelles. Prenons l'exemple d'un article centré sur des sujets liés à la "Médecine" : il est peu probable qu'il reçoive le label "Divertissement". Ces corrélations entre les labels sont cruciales à capturer pour une modélisation précise. Il convient de souligner que ces problèmes sont souvent exacerbés par les méthodes de transformation binaire [35], ce qui rend la tâche de modélisation encore plus complexe.

2.4.2 Mauvaise gestion du déséquilibre entre les classes des labels

Ce problème est particulièrement répandu dans les domaines médical et bioinformatique, où certains labels sont fréquemment représentés dans l'ensemble de données, tandis que d'autres restent relativement rares. Ce déséquilibre des labels peut poser des défis significatifs lors de la modélisation des données.

Il est important de noter que les méthodes utilisées dans l'approche de TP peuvent accentuer le degré de déséquilibre des labels à chaque étape de la transformation [36]. Par exemple, lors de la sélection des caractéristiques des données, les méthodes conventionnelles peuvent accentuer l'impact de ce déséquilibre, ce qui peut conduire à des prédictions biaisées ou peu fiables.

4.2.3 Le fléau de la dimensionnalité de l'espace de sortie

Ce défi découle de la croissance exponentielle du nombre de labels, ce qui entraine une augmentation exponentielle du nombre potentiel d'ensembles de labels. Par exemple, dans un espace de labels comprenant 30 labels, le nombre total possible de combinaisons de labels peut atteindre 2^{30}, soit plus d'un milliard de combinaisons différentes. Cette croissance exponentielle de la dimensionnalité aggrave considérablement la complexité des algorithmes utilisés pour la modélisation et le traitement des données [37]. Cependant, cet inconvénient rend la tâche de modélisation de plus en plus ardue à mesure que le nombre de labels augmente. Cela nécessite des ressources informatiques considérables pour manipuler des ensembles de données de grande échelle.

4.2.4 Structure hiérarchique :

Dans certains scénarios de CML, il est nécessaire de structurer les labels pour classer les instances en une hiérarchie ou en arborescence. Cette structure hiérarchique implique que chaque label peut avoir un ou plusieurs labels parent (super-labels) et des labels enfants (sous-labels) [34]. Un exemple typique de cette structure est celui d'une bibliothèque numérique où les documents couvrent une variété de sujets. Chaque document peut se voir attribuer plusieurs labels pour représenter son contenu.

Cependant, ces labels ne sont pas indépendants les uns des autres ; ils sont plutôt en relation hiérarchique. Par exemple, un document peut être étiqueté avec "Mathématique", qui est un label parent, et ensuite avec des sous-labels tels que "Algèbre" ou "Géométrie", qui sont des catégories plus spécifiques relevant de "Mathématique".

La Table 2.3 présente une synthèse des méthodes les plus connues des approchas TP et AA mettant en évidence la problématique abordée par chacune d'entre elles.

Table.2.3 Défis engendrés par les méthodes des approches TP et AA.

Problème Posé	Méthode	Raison
Difficulté de modéliser les dépendances entre labels	BR [25]	Chaque label est traité de manière indépendante.
	CC [26]	Ne pas capturer toutes les dépendances de manière efficace, puisque elles sont capturées selon l'ordre des labels dans la chaine.
	LP [18]	Les combinaisons de labels sont considérées comme des classes distinctes, en ignorant les dépendances entre labels.
	PS [27]	Modélisation implicite des dépendances entre labels à cause de la perte d'information lors du processus élagué.
	ML-C4.5 [4]	Capture implicite des dépendances entre labels uniquement au niveau de chaque arbre de la structure.
Mauvaise gestion du déséquilibre entre les classes	BR [25]	Ne pas tenir compte de la distribution des classes de chaque label lors de l'apprentissage des classifieurs binaires.
	CC [26]	Déséquilibre non traité entre les classes des labels. entraînant des prédictions biaisées en faveur de la classe majoritaire pour chaque label.
	LP [18]	Déséquilibre traité implicitement entre les classes des labels puisque chaque combinaison de labels est traitée comme une classe distincte sans tenir compte du déséquilibre au sein de chaque classe.
	PS [27]	Espace de recherche réduit par élimination des labels non pertinents, n'implique pas un traitement direct
	PCT [28]	Le partitionnement de l'espace des caractéristiques lors de la prédiction des labels, les distributions de classes déséquilibrées au sein de chaque classe ne sont pas gérées.
	BP-MLL [29]	Nécessite des techniques d'équilibrages (l'échantillonnage ou pondération des classes) pour traiter le déséquilibre des classes de labels.
	ML-KNN [30]	Gérer implicitement le déséquilibre entre classes de labels via des techniques de pondération des instances.

Problème Posé	Méthode	Raison (Suite)
Dimensionnalité de l'espace de sortie	BR [25]	Dépend au nombre de labels.
	CC [26]	
	ML- KNN [30]	
	LP [18]	Importante en particulier avec de nombreux labels.
	PS [27]	Dépend du nombre de labels pertinents pour chaque instance.
	ML-C4.5 [4]	Dépend de la structure de l'arbre de décision.
	PCT [28]	Dépend du nombre de clusters formés par les arbres.
	BP-MLL [29]	Dépend du nombre de labels et de l'architecture du réseau neuronal.

En se basant sur la synthèse ci-dessus, il est évident que les algorithmes mentionnés continuent de faire face à des défis persistants, notamment les dépendances entre les labels, le déséquilibre des classes, ainsi que la grande dimensionnalité de l'espace de sortie. Ces défis ont souvent pour conséquence des performances de classification insatisfaisantes dans des applications du monde réel.

Pour surmonter ces lacunes, une troisième approche, appelée approche Ensemble, a été développée en tirant parti des méthodes des approches TP et AA. Cette approche permet de combiner les avantages des deux approches précédentes tout en atténuant leurs inconvénients. L'approche Ensemble vise à analyser les interactions entre divers classifieurs et à exploiter leurs compétences respectives afin d'améliorer les performances de CML. Dans le chapitre suivant (*Chapitre 3*), nous allons explorer cette approche en détail, dans le but de proposer des solutions plus efficaces et mieux adaptées aux défis rencontrés dans le domaine.

2.5 Domaines d'Application

Les progrès technologiques récentes ont considérablement favorisé la disponibilité de grandes quantités de DML hétérogènes, extraites de divers référentiels tels que MILAN[1], MEKA[2] et CLUS[3]. Cette abondance de données a facilité une évaluation approfondie des performances des algorithmes d'apprentissage multi-label dans divers domaines d'application tels que, la catégorisation de texte, la bioinformatique, le multimédia, l'exploration des réseaux sociaux, E-Learning, et bien d'autres encore.

2.5.1 La catégorisation de texte

Dans le contexte de CML, chaque document tel que un article, un rapport ou email, appartenant à l'ensemble de données représente un fragment de texte étiqueté par un ou plusieurs labels simultanément. Ces labels représentent ainsi les sujets auxquels le document appartient. Par exemple, pour les textes cliniques tels que les dossiers médicaux électroniques, plusieurs labels peuvent être attribués, tels que les codes de diagnostic, les antécédents médicaux, les listes de problèmes et des détails sur les procédures pratiquées sur les patients.

Ainsi, le contenu de chaque document est représenté sous la forme d'une chaine de texte, stockée dans un format structuré tel que CSV. Chaque ligne de ce format correspond à un document distinct (instance), tandis que les colonnes représentent les différentes caractéristiques extraites de chaque document, suivies des colonnes de labels associés à chaque document. Chaque label est binaire, indiquant la présence ou l'absence du document dans une catégorie donnée.

2.5.2 La Bioinformatique

La multidimensionnalité des données protéiniques, caractérisée par la présence de multiples fonctions et interactions entre les protéines, nécessite l'utilisation de la CML. Ce paradigme de classification a réussi à satisfaire les exigences inhérentes du domaine en permettant à une protéine d'être associée à plusieurs labels simultanément, reflétant ainsi sa diversité fonctionnelle et son implication dans divers scénarios biologiques.

[1] http://mulan.sourceforge.net/
[2] http://meka.sourceforge.net/
[3] http://clus.sourceforge.net/

Par exemple, un gène peut appartenir à une liste prédéfinie de fonctions telles que le métabolisme, la synthèse des protéines et la transcription, la synthèse des protéines et la transcription.

La représentation d'une donnée protéiniques dans un ensemble de DML est basée sur une structure tabulaire, où chaque ligne représente une protéine et chaque colonne représente un attribut de la protéine. D'autres colonnes de labels sont incluses pour indiquer la présence ou l'absence de la protéine dans cette catégorie donnée.

2.5.3 Multimédia

En raison de la grande diversité des types de ressources multimédias, comprenant des images, des vidéos, et des audio, ainsi qu'à la complexité inhérente à leurs données, CML s'est révélée être une approche efficace et adaptable à ce domaine.

Dans le cas de CML d'images, chaque image peut être associée à plusieurs labels simultanément, ce qui permet de décrire différents éléments présents dans l'image. Prenons le cas d'une plateforme de vente d'électroménager en ligne qui souhaite améliorer son système de recherche d'images en attribuant les labels suivants : Type (aspirateur, lave-vaisselle, machine à laver), Couleur (blanc, gris, noir), Marque (LG, Brand, Samsung, Artur Martin, Condor).

La CML des vidéos pour un système de recherche et de recommandation en ligne permet d'attribuer à chaque vidéo une liste prédéfinie de labels décrivant son contenu, à savoir : Genre (action, comédie, drame, science-fiction), Thème (romantique, familial, guerre), Public-visé (enfants, adultes, femmes, hommes), Humeur (drôle, triste, effrayant, inspirant).

La CML des sons dans une application de reconnaissance sonore pour les smartphones permet d'attribuer plusieurs labels à un événement audio pour le décrire. Ces labels peuvent avoir les désignations suivantes : Bruit (trafic, conversation, musique), Evénements (alarme, sirène, pleurs de bébé), Sources (voix humaine, moteur de voiture, instrument de musique).

2.5.4 E-Learning

La CML a également contribué à améliorer le processus d'apprentissage et l'enseignement dans divers domaines de l'E-learning.

Ces améliorations incluent la personnalisation de l'apprentissage, où le système CML peut analyser les résultats des tests d'apprenants et leur recommander des exercices supplémentaires pour cibler leurs points faibles.

De plus, dans le domaine de l'évaluation des apprenants, un système CML peut analyser un essai écrit par un apprenant pour identifier les concepts clés abordés, évaluer la structure de son argumentation et évaluer la qualité de son écriture. Enfin, l'analyse des données d'apprentissage permet au système CML d'examiner les données d'apprentissage d'un groupe d'étudiants pour identifier les ressources pédagogiques les plus pertinentes pour un sujet particulier ou les types d'erreurs les plus fréquemment commises par les étudiants.

2.5.5 Exploration des réseaux sociaux

Le domaine de l'exploration des réseaux sociaux a fait des progrès considérables grâce à CML, permettant d'organiser plus finement le contenu et les utilisateurs, ce qui conduit à une meilleure compréhension des contenus sociaux en ligne.

Ce domaine d'application a pu prouver ses compétences dans différents secteurs sociaux, tels que l'identification des sentiments et des opinions exprimés par les utilisateurs dans leurs publications, commentaires et messages. En outre, il a été appliqué pour classer les publications et les messages en fonction de leur type de contenu (nouvelles, humour, marketing, etc.). Le secteur de la détection des communautés et des groupes d'utilisateurs qui partagent des intérêts et des préoccupations similaires a également bénéficié des techniques avancées de la CML.

2.5.6 Autres domaines

D'autres domaines d'application ont également bénéficié de l'utilisation de la CML, démontrant ainsi sa polyvalence et son potentiel dans une variété de contextes. Par exemple, dans le domaine de la recommandation de camions de restauration, la CML est utilisée pour offrir une variété de cuisines et de plats, ainsi que pour recommander les camions adaptés aux préférences multiples et variées des consommateurs.

De même, pour la reconnaissance des activités humaines dans les maisons intelligentes, la CML a été adaptée pour reconnaître les activités humaines à l'aide de données sensorielles captées et classées par des caméras, des microphones et des capteurs de mouvements.

Ces données permettent aux systèmes domestiques d'ajuster leur comportement en fonction des besoins et des préférences des occupants, améliorant ainsi le confort et la commodité.

Suite à cette présentation des différents domaines de la CML, un résumé des principales applications rapportées du domaine est décrit dans la Table 2.4.

Table.2.4 Applications rapportées de la Classification Multi-Label.

Application	Domaines	Rapporté dans
Catégorisation de texte	Indexation des documents, Suggestion de tags, Codage médical, Classification des activités économiques, Filtrage des e-mails, Rapports aéronautiques, Textes juridiques, Documents web, Classification des brevets, Classification des nouvelles.	[3], [3], [38], [39], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55]. [41],
Bioinformatique	Prédiction de la fonction génique, Prédiction de la fonction des protéines, Multi-location subcellulaire des protéines, Prédiction de la structure 3D des protéines, Diagnostics Medical, Segmentation des régions tissulaires, Classification des lésions cutanée, Découverte de médicaments, Effets indésirables des médicaments.	[29], [56],[57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68], [69], [70], [71].
Multimédia	Annotation des images et vidéos, La vérification faciale, Classification des émotions dans la musique et la parole, Extraction de méta-données musicales.	[66], [72], [73], [9], [74], [75], [76], [77], [78], [79], [80], [81], [82], [83], [84]
E-Learning	Classification des styles d'apprentissage en fonction des profils des apprenants, Comprendre les expériences d'apprentissage des élèves.	[85], [86].
Exploration des réseaux sociaux	Apprentissage du comportement collectif, Publicité sur les réseaux sociaux, Annotation automatique.	[87], [88], [89], [90], [91], [92].

2.6 Ensembles de Données Multi-Label

Une donnée multi-label (DML) peut être mesurée par un nombre d'instances *m*, un nombre de labels (*q*), et nombre d'attributs (*d*). Chaque ensemble de données multi-label est caractérisé par ses propres indicateurs de base, regroupés en trois catégories :

La première catégorie comprend les indicateurs basés sur la fréquence des labels, définis par les équations (2.4) à (2.6), tels que la dimensionnalité (Dim), la cardinalité des labels (Card) et la densité (Dens).

La deuxième catégorie englobe les indicateurs basés sur le déséquilibre, comme avgIR défini par l'équation (2.7). Enfin, la troisième catégorie comprend les indicateurs basés sur la dépendance entre labels, tel que rDep défini par l'équation (2.8). La Table 2.6 décrit les caractéristiques de quelques datasets de référence couvrant différents domaines d'application.

- La dimensionnalité (Dim) de l'ensemble de données (*D*), définie par l'équation (2.4), représente l'étendue de la complexité des données [93]. Elle est calculée comme le produit du nombre d'instances (*m*), de caractéristiques (*d*), et de labels (*q*) [94].

$$Dim = m \times \mathrm{d} \times \mathrm{q} \tag{2.4}$$

- La cardinalité des labels (*Card*), définie par l'équation (2.5), indique le nombre moyen de labels (*q*) associées à chaque exemple (*m*) [16]. Cette valeur est directement liée au nombre de labels.

$$Card(D) = \frac{1}{m} \sum_{i=1}^{m} |Y_i| \tag{2.5}$$

- La densité (*Dens*), telle que définie dans l'équation (2.6), quantifie la proportion des labels possibles (*q*) associées à chaque instance (*m*) [16], tout en tenant compte des dimensions de l'espace de labels.

$$Dens(D) = \frac{1}{q}\, Card(D) \tag{2.6}$$

- Le rapport de déséquilibre moyen (avgIR), défini par l'équation (2.7), calcule pour chaque label le rapport de fréquences entre le label le plus fréquent ($f\lambda'$) et le label réel f_l [95].

$$avgIR(D) = \frac{1}{q}\sum_{l=1}^{q}\frac{\arg\max(f_{\lambda'})}{f_l} \tag{2.7}$$

- Le ratio des paires de labels dépendants (rDep), tel que défini dans l'équation (2.8) [96], indique la proportion de paires de labels fortement interdépendants. La moyenne est ensuite calculée sur le nombre total de paires de labels [97]. Dans cette équation, $x^2(\lambda_l, \lambda_j)$ est une mesure qui évalue la relation entre deux labels spécifiques (λ_l) et (λ_j). Ces labels seront fortement indépendants si leur mesure est supérieure au seuil de dépendance (6.635).

$$rDep(D) = \left(\sum_{l=1}^{q-1}\sum_{j=l+1}^{q}\left[x^2(\lambda_l, \lambda_j) > 6.635\right]\right) x \left(\frac{q(q-1)}{2}\right)^{-1} \tag{2.8}$$

Selon la littérature, ces ensembles de données sont au format original et sont subdivisés en parties d'apprentissage et de test. L'ensemble d'apprentissage représente les 2/3 des données, et le tiers restant représente l'ensemble de test. La Table 2.5 illustre un aperçu de divers ensembles DML de référence, ainsi que leurs caractéristiques respectives.

Table.2.5 Aperçu de quelques datasets Multi-Label de référence.

Datasets	Domain	m	q	d	Card	Dens	avgIR	rDep
Birds	Audio	645	19	260	1.014	0.053	5 .407	0 .123
CHD_49	Medicine	555	6	49	2.580	0.430	5.766	0.267
Genbase	Biology	662	27	1186	1.252	0.046	37.315	0.157
HumanPse	Biology	3106	14	440	1.185	0.085	15.289	4.418
Medical	Text	978	45	1449	1.245	0.028	89.501	0.039
PlantPse	Biology	978	12	440	1.079	0.090	6.690	0.318
VirusGo	Biology	207	6	749	1.217	0.203	4.041	0.400
Yeast	Biology	2417	14	103	4.237	0.303	7.197	0.670

2.7 Mesures d'évaluation

L'évaluation des modèles de CML est une tâche très complexe nécessitant un traitement particulier. À cet effet, des mesures spécifiques ont été conçues pour évaluer les prédictions obtenues, qu'elles soient entièrement précises, partiellement précises ou complètement erronées. Pour relever ce défi, Tsoumakas et al. [25] ont introduit une vaste collection de mesures d'évaluation, classées en deux groupes : les mesures basées sur les exemples et celles basées sur les labels.

Formellement, pour chaque mesure d'évaluation, l'ensemble de données de test se compose de paires (x_i, y_i) où 1 ≤ i ≤ n. Ici, $y_i \in \{0, 1\}$ représente les labels réels du i-ème exemple de test, et $\widetilde{y_i} = h(x_i)$ représente ses labels prédits.

2.7.1 Mesures basées sur les exemples

Les mesures basées sur les exemples impliquent le calcul des écarts moyens entre les ensembles de labels réels y_i et ceux prédits $\tilde{y}$ pour chaque exemple de test x_i appartenant à la base de test. Ils calculent ensuite la moyenne sur tous les exemples de l'ensemble de test *S*. Cette catégorie de mesures comprend : Hamming_Loss, Accuracy, Subset-Accuracy, Precision, Recall et F1.

- **Hamming_Loss**: évalue le nombre de fois où une paire d'exemple-label est mal classés. Lorsque cette mesure tend vers le zero. Cette mesure est définie par l'équation (2.9), où Δ représente la différence symétrique entre l'ensemble de labels réels y_i et l'ensemble des prédictions $\tilde{y}$, *m* est le nombre d'exemples et *q* est le nombre total de labels possibles.

$$HLoss \downarrow = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{q} \left| h(x_i) \, \Delta \, y_i \right| \tag{2.9}$$

- **Accuracy**: pour chaque instance de test x_i, elle permet d'évaluer la similarité de *Jaccard* entre l'ensemble de labels réels y_i et l'ensemble de labels prédits $\tilde{y}$. Cette mesure est définie par l'équation (2.10).

$$Accuracy \uparrow = \frac{1}{m} \sum_{i=1}^{m} \frac{|h(x_i) \cap y_i|}{|h(x_i) \cup y_i|} \tag{2.10}$$

- **Subset Accuracy:** pour chaque instance de test x_i, il évalue si les labels prédits $\tilde{y}$ correspondent exactement aux vrais labels y_i. Cette évaluation est déterminée par l'équation (2.11), où laquelle I(vrai) équivaut à 1 et I(faux) équivaut à 0.

$$Subset\_accuracy \uparrow = \frac{1}{m} \sum_{i=1}^{m} I\left(h(x_i) = y_i\right) \tag{2.11}$$

- **Precision**: pour chaque exemple test x_i, elle mesure la proportion des labels correctement prédites par rapport au nombre total de labels prédits. Cette mesure est définie par l'équation (2.12).

$$Precision \uparrow = \frac{1}{m}\sum_{i=1}^{m}\frac{|h(x_i) \cap y_i|}{|y_i|} \quad (2.12)$$

- **Recall**: pour chaque exemple test x_i, elle mesure le ratio des labels correctement prédits par rapport à tous les vrais labels. Cette mesure est définie par l'équation (2.13).

$$Recall \uparrow = \frac{1}{m}\sum_{i=1}^{m}\frac{|h(x_i) \cap y_i|}{|h(x_i)|} \quad (2.13)$$

- **F1-score**: pour chaque exemple test x_i, elle mesure la moyenne harmonique entre la précision et le rappel et est défini par l'équation (2.14).

$$F1 - score \uparrow = \frac{1}{m}\sum_{i=1}^{m}\frac{2 \times |h(x_i) \cap y_i|}{|h(x_i)| + |y_I|} \quad (2.14)$$

2.7.2 Mesures basées sur les labels

Les mesures basées sur les labels évaluent la performance prédictive de chaque label indépendamment, en se basant sur le nombre de vrais positifs (VP), de vrais négatifs (VN), de faux positifs (FP) et de faux négatifs (FN). Ensuite, les performances obtenues pour les différents labels sont combinées en utilisant deux types de moyennes : la micro-moyenne et la macro-moyenne. La première moyenne attribue le même poids à chaque label, quelle que soit sa fréquence, tandis que la deuxième attribue le même poids à chaque observation. Les mesures basées sur les labels englobent : Macro_precision, Micro_precision, Macro_recall, Micro_recall, Macro_F1 et Micro_F1.

Micro_precision et Micro_recall calculent la mesure moyenne sur toutes les paires (exemple, label). En revanche, Macro_precision et Macro_recall calculent la mesure moyenne sur tous les labels et sont définies par les équations (2.15) à (2.18). En ce qui concerne la mesure Micro_F1, elle représente une moyenne harmonique entre Micro_precision et Micro_recall. D'un autre côté, Macro_F1 exprime une moyenne harmonique entre la précision et le rappel, calculée par label et ensuite moyennée sur tous les labels. Ces mesures sont définies par les équations (2.19) et (2.20).

$$Micro\_precision = \frac{\sum_{j=1}^{q} vp_j}{\sum_{j=1}^{q} vp_j + \sum_{j=1}^{q} fp_j} \quad (2.15)$$

$$Micro\_recall = \frac{\sum_{j=1}^{q} vp_j}{\sum_{j=1}^{q} vp_j + \sum_{j=1}^{q} fn_j} \quad (2.16)$$

$$Macro\_precision = \frac{1}{q}\sum_{j=1}^{q} \frac{vp_j}{vp_j + fp_j} \quad (2.17)$$

$$Macro\_recall = \frac{1}{q}\sum_{j=1}^{q} \frac{vp_j}{vp_j + fn_j} \quad (2.18)$$

$$Micro\_F1 = \frac{2 \times Micro_{precision} \times Micro_{recall}}{Micro_{precision} + Micro_{recall}} \quad (2.19)$$

$$Macro_{F1} = \frac{1}{q}\sum_{j=1}^{q} \frac{2 \times p_j \times r_j}{p_j + r_j} \quad (2.20)$$

2.7.3 Autres mesures

La courbe AUC (Area Under Cuve) ROC (Receiver Operating Characteristic) est une autre mesure de performance des modelés de classification multi-label [98]. Cette courbe est basée sur les taux TP tracés en fonction des taux FP, où chaque courbe AUC_i évalue la performance de la classification du label *i* par rapport aux autres labels $|q|$ afin identifier les labels pour lesquels le modèle fonctionne bien.

La courbe AUC peut être calculée en utilisant soit la formule Micro-AUC, soit celle de Macro-AUC. La mesure Micro-AUC permet de combiner les prédictions de chaque instance pour calcule l'AUC globale. En revanche, la mesure Macro-AUC évalue la performance moyenne par label en calculant l'AUC pour chaque label *i*, ensuite prend la moyenne de ces valeurs. La formule générale de la courbe AUC est définie par l'équation (2.21).

$$AUC = \frac{1}{q}\sum_{i=1}^{q} AUC_i \qquad (2.21)$$

2.8 Classification Multi-Label à grande échelle

Avec la croissance exponentielle des DML, tant en taille qu'en complexité, les algorithmes de CML doivent désormais traiter ces volumes massifs de données. Les plateformes de streaming comme Netflix ou Spotify illustrent bien ce phénomène, où chaque choix de l'utilisateur est associé à plusieurs labels, telles que les genres de films ou de musique préférés. À mesure que la base d'utilisateurs et de contenus disponibles augmente, les modèles de classification doivent gérer des milliards d'interactions possibles, ce qui accroît considérablement la charge de calcul. Cependant, les algorithmes de CML conventionnels traitent les données sur un seul serveur ou via une infrastructure centralisée. Face à des volumes de données gigantesques et à des espaces de sortie hautement dimensionnels, ces algorithmes deviennent lents et inefficaces, tant en termes de temps de calcul qu'en termes de gestion des ressources mémoire.

Pour répondre à ces défis, des approches distribuées et parallèles ont été introduites dans divers domaines d'application, tels que l'image [99], le texte [100], [101], [102], la vidéo [103], et bien d'autres afin de traiter des DML à grande échelle. En divisant la charge de travail et en parallélisant les calculs sur plusieurs nœuds, ces techniques permettent d'exploiter efficacement de grandes collections de données tout en garantissant une meilleure évolutivité et une efficacité accrues des algorithmes de CML.

Divers travaux cités par la littérature ont mis l'accent sur l'importance de distribuer la charge de travail et de paralléliser les calculs sur plusieurs nœuds, permettant ainsi d'améliorer l'évolutivité et l'efficacité des algorithmes de CML. Ils se concentrent principalement sur deux axes : la sélection distribuée des caractéristiques et la CML distribuée, en utilisant des framework comme *Apache Spark* et *MapReduce*.

Parmi les contributions notables, Gonzalez-Lopez et al. [104] ont développé une version distribuée et optimisée de l'algorithme ML-kNN avec trois différentes stratégies distinctes pour rechercher les plus proches voisins dans un environnement Spark, à savoir : la diffusion itérative des instances, l'utilisation d'une structure d'indexation basée sur un arbre

et la construction de tables de hachage pour regrouper les instances. Leur étude montre que la stratégie d'indexation basée sur un arbre offre les meilleures performances pour traiter des ensembles de DML volumineux et complexes.

D'autres travaux, comme ceux d'Amazal et al. [101], se sont concentrés sur la CML distribuée pour des données textuelles massives en utilisant la méthode LP couplée à l'algorithme *MapReduce* pour attribuer des labels positifs aux documents. L'approche utilise également la technique de sélection des caractéristiques TFIDF, pour réduire les dimensions des données textuelles et améliorer la précision de la classification. Les résultats expérimentaux montrent que l'approche proposée améliore la précision de la CML tout en gérant efficacement les données volumineuses. De même, Biswas et Devi [105] ont proposé une approche parallèle basée sur *MapReduce* pour accélérer le processus de classification et réduire la dimensionnalité dans les ensembles de données à grande échelle. Ils ont utilisé des techniques de sélection de caractéristiques, telles que le score de Fisher, pour éliminer les caractéristiques redondantes. Les résultats montrent que le traitement parallèle améliore considérablement la précision et le temps de traitement, rendant ainsi cette approche plus adaptée aux ensembles de DML à grande échelle.

Amazal et al. [102] ont également exploré une approche de sélection de caractéristiques distribuée basée sur le chi-carré pondéré pour attribuer des poids aux caractéristiques en fonction de leur fréquence. L'approche implémentée en utilisant le modèle *MapReduce* a prouvé son efficacité et sa robustesse à travers des expérimentations sur des données textuelles de référence.

Ces travaux montrent que l'approche distribuée a permis de surmonter les limitations liées à la scalabilité et à la complexité des DML. Les techniques comme *MapReduce* et *Spark* facilitent le parallélisme et permettent de traiter des ensembles de données volumineux tout en réduisant le temps d'exécution et en optimisant l'utilisation des ressources.

2.9 Conclusion

Ce chapitre offre une vue d'ensemble exhaustive des différentes méthodes issues des deux premières approches de la CML, à savoir TP et AA. Après avoir introduire les concepts fondamentaux de la CML, nous avons exploré en détail les méthodes les plus connues.

Par la suite, une analyse comparative des deux approches a été réalisée, suivie d'une présentation des différents défis scientifiques inhérents au domaine de la CML. De plus, nous avons examiné les différentes caractéristiques spécifiques des DML, ainsi que les mesures d'evaluation conçues pour évaluer la performance et la robustesse des modèles de CML.

Suite aux défis scientifiques introduits par les approches TP et AA de CML, le chapitre suivant se concentre sur l'approche Ensemble, qui est considérée comme une solution efficace et adaptée pour résoudre les différents défis rencontrés par le domaine de la CML.

3

CLASSIFICATION EN ENSEMBLE

Sommaire

3.1 Introduction

L'approche ensembliste proposée dans l'apprentissage conventionnel constitue une avancée significative dans le domaine de l'IA [106]. Elle a été caractérisée particulièrement par son efficacité à améliorer les performances prédictives des systèmes d'apprentissage. Son objectif est de surmonter les lacunes inhérentes aux méthodes individuelles en combinant leurs forces et en compensant leurs faiblesses [107].

Un modèle Ensemble regroupe plusieurs classifieurs de base, qui peuvent être homogènes ou hétérogènes en termes d'algorithme d'apprentissage ou de paramètres. La prédiction finale de l'ensemble est obtenue en combinant les prédictions générées par tous les classifieurs en utilisant des techniques d'agrégation simples telles que le vote majoritaire ou la moyenne des prédictions, ou des techniques avancées comme le Bagging, le Boosting, le Random Forests et le Stacking. Cependant, l'approche d'ensemble conventionnelle n'a pas toujours réussi à répondre aux exigences découlant de la diversité des données provenant de multiples sources.

L'approche Ensemble de l'apprentissage Multi-Label a également joué un rôle crucial en raison de la complexité inhérente à ce type de problème. Elle a prouvés ses grandes compétences en attenant les défis scientifiques émergent par le domaine tels que, la dépendance entre labels, le déséquilibre des labels ainsi que la haute dimensionnalité de l'espace de sortie [108]. De plus, l'une des principales forces de cette approche réside dans son aptitude à exploiter la diversité des modèles individuels en utilisant des ensembles de données contenant des caractéristiques très spécifiques. En combinant les modèles de base, chacun formé avec ses propres propriétés et biais, l'ensemble est capable de capturer une gamme plus large de relations entre les caractéristiques des instances et les labels associés. Cela permet au modèle global de fournir des prédictions adaptées à la complexité des DML. Il est important de noter que la majorité des modèles ensemble pour la CML ont été principalement développés à partir des méthodes des approches de Transformation de Problème (TP) ou d'Adaptation d'Algorithme (AA).

Dans ce chapitre, nous introduisons tout d'abord le concept de classification en ensemble conventionnel, suivi d'une exploration approfondie des méthodes populaires de l'approche Ensemble pour la CML. Cette exploration offre une vue d'ensemble du fonctionnement des différentes méthodes de cette approche, ainsi que de leur pertinence dans ce domaine.

Nous présentons également les différentes taxonomies des Classifieurs Ensemble Multi-Label (CEML) proposées par la littérature, suivie d'une nouvelle proposition basée sur les problèmes posés par le domaine. Enfin, nous clôturons ce chapitre par une analyse approfondie des performances de plusieurs méthodes ensemblistes dans le domaine de la Bioinformatique, afin de déterminer le choix de la méthode la plus appropriée face à la nature complexe des DML utilisées.

3.2 Paradigme d'Ensemble Conventionnel

Les modèles d'ensemble conventionnels se sont révélés extrêmement utiles en combinant les forces et en compensant les faiblesses des classifieurs individuels [106]. Leur efficacité est grandement influencée par la sélection minutieuse des classifieurs de base et par l'utilisation d'une stratégie d'agrégation optimale. À cet égard, plusieurs stratégies avancées sont disponibles, telles que le Bagging [109], le Boosting [110], le Random Forests [111] et le Stacking [112].

3.2.1 Le Bagging

L'ensachage [109], ou Bagging (**B**ootstrap **Agg**regat**ing**) en anglais, est une technique d'ensemble en Machine Learning qui vise à améliorer la performance des modèles prédictifs en réduisant la variance et en améliorant la généralisation du modèle global. Le Bagging permet d'entraîner plusieurs modèles de base indépendants sur des ensembles d'échantillons bootstrap, créés aléatoirement avec remplacement à partir des données d'apprentissage. Chaque modèle de base formé sur un ensemble bootstrap est capable de générer sa propre prédiction pour de nouveaux exemples. Pendant la classification d'un nouvel exemple, les prédictions produites par les différents sous-modèles sont agrégées par un vote majoritaire.

En introduisant de la diversité dans les modèles par le rééchantillonnage [113], le Bagging permet d'atténuer le surajustement aux données d'apprentissage spécifiques et d'améliorer la capacité des modèles à généraliser correctement sur de nouvelles données.

3.2.2 Le Boosting

Le renforcement [110], ou Boosting en anglais, est une technique d'apprentissage faible homogène qui se déroule de manière séquentiellement et adaptative.

En se concentrant principalement sur l'algorithme AdaBoost, le processus commence par entraîner un premier modèle faible sur la base d'apprentissage. Un deuxième modèle faible est alors entraîné en tenant compte des erreurs précédentes, de sorte que le modèle M_i est ajusté pour corriger les erreurs faites par son prédécesseur M_{i-1}. Ce processus est répété séquentiellement pour former une série de modèles, où chaque modèle M_i tente d'améliorer sa prédiction en corrigeant les erreurs de son prédécesseur.

Le processus itératif de Boosting poursuit ses itérations jusqu'à ce que l'erreur globale du modèle soit suffisamment réduite pour produire un modèle prédictif solide. Son principal objectif est de réduire les biais et d'améliorer la généralisation du modèle final.

3.2.3 Le Random Forests

Les forets aléatoires [111], ou Random Forests en anglais, est une technique d'ensemble avancée reposant sur le principe de "diviser pour mieux régner". Cette stratégie comporte un grand nombre d'arbres de décision, chacun formé sur une sélection aléatoire d'un sous ensemble de données d'entraînement en utilisant le Bagging à chaque modèle. Cette opération est réalisée à l'aide de l'algorithme CART (Classification And Regression Trees), qui divise récursivement les données en sous-ensembles homogènes.

Lors de la construction de chaque arbre, une autre sélection est effectuée sur un sous-ensemble aléatoire de caractéristiques, utilisé pour rechercher le meilleur split à chaque nœud de l'arbre. Ensuite, la prédiction finale de la forêt est obtenue par l'agrégation des prédictions générées par les arbres individuels en utilisant un vote majoritaire. Cette technique d'ensemble est largement sollicitée en raison de sa robustesse face aux problèmes liés à la grande dimensionnalité des données.

3.2.4 Le Stacking

L'empilement [114], ou Stacking en anglais, est une technique d'ensemble avancée qui permet d'explorer une large gamme de modèles pour résoudre un problème spécifique. Ainsi, elle adopte une approche multi-modèle qui peut appréhender différentes facettes du problème.

Le Stacking consiste à combiner les prédictions de plusieurs classifieurs de base à l'aide d'un méta_classifieur.

Les modèles de base sont des algorithmes d'apprentissage supervisé traditionnels entraînés sur l'ensemble complet des données d'apprentissage. Ensuite, un méta-modèle est formé sur les sorties des modèles du niveau précédent (niveau de base) pour générer la prédiction finale de l'ensemble. Pour cela, le méta-modèle apprend à exploiter les forces des différents modèles de base et à atténuer leurs faiblesses.

Cette technique d'agrégation constitue une approche flexible et puissante capable d'exploiter la diversité des algorithmes pour obtenir des prédictions plus précises contribuant ainsi à améliorer la performance globale de l'ensemble. Il est évident de constater que le choix entre les méthodes ensemblistes citées précédemment dépend de plusieurs critères importants [106], [115], [116], tels que : la taille et la nature des données traitées, le type de modèle de base, l'objectif de la modélisation, ainsi que le temps de calcul et la complexité de la méthode. La Table 3.1 résume un état comparatif de ces méthodes.

Table.3.1 Analyse comparatif des méthodes ensemblistes conventionnelles les plus populaires

Critère	Bagging	Boosting	Stacking	Random Forest
Principe de fonctionnement	Échantillonnage (avec remplacement) pour créer des sous-ensembles aléatoires de données	Réduit l'erreur en corrigeant les erreurs des classifieurs précédents	Combinaison de plusieurs modèles hétérogènes	Variante de Bagging avec sélection aléatoire des caractéristiques
Taille et nature des données	Adapté aux grandes bases de données et caractéristiques	Sensible aux données déséqui-librées, et efficace sur des ensembles de taille moyenne	Nécessiter plus de données et un bon méta-modèle pour éviter le sur-apprentissage.	Adapté aux grandes bases de données et caractéristiques
Objectif	Réduction de la Variance	Réduction du Biais	Combinaison de Modèles	Réduction de la variance

Critère	Bagging	Boosting	Stacking	Random Forest
Temps de calcul et complexité	Coûts computationnels modérés	Plus gourmands en temps de calcul avec un grand nombre de modèles	Plus gourmands en temps de calcul avec un grand nombre de modèles	Coûts computationnels modérés

3.3 Approche d'Ensemble pour CML

Dans cette section, nous commençons par présenter une définition formelle de l'approche Ensemble pour la CML. Ensuite, nous passons en revue neuf méthodes ensemblistes connues dans le domaine, développées à partir des deux approches principales TP et AA [35], [117].

3.3.1 Définition Formelle

Le modèle ensemble pour la classification multi-label implique d'entraîner N classifieurs multi- label H_1, H_2, .., H_N. Pour une nouvelle instance $x_i \in$ X (ensemble d'instances), chaque modèle individuel k de l'ensemble diversifié de N modèles génère un vecteur P_k de *D_dimensions*, tel que $P_k = [P_{1k}, P_{2k}, \ldots, P_{mk}]$. Notant que, P_{Lk} représente la probabilité attribuée au label de classe L par le classifieur C_K. Pour obtenir la prédiction multi-label finale d'une nouvelle instance, les prédictions générées par chaque classifieur de base au sein de l'ensemble sont combinées en utilisant une stratégie d'agrégation spécifique [108].

3.3.2 Ensembles basés sur l'approche TP

Les méthodes ensemblistes de cette catégorie sont fondées sur des algorithmes indépendants qui transforment les problèmes de CML en plusieurs problèmes de classification mono-label. Ensuite, la prédiction Multi-Label de l'ensemble est obtenue en combinant des prédictions de ces sous-problèmes à l'aide d'un classifieur multi-label. Dans cette catégorie, nous recensons six méthodes principales, à savoir : EBR, ECC, EPS, RAKEL, HOMER et MLS.

La méthode EBR (Ensemble of Binary Relevance classifiers) [118] construit plusieurs classifieurs BR [25] à partir d'une sélection aléatoire de sous-ensembles d'instances. Chaque classifieur de base transforme le problème de CML en N problèmes de classification mono-label, en utilisant un classifieur par label. Dans le processus de création de la méthode EBR, chaque classifieur binaire doit déterminer si son label assigné est pertinent

ou non pour une nouvelle instance x_i en se basant sur les données d'entraînement. Ensuite, un seuil ε est appliqué pour décider les labels pertinents, ce qui génère l'ensemble final des labels prédits pour cette instance.

La méthode ECC (Ensemble of Classifier Chains) [26] entraîne plusieurs classifieurs CC, construits à partir d'une selection aléatoire de sous-ensembles d'instances en utilisant une séquence aléatoire de classifieurs binaires. Chaque classifieur successif dans la chaîne étend l'espace des caractéristiques du classifieur précédent, permettant ainsi de capturer efficacement les dépendances entre les labels. Pendant la classification, les prédictions de tous les classifieurs CC sont combinées via la moyenne des valeurs de confiance pour chaque label. Ensuite, une fonction de seuillage est appliquée pour identifier les labels les plus pertinentes, formant ainsi l'ensemble final des labels prédits.

La méthode EPS (Ensemble of Pruned Sets) [17] permet de construire plusieurs classifieurs PS [27] à partir d'une selection aléatoire de sous-ensembles d'instances sans remplacement. Son premier objectif est d'éliminer les échantillons contenant des combinaisons de labels rares, afin de permettre au modèle de se concentrer sur les ensembles de labels les plus pertinents. Ensuite, elle compense la perte d'informations en réintroduisant les échantillons élagués associés aux autres sous-ensembles de labels pertinents. Pendant la classification, les prédictions générées par tous les classifieurs PS sont combinées par un vote, et une fonction de seuillage ε est appliquée pour identifier les labels pertinents de l'ensemble final des labels prédits.

La méthode RAKEL (RAndom k-labELsets) [18] permet de générer un groupe de classifieurs LP en divisant aléatoirement un grand ensemble de label en N partitions de petite taille k (k-labels). Pour chaque partition, un classifieur LP est entrainé pour fournir des prédictions binaires pour chaque label de son ensemble de k-labels correspondant. Ensuite, ces prédictions sont combinées pour obtenir une prédiction multi-label via un vote majoritaire pour chaque label, comme illustré dans la Figure 3.1. RAkEL est nettement plus simple que la méthode LP, puisqu'elle ne gère que de petits sous-ensembles de labels simultanément. En outre, cette méthode aborde de manière efficace les ensembles de labels non présents dans l'ensemble d'apprentissage. Ainsi, lorsque l'on rencontre un nouvel ensemble de labels lors de la prédiction, la méthode se base sur les prédictions des classifieurs de base formés sur des ensembles de k-labelsets similaires pour prédire l'ensemble de labels inédit.

Modèle	3-Labelsets	Prédictions					
		L1	L2	L3	L4	L5	L6
M1	{L1, L2, L3}	0	1	1	-	-	-
M2	{L2, L3, L4}	-	1	0	1	-	-
M3	{L1, L5, L6}	1	-	-	-	0	1
M4	{L3, L5, L6}	-	-	0	-	1	1
M5	{L1, L4, L5}	0	-	-	0	1	-
M6	{L1, L2, L4}	1	0	-	1	-	-
M7	{L3, L5, L6}	-	-	1	-	0	0
Moyenne des votes		2/4	2/3	2/4	2/3	2/4	2/3
Prédiction Finale		0	1	0	1	0	1

Figure 3.1 Exemple d'application de la méthode RAKEL

La méthode HOMER (Hierarchy Of Multi-label classifiERs) [20] a été développée pour gérer les ensembles de DML extrêmes. Cette méthode utilise le Clustering équilibré pour maintenir une distribution uniforme d'un ensemble de labels en sous-ensembles disjoints, de sorte que les labels similaires soient regroupés ensemble et les labels dissimilaires séparés. Pendant la classification, HOMER utilise un classifieur BR [25] par nœud et transmet l'instance à chaque nœud-enfants lorsque le parent a prédit l'un de ses labels. Les labels prédits par les feuilles sont ensuite combinées pour générer la prédiction finale de l'ensemble. La Figure 3.2 illustre le fonctionnement de la méthode HOMER comportant neuf labels.

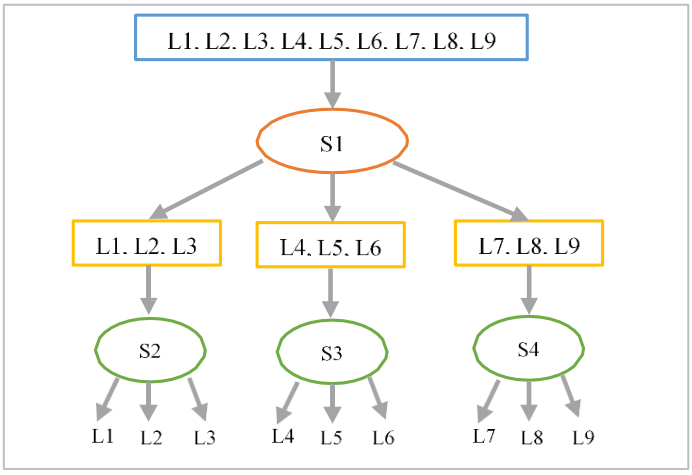

Figure 3.2 Exemple d'application de la méthode HOMER

La méthode MLS (Multi-Label Stacking) [119] est un empilement à deux niveaux : le niveau de base et le méta-niveau. Au niveau de base, *N* classifieurs BR [25] indépendants sont entrainés, chacun étant associé à un label spécifique. Le méta-modèle construit à partir d'un nouvel ensemble de BRs, utilise les prédictions du niveau précédent avec ou sans augmentation de l'espace des caractéristiques dans le pipeline. Ainsi, la sortie générée par le méta-modèle représente la prédiction Multi-Label finale d'un nouvel exemple, comme illustré dans la Figure 3.3.

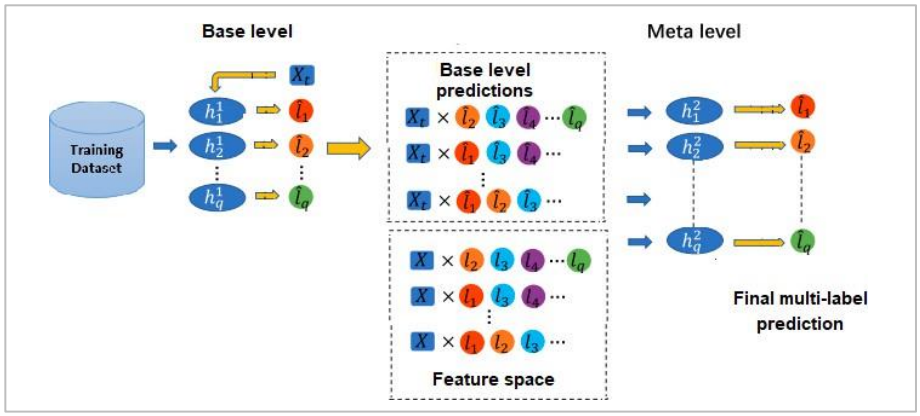

Figure 3.3 Exemple d'application de la méthode MLS

3.3.3 Ensembles basés sur l'approche AA

Les méthodes ensemblistes de cette catégorie sont fondées sur des algorithmes classiques adaptés pour traiter efficacement des ensembles de DML. Dans cette catégorie, nous recensons trois méthodes principales, à savoir : RF-PCT, RFML-C4.5, TREML.

La méthode RF-PCT (Random Forest of Predictive Clustering Trees) [120] combine plusieurs classifieurs PCT (Predictive Clustering Trees) [28] construits à partir d'une sélection aléatoire d'instances et de caractéristiques. PCT étant un arbre de décision ancré dans une hiérarchie de clusters, son nœud racine, qui contient toutes les données, subit un partitionnement itératif en sous-clusters au sein de ses nœuds descendants. Ainsi, l'arbre prédit simultanément plusieurs labels, en utilisant un processus d'induction dans PCT pour identifier les sélections optimales à chaque nœud. Pendant la classification, les prédictions multi-label générées par les PCT sont regroupées par label et le sous-ensemble final de labels prédits est déterminé par un mécanisme de vote basé sur les probabilités pour chaque label.

La méthode RFML-C4.5 (Random Forest of Multi-Label C4.5 Decision Trees) [117] fonctionne selon le même principe que la méthode RF-PCT. Elle consiste à regrouper une collection de classifieurs ML-C4.5 [4], construits sur une sélection aléatoire de sous-ensembles de caractéristiques à chaque nœud de l'arbre. Le classifieur de base ML-C4.5 est une adaptation de l'algorithme C4.5 [31] dans un contexte de CML. Dans cette adaptation, les nœuds de l'arbre englobent les échantillons associés à un ensemble spécifique de labels. Pendant la classification, les prédictions générées par tous les ML-C4.5 sont combinées à l'aide d'un mécanisme de vote précis ou probabiliste pour chaque label.

La méthode TREMLC (Triple Random Ensemble for Multi-Label Classification) [121] construit son ensemble en utilisant des sous-ensembles de caractéristiques, de labels et d'instances. Cette approche combine trois méthodes de randomisation distinctes : la méthode du sous-espace aléatoire pour la sélection des sous-ensembles de caractéristiques, le Bagging pour sélectionner les sous-ensembles d'instances, et la méthode RAkEL [18] pour sélectionner les sous-ensembles de labels. Cependant, TREMLC fonctionne de manière itérative, en sélectionnant divers sous-ensembles à chaque itération. Les paramètres optimaux sont alors déterminés à l'issue de ces itérations pour générer l'ensemble final. Ensuite, une stratégie de vote majoritaire est appliquée pour prédire le sous-ensemble final des labels prédits.

3.3.4 Analyse comparative des méthodes ECML

Bien que de nombreuses recherches de référence se soient principalement concentrées sur les méthodes individuelles de CML, menant des expériences sur une diversité d'ensembles de données de référence [24], [122], [123], [124], [125] [126], peu d'études ont abordé les méthodes ensemble [127], [128], [129], [130], [131]. Les Tables.3.2 et 3.3 présentent une analyse comparative des méthodes ECML les plus connues dans le domaine, basées sur les deux approches TP et AA. Cette analyse peut s'avérer très utile pour la selection de la meilleure méthode en fonction des besoins spécifiques d'une tâche donnée.

Table.3.2 Analyse comparative des méthodes ECML basées dur l'approche TP

Méthode	Avantages	Inconvénients
EBR [118]	▪ La génération des classifieurs de base à partir des sélections aléatoires d'instances améliore nettement les performances prédictives.	▪ Ignore la corrélation entre les labels.
ECC [26]	▪ Capture les dépendances entre labels en utilisant une séquence ordonnée de prédictions.	▪ Performance affectée par l'ordre dans lequel les labels sont chaînés, nécessitant potentiellement une optimisation supplémentaire. ▪ Entraîner plusieurs chaînes de classifieurs peut être coûteux en temps et en ressources
EPS [17]	▪ Maintenir un bon équilibre entre les classes déséquilibrées et la préservation de la diversité des labels en réintroduisant les échantillons élagués. ▪ Prédire des combinaisons de labels qui ne sont pas présentes dans l'ensemble de données d'apprentissage.	▪ Requiert une autre classification lorsque la prédiction multi-label finale n'est pas identifiée par l'un des classifieurs de base. Cette classification supplémentaire est obtenue en ajustant les critères de sélection pour inclure d'autres PSs dans l'ensemble final.
RAKEL [18]	▪ Gère le déséquilibre des classes de données en réduisant l'impact des labels rares et les inclure dans des sous-ensembles spécifiques. ▪ Possibilité de prédire les ensembles de labels non présents dans l'ensemble d'apprentissage.	▪ Gestion partielle du problème de dépendance entre labels au sein des sous-ensembles de labels. ▪ La performance du modèle dépendre d'une sélection soignée de la taille des sous-ensembles de labels.
HOMER [20]	▪ Modélise les dépendances de haut niveau entre labels. . Bonne gestion des données à grande échelle via une hiérarchie des labels, réduisant la complexité du problème.	▪ Adapté uniquement aux structures hiérarchiques de labels. ▪ Complexité de mise en œuvre de la hiérarchie nécessitant des connaissances spécifiques sur les relations inter-labels.
MLS [119]	▪ Simplifie la gestion de la dimensionnalité en utilisant un méta-modèle pour combiner les résultats de plusieurs classifieurs de base.	▪ Néglige la corrélation par paire entre labels. ▪ Ignore les poids des classifieurs individuels lors de leur sélection, conduisant à une perte potentielle d'informations sur les classifieurs.

Table.3.3 Analyse comparative des méthodes ECML basées sur l'approche AA

Méthodes	Avantages	Inconvénients	Ref
RF-PCT [120]	▪ Possibilité de réduire la complexité de la dimensionnalité de l'espace de sortie en regroupant les labels similaires dans des sous-ensembles plus petits.	▪ Le partitionnement en clusters peut être complexe à optimiser, surtout avec des données de grande dimension.	[120]
RFML-C4.5 [117]	▪ Offre une approche robuste pour gérer les données déséquilibrées en utilisant une forêt d'arbres diversifiés.	▪ Performance sensible à la sélection des sous-ensembles de caractéristiques qui nécessite un réglage fin.	[117]
TREMLC [121]	▪ Gestion des sous-ensembles diversifiés réduit le sur-apprentissage et gère les dépendances entre les labels.	▪ Augmentation du coût computationnel et du temps d'apprentissage via la combinaison des trois méthodes de randomisation. ▪ Difficulté de paramétrage optimal pour les trois randomisations (sous-espaces aléatoires, Bagging, et RAkEL).	[121]

3.4 Taxonomies des méthodes ECML

De nombreuses études dans le domaine de CML se sont principalement concentrées sur les approches de TP et AA [123], [124], [125], [126], [24], [122], mentionnant brièvement les méthodes d'ensemble avec peu de taxonomies citées dans la littérature.

La première taxonomie des méthodes ECML, suggérée par Madjarov et al. [117], était fondée sur les approches TP et AA, comme illustré sur la Figure 3.4. Ensuite, Herrera et al. [129] ont introduit une nouvelle catégorisation des méthodes en fonction du type de classifieur de base, qu'il soit binaire, multi-classe ou hiérarchique, comme le montre la Figure 3.5. Cette dernière taxonomie s'inspire de celle proposée par Zhang et Zhou [35] pour les méthodes des approches TP et AA.

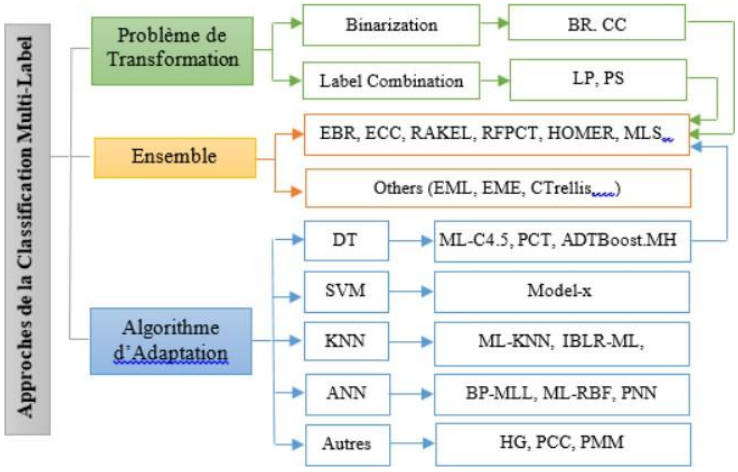

Figure.3.4 Taxonomie des méthodes ECML basée sur les approches TP et AA

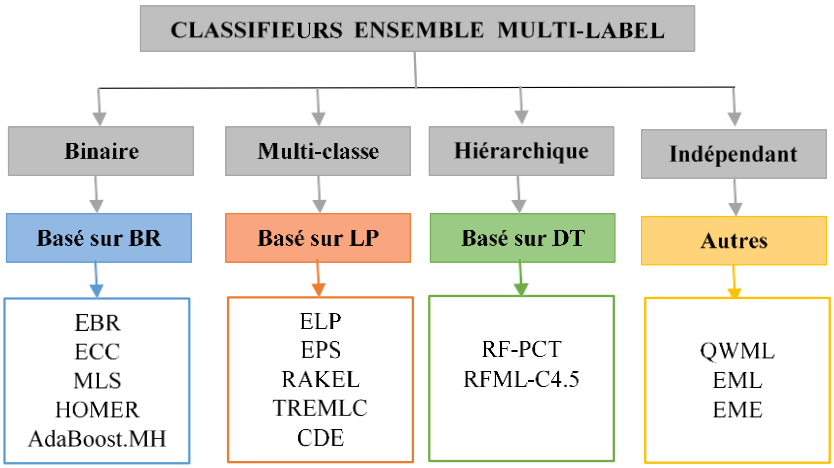

Figure.3.5 Taxonomie des méthodes ECML basée sur le type de classifieur de base

En plus des taxonomies précédentes, Moyano et al. [127] ont proposé une catégorisation intrigante de ces méthodes basée sur le niveau de diversité pour générer l'ensemble. Comme le montre la Figure 3.6, cette catégorisation définit deux niveaux de diversité :

i) Type d'algorithmes d'apprentissage : à ce niveau de diversité, les classifieurs de base de l'ensemble peuvent être construits avec différents algorithmes [132], ou avec le même algorithme en utilisant différents paramètres [26].

ii) Sélection aléatoire d'espaces d'échantillonnages : chaque classifieur de base est formé sur un sous-ensemble différent de l'espace d'instances, de labels ou de caractéristiques.

Cependant, certains CEML peuvent construire leurs membres sur deux niveaux de diversité ou plus. Pour une explication plus détaillée, la Table 3.4 résume les différents niveaux de diversité de quelques méthodes ensemblistes.

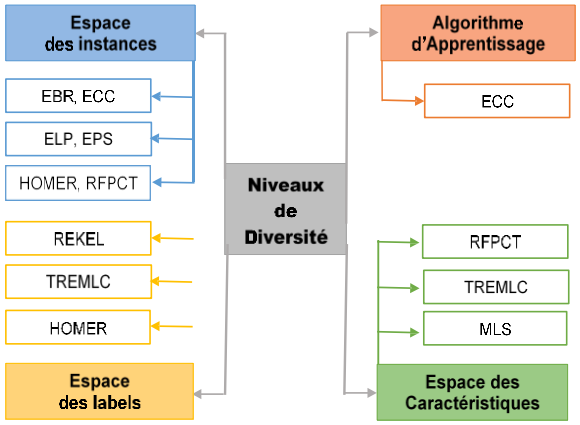

Figure 3.6 Taxonomie des méthodes ECML basée sur le niveau de la diversité

Table.3.4 Niveaux de diversité de quelques méthodes ECML connues.

Méthode	Niveau de Diversité	Performances
ECC	Espace d'instances Algorithme d'apprentissage	Utilisation d'un ordre aléatoire des chaînes et une sélection aléatoire de sous-ensembles d'instances.
MLS	Espace de caractéristiques	Utilisation d'un sous-ensemble distinct de caractéristiques par chaque classifieur BR.
EPS	Espace d'instances	Sélection aléatoires de sous-ensembles d'instances par chaque classifieur PS.
RF-PCT	Espace d'instances Espace de caractéristiques	Sélection aléatoire d'un sous-ensemble d'instances par le Bagging à chaque modèle, et une autre sélection d'un sous-ensemble de caractéristiques à chaque nœud de l'arbre.
RakEL	Espace de labels	Sélection aléatoires de sous-ensembles de labels par chaque classifieur LP.

Certaines études intéressées par l'exploitation des dépendances entre les labels [133], [35], conduisant à la proposition d'une taxonomie de méthodes d'ensemble basée sur différents degrés de dépendance entre les labels. Ainsi, trois degrés ont été distingués : le Premier Ordre, le Deuxième Ordre et l'Ordre Elevé. Il convient de noter que seuls le premier et le dernier ordre concernent les classifieurs EML. Pour le premier ordre, il ignore la coexistence des labels supplémentaires et analyse chacun d'eux indépendamment. Les méthodes d'ensemble concernées par cet ordre sont EBR et RFML-C4.5. En ce qui concerne l'ordre élevé, l'impact de chaque label sur les autres est pris en compte, et un ordre élevé est détecté parmi tous les labels ou parmi des sous-ensembles de labels sélectionnés au hasard. EPS, ECC et RAKEL sont les méthodes concernées par ce type d'ordre.

Les taxonomies évoquées précédemment révèlent que les défis inhérents à l'apprentissage Multi-Label ne sont pas explicitement pris en compte par les auteurs. Dans cette optique, nous proposons une nouvelle taxonomie des méthodes ECML, présentée dans la Table 3.5, axée sur les problèmes introduits par le domaine, notamment la dépendance entre labels (A), le déséquilibre des classes (B), et la dimensionnalité de l'espace de sortie (C).

La synthèse résultant de l'analyse de cette table permet de conclure que le choix de la méthode appropriée dépend étroitement du problème spécifique abordé, en tenant compte des caractéristiques complexes des données expérimentées. Par conséquent, il serait bénéfique de développer un système polyvalent et adaptable, capable de relever les défis variés de l'apprentissage Multi-Label en exploitant la collaboration de sous-modèles diversifiés et complémentaires.

Table 3.5 Taxonomie des méthodes ECML de pointe basée sur le problème traité, la dépendance entre labels (A), le déséquilibre des classes (B), et la dimensionnalité de l'espace de sortie (C).

Travail de Recherche	Approche Proposée	Classifieurs de base	Problème Traité	Explication
[11] /2007/	RAKEL	LPs	B, A	▪ Réduit l'impact des labels rares en ne les incluant que dans des sous-ensembles spécifiques. ▪ Capture certaines dépendances locales entre les labels au sein de ces sous-ensembles.
[14] /2007/	RF_PCT	PCTs	B	▪ Simplifie la gestion de la dimensionnalité de l'espace de sortie en regroupant les labels similaires dans des sous-ensembles plus petits.
[9] /2008/	EPS	PSs	B	▪ Maintenir un bon équilibre entre la gestion des classes déséquilibrées et la préservation de la diversité des labels en réintroduisant les échantillons élagués.
[12] /2008/	HOMER	BRs	A	▪ Gestion des dépendances entre labels en utilisant une hiérarchie de classifieurs pour modéliser les relations entre les labels.
[8] /2009/	ECC	CCs	A	▪ Gestion des dépendances entre labels en prenant en compte l'ordre dans lequel ils sont prédits.
[13] /2009/	MLS	2BRs	A, C	▪ Gestion de la dépendance globale de labels au niveau méta en combinant toutes les prédictions issues de niveau de base. ▪ Simplifie la gestion de la dimensionnalité en utilisant un méta-modèle pour combiner les résultats de plusieurs classifieurs de base.
[19] /2010/	TREMLC	LPs	A	▪ Gestion des dépendances entre labels en utilisant un ensemble de classifieurs randomisés qui s'adaptent aux interactions entre les différentes labels et de mieux les modélisés.
[16] /2012/	RFML-C4.5	ML-C4.5	B	▪ Bonne gestion des données déséquilibrées en utilisant une forêt d'arbres diversifiés.

3.5 Construction d'un modèle ECML

3.5.1 Théorie de construction d'un ECML

Dans le contexte général des modèles de CML, chaque classifieur individuel est chargé de prédire la présence ou l'absence d'un label spécifique. Les prédictions de tous ces classifieurs sont ensuite combinées pour prédire le sous ensemble final de labels d'une instance donnée. Cependant, dans les modèles ensemble pour CML, un pool initial est généré par Q classifieurs multi-label CML_1, CML_2,.., CML_Q, chacun fournissant une décision pour chaque label L_i. Ensuite, les sous-ensembles de labels prédits par tous les classifieurs sont combinés pour générer le sous-ensemble final des labels prédits pour une nouvelle instance x_i [134], comme défini par la Figure 3.7.

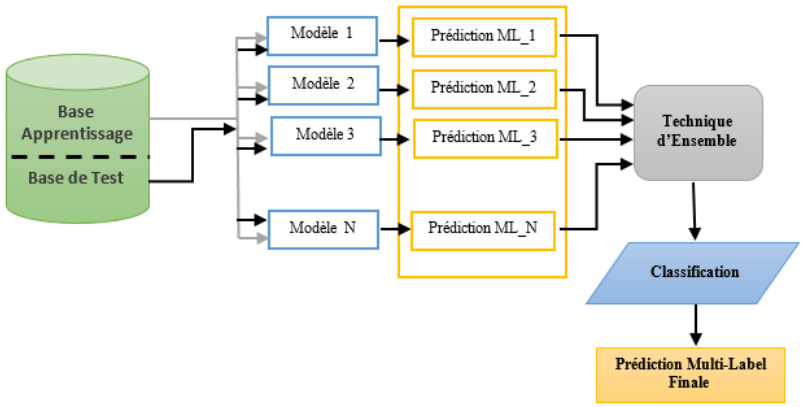

Figure 3.7 Processus de construction d'un modèle ECML

3.5.2 Stratégies de combinaison de classifieurs individuels

La construction des méthodes ECML repose sur des techniques d'agrégation simples telles que, *la Moyenne ou le Vote*, ou sur des stratégies avancées comme *le Bagging* ou *le Stacking*.

La *Vote Majoritaire* constitue la première stratégie d'agrégation commune pour construire un classifieur EML [17], [18], [121], [135], [136]. Chaque classifieur de base, entraîné sur l'ensemble des données d'apprentissage, génère des prédictions pour chaque label de sortie [137]. Ensuite, les prédictions de tous les classifieurs de base sont agrégées en attribuant à chaque label la valeur prédite par le vote majoritaire parmi tous les classifieurs de base. Afin d'obtenir des prédictions multi-label plus précises et plus efficaces, le *Vote Pondéré* a été adapté par un certain nombre d'études [26].

L'utilisation des seuils spécifiques dans cette stratégie a permis de déterminer si le label est pertinent ou non, en fonction du nombre de votes reçus.

Il convient de noter que le schéma de vote permet de réduire l'impact du bruit et l'incertitude dans les prédictions. Néanmoins, il ignore les dépendances entre labels en traitant tous les labels avec la même importance, ce qui ne correspond pas aux scénarios de CML où la pertinence des labels peut varier [138].

Le Bagging est également l'une des techniques sur laquelle repose la majorité des méthodes ECML [118], [20], [120], [121]. Elle permet de générer divers classifieurs de base et combine leurs prédictions par un vote majoritaire pour chaque label. Pendant la prédiction, les valeurs de confiance de tous les sous-modèles pour chaque label sont moyennées. Ce schéma d'agrégation à une grande capacité à améliorer la généralisation du modèle global et à réduire le risque de surajustement. Néanmoins, il ne tiens pas compte des dépendances locale entre les paires de labels.

Récemment, une attention considérable a été accordée aux progrès de l'approche *Stacking* dans l'apprentissage Multi-Label [130], [119], [139], [140]. Cette stratégie consiste à construire une pile de deux niveaux de classifieurs de base, où les sous-modèles du niveau de base sont entraînés à prédire la probabilité associée à chaque label et le méta-modèle est entraîné pour combiner ces probabilités en un sous-ensemble de labels.

Le *Stacking* a démontré des résultats particulièrement prometteurs par rapport à d'autres techniques d'ensemble, d'où il a été appliqué de diverses manières. Par exemple, il était utilisé pour capturer les corrélations entre les labels et faciliter la sélection des caractéristiques pour identifier les sous-ensembles des labels spécifiques [139], [141]. De même, il a été utilisé pour combiner les prédictions des classifieurs sous-jacents utilisant des règles différentes [142], [143]. En outre, cette stratégie a été employée également dans des processus hiérarchiques pour réduire la complexité de l'échantillonnage [144], et dans des méthodes d'élagage pour sélectionner les labels les plus pertinents afin de préserver les informations utiles pour des prédictions précises [119], [140].

Cette technique a prouvé son efficacité en tenant compte de l'historique des échantillons analysés et en permettant aux sous-modèles d'ajuster leurs prédictions sur la base des erreurs corrigées. Cependant, il néglige l'effet de la dépendance des labels par paire [134]. La Table 3.6 présente un résumé de quelques méthodes ECML récentes construites par les techniques d'agrégations élaborées ci-dessus, telles que citées dans la littérature.

Table.3.6 Stratégies de constructions des méthodes ECML.

Travail de Recherche	Année de Recherche	Approche Proposée	Technique d'agrégation	Performances	Jeux de données
[145]	2020	PTC et CTP	Binary Majority Voting (BMV) Graded Majority Voting (GMV)	PTC (Predict Then Combine) et CTP (Combine Then Predict) sont plus performants que les méthodes ensemble basées sur les techniques de vote standard.	Cal500, Emotions, Scene, Yeast, Bibtex, Mediamill, Medical.
[136]	2020	Ensemble Classifier Chain	Vote Majoritaire	L'approche utilisant DT comme classifieur de base a donné de bons résultats avec une précision moyenne de 0,97 %	ARAS
[146]	2022	Ensemble de Bayésiens Naïfs		Meilleur score obtenu de Hloss avec une valeur de 0.116.	Les versets du Coran traduits en anglais.
[147]	2022	CML sur les versets de Hadiths	Bagging et CART	Le Bagging a donné les meilleurs scores obtenus pour Hloss =0,191% et Accuracy =80,86 %.	Les versets du Hadiths traduits en indonésien.
[148]	2023	MLDE (MLC with Dynamic Ensemble learning)	Bagging	La selection dynamique des classifieurs de base utilisant le Bagging a prouvé sa grande efficacité par apport aux méthodes connexes.	Vingt-quatre jeux de données du Référentiel Mulan.
[140]	2020	RFS (ReliefF-based Stacking)	Stacking	L'approche est plus efficace que les autres méthodes d'ensemble avancées en capturant les dépendances entre labels, puis en sélectionnant les plus pertinents.	Emotions, Genbase, Yeast, Birds, Medical.
[139]	2021	SMLS (Stacking Model with Label Selection)		L'approche a démontré sa performance par rapport aux méthodes de pointe et connexes en réduisant la sensibilité au bruit et la durée d'exécution.	Education, Genbase, Recreation, Science, Yeast, Arts, Medical.
[149]	2023	StaC (Stacked Chaining)		L'approche proposée est plus performante que les autres méthodes de pointe pour différents paramètres d'évaluation des performances.	Douze datasets ML du Référentiel Mulan.

Pour atténuer les limitations des techniques d'ensemble mentionnés précédemment, l'approche pondérée a été appliquée dans diverses applications réelles de CML [150], [151], permettant ainsi au modèle de s'ajusté plus précisément aux caractéristiques et aux exigences spécifiques du problème. Ainsi, pour obtenir des systèmes plus précis, l'approche Pondérée s'est avérée bien adaptée au schéma de Stacking [138], [130], [152], [153]. L'estimation de pondérations distinctes aux différents niveaux a permis d'améliorer la précision au niveau de base du processus Stacking en générant des prédictions plus pertinentes.

Bien que ces études aient permis d'évaluer la performance des modèles de CML en intégrant un mécanisme de pondération à différents niveaux, aucune d'entre elles n'a examiné l'impact de la pondération des labels sur la robustesse des systèmes d'apprentissage. Certaines études se sont concentrés sur la pondération des classifieurs de base [130], [138], [132], tandis que d'autres ont porté sur la pondération des caractéristiques [152]. La Table 3.7 présente un résumé des méthodes ECML récentes basées sur les différentes techniques du Stacking Pondéré.

Table.3.7 Synthèse des méthodes ECML connexes basées sur le Stacking Pondéré.

Travail de Recherche	Approche Proposée	Niveau de Pondération	Performances Obtenues	Jeux de données
[132] /2012/	EML (Ensemble Multi-Label)	Classifieur	L'approche a fourni une solution très précise et efficace par rapport aux méthodes de pointe. Cependant, la méthode de sélection des classifieurs de base peut avoir un impact sur les techniques de combinaison utilisées, entraînant ainsi des performances différentes pour des domaines de problèmes spécifiques.	Enron, Medical, Scene, Pascal07, Yeast, Emotions.
[130] /2018/	GOOWE-ML (Geometrically Optimum Online Weighted Ensemble for Multi-Label classification)	Classifieur	L'approche a prouvé son efficacité par rapport aux principaux modèles de Bagging. Cependant, l'ajustement de la taille optimale du pool n'est pas pris en compte par rapport à la dimension de l'espace des labels et des caractéristiques, ainsi que la cardinalité des labels, ce qui peut entraîner une dégradation des performances.	20NG, Yeast, Ohsumed, IMDB, Slashdot, Reuters, TMC2007.
[152] /2021/	WKNNS (Staking Weighted k-Nearest Neighbour)	Caractéristique	L'approche a réalisé de meilleures performances par rapport aux autres algorithmes de pointe en résolvant le problème des données déséquilibrées. Cependant, étant donné que KNN est sensible aux données bruitées, leur présence peut influencer les prédictions du modèle, et les poids attribués par le modèle peuvent amplifier les effets du bruit, entraînant une dégradation des performances.	Vingt-deux datasets ML du référentiel Mulan.
[138] /2021/	MLWSE (Multi-Label Weighted Stacked Ensemble)	Classifieur	L'approche a confirmé ses performances compétitives par apport aux méthodes de pointe ainsi que son efficacité dans les applications de diagnostic médical assisté. Cependant, l'approche n'est pas adaptée aux données à grande échelle.	Treize datasets ML et datasets relatifs aux maladies cardiovasculaires/ cérébrovasculaires

3.6 Cas pratique : Analyse de l'impact des méthodes ECML en Bioinformatique

En Bioinformatique, les ensembles de données diversifiés requièrent souvent l'attribution de plusieurs labels afin de référencer précisément une instance donnée. La génomique fonctionnelle offre un exemple frappant, où un gène peut être associé à une multitude de fonctions incluant le métabolisme, la synthèse des protéines et la transcription. De même, dans le contexte des données textuelles cliniques, telles que les dossiers médicaux électroniques des patients, une large gamme de labels peut être attribuée, comprenant notamment les codes de diagnostic, les antécédents médicaux, les listes de maladies, ainsi que des détails sur les procédures pratiquées sur les patients.

Ainsi, la CML est devenue un paradigme puissant en Bioinformatique, permettant d'analyser des données biologiques complexes et d'obtenir des informations précieuses sur les relations entre différents phénomènes biologiques. Ce paradigme de classification a été appliqué avec succès dans divers domaines de la Bioinformatique, notamment pour l'analyse des données textuelles cliniques [7], des données géniques [58], et des données sur les protéines [156].

Bien que CML ait réalisé des progrès significatifs dans le domaine de la Bioinformatique [156], [157], [158], [159], [160], [161], elle a également introduit de nouveaux défis qui doivent être relevés pour une analyse efficace des données biologiques complexes. Parallèlement, l'approche Ensemble est révélée être un succès notable dans ce domaine [160], [50], [162]. Malheureusement, le nombre d'articles de recherche qui fournissent des vue d'ensemble [163], des études comparatives [117], [164] et des enquêtes [165] sur l'approche ECML reste relativement limité.

Pour évaluer l'impact de l'approche Ensemble dans le domaine de la Bioinformatique, nous avons effectué une comparaison exhaustive des performances des méthodes ensemblistes les plus répandus de cette approche sur des ensembles de données Bioinformatique multi-label de référence. Notre objectif est d'identifier la méthode ensembliste la plus performante dans divers scénarios, tout en fournissant des directives pour la sélection de l'algorithme le plus approprié en fonction des caractéristiques complexes des ensembles de DML expérimentés.

Pour mener cette analyse comparative, nous avons sélectionné sept ensembles de données provenant du référentiel Mulan (https://www.uco.es/kdis/mllresources/), couvrant divers domaines d'application de la Bioinformatique, notamment la Biologie et la Médecine. Les ensembles de données biologiques comprennent Genbase [58], HumanPse [166], PlantPse [166], et Yeast [56], tandis que les ensembles de données médicales incluent CHD_49 [158] et Medical [157]. Pour évaluer notre analyse, nous avons utilisé cinq différentes mesures d'évaluation, à savoir trois métriques basées sur les exemples (Hloss, Accuracy et F1_score), ainsi que deux métriques basées sur les labels (Micro-F1 et Macro-F1), telles que définies dans le Chapitre 2, Section 2.7.

Dans ce qui suit, nous présentons les résultats de l'évaluation expérimentale de huit méthodes ECML les plus connus, dressés sur les cinq Tables 3.7, 3.8, 3.9, 3.10 et 3.11. Pour chaque type de mesure d'évaluation, nous présentons et discutons les résultats des tests suivants :

La Table 3.8 présente les performances prédictives des algorithmes ECML sélectionnées en termes de Hamming loss (Hloss). Pour le dataset *CHD_49*, EBR a la plus faible Hloss (0.294), tandis que TREMLC et ECC ont les plus élevées (0.329). Sur le dataset *Genbase*, ECC, RAKEL et HOMER se démarquent avec une Hloss de 0.001, tandis que RF-PCT a une performance inférieure avec 0.049. Pour le dataset *HumanPse*, les algorithmes ont des valeurs proches, EPS étant légèrement meilleur (0.082) et HOMER moins performant (0.121). Dans le cas du dataset *Medical*, ECC et HOMER ont les meilleures performances (0.010), alors que RF-PCT se situe au-dessus de la moyenne avec 0.025. Sur le dataset *PlantPse*, EPS, MLS et RF-PCT affichent une Hloss de 0.089, démontrant leur efficacité, contrairement à HOMER (0.138). Concernant le dataset *VirusGo*, RAKEL et MLS se démarquent avec une Hloss de 0.038, tandis que TREMLC a une performance nettement inférieure (0.143). Enfin, pour le dataset *Yeast*, EBR a la meilleure performance (0.201) et HOMER est la moins bonne (0.254).

Ces résultats soulignent clairement l'importance de la sélection des algorithmes en fonction des caractéristiques spécifiques du jeu de données utilisé pour optimiser les performances prédictives. Ainsi, les méthodes EPS et RAKEL étant particulièrement performantes sur plusieurs ensembles de données.

Table.3.8 Performance prédictives des EMLC en termes de Hloss.

Datasets	EBR	ECC	EPS	RAKEL	HOMER	MLS	RF-PCT	TREMLC
CHD_49	**0.294**	0.329	0.302	0.320	0.321	0.321	0.307	0.329
Genbase	0.002	**0.001**	0.002	**0.001**	**0.001**	**0.001**	0.049	0.010
HumanPse	**0.081**	0.084	0.082	0.092	0.121	0.114	0.085	0.089
Medical	0.013	**0.010**	0.012	0.011	0.126	**0.010**	0.025	0.014
PlantPse	0.088	0.093	0.089	0.104	0.138	0.132	0.089	0.102
VirusGo	0.041	0.041	0.042	0.038	0.126	0.038	**0.035**	0.143
Yeast	**0.201**	0.204	0.208	0.243	0.254	0.244	0.214	0.225

La Table 3.9 présente les performances prédictives des algorithmes ECML en termes d'Accuracy. Pour le dataset *CHD_49,* l'algorithme ECC se distingue avec la meilleure Accuracy (0.535), tandis que MLS a la performance la plus faible (0.452). Sur le dataset *Genbase*, RAKEL et MLS se démarquent (0.240 et 0.243 respectivement), montrant une meilleure précision que les autres algorithmes. Pour le dataset *HumanPse*, MLS obtient la meilleure Accuracy (0.189), indiquant une performance supérieure, tandis qu'EBR a le plus faible score (0.108). Concernant le dataset *Medical*, ECC et RAKEL partagent la meilleure Accuracy (0.768), démontrant leur efficacité, tandis que RF-PCT a une performance nettement inférieure (0.593). Pour le dataset *PlantPse*, ECC et RAKEL sont encore parmi les meilleurs (0.734 et 0.732) algorithmes, alors que RF-PCT affiche la plus faible Accuracy (0.572). Sur le dataset *VirusGo,* RF-PCT se distingue avec une Accuracy exceptionnelle (0.891), tandis que TREMLC a la performance la plus faible (0.734). Enfin, pour le dataset *Yeast*, HOMER montre la meilleure précision (0.564), alors que MLS est le plus faible des algorithmes.

Ces résultats mettent en lumière les variations de performances des algorithmes en fonction des caractéristiques des datasets utilisé, soulignant qu'ECC et RAKEL sont fréquemment parmi les meilleurs algorithmes en termes d'Accuracy, tandis que MLS et RF-PCT présentent des performances plus variables.

Table.3.9 Performance prédictive des EMLC en termes d'Accuracy.

Datasets	EBR	ECC	EPS	RAKEL	HOMER	MLS	RF-PCT	TREMLC
CHD_49	0.508	**0.535**	0.525	0,522	0.490	0.452	0.530	0.518
Genbase	0.172	0.218	0.167	0.240	0.198	**0.243**	0.171	0.194
HumanPse	0.108	0.151	0.160	0.162	0.138	**0.189**	0.113	0.145
Medical	0.756	**0.768**	0.767	**0.768**	0.740	0.749	0.593	0.625
PlantPse	0.720	**0.734**	0.718	0.732	0.681	0.689	0.572	0.631
VirusGo	0.856	0.844	0.867	0.856	0.761	0.856	**0.891**	0.734
Yeast	0.488	0.546	0.491	0.534	**0.564**	0.434	0.529	0.438

Comme la mesure F1-score offre des indications sur la gestion de la tâche de CML par les modèles, en tenant compte des relations entre labels. Les résultats de la Table 3.10 montrent que les datasets ayant une faible corrélation entre labels tels que *Medical (rDep = 0.039)*, *Genbase* (rDep = 0.157) et *CHD_49* (rDep = 0.267), les algorithmes ECC et MLS sont les plus performants. Pour les datasets modérément corrélés comme *VirusGo* (rDep = 0.400), *HumanPse* (rDep = 0.418) et *PlantPse* (rDep = 0.318), les algorithmes RAKEL et RF-PCT se démarquent. En revanche, le dataset fortement corrélés tel que *Yeast* (rDep = 0.670), HOMER est le plus efficace des algorithmes.

En outre, pour les datasets à haute dimensionnalité tels que *HumanPse* et *Medical*, qui comportent un grand nombre d'instances, de caractéristiques et de labels, EPS se distingue par des résultats de F1_score comparativement plus élevés que ceux des autres algorithmes. Cela s'explique par le fait que l'algorithme EPS est capable de gérer efficacement la complexité de la haute dimensionnalité des datasets en utilisant un processus d'élagage pour supprimer les échantillons dont les ensembles de labels sont peu fréquents.

Table.3.10 Performance prédictive des EMLC en termes de F1-score.

Datasets	EBR	ECC	EPS	RAKEL	HOMER	MLS	RF-PCT	TREMLC
CHD_49	0.625	0.640	0.640	0.584	0.611	0.584	0.647	0.635
Genbase	0.177	0.228	0.173	0.259	0.231	**0.273**	0.180	0.207
HumanPse	0.107	0.151	0.204	**0.209**	0.159	0.168	0.109	0.149
Medical	0.780	**0.793**	0.779	0.781	0.759	0.781	0.788	0.659
PlantPse	0.526	0.551	0.518	0.543	0.510	0.517	**0.658**	0.459
VirusGo	0.880	0.876	0.890	0.877	0.934	0.877	**0.935**	0.920
Yeast	0.584	0.667	0.584	0.658	**0.684**	0.577	0.611	0.609

Les mesures Micro-F1 et Macro-F1 sont souvent utilisées pour évaluer les performances des algorithmes sur des datasets équilibrés et déséquilibrées. Macro-F1 est particulièrement utile pour obtenir des performances équilibrées pour tous les labels en leurs accordant la même importance. D'autre part, Micro-F1 se concentre sur la performance globale basée sur le nombre total d'instances correctement classées. Les résultats présentés dans les Tables 3.11 et 3.12 montrent que les performances des différents algorithmes dépendent du degré de déséquilibre des datasets testés. Pour les datasets présentant un faible degré de déséquilibre, tels que *VirusGo* (avgIR = 4.041) et *CHD_49* (avgIR = 5.766), l'algorithme RF-PCT est le plus performant.

Pour les datasets présentant un degré de déséquilibre moyen, tels que *PlantPse* (avgIR = 6.690) et *Yeast* (avgIR = 7.197), les méthodes MLS et HOMER sont les algorithmes les plus performants. Enfin, pour les datasets possédant un degré élevé de déséquilibre, tels que *HumanPse* (avgIR = 15.289), *Genbase* (avgIR = 37.315) et *Medical* (avgIR = 89.501), les algorithmes ECC et RAKEL sont les plus performants.

Table.3.11 Performance prédictive des EMLC en termes de Micro-F1.

Datasets	EBR	ECC	EPS	RAKEL	HOMER	MLS	RF-PCT	TREMLC
CHD_49	0.645	0.658	0.654	0.610	0.635	0.610	**0.667**	0.656
Genbase	0.979	0.979	0.968	**0.980**	0.978	0.976	0.010	0.893
HumanPse	0.237	0.283	0.231	**0.307**	0.261	0.291	0.239	0.257
Medical	0.805	**0.810**	0.775	0.808	0.768	0.808	0.185	0.719
PlantPse	0.166	0.210	0.153	0.223	0.208	**0.246**	0.166	0.205
VirusGo	0.890	0.890	0.881	0.897	**0.912**	0.897	**0.912**	0.901
Yeast	0.617	0.628	0.616	0.572	**0.664**	0.572	0.608	0.601

Table.3.12 Performance prédictive des EMLC en termes de Macro-F1.

Datasets	EBR	ECC	EPS	RAKEL	HOMER	MLS	RF-PCT	TREMLC
CHD_49	0.488	0.504	0.499	0.460	0.482	0.460	**0.511**	0.494
Genbase	0.730	0.735	0.668	**0.739**	0.736	0.736	0.001	0.611
HumanPse	0.096	0.113	0.085	**0.155**	0.134	**0**.138	0.078	0.117
Medical	0.643	**0.659**	0.606	**0.659**	0.333	0.659	0.026	0.274
PlantPse	0.079	0.095	0.063	0.115	0.141	**0.158**	0.057	0.105
VirusGo	0.785	0.822	0.833	0.784	0.766	0.784	**0.834**	0.633
Yeast	0.385	0.398	0.374	0.383	**0.401**	0.384	0.393	0.385

A travers ces expériences nous pouvons conclure que le choix d'une méthode ECML appropriée, adaptée aux données Bioinformatique spécifiques, dépend de plusieurs facteurs clés tels que, la nature des DML testés (les dépendances entre labels, le déséquilibre des classes et la dimensionnalité de l'espace de sortie), les caractéristiques de la méthode utilisée, ainsi que les exigences spécifiques du domaine traité. L'analyse des résultats obtenus montre clairement qu'aucune méthode ensembliste n'a pu démontrer sa supériorité sur les autres pour relever efficacement la majorité des défis posés par le domaine. Chaque méthode est capable de résoudre un problème spécifique, tel que les méthodes EPS et RAKEL qui abordent le problème du déséquilibre des classes de manière différente.

EPS gère ce problème en réintroduisant les échantillons élagués dans le processus de prédiction, tandis que RAKEL réduit l'impact des labels rares en les incluant dans des sous-ensembles spécifiques de labels plus fréquents, assurant ainsi une représentation plus équilibrée. ECC excelle dans la gestion des dépendances entre labels en utilisant une séquence ordonnée de prédictions, permettant ainsi de capturer les interactions complexes entre les labels. Enfin, la hiérarchie dans RF-PCT permet de structurer et de réduire l'espace de sortie en regroupant les labels similaires dans des sous-ensembles plus petits et en sélectionnant les attributs de manière efficace.

Il est donc recommandé de développer une approche généralisable qui puisse répondre aux besoins du domaine, tout en prenant en compte les facteurs mentionnés ci-dessus.

3.7 Conclusion

Ce chapitre constitue une exploration approfondie de l'approche ECML. Nous avons commencé par fournir une vue d'ensemble des méthodes les plus connues de cette approche Ensuite, une analyse comparative détaillée de ces différentes méthodes est présentée, permettant ainsi d'évaluer leurs forces et leurs faiblesses respectives. Par la suite, nous avons exploré les multiples techniques d'agrégation utilisées pour construire ces méthodes. Enfin, nous avons analysé l'impact des méthodes ECML en Bioinformatique afin de déterminer la meilleure méthode capable de répondre efficacement aux besoins et aux exigences de ce domaine.

L'analyse des résultats obtenus a confirmé que la sélection d'une méthode ECML appropriée dépend de plusieurs facteurs, à savoir : les exigences spécifiques du domaine traité, les caractéristiques des DML testées, la nature des classifieurs de base entrainés ainsi que les spécificités de la technique d'agrégation employée pour concevoir l'ensemble.

Il serait donc intéressant de développer un système idéal et généralisable capable de répondre aux exigences du domaine en considérant les facteurs mentionnés précédemment.

4

Optimisation des Performances de la Classification Multi-Label par un Méta-Modèle

Sommaire

4.1 Introduction

Pour résoudre les problèmes inhérents à CML, les limitations des techniques de combinaison d'ensemble, la nature des classifieurs entrainés ainsi que les exigences spécifiques du domaine traité, nous avons développé un méta-modèle capable de résoudre ces divers défis tout en améliorant la performance prédictive et la généralisation du modèle global. Notre approche, *ConfBoost*, est une nouvelle méthode ECML, basée sur un paradigme de Stacking pondéré intégrant la confiance des labels et des seuils ajustés.

D'abord, nous avons soigneusement sélectionné quatre méthodes ECML hétérogènes et complémentaires : ECC, EPS, RAKEL et RF-PCT, en tenant compte de trois critères principaux : (i) L'hétérogénéité des techniques sous-jacentes des classifieurs individuels, étant donné que ECC repose sur une technique de binarization, EPS et RAKEL utilisent des combinaisons de labels, tandis que RF-PCT suit une stratégie hiérarchique ; (ii) Le niveau de diversité utilisé pour générer chaque méthode ensembliste, déterminé par le type d'algorithme d'apprentissage ou par la sélection aléatoire de différents espaces d'échantillonnage (Labels, caractéristique, instances), comme détaillé dans la Table 3.3 du Chapitre 3, Section 3,4 ; (iii) La complémentarité des méthodes ensemble, dont chacune traite un problème spécifique de MLC, comme expliqué dans la Table 3.4 du Chapitre 3, Section 3,4.

Ensuite, en amalgamant les techniques de Stacking et de pondération, notre approche génère des prédictions plus pertinentes et améliore la précision au niveau de base, atténuant ainsi l'impact des labels non pertinents lors du processus de Stacking. De plus, l'utilisation de seuils ajustés permet au modèle de générer des prédictions adaptées à chaque label, facilitant le traitement des distributions déséquilibrées des labels en accordant plus de sensibilité aux classes rares sans sacrifier la précision sur les classes plus courantes.

Dans ce chapitre, nous détaillons l'approche proposée, *ConfBoost*, en expliquant ses différentes étapes de conception et les techniques déployées. Enfin, nous analysons en profondeur les résultats expérimentaux issus des scénarios menés sur des ensembles de DML de référence, mettant en évidence les performances et les avantages de notre approche par rapport aux objectifs définis.

4.2 Modèle ConfBoost

ConfBoost repose sur une architecture de Stacking Pondéré qui intègre la confidence des labels associée à des seuils ajustés. Cette approche construit un méta-modèle en utilisant des CEMLs de manière synergique, chacun se concentre sur la prédiction d'un sous-ensemble spécifique de labels. Comme illustré dans la Figure 4.1, l'algorithme *ConfBoost* comprend deux modules principaux : l'Apprentissage et la Classification.

Durant la phase d'Apprentissage, une étape initiale, appelée **"*Générateur d'un Pool de Classifieurs Ensemble*"**, est réalisée pour créer des sous-modèles, chacun se concentre sur la prédiction d'un sous-ensemble spécifique de labels. Ensuite, la base de test est divisée en deux segments : *TestBV* et *TestAG*. ***TestBV*** est utilisé pour générer les vecteurs de confiance des labels (BVCL), tandis que ***TestAG*** est considérée comme une base de test indépendante servant à évaluer les performances du modèle final. Après avoir ajusté les seuils et pondéré les prédictions en fonction des vecteurs de confiance générés par TestBV, la base TestAG permet de tester la capacité du modèle global à faire des prédictions multi-label de manière cohérente. Dans l'étape suivante, le module **"*Générateur de Vecteurs de Confidence par Label*"** construit la base des vecteurs de confidence des labels *(BVCL*). Chaque vecteur contient les scores de confidences des différents sous-modèles associés à un label spécifique. Ces vecteurs sont ensuite utilisés dans la phase de Classification pour pondérer les prédictions des sous-modèles.

Dans la phase de Classification, les prédictions générées par la première étape sont combinées à l'aide du module ***"Classification Multi-Label"***, en utilisant BVCL pour prédire le sous-ensemble final de labels d'un nouvel échantillon. A ce niveau, trois stratégies de combinaison différentes sont utilisées : le *Vote Majoritaire (VM)*, le *Vote Pondéré* (VP) et le *Stacking Augmenté (SA)*, afin de comparer leur efficacité à celle de l'approche proposée, ***ConfBoost***, qui repose sur une stratégie de *Stacking Pondéré Augmenté (SPA)*. Pour les approches pondérées, le mécanisme de pondération est adapté aux labels, en étant calculé en fonction de leur scores de confidence.

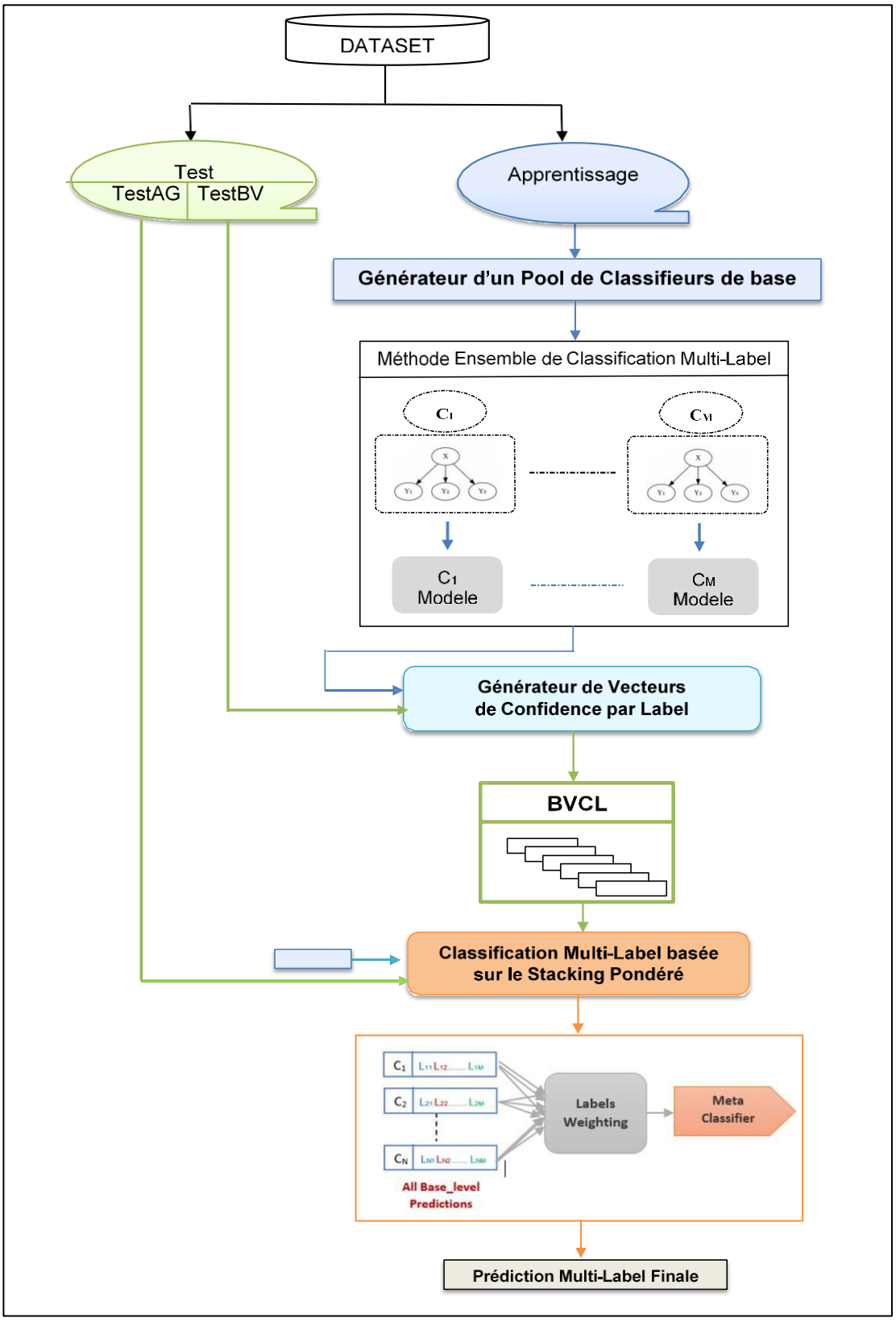

Figure 4.1 Architecture du méta-modèle ConfBoost [134]

4.2.1 Phase d'Apprentissage

4.2.1.1 Générateur d'un Pool de Classifieurs Individuels

Pour la phase initiale de l'apprentissage, un pool initial *P* est créé, contenant *N* classifieurs individuels C_1, C_2,.., C_N, chacun d'eux est conçu pour prédire un sous-ensemble spécifique de labels. La création de ce pool vise à générer des CEML de base, diversifiés capables de traiter efficacement différentes aspects de données et d'améliorer la précision globale des prédictions. De plus, la sortie de pool sert de base pour les étapes d'apprentissage ultérieures, telles que la génération du BVCL et à l'étape d'agrégation pour déterminer la prédiction multi-label finale.

4.2.1.2 Générateur de Vecteurs de Confiance par Label

Après la création du pool initial *P*, le processus d'apprentissage passe à la génération de la Base des Vecteurs de Confidences de Labels (*BVCL)*. Les sous modèles de *P* utilisent les données issues de la base TestBV pour générer le vecteur $\overrightarrow{Vi}$ des scores de confiance pour chaque label L_i, tel que défini par l'équation (4.1).

$$\overrightarrow{Vi} = [Conf(L_i, C_1), Conf(L_i, C_2), \dots, Conf(L_i, C_N)] \quad (4.1)$$

Chaque élément du vecteur $\overrightarrow{Vi}$, représente le score de confidence associée au label L_i fourni par le sous-modèle C_j du pool *P*. Ces scores expriment la probabilité que le *j-ème* sous-modèle prédise correctement le *i-ème* label pour une instance donnée x_i. Ces scores sont calculés à l'aide de la fonction *Softmax*, qui normalise les sorties brutes des sous-modèle (y_i) pour produire des probabilités, comme défini par l'équation (4.2) :

$$Conf(L_i, C_j) = \frac{\exp(y_i)}{\sum_{j=1}^{q} \exp(y_j)}. \quad (4.2)$$

4.2.2 Phase de Classification

Notre approche, *ConfBoost*, qui est un méta-modèle basé sur un schéma de Stacking pondéré, combine les CEMLs générés par le pool initial *P*.

Ainsi, chaque sous-modèle est entraîné en utilisant l'espace de caractéristiques original pour prédire son sous-ensemble de labels respectif P_i. Les sous-ensembles de labels prédits P_1, P_2,..., P_N générés par tous les sous-modèles sont combinés pour générer le sous-ensemble final de labels d'une nouvelle instance donnée x_i. A ce niveau, BVCL et TestAG sont utilisés pour prédire et valider le sous-ensemble final de labels d'un nouvel échantillon.

Dans notre expérimentation, différentes stratégies de combinaison telles que *VM*, *VP*, *SA* et *SPA* sont utilisées pour évaluer leurs performances et identifier la meilleure stratégie, qui constitue l'approche *ConfBoost* proposée.

4.2.2.1 Agrégation basée sur le Vote Majoritaire (VM)

Dans le processus d'agrégation qui repose sur une approche VM, comme illustré dans la Figure.4.2, chaque sous-modèle *Cj* du pool initial *P* peut générer un sous-ensemble de labels prédits *Pij* pour une nouvelle instance x_i. Pour chaque sous ensemble prédit *Pij* nous comptant le nombre d'occurrences de chaque label spécifique *Li* selon l'équation (4.3). *m* représente le nombre total des sous-modèles et (**1** [$Li \in Pij(cj(xi))$]) indique une fonction indicatrice, qui vaut **1** si L_i est présent dans le sous ensemble prédit par le *j-ème* sous-modèle pour x_i , et 0 sinon.

$$Occ(L_i) = \sum_{j=1}^{m} 1\left[L_i \in P_{ij}\left(C_j(x_i)\right)\right] \qquad (4.3)$$

Le sous-ensemble final des labels prédits (xi) pour une nouvelle instance x_i est formé en sélectionnant les labels dont le nombre d'occurrences dépasse le seuil d'acceptation Ɛ, tel que défini par l'équation (4.4). Ce seuil d'acceptation est appliqué pour déterminer l'inclusion ou l'exclusion du label *Li* dans la prédiction finale $S(x_i)$. Pour cela, si le nombre d'occurrences d'un label *Li* dépasse ou égale à ε, alors ce label est retenu dans la prédiction finale de xi, sinon il est exclu.

$$S(x_i) = \bigcup_{i=1}^{q}\{L_i | \, Occ\,(L_i) \geq Ɛ\} \qquad (4.4)$$

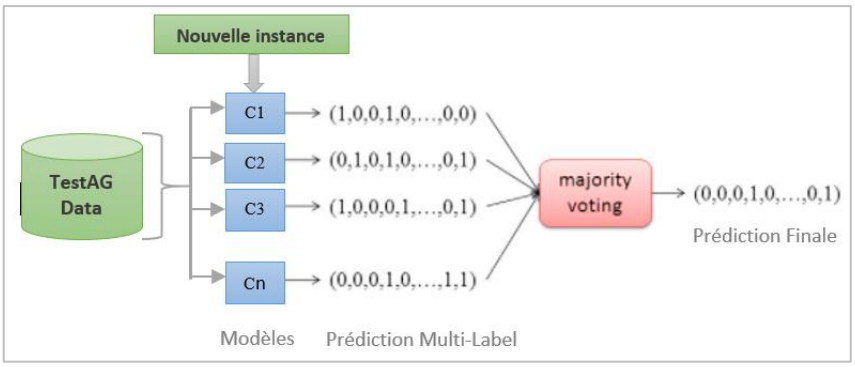

Figure 4.2 Agrégation basée sur VM [134]

4.2.2.2 Agrégation basée sur le Vote Pondéré (VP)

Dans le schéma VP, chaque sous-modèle C_j produit une prédiction multi-label Pij pour une instance donnée x_i. Chaque label L_i prédit par C_j se voit attribuer un score de confiance *Conf (Li, Cj)*, extrait de BVCL. Ce score reflète la certitude du classifieur C_j concernant la prédiction de ce label. Le poids d'un label L_i est ensuite calculé en additionnant les contributions des différents sous-modèles pour ce label, pondérées par leur scores de confiance associés *Conf (Li, Cj)*, conformément à l'équation (4.5).

$$W(L^*) = \sum_{i=1}^{q} \sum_{j=1}^{m} C_j(L_i) \text{ X } Conf(L_i, C_j) \qquad (4.5)$$

Où :

- q représente le nombre total de labels et m le nombre de classifieurs, $Cj(Li)$ vaut 1 si le classifieur Cj a prédit le label Li, et 0 sinon,
- $Conf\ (L_i, C_j)$ représente le score de confidence du classifieur C_j dans la prédiction du label L_i.

Pour déterminer l'inclure ou l'exclure d'un label particulier L_i dans le sous-ensemble final de labels, une fonction de seuillage ajustée *g(s)* est appliquée, conduisant à une bipartition des labels. Cette fonction compare le poids accumulé du label $W(L^*)$ à la moyenne des scores de confiance des m classifieurs pour ce label, conformément à l'équation (4.6).

$$g(s) = \begin{cases} 1, & W(L^*) \geq \frac{1}{m} \sum_{\substack{1 \leq i \leq q \\ 1<j<m}} Conf(L_i, C_j) \\ 0, & sinon \end{cases} \qquad (4.6)$$

Par conséquent, si le poids du label *Li* est supérieur ou égal à cette moyenne, alors la fonction de seuillage *g(s)* renvoie 1, indiquant ainsi l'inclusion du label *Li* dans la prédiction finale. Sinon, le label *Li* sera exclu. Le sous-ensemble final des labels prédits $S(x_i)$ pour l'instance x_i, est alors formé en rassemblant tous les labels *Li* pour lesquels $g(L_i)=1$, conformément à l'équation (4.7). Cette approche d'agrégation, telle qu'illustrée par la Figure 4.3, peut être implémenté par l'Algorithme 4.1 [134].

$$S(x_i) = \bigcup_{i=1}^{q} \{L_i | \, g\,(L_i) = 1\} \tag{4.7}$$

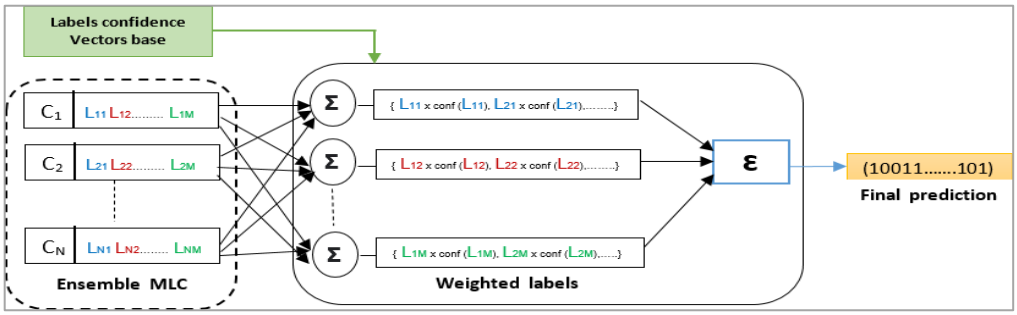

Figure 4.3 Agrégation basée sur VP [134]

Algorithme 4.1 : Agrégation basée sur VP

1 : Entrée :
Ensemble de *m* classifieurs EML $\{C_1, C_2, ..., Cm\}$;
Ensemble de *q* labels $\{L_1, L_2, ..., L_q\}$;
Conf (Li, Cj) score de confiance de *Cj* pour le label *Li ;*

2 : initialiser la matrice des poids $W(Li^*)$ à zéro ;
3 : **Pour** chaque *Li* allant de 1 à *q* **faire**
4 : **Pour** chaque *Cj* allant de 1 à *m* **faire**
5 : **Si** *Cj (Li)*=1 **alors** $W(Li^*)$ += *Li(Cj)*× *Conf (Li, Cj);*
6 : **Fin pour**
7 : initialiser la liste *S (xi)* à vide ;
8 : **Pour** chaque *Li* allant de 1 à *q* **faire**
9 : *Avg_Conf(Li, Cj)* ⟵ *1/m *Sum (Conf (Li,Cj)) ;*
10 : **Si** $W(Li^*)$ > *Avg_Conf(Li,Cj)* **alors**
11 : ajouter *Li* à *S (xi);*
12: **Fin pour**
13: Retourner *S (xi).*

4.2.2.3 Agrégation basée sur le Stacking Augmenté (SA)

Dans notre expérimentation, le niveau de base est constitué de classifieurs EML individuels C_1, C_2,.., C_N, où chacun d'eux génère un vecteur de prédiction $f_i^1(x)$ pour une instance donnée *xi*. Ces prédictions sont encapsulées dans la fonction de prédiction globale $f^1(x_i)$ au niveau de base, représentée par l'équation (4.8).

$$f^1(x_i) = \left(f_1^1(x_i), f_2^1(x_i), .., f_m^1(x_i)\right) \quad (4.8)$$

Où :

- $f_i^1(x)$ est un vecteur de prédiction multi-label générée par l'i-ème classifieur individuel pour l'instance *xi*.
- m indique le nombre total des classifieurs individuels utilisés au niveau de base.

Les prédictions multi-label générées par le niveau de base sont combinées dans un vecteur global qui est ensuite fusionné avec le vecteur de caractéristiques original *X* de *xi* pour créer un espace de caractéristiques étendu ($X \times f^1(x_i)$). Cet espace augmenté (étendu) enrichi par les prédictions des classifieurs du niveau de base, constitue l'entrée du classifieur de méta-niveau. La fonction d'augmentation représentant cet espace est définie par l'équation (4.9), où f^2 est le méta_classifieur et *y'* représente sa sortie.

$$f^2(X \times f^1(x_i)) \rightarrow y' \quad (4.9)$$

Le méta_classifieur est ensuite entraîné sur l'espace de caractéristiques étendu pour apprendre les relations entre labels. Il utilise à la fois les informations provenant des prédictions de base et celles du vecteur d'origine pour capturer les dépendances complexes entre les labels. Pour une nouvelle instance *xi*, le méta_classifieur f^2 génère la prédiction *Pij* finale en appliquant la fonction (4.10). y est la vérité terrain utilisée pour évaluer le méta_classifieur f^2.

$$f(x_i) = f^2(x_i \times f^1(x_i)) \rightarrow y \quad (4.10)$$

Il est important de noter que ce schéma d'agrégation, tel qu'illustré par la Figure.4.4, est souvent privilégié dans les tâches de CML. Il permet d'exploiter une gamme plus large de caractéristiques potentiellement informatives pour la prédiction finale en capturant les dépendances complexes entre les labels. Ainsi, l'algorithme 4.2 permet de bien détailler le déroulement des différentes étapes de ce schéma d'agrégation.

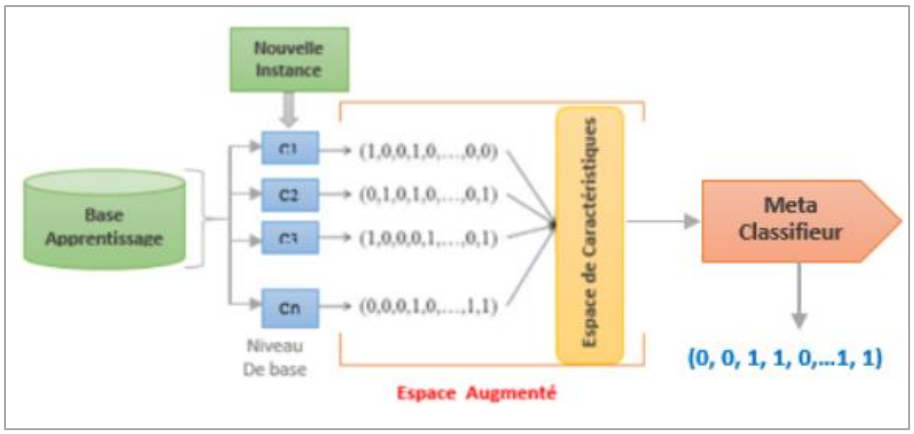

Figure 4.4 Agrégation basée sur SA [134]

Algorithme 4.2 : Agrégation basée sur SA

1 : Entrée :
Ensemble de *m* classifieurs EML {C_1, C_2,.., C_m};
Ensemble de *q* labels {L_1, L_2, ..., L_q} ;
Ensemble d'instances *(X) ;*

2 : initialiser *Base_Pred (X, m) à zéro,*
3 : **Pour** chaque *xi* appartenant à *X* **faire**
4 : **Pour** chaque *Cj* allant de 1 à *m* **faire**
5 : stocker les prédictions de *Cj dans Base_Pred (xi, Cj) ;*
6 : **Fin pour**

7 : **Pour** chaque *xi* appartenant à *X* **faire**
8 **:** **Pour** chaque *Cj* allant de 1 à *m* **faire**
9 : construire *Extend_feature* (*xi*);
10: *Extend_feature (xi)* ⟵ *xi* ∪ *Base_Pred (xi, Cj);*
11: **Fin pour**
12: **Fin pour**

13: initialiser la matrice *Pred_Final (xi)* à vide ;
14: **Pour** chaque *xi* appartenant à *X* faire **faire**
15: appliquer le méta_classifieur sur *Extend_feature (xi)* ;
16: **Fin pour ;**
17: Retourner *Pred_Final (xi).*

4.1.1.1 Agrégation basée sur le Stacking Pondéré_Augmenté (SPA)

Basée sur l'intégration d'un mécanisme de pondération des labels et d'une fonction de seuillage ajustée dans le processus du SA, l'approche ***ConfBoost*** est développée comme suit :

Les classifieurs ensembliste du niveau de base C_1, C_2,..,C_m sont entraînés pour générer des prédictions *Pij* pour une instance donnée *xi*. Chaque label *Li* de *Pij* est accompagné de son score de confiance *Conf (Li, Cj)*, tiré de la base BVCL. Ces scores sont ensuite utilisés pour pondérer les labels de chaque *Pij* et générer un vecteur de labels pondérés $\overrightarrow{V}ji$ pour chaque *Pij* associé, comme défini par l'équation (4.11), où *WLji* représente la pondération de l'i-ème label pour le j-ème sous modèle.

$$\overrightarrow{V}ji = [WL_{j1}, WL_{j2}, \dots, WL_{jk}] \quad (4.11)$$

Une fonction de seuillage ajustée *g(s)*, définie par la fonction (4.6), est appliquée à chaque vecteur pondéré $\vec{V}_{ji}$ pour sélectionner uniquement les labels pertinents. Les vecteurs filtrés ainsi obtenus sont ensuite combinés pour former un vecteur global $\overrightarrow{V}$, tel que défini par l'équation (4.12).

$$\overrightarrow{V} = [\vec{V}_{j1}, \vec{V}_{j2}, \dots, \vec{V}_{jn}] \quad (4.12)$$

Ce vecteur $\vec{V}$ est alors combiné avec l'espace de caractéristiques original *X*, formant ainsi un espace étendu $(X \times \vec{V})$. Sur cet espace étendu, le méta_classifieur f^2 est entraîné pour capturer les relations complexes entre les labels tout en utilisant les caractéristiques originales et les prédictions pondérées du niveau de base.

Pour une nouvelle instance *xi*, une fois l'espace de caractéristiques étendu généré, le méta_classifieur f^2 fourni la prédiction finale des labels conformément à l'équation (4.10). Cette prédiction finale représente le sous-ensemble des labels optimisés, tenant compte à la fois des informations des sous-modèles de base et des relations entre labels capturées par le méta_classifieur.

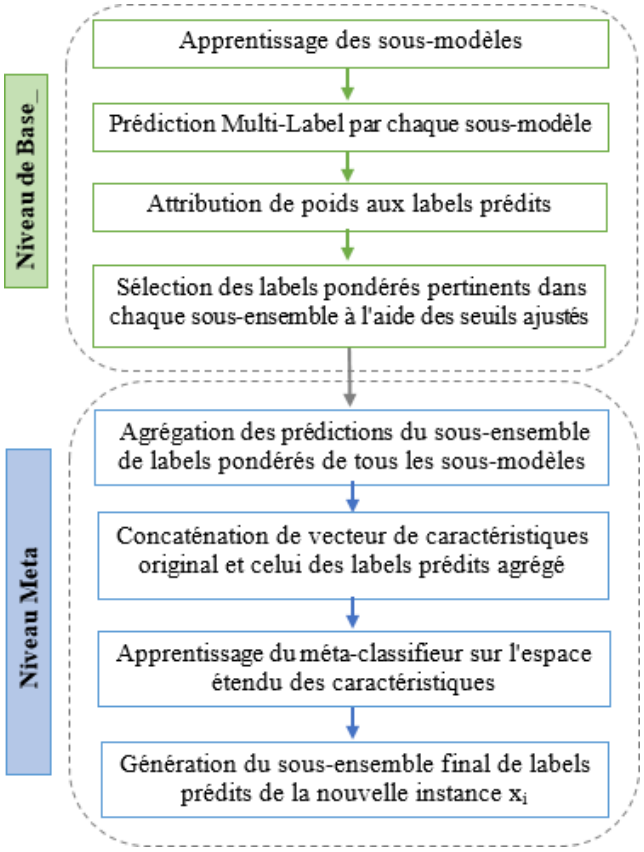

Figure.4.5 Agrégation basée sur le Stacking Pondéré: ConfBoost [1]

En se référant à la figure 4.5 ci-dessus, l'approche proposée fonctionne à travers deux niveaux distincts : le niveau de base et le niveau-méta, chacun englobant des tâches spécifiques destinées à faciliter le processus de CML.

Au niveau de base, l'étape initiale consiste à former les sous-modèles, formés individuellement à l'aide de l'espace de caractéristiques original. Une fois formé, chaque sous-modèle prédit un sous-ensemble de labels pour une instance donnée. Par la suite, des niveaux de confiance sont attribués aux labels prédits sur la base de leur confidence. En outre, une fonction de seuillage ajustée est appliquée pour déterminer les labels pondérés pertinents dans chaque sous-ensemble de label prédits.

Au méta-niveau, le processus d'agrégation des prédictions multi-label pondérées de tous les sous-modèles est lancé. Ces prédictions agrégées, qui représentent les labels potentielles pour la classification d'une instance donnée, sont ensuite combinées avec le vecteur de caractéristiques original. Cette fusion forme un espace de caractéristiques étendu, comprenant à la fois les caractéristiques originales et les labels prédits du niveau précédent.

A ce niveau, un méta_classifieur est formé sur cet espace de caractéristiques étendu, en tirant parti des informations combinées pour générer des prédictions plus précises. Enfin, le méta_classifieur entraîné est déployé pour produire le sous-ensemble final de labels prédits pour une nouvelle instance, complétant ainsi le processus de CML.

4.2 Etude Expérimentale

Dans cette section, nous présentons les jeux de données Multi-label utilisés, les mesures d'évaluation employées pour mener les études empiriques, ainsi que les paramètres expérimentaux des classifieurs utilisés. Les résultats des différentes expériences sont illustrés par des figures et des tables, puis analysés et discussion en détail pour tirer des conclusions significatives et éclairer les implications de nos travaux.

4.2.1 Ensemble de données Multi-Label

Pour évaluer notre étude expérimentale, nous avons utilisé quatre jeux de données Multi-Label (datasets) hétérogènes provenant de trois domaines différents répertoriés dans le référentiel MULAN[1], tels que l'audio, la biologie, et la catégorisation de texte.

- **Audio** : le dataset *Birds* [167] est une collection d'espèces d'oiseaux annotée par 19 extraits de chant.
- **Biologie**: les datasets de ce domaine représentent généralement des protéines en tant que variables descriptives, ainsi que des cibles pour la prédiction de la fonction génique ou la localisation subcellulaire. Par exemple, le dataset *Genbase* [58] est composé d'instances étiquetées avec 27 fonctions de protéines, tandis que le dataset *Yeast* [56] comporte 14 fonctions biologiques attribuées à chaque instance génique.
- **Catégorisation de texte**: le dataset *Medical* [157] est une collection de rapports cliniques en texte libre associés à 45 codes de maladie par instance. Il offre une vue globale de la santé du patient basée sur des mesures vitales.

Selon la littérature, les ensembles de DML sont conservés dans leurs formats originaux et sont préalablement divisés en ensembles d'apprentissage et de test. L'ensemble d'apprentissage représente les deux tiers du jeu de données globale, tandis que le tiers restant constitue l'ensemble de test.

[1] https://www.uco.es/kdis/mllresources/

De plus, dans notre étude, nous avons subdivisé l'ensemble de test en deux sous-ensembles égaux appelés TestBV et TestAG. Une vue d'ensemble des datasets expérimentaux ainsi que leurs caractéristiques respectives est présentée dans la Table 4.1.

Table 4.1. Caractéristiques des datasets Multi-Label expérimentaux.

Datasets	Domain	m	q	d	Card	Dens	avgIR	rDep
Birds	Audio	645	19	260	1.014	0.053	5 .407	0 .123
Genbase	Biology	662	27	1 186	1.252	0.046	37.315	0.157
Yeast	Biology	2 417	14	103	4.237	0.303	7.197	0.670
Medical	Text	978	45	1 449	1.245	0.028	89.501	0.039

m : nombre d'instances, **Card** : nombre de labels par instance,
q : nombre de labels, **Dens** : ratio entre card et m,
rDep : dépendance entre paire de labels,
d : nombre d'attribues, **avgIR** : dégrée moyen du déséquilibre de tous les labels.

4.2.2 Paramètres Expérimentaux

4.2.2.1 Mesures d'évaluation

L'évaluation des performances de nos expériences s'est concentrée sur huit différentes métriques, comprenant trois mesures basées sur les exemples (Hloss, Accuracy et F1_score), ainsi que quatre métriques basées sur les labels (Micro-F1, Macro-F1, Macro_precision et Macro_recall). La courbe AUROC a également été utilisée pour évaluer les performances des modèles. L'ensemble de ces métriques est défini au niveau de la section 2.7.

4.2.2.2 Configuration des Méthodes

Les méthodes composant l'approche *ConfBoost* ont été entraînées avec les paramètres par défaut comme suit :

- ECC est composé de 10 classifieurs CC, l'ordre des chaînes de chaque CC étant généré aléatoirement.
- EPS est composée de 10 classifieurs PS, la valeur de p étant fixée à 1 (p=1), et chaque classifieur utilise une sélection aléatoire d'instances échantillonnées.
- RF-PCT est composés de 10 arbres, et chaque classifieur utilise une double sélection aléatoire d'instances et de caractéristiques échantillonnées.
- RAkEL est composé de 2q classifieurs LP, chacun d'eux utilise un petit sous-ensemble aléatoire de k-label, et la taille des k-ensembles label est fixée à 3 (k=3).

4.3 Résultats Expérimentaux

Pour évaluer le potentiel de notre approche, des expériences ont été menées dans trois scénarios distincts: dans le premier, nous avons évalué les performances des méthodes EML, comprenant ECC, EPS, RF-PCT et RAKEL. Dans le second scénario, nous avons combiné les classifieurs EML en utilisant diverses techniques de combinaison, notamment le Vote Majoritaire, le Vote Pondéré, le Stacking et le Stacking Pondéré, puis évalué les performances de chaque méthode combinée. Enfin, nous avons évalué les performances de notre approche, *ConfBoost*, par rapport aux méthodes connexes en utilisant les deux datasets *Yeast* et *Medical* possédant des caractéristiques spécifiques. Il convient de souligner que les méthodes connexes comparées reposent sur des stratégies différentes de pondérations telles que la pondération des classifieurs et des caractéristiques.

4.3.1 Scénario 1: Comparaison des performances des méthodes ensembles individuelles

Cette expérience permet de comparer les performances des méthodes ECML individuelles sélectionnées en fonction des caractéristiques spécifiques des datasets testés, à savoir : la dépendance des labels, le déséquilibre des classes de données et la dimensionnalité de l'espace de sortie, désignés par les paramètres rDep, avgIR et dim (m*q*d).

Les résultats présentés dans la Table 4.2 montrent que, pour la mesure Hloss, la méthode ECC s'est révélé être la plus performante pour tous les datasets expérimentaux, notamment pour ceux comportant un grand nombre de labels, tels que *Genbase* (27 labels) et *Medical* (45 labels). Le succès d'ECC est également attribué à sa capacité à prédire avec précision un nombre plus élevé de labels par rapport aux autres méthodes. Par ailleurs, EPS a systématiquement surpassé RF-PCT pour tous les datasets testés. Il est évident de noter que tous les classifieurs ensemblistes ont montré des performances relativement réduites lorsqu'ils ont été testés avec le dataset *Yeast*. Cependant, la métrique Hloss entraîne des évaluations différentes en fonction des datasets ayant des longueurs différentes de labels.

Concernant la métrique de l'exactitude (Accuracy), calculée pour chaque instance, elle détermine la proportion entre le nombre de labels correctement prédits et le nombre total de labels. Selon les résultats indiqués dans la Table 4.2, RAKEL a démontré une performance excellence sur la plupart des datasets testés, à l'exception du dataset *Medical* où ECC a surpassé toutes les autres méthodes.

La méthode EPS a également surpassé RF-PCT dans tous les cas. Les scores d'Accuracy les plus élevés ont été atteints par tous les classifieurs avec le dataset *Medical* (45 labels), suivi du dataset *Genbase* (27 labels). En revanche, les datasets *Bird* (19 labels) et *Yeast* (14 labels) ont donné des scores faibles. Cette sensibilité au nombre de labels explique leur impact sur la métrique Accuracy. Ainsi, avec un nombre plus élevé de labels, cette métrique est plus performante, tandis qu'elle se détériore avec un nombre plus faible.

F1-score est une métrique couramment utilisée pour évaluer l'efficacité d'un sous-ensemble de labels prédits, en considérant les dépendances entre labels. Dans des scénarios où il existe une forte dépendance entre labels, comme observé dans le dataset *Yeast* (rDep = 0,670) illustré par la Figure 4.6, la méthode ECC est en moyenne le choix optimal. Cependant, en considérant simultanément tous les labels, ECC permet une exploration complète des dépendances entre labels. À l'inverse, le dataset *Medical*, présentant une faible dépendance entre labels (rDep = 0,039), comme illustré à la Figure 4.7, désigne ECC et RAkEL comme des méthodes performantes. Ces algorithmes gèrent efficacement les dépendances entre labels en les considérants de manière séquentielle, ce qui conduit à des prédictions précises dans des scénarios où les labels sont moins interconnectés.

Macro_precision et Macro_recall, deux mesures essentielles, permettant d'évaluer la performance d'un modèle sur chaque label indépendamment, et de calculer par la suite la moyenne de ces performances sur l'ensemble des labels. Cela donne une vue d'ensemble sur le comportement du modèle sur tous les labels, sans donner plus de poids aux labels plus fréquents. La métrique Macro_precision permet de mesurer la capacité du modèle à ne pas faire de fausses prédictions positives. Par contre la métrique Macro_recall évalue la capacité du modèle à identifier correctement toutes les occurrences d'un label, indépendamment de sa fréquence. Selon les résultats présentés par la Table 4.2, ces deux mesures ont systématiquement positionné RAKEL avec les meilleurs scores dans la plupart des datasets. ECC est apparu comme le deuxième meilleur classificateur, surpassant tous les autres en termes de Macro_recall (64,4 %) et de Macro_precision (64,9 %) pour le dataset *Medical*. En revanche, EPS et RF-PCT ont obtenu les scores les plus faibles pour ces mesures dans tous les datasets testés.

Les mesures F1 (Macro et Micro) sont considérées comme des métriques plus efficaces et plus fiables que l'Accuracy pour évaluer les performances des modèles sur des datasets déséquilibrés selon deux points de vue différents.

Bien que Macro-F1 accorde la même importance à tous les labels indépendamment de leur fréquence, Micro-F1 est influencée par la fréquence d'occurrence de chaque label. Pour la mesure Micro-F1, ECC a affiché d'excellentes performances sur des dataset légèrement déséquilibrés, tels que les datasets *Bird* (avgIR=5.407) et *Yeast* (avgIR=7.197). Cette performance peut être attribuée à la prédiction précise des labels fréquents par ECC par rapport aux autres méthodes. En revanche, pour la mesure Macro_F1, RAKEL a surpassé les autres méthodes dans les datasets fortement déséquilibrés, tels que *Genbase* (avgIR=37.315) et *Medical* (avgIR=89.501). Cela est dû au fait que RAKEL est une méthode robustesse face au déséquilibre des classes. Elle permet de réduire l'impact des labels rares en les intégrant de manière contrôlée dans les sous-ensembles spécifiques.

.

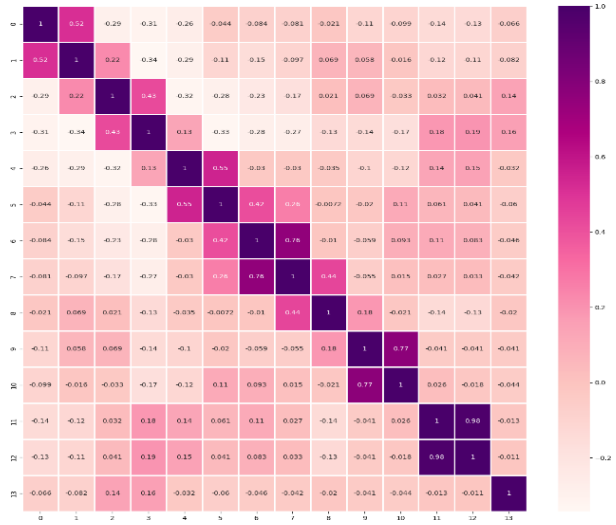

Figure 4..6 Matrice de correlation du dataset Yeast (14 labels).

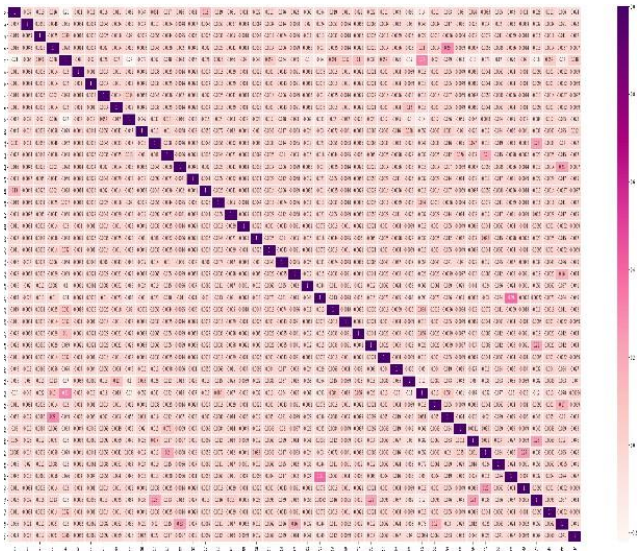

Figure 4.7 Matrice de correlation du dataset Medical (45 labels).

Table 4.2. Comparaison des performances des méthodes Ensemble individuelles en terme de mesures Hloss, Accuracy, F1score, Macro- F1, Micro-F1, Macro_precision, Macro_recall.

Dataset	CEML	Hloss ↓	Accuracy ↑	F1-score ↑	Macro↑ Precision	Macro recall ↑	Micro F1↑	Macro-F1↑
Birds	ECC	**0.041**	0.116	**0.178**	0.432	0.225	**0.443**	**0.242**
	EPS	0.042	0.148	0.123	0.331	0.201	0.416	0.210
	RFPCT	0.043	0.047	0.051	0.400	0.220	0.413	0.233
	RAKEL	0.046	**0.162**	0.157	**0.433**	**0.249**	0.425	0.239
Yeast	ECC	**0.209**	0.094	0.097	0.492	0.387	**0.645**	**0.411**
	EPS	0.210	0.140	**0.156**	0.507	0.358	0.628	0.378
	RFPCT	0.219	0.044	0.052	0.466	0.400	0.608	0.390
	RAKEL	0.223	**0.145**	0.142	**0.539**	**0.406**	0.639	0.409
Genbase	ECC	**0.001**	0.210	**0.259**	0.925	0.926	0.986	0.746
	EPS	0.002	0.164	0.171	0.771	0.757	0.979	0.679
	RFPCT	0.043	0.162	0.178	0.221	0.218	0.005	0.008
	RAKEL	**0.001**	**0.234**	0.225	**0.929**	**0.934**	**0.988**	**0.747**
Medical	ECC	**0.010**	**0.768**	**0.799**	**0.649**	**0.644**	0.815	0.363
	EPS	0.012	0.757	0.781	0.627	0.598	0.786	0.330
	RFPCT	0.025	0.106	0.115	0.381	0.335	0.183	0.029
	RAKEL	0.011	0.761	0.787	0.648	0.642	**0.819**	**0.376**

Comme mentionné précédemment, pour évaluer les performances des méthodes ensemblistes en fonction de la dimensionnalité de l'espace de sortie (dim), nous avons organisé les datasets de la Table 4.3 selon un ordre croissant de dimensionnalité. La Table 4.3 présente les temps d'apprentissage (T_app) et de test (T_test) pour les quatre méthodes analysées sur les datasets testés.

Les résultats obtenus dans la Table 4.3 montrent que la méthode RFPCT se distingue par ses performances supérieures sur tous les datasets, bien qu'il soit reconnu comme l'algorithme le plus efficace pour traiter des espaces de sortie de grande dimension. Ceci est dû à sa structure hiérarchique permettant de réduire l'espace de sortie en regroupant les labels et en sélectionnant les attributs de manière efficace à chaque nœud de l'arbre, tout en tenant compte des interactions entre labels.

Pour les petits datasets (*Bird, Yeast*), la méthode EPS se distingue par sa rapidité, offrant des temps d'exécution acceptables comparés à ceux des méthodes ECC et RAKEL. Ainsi, la méthode RFPCT étant privilégiée pour les grands et complexes datasets, et EPS est bien adaptée pour des petits datasets où la rapidité est cruciale, vu qu'elle simplifie le modèle en éliminant les combinaisons de labels rares, réduisant ainsi la complexité computationnelle et accélérant les temps d'apprentissage et de prédiction.

Table 4.3. Temps d'apprentissage et de test des CEML individuels.

Dataset	dim	T	ECC	EPS	RFPCT	RAKEL
Birds	3 186 300	T_app	5.90E+00	1.63E+00	**1.37E+00**	7.67E+00
		T_test	5.41E+00	1.09E+00	**6.91E-01**	6.92E+00
Yeast	3 485 314	T_app	1.85E+01	7.39E+00	**5.19E+00**	1.99E+01
		T_test	1.78E+01	6.77E+00	**1.01E+00**	1.83E+01
Genbase	21 198 564	T_app	5.05E+00	2.94E+00	**1.61E+00**	6.12E+00
		T_test	4.91E+00	2.31E+00	**1.11E+00**	4.58E+00
Medical	63 770 490	T_app	3.45E+01	7.62E+00	**2.34E+00**	3.32E+01
		T_test	3.34E+01	7.23E+00	**2.91E+00**	3.20E+01

Pour approfondir davantage cette analyse comparative, les performances des modèles individuels (ECC, RAKEL, EPS et RFPCT) ont été évaluées par la courbe AUROC en testant le dataset *Yeast*, comme illustré dans la Figure 4.8. Il est observé que le modèle ECC a produit la plus haute AUROC, atteignant 84 %, suivi de près par RAKEL avec une AUROC légèrement inférieure.

D'autre part, le modèle EPS a obtenu une courbe moyenne plus basse que celles de RAKEL et ECC, mais a tout de même surpassé la courbe de RFPCT, affichant une AUROC moyenne de 41 %. Il est intéressant de noter qu'une analyse par classe révèle que la classe 14 présente l'AUROC la plus élevée, indiquant une excellente capacité de prédiction pour cette classe spécifique. En revanche, la classe 11 a un AUROC relativement plus faible, suggérant des performances de prédiction moins satisfaisantes pour cette classe particulière.

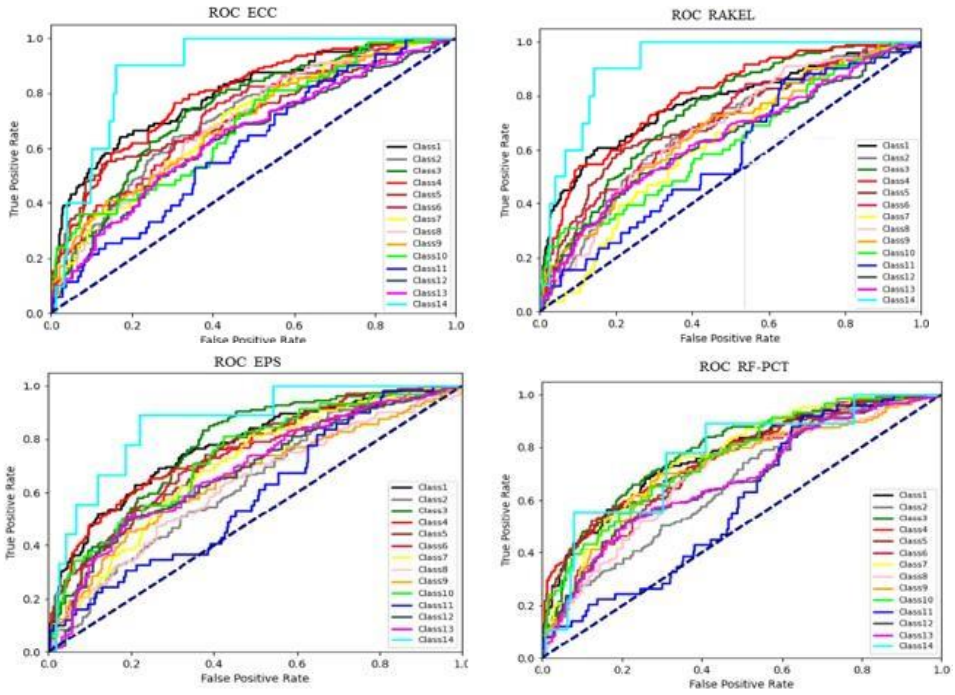

Figure 4.8 Courbes AUROC de quatre méthodes d'ensemble (ECC, EPS, RAKEL et RF-PCT) entraînées sur le dataset Yeast.

L'analyse des résultats obtenus dans le premier scénario confirme que chaque méthode individuelle, bien qu'elle s'inscrive dans une approche d'ensemble, apporte une contribution spécifique en fonction de la complexité du dataset testé et des caractéristiques propres à la méthode utilisée. Cependant, il est important de noter qu'aucune méthode ne se révèle systématiquement supérieure aux autres, ni capable de répondre efficacement à toutes ces contraintes simultanément. Cette constatation met en lumière la complexité et la diversité des problèmes posés par la CML, ainsi que la variété des méthodes employées. Par conséquent, il est nécessaire de développer une approche avancée basée sur une structure de méta-modèle, qui exploite la complémentarité des modèles d'ensemble hétérogènes pour atteindre des performances optimales.

4.3.2 Scénario 2: Comparaison des performances des différentes approches combinées

Le méta-modèle proposé dans ce scénario repose sur la collaboration des méthodes ensemblistes ECC, EPS, RAKEL et RFPCT, intégrant ainsi quatre techniques ensemble différentes. Cette collaboration a pour objectif de tirer pleinement parti des avantages spécifiques de chaque méthode, tout en atténuant les défis posés par la CML. Par exemple, les méthodes EPS et RAKEL abordent le problème du déséquilibre des classes de données de manière distincte. EPS gère ce problème en réintroduisant les échantillons élagués dans le processus de prédiction. En revanche, RAKEL réduit l'impact des labels rares en les incluant dans des sous-ensembles spécifiques de labels plus fréquents, assurant ainsi une représentation plus équilibrée. ECC excelle dans la gestion des dépendances entre labels en utilisant une séquence ordonnée de prédictions, permettant de capturer les interactions complexes entre les labels. Enfin, la hiérarchie dans RF-PCT permet de structurer et de réduire l'espace de sortie en regroupant les labels similaires dans des sous-ensembles plus petits et en sélectionnant les attributs de manière efficace. Cela rend la méthode RF-PCT particulièrement adaptée à la gestion des problèmes de grande dimensionnalité en CML, en tenant compte des interactions entre labels tout en réduisant la complexité du modèle.

Ces classifieurs individuels sont combinés à l'aide de diverses techniques ensemble telles que le Vote Majoritaire, le Vote Pondéré, le Stacking et le Stacking Pondéré. Chaque technique présente ses propres avantages et inconvénients, et leur performance peuvent variés en fonction des caractéristiques spécifiques de chaque dataset utilisé. Comme présenté dans la Table 4.4, la technique de Vote Majoritaire a produit les meilleurs scores avec les datasets *Genbase* et *Medical*. Cependant, malgré ses résultats prometteurs, le dataset *Yeast* a donné les scores les plus faibles en raison de ses caractéristiques spécifiques.

D'autre part, le Vote Pondéré a donné les meilleurs scores pour tous les datasets, en particulier pour *Genbase*, avec des scores de Hloss (0,6 %), Accuracy (92,6 %), Macro_precision (88,7 %) et Macro_recall (85,3 %). Cette approche accorde des poids différents aux prédictions de chaque classifieur en fonction de leur confiance, ce qui peut améliorer la robustesse et la précision de la prédiction finale. Néanmoins, le dataset *Yeast* a présenté des scores relativement bas en termes de HLoss (28,1 %) et d'Accuracy (39,8 %), soulignant les défis spécifiques associés à cet ensemble de données.

Le dataset *Bird* a présenté des résultats relativement faibles avec toutes les techniques d'ensemble, ce qui peut indiquer des caractéristiques particulières de ce dataset rendant la tâche de classification plus complexe. En revanche, le dataset *Medical* a affiché des performances satisfaisantes avec toutes les techniques, ce qui suggère que ces techniques d'ensemble peuvent être efficaces dans un large éventail de contextes.

Les résultats obtenus avec les stratégies de Stacking ont démontré une efficacité remarquable dans la plupart des cas. Bien que les performances puissent être moins satisfaisantes pour le dataset Yeast par rapport aux autres, les stratégies de Stacking surpassent néanmoins les scores obtenus avec les stratégies de Vote, soulignant ainsi la robustesse de l'approche de Stacking. En outre, la technique de Stacking Pondéré, désignée par *ConfBoost*, se distingue par ses performances supérieures par rapport à toutes les autres techniques de combinaison, affichant des résultats prometteurs sur la plupart des datasets testés. En particulier, avec le dataset *Genbase*, *ConfBoost* a surpassé toutes les autres méthodes de combinaison en termes d'Accuracy (98,2 %), de Macro_recall (94,2 %) et de HLoss (0,1 %). Ces scores élevés témoignent de la capacité de *ConfBoost* à fournir des prédictions précises et fiables. L'aspect distinctif de l'approche proposée réside dans son potentiel à améliorer les performances prédictives en introduisant le concept de pondération des labels. Cette intégration permet d'accorder plus d'importance à certains labels lors de la phase de classification, en fonction de leur pertinence et de leur confiance.

Les résultats expérimentaux obtenus ont validé deux observations importantes. Tout d'abord, ils ont mis en évidence l'impact significatif des caractéristiques intrinsèques de chaque dataset expérimenté sur les performances des algorithmes de CML. Des facteurs tels que le nombre de labels, les dépendances entre les labels et le déséquilibre des classes de label sont des éléments clés influençant les performances des modèles. Cette observation souligne l'importance de considérer ces caractéristiques lors du développement et de l'évaluation des méthodes, afin d'obtenir des résultats précis et fiables.

Deuxièmement, les résultats ont également démontré l'efficacité de l'intégration des pondérations aux labels basées sur leurs scores de confiance, couplée à des seuils ajustés. Cette approche permet à *ConfBoost* d'attribuer une priorité différente à chaque label pendant le processus de classification, en atténuant ainsi l'effet des labels non pertinents. Cette particularité permet au modèle de prendre des décisions plus éclairées et de générer des prédictions plus précises.

Table 4.4. Comparaison des performances de différentes approches combinées.

Dataset	Techniques Ensemble	Méthodes Ensemble	Hloss ↓	Accuracy ↑	Macro↑ precision	Macro recall ↑	Micro - F1↑	Macro F1↑
Birds	Voting	Simple Voting	0.061	0.462	0.161	0.395	0.395	0.209
		Weighted Voting	0.054	0.559	0.293	0.403	0.403	0.319
	Stacking	Stacking	0.045	0.571	0.391	0.456	0.456	0.387
		ConfBoost	**0.040**	**0.815**	**0.621**	**0.647**	**0.647**	**0.630**
Yeast	Voting	Simple Voting	0.287	0.316	0.270	0.489	0.489	0.209
		Weighted Voting	0.281	0.398	0.375	0.501	0.501	0.260
	Stacking	Stacking	0.264	0.429	0.381	0.553	0.553	0.366
		ConfBoost	**0.194**	**0.809**	**0.647**	**0.646**	**0.646**	**0.624**
Genbase	Voting	Simple Voting	0.010	0.781	0.721	0.806	0.806	0.655
		Weighted Voting	0.006	0.926	0.887	0.831	0.831	0.701
	Stacking	Stacking	0.003	0.973	0.930	0.882	0.882	0.822
		ConfBoost	**0.001**	**0.982**	**0.952**	**0.991**	**0.991**	**0.843**
Medical	Voting	Simple Voting	0.013	0.603	0.521	0.655	0.655	0.627
		Weighted Voting	0.011	0.691	0.578	0.789	0.789	0.704
	Stacking	Stacking	0.006	0.779	0.649	0.824	0.824	0.745
		ConfBoost	**0.004**	**0.976**	**0.662**	**0.837**	**0.837**	**0.770**

Cette autonomie dans la prise de décision donne à *ConfBoost* un avantage significatif par rapport à d'autres techniques de combinaison, comme en témoigne la Figure 4.9, où l'on peut observer une nette amélioration des performances par rapport aux approches traditionnelles. En conséquence, cette capacité à prioriser les labels en fonction de leur confiance contribue à renforcer la précision et la fiabilité des prédictions de *ConfBoost* dans divers scénarios d'application.

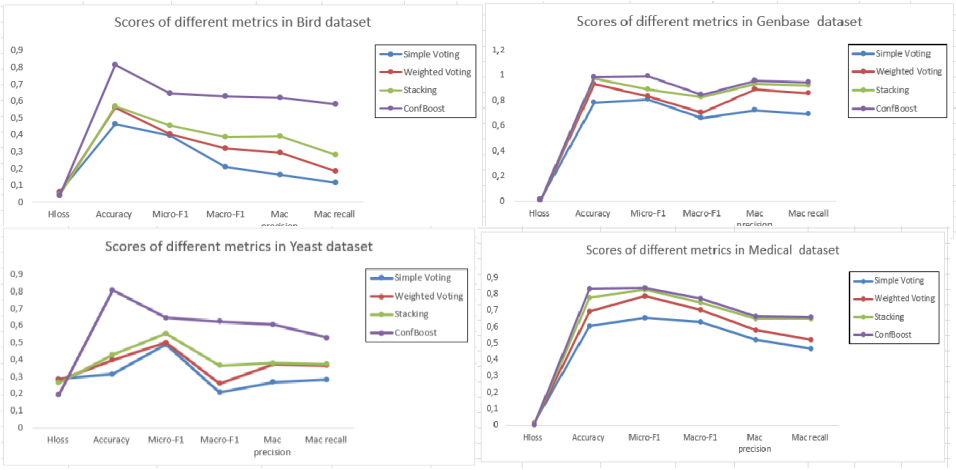

Figure 4.9 Performance prédictive des méthodes combinées en termes des métriques utilisées sur les datasets expérimentés (Bird, Yeast, Genbase, et Medical).

En se basant sur les résultats présentés dans les Tables 4.3 et 4.4, une observation importante émerge : les modèles combinés surpassent systématiquement les modèles individuels dans tous les scénarios évalués. Cette conclusion est renforcée par le constat que les performances obtenues dans le deuxième scénario dépassent même les meilleurs résultats observés dans le premier scénario. Cette tendance est illustrée dans la Figure 4.10, où l'on constate que le méta-modèle *ConfBoost* a atteint une AUROC moyenne nettement supérieure à celle d'ECC, qui est pourtant considérée comme la méthode individuelle la plus performante sur le dataset *Yeast*.

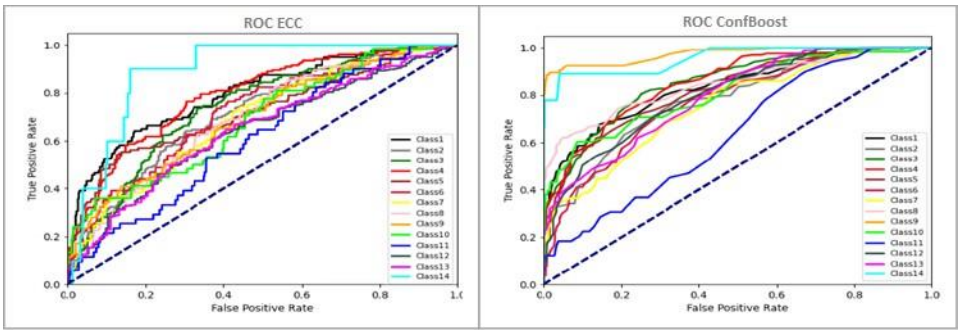

Figure 4.10 Comparaison entre les courbes AUROC moyenne de ConfBoost et ECC entrainés sur l'ensemble de données Yeast.

Ces résultats soulignent l'efficacité et la valeur ajoutée de la stratégie de combinaison adoptée par *ConfBoost*. En fusionnant les prédictions des sous-modèles hétérogènes et complémentaires, *ConfBoost* exploite la diversité des méthodes ensemblistes et tire parti de leurs forces respectives pour atténuer les défis posés par la CML. De plus, la capacité de *ConfBoost* à intégrer des pondérations aux labels basées sur les scores de confiance, couplée à des seuils adaptés, contribue à renforcer encore davantage ses performances.

4.4.3 Scénario 3: Comparaison des performances entre ConfBoost et les méthodes connexes

Ce scénario évalue et compare les performances de cinq méthodes de pointe afin de déterminer l'efficacité de l'approche proposée, *ConfBoost*. Les méthodes examinées comprennent trois approches basées sur le Stacking simple, à savoir MLS [119], RFS [140] et ABC-based Stacking [168]. D'autre part, deux autres méthodes basées sur le Stacking Pondéré sont également considérées : MLWSE [138] et WKNNS [152]. Il est important de noter que MLWSE utilise une pondération de classifieurs, tandis que WKNNS se concentre sur la pondération des caractéristiques. Cette différence dans les approches de pondération peut avoir un impact significatif sur les performances des algorithmes, en fonction des caractéristiques spécifiques des datasets utilisés.

Par ailleurs, le choix des datasets testés, *Yeast* et *Medical*, est motivé par leurs particularités distinctes, qui peuvent considérablement affecter les performances des algorithmes évalués. Par exemple, le dataset *Yeast* présente un déséquilibre modéré, avec un indice moyen (avgIR =7,197), ainsi qu'un niveau élevé de dépendance entre les labels (rDep = 0,670).

En revanche, le dataset *Medical* se distingue par un déséquilibre beaucoup plus marqué, avec un indice moyen (avgIR= 89,501), et un degré de dépendance entre labels nettement plus faible (rDep = 0,039). De plus, le nombre de labels et d'attributs dans le dataset *Medical* dépasse celui de *Yeast*, ce qui peut présenter des défis supplémentaires en termes de complexité et de dimensionnalité des données.

Les résultats présentés dans la Table 4.5 offrent une clarté saisissante quant à la performance des différentes méthodes évaluées. Les méthodes de Stacking simple, notamment MLS, RFS et ABC based-Stacking, démontrent des performances modérées à louables selon les diverses métriques d'évaluation. Cependant, lorsqu'elles sont confrontées aux méthodes de Stacking pondérées, telles que MLWSE et WKNNS, ces dernières les surpassent systématiquement sur l'ensemble des datasets utilisés. Comme illustré dans les Figures 4.11 et 4.12, l'approche proposée, *ConfBoost*, se distingue en affichant des résultats impressionnants, surpassant à la fois les méthodes de Stacking simple et pondérées sur la plupart des mesures d'évaluation pour les deux datasets examinés,

La performance remarquable de l'approche *ConfBoost* découle de plusieurs facteurs clés. Tout d'abord, la pondération des labels représente une innovation cruciale. En attribuant des poids plus élevés à certains labels, le modèle peut mieux tenir compte de l'importance relative de chaque classe, ce qui se traduit par des prédictions plus précises et une réduction du risque de surajustement. Cette stratégie permet de mieux équilibrer la contribution de chaque label à la fonction de coût globale, ce qui conduit à des décisions plus informées.

En outre, l'utilisation de seuils ajustés basés sur les scores de confiance des labels constitue une autre avancée significative. En adaptant dynamiquement ses seuils de décision au modèle, l'approche *ConfBoost* parvient à générer des prédictions adaptées aux caractéristiques spécifiques de chaque label. Ainsi, le modèle peut s'adapter de manière plus précise aux nuances des données, améliorant ainsi sa capacité à généraliser et à produire des prédictions fiables dans des contextes variés.

De plus, la sélection minutieuse des classifieurs de base constitue un autre facteur essentiel de l'approche *ConfBoost*, contribuant de manière significative à l'optimisation de sa performance prédictive global. Cette sélection rigoureuse vise à identifier et à intégrer des classifieurs de base diversifiés et complémentaires dans leurs prédictions.

La diversité est un paramètre crucial permettant à *ConfBoost* de bénéficier d'une couverture plus large de l'espace des caractéristiques, ce qui renforce sa capacité à généraliser efficacement sur de nouvelles données. D'autre part, la complémentarité entre les classifieurs permet à ConfBoost d'atténuer les défis posés par le domaine tels que le déséquilibre des classes de données (EPS et RAKEL), la gestion des dépendances entre labels (ECC) ainsi que la dimensionnalité de l'espace de sortie (RFPCT).

Table 4.5. Comparaison des performances de ConfBoost avec les méthodes connexes.

Datasets	Techniques Ensemble	Méthodes Ensemble	Hloss ↓	Accuracy ↑	F1-score ↑	Micro - F1↑	Macro-F1↑
Yeast	Stacking	MLS	0.249	0.434	0.556	0.581	0.384
		RFS	0.198	0.669	0.500	0.592	0.406
		ABC Stacking	**0.192**	0.535	0.640	0.520	0.312
	Stacking Pondéré	MLWSE	0.199	0.801	0.625	0.621	0.593
		WKNNS	0.210	0.744	0.599	0.605	0.472
		ConfBoost	0.194	**0.809**	**0.647**	**0.646**	**0.624**
Medical	Stacking	MLS	0.100	0.752	0.783	0.813	0.669
		RFS	0.009	0.817	0.778	0.609	0.476
		ABC Stacking	0.110	0.767	**0.797**	0.620	0.342
	Stacking Pondéré	MLWSE	0.130	**0.987**	0.770	0.759	0.755
		WKNNS	0.141	0.891	0.744	0.784	0.418
		ConfBoost	**0.004**	0.976	0.781	**0.837**	**0.770**

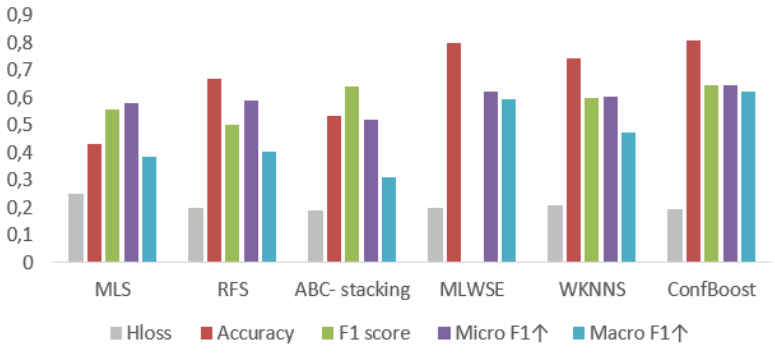

Figure 4.11 Résultats expérimentaux de *ConfBoost* et les méthodes connexes entraînées sur le dataset Yeast.

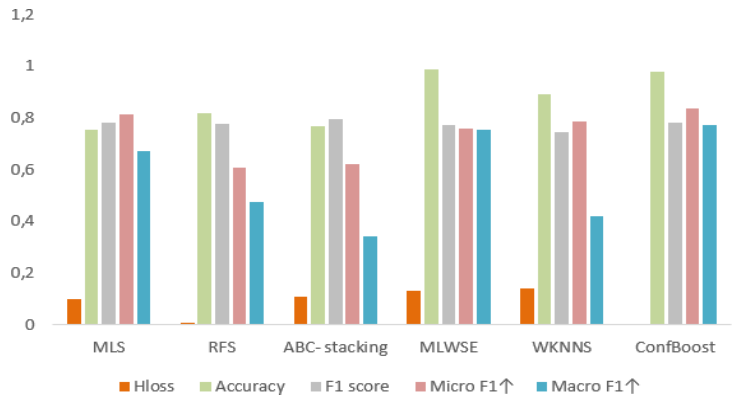

Figure 4. 12 Résultats expérimentaux de *ConfBoost* et les méthodes connexes entraînées sur le dataset Medical.

À la lumière des résultats obtenus dans les différents scénarios 1, 2 et 3, nous pouvons affirmer avec assurance que *ConfBoost* démontre des performances compétitives par rapport aux méthodes ensemblistes individuelles, ainsi qu'aux approches combinées conventionnelles telles que le Vote Simple, le Vote Pondéré et le Stacking, sur les divers ensembles de DML du référentiel Mulan. L'intégration de la pondération des labels basés sur leurs scores de confiance, associée à des seuils ajustés, attestent que *ConfBoost* a surpassé même les méthodes connexes utilisant des pondérations différentes. De plus, la sélection minutieuse des classifieurs ensemble de base, hétérogènes et complémentaires, a permis de relever les défis spécifiques posés par l'apprentissage multi-label, étant donné que chaque méthode individuelle est capable de résoudre un problème spécifique émergent dans le domaine d'application.

Les facteurs susmentionnés contribuent ensemble à améliorer la capacité de généralisation des systèmes d'apprentissage multi-label. En conséquence, nous affirmons que l'approche proposée, *ConfBoost*, a permis d'explorer de nouvelles voies pour l'application des méthodes d'ensemble dans le domaine de la CML, offrant ainsi des solutions plus robustes et adaptatives pour traiter des DML complexes.

4.4 Conclusion

Ce chapitre a présenté une analyse détaillée d'une nouvelle approche d'ensemble *ConfBoost*, qui est une collaboration de méthodes ECML. Tout d'abord, nous avons détaillé le fonctionnement de *ConfBoost* avec une description approfondie de ses phases.

Ensuite, une étude expérimentale de l'approche proposée est présentée, incluant les ensembles de DML utilisés et les paramètres expérimentaux définis. Les résultats expérimentaux sont ensuite examinés à travers trois scénarios différents : i) Une comparaison des performances des méthodes ECML individuelles, ii) Une comparaison des approches combinées, iii) Une comparaison entre *ConfBoost* et les méthodes connexes. Ce chapitre se clôture par un résumé sur les principaux résultats obtenus, soulignant ainsi les forces et les avantages de l'approche proposée dans les différents scénarios.

Dans le but d'examiner la portée de l'approche ensembliste sur des ensembles de DML à grande échelle, nous proposerons dans le chapitre suivant une nouvelle approche distribuée parallèle basée sur un framework de *MapReduce*.

5

Approche Distribuée Parallèle basée sur des Classifieurs Ensemble Multi-Label

Sommaire

5.1 Introduction

L'évolution croissante des technologies de l'information a conduit le monde vers une augmentation exponentielle des données, dépassant les capacités des structures traditionnelles. Dans le contexte de CML, certains ensembles de données se distinguent par leur haute dimensionnalité en termes d'instances, de labels et d'attributs. La gestion efficace et efficiente du déluge des données est donc devenue un défi majeur. Cela à inciter la communauté scientifique à adopter des approches robustes, permettent de répondre aux exigences de ces environnements complexes

Bien que le développement des méthodes ECML ait améliorer la robustesse et la capacité de généralisation des systèmes d'apprentissage Multi-Label en surmontant les défis introduits par ce domaine [108], la complexité informatique croissante des modèles ensemble pour gérer des ensembles de DML à grande échelle nécessite de nouvelles stratégies pour traiter efficacement cette grande masse de données. Par exemple, des ensembles de données tels que MS COCO (Microsoft Common Objects in COntext) ou OI (Open Images) contiennent des millions d'images, chacune annotée avec de multiples labels décrivant les objets présents dans l'image. En raison du volume et de la complexité de ces ensembles de données, l'approche ECML avait ses limitations pour traiter avec précision ce genre de données, nécessitant des techniques et des ressources informatiques spécialisées.

Pour traiter efficacement des ensembles de DML à grande échelle, l'informatique parallèle dans un environnement distribué se révèle être une solution prometteuse, exploitant les ressources de calcul et de mémoire partagées [101]. L'introduction d'un framework *MapReduce* dans des scénarios de CML a entraîné une amélioration significative des performances en termes de temps d'exécution et de capacité à gérer des volumes de données massifs [101], [105], [169]. En particulier, l'intégration de l'approche Ensemble pour la CML au modèle MapReduce a permis d'améliorer nettement la robustesse et la précision des prédictions. En répartissant le processus d'apprentissage des différents classifieurs sur plusieurs nœuds, *MapReduce* facilite le traitement parallèle, accélérant l'apprentissage et permettant de gérer efficacement de très grandes quantités de données. L'intégration des méthodes ECML avec l'approche distribuée a été explorée dans plusieurs recherches citées par la littérature [104], [170].

Dans ce chapitre nous présentons une approche distribuée, nommée *DisEMLC* (Distributed Ensemble Multi-Label Classifiers), utilisant une architecture de *MapReduce* qui intègre des CEML. L'approche proposée vise à gérer des ensembles de DML à grande échelle mis en œuvre par Hadoop, tout en tirant partie de la diversité des méthodes ensemblistes utilisées. D'abord, nous commençons par introduire les concepts de Big Data et de *MapReduce*. Ensuite, nous détaillons l'architecture de *DisEMLC*, suivie d'une étude expérimentale menée sur cinq ensembles de DML à grande échelle de référence. Cette expérimentation a montré que l'approche proposée offre des avantages en termes de précision, d'efficacité et d'évolutivité par rapport à notre précédente approche *ConfBoost*. Enfin, en tirant parti du modèle distribué et des performances des méthodes ECML, notre approche aide à surmonter les défis d'évolutivité couramment rencontrés dans des environnements distribués multi-label.

5.2 Big Data et CML

Selon les statistiques de la National Security Agency [171], le volume de données au niveau mondial sera multiplié par 50 au cours de la prochaine décennie. Chaque jour, environ 1826 Péta octets de données sont traités sur l'internet. Le moteur de recherche Google représente 77 % des 5 milliards de recherches effectuées sur internet en une seule journée [172]. Facebook, le célèbre réseau social, comporte environ 1,5 milliard d'utilisateurs actifs, contribuant à un énorme flux de données en téléchargeant jusqu'à 300 millions de photos, en publiant 510 000 commentaires et environ 293 000 mises à jour de statut quotidiennement [173].

De nombreux secteurs tels que le marketing, l'économie, l'industrie, la santé, la recherche scientifique, etc. sont touchés par la démultiplication de données, tandis que d'autres ont émergé en réponse à ce phénomène. L'importance du Big Data, reconnue par de nombreux organismes publics, lancent régulièrement de nouveaux appels aux projets industriels et scientifiques sur cette thématique. Par exemple, dans le secteur économique, les technologies du Big Data sont largement adopte pour fournir des informations nécessaires et à jour aux banques, permettant ainsi une gestion efficace des risques bancaires. Cette utilisation contribue à renforcer la sécurité au sein des institutions bancaires en réduisant les risques d'erreurs et de fraudes. De plus, la centralisation des tâches bancaires constitue une source importante de données massives, aidant les banques à prendre de meilleures décisions basées sur l'analyse de ces données [174].

Le concept de Big Data est apparu pour la première fois dans la revue *Nature*, désignant des données à grande échelle qui doivent être présentées, traitées et analysées à l'aide de techniques et de méthodes avancées [175] pour en extraire des informations utiles, synthétiques et exploitables [176]. Il s'agit d'un écosystème [177] représentant l'ensemble de la chaîne de valeur, qui comprend plusieurs étapes par lesquelles les données passent, depuis leur création ou collecte jusqu'à leur visualisation. Bien que les Big Data aient gagné en popularité dans les applications du monde réel, plusieurs défis scientifiques sont émergés en matière de manipulation, de traitement et de gestion des données.

5.2.1 Défis liés aux caractéristiques des données

Le Big Data est caractérisé par les 5V, comme illustré dans la Figure 5.1, fournissant ainsi une taxonomie pour classer les données selon *le Volume, la Variété, la Vitesse, la Valeur* et *la Véracité* [178]. D'autres caractéristiques, telles que la Variabilité, la Volatilité, la Validité du Big Data, ont également été mises en évidence dans certains travaux [179]. Ces 5V doivent être pris en compte et exploités dans le cadre d'une démarche d'optimisation de la gestion du Big Data.

- **Volume**: représente la quantité massive de données générées, stockées et traitées quotidiennement par les activités numériques. Ces données proviennent de diverses sources, telles que les réseaux sociaux, les journaux log, les transactions en ligne, etc., et peuvent être structurées, semi-structurées et non structurées. Cette caractéristique met l'accent sur la quantité de données plutôt que sur leur contenu. Il est important de noter que les principaux défis associés au *volume* sont les limites de stockage, les exigences des traitements effectués et la bande passante [180].

- **Variété**: représente la diversité des données en types et en formats, tels que les tweets, les vidéos, les photos, les textes, des audio, etc., qui ne sont pas faciles à structurer et à organiser. Par exemple, les données structurées sont généralement organisées dans des bases de données relationnelles, contrairement aux données semi-structurées et non structurées qui sont difficiles à traiter. Cette caractéristique présente un défi supplémentaire, car il existe de nombreuses sources de données hétérogènes qui ne sont pas correctement organisées et gérées.

- **Vélocité**: cette caractéristique est un facteur crucial dans l'évaluation des Big Data, il se réfère à la vitesse à laquelle les données sont générées, capturées et partagées, émergeant continuellement à tout moment. Les données qui arrivent sous forme de flux continus doivent être traitées en temps réel pour répondre aux besoins des processus soumis à des contraintes temporelles. Ainsi, cette vitesse peut être significativement influencée par la quantité de données entrantes. Plus le volume de données n'est important, plus les systèmes de traitement des données sont susceptibles d'être surchargés. Cela peut entraîner des goulots d'étranglement et des retards dans le traitement.

- **Véracité** : représente la qualité et la fiabilité des données sur lesquelles nous fondons nos prises de décision ou nos conclusions pertinentes. Cependant, les données massives provenant de diverses sources sont souvent incohérentes, incomplètes, ou même ambiguës, ce qui peut conduire à des résultats trompeurs. Il est donc crucial de garantir la véracité des données en les nettoyant et en les vérifiant avant de les utiliser dans des analyses ou des décisions critiques.

- **Valeur** : représente l'importance et l'utilité des données pour une organisation donnée. La valeur du Big Data provient généralement de l'identification et de l'extraction d'informations importantes et significatives, conduisant à une prise de décision plus éclairée. Par exemple, la Valeur d'une collecte de données apportée à une entreprise et à son public potentiel peut porter sur la personnalisation des services, l'amélioration des produits, le Feedback des clients, la tendance du marché, etc.

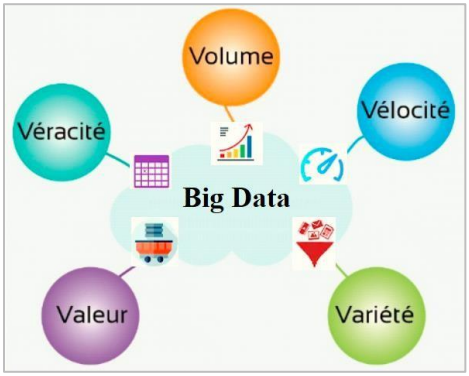

Figure 5.1 Caractéristiques des Big Data

5.2.2 Défis liés au traitement des données

Les données massives, qui varient en types et en structures, sont présentées sous des formes variées et doivent être converties en formats standards pour faciliter leur traitement. Cependant, le traitement de ces données nécessite plusieurs phases spécifiques, tel que illustré par la Figure 5.2.

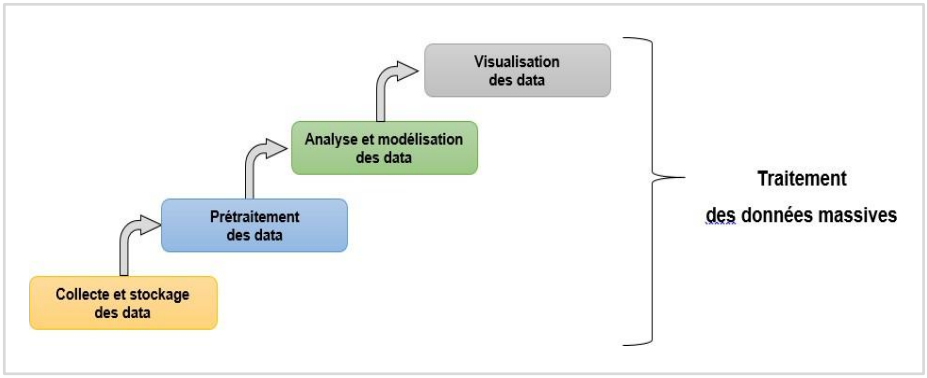

Figure 5.2 Processus de traitement des données

Tout d'abord, des données provenant de sources et applications hétérogènes, telles que des capteurs, des réseaux sociaux, des dispositifs de surveillance, et des transactions commerciales, sont collectées. Ensuite, elles sont stockées dans des infrastructures adaptées telles que des data lakes, des bases de données NoSQL, ou des systèmes distribués, qui permettent un accès flexible et scalable. Après le stockage, les données sont prétraitées [181] par le biais de techniques de filtrage pour éliminer les données inutiles, corriger les erreurs, et éliminer les incohérences. L'objectif de cette phase est de fournir des informations fiables et exploitables pour une étape ultérieure de traitement et d'analyse de données.

L'étape d'analyse repose sur des algorithmes et des techniques avancées, telles que le Data Mining, le Machine Learning, le Deep Learning, et d'autres méthodes analytiques. Elle vise à extraire des informations pertinentes et des connaissances exploitables pour améliorer la prise de décision. Le traitement et l'analyse des données incluent les tâches de segmentation, de classification, de prédiction, ainsi que la découverte de motifs cachés.

Enfin, ce cycle de vie se conclut par la phase de visualisation, qui est essentielle pour interpréter et communiquer les résultats de manière claire et concise issus de la phase d'analyse. Pour cela, des graphiques, des tableaux de bord interactifs, et d'autres outils visuels sont utilisés pour présenter les résultats de façon compréhensible, facilitant ainsi l'évaluation de la valeur des informations extraites.

Cette étape permet également de détecter rapidement des tendances, des anomalies, et d'autres insights essentiels à la prise de décision en utilisant divers types de graphiques pour représenter les informations utiles [182], [183].

5.2.3 Défis liés à la gestion des données

Les défis liés à la gestion des données massives concernent plusieurs aspects clés de leur administration au sein des organisations [184]. La préservation de la confidentialité, la sécurisation des données, et leur gouvernance figurent parmi les principaux défis auxquels les entreprises sont confrontées. La confidentialité des données est cruciale pour garantir que les informations sensibles ne soient pas compromises ou divulguées de manière inappropriée. Par ailleurs, la sécurisation des données est d'une importance capitale pour protéger celles-ci contre les menaces, telles que les cyberattaques et les fuites de données. De plus, la gouvernance des données vise à établir des politiques, des procédures et des normes pour garantir l'intégrité, la qualité et l'utilisation éthique des données.

Un autre défi majeur en matière de gestion du Big Data est le manque de personnel qualifié capable de gérer efficacement chaque phase du traitement des données massives. Avec l'évolution rapide des technologies et l'émergence de nouvelles techniques d'analyse et de traitement des données, il est essentiel pour les entreprises de disposer de professionnels compétents et formés, capables d'exploiter pleinement le potentiel de leurs données.

Le Big Data accorde également une grande importance aux concepts de sécurité et de confidentialité. Les données massives contiennent souvent des informations sensibles et confidentielles, dont l'accès non autorisé pourrait entraîner des conséquences néfastes, tant pour les individus concernés que pour les entreprises qui les détiennent. Il est donc important de mettre en œuvre des mesures de sécurité robustes et des protocoles de confidentialité efficaces afin de garantir la protection et l'intégrité des données dans le contexte de la cyber-sécurité.

En plus des défis généraux liés aux Big Data décrits ci-dessus, le domaine de la CML présente des défis supplémentaires en raison de la nature complexe des DML, comme détailler dans le Chapitre 2, Section 2,4, rendant ainsi la tâche de classifier des ensembles de DML massifs encore plus ardue en raison des exigences accrues en stockage et en calcul.

Ainsi, la gestion de données massives, combinée au besoin de stocker des données multi-label associées à chaque exemple, peut épuiser la capacité de stockage disponible. De plus, l'entraînement et l'évaluation des modèles de CML à grande échelle peuvent être coûteux en termes de temps et de ressources. Cela nécessite l'adoption d'infrastructures distribuées, telles que le modèle *MapReduce* [185], [186], pour traiter efficacement les volumes de données tout en minimisant les coûts.

L'approche *MapReduce* a permis de surmonter les défis posés par le Big Data et la CML tout en offrant une évolutivité capable d'ajuster les ressources de calcul en fonction de la taille croissante des datasets multi-label. De plus, cette approche assure une rapidité accrue dans le traitement des données en exploitant le parallélisme massif, répartissant les tâches de calcul sur de multiples nœuds pour minimiser le temps d'exécution global.

Bien que plusieurs travaux cités dans la littératures aient adapté l'approche distribuée parallèle à CML [99], [100], [101], [102], [103], [104], [105], peu de recherches ont adapté l'approche ensemble. Par exemple, Wu et al. [170] ont proposé l'approche Adaptative Random Forest (ARF) pour la CML, dans laquelle la méthode de forêt aléatoire a été parallélisée à l'aide du paradigme *MapReduce*. Dans cette approche, les primitives Map et Reduce ont contribué à la construction des arbres, chaque arbre étant traité comme une entité parallèle indépendante, et les prédits de tous les arbres ont été agrégés simultanément. Dans une étude distincte, Gonzalez-Lopez et al. [104] ont développé des implémentations basées sur le framework Spark, intégrant la méthode RAKEL, afin d'améliorer l'évolutivité des méthodes ECML. Leur étude a mené une analyse comparative entre les différentes approches parallèles distribuées, en se concentrant sur les performances des modèles de classification ainsi que sur les temps d'exécution obtenus. Cependant, ces travaux n'ont pas exploité l'impact de la diversité des CEML ni la mise en œuvre des mécanismes de pondération spécifiques pour optimiser les performances.

5.3 Modèle MapReduce

5.3.1 Définition

Le modèle *MapReduce* [187] a été conçu pour traiter des données massives en utilisant un traitement parallèle et distribué, permettant leur répartition sur de multiples nœuds de calcul au sein d'un cluster. En divisant le processus en deux étapes clé, le mappage et la réduction, ce modèle permet un traitement parallèle et rapide sur plusieurs machines.

En exploitant la puissance de calcul distribuée, *MapReduce* réduit considérablement le temps d'exécution, un avantage crucial dans le contexte du Big Data. Il a été largement utilisé dans divers domaines d'application, notamment l'analyse de données, le traitement de texte, l'indexation web, l'apprentissage automatique, et bien d'autres domaines. De plus, pour assurer la fiabilité des systèmes, le modèle *MapReduce* intègre une tolérance aux pannes [188] grâce à des mécanismes robustes permettant d'assurer les tâches suivantes :

i) **Détection des pannes** : Lorsqu'une panne est détectée, le système peut automatiquement réaffecter les tâches à d'autres nœuds de calcul fonctionnels.

ii) **Réplication des tâches sur plusieurs nœuds de calcul** : Lorsque l'un des nœuds tombe en panne, les tâches répliquées sur d'autres nœuds peuvent continuer leur exécution. Ainsi, les données traitées par le nœud défaillant peuvent être distribuées aux autres nœuds disponibles.

5.3.2 Principe de fonctionnement

Le modèle *MapReduce* contient un ensemble de composants interconnectés qui travaillent ensemble, comme détaillé dans la Figure 5.5, pour permettre un traitement efficace et parallèle des données massives sur un cluster de calcul distribué.

- ***Prétraitement des données d'entrées***

Ce processus commence par une phase de Prétraitement des données d'entrées, comprenant des opérations spécifiques aux données massives, telles que la décompression des fichiers, la conversion des données vers des formats standards, et d'autres opérations nécessaires pour faciliter le traitement de ces données. Ensuite, l'ensemble de données prétraitées D est subdivisé en blocs ou splits plus petits S_1, S_2,.., S_n dans une phase appelée la fragmentation (Splitting).

Ces splits seront par la suite distribués aux nœuds de calcul dans le cluster. Dans la phase de Splitting, chaque split S_i est transformé en paire (k_{ij}, val_{ij}), où la clé k_{ij} est la position de départ du split (offset) dans le fichier d'origine et val_{ij} représente la valeur associée à k_{ij} pour le split S_i. La phase de Splitting peut être définie par l'équation (5.1), où (k_{ij}, val_{ij}) représente la paire clé-valeur associées au split S_i.

$$(S_i) \rightarrow \{(k_{i1}, val_{i1}), (k_{i2}, val_{i2}), \dots, (k_{in}, val_{in})\} \quad (5.1)$$

- ***Phase de Mappage***

À cette étape, une fonction *Map* est appliquée en prenant en entrée toutes les paires (k_{ij}, val_{ij}) formées à partir de l'étape précédente pour générer une liste de paires (k_{ij}, val_{ij}) intermédiaires, comme illustré dans la Figure 5.3. Ainsi, la liste des paires intermédiaire servira d'entrée à la fonction *Reduce*, ou chaque paire d'entrée (k_{ij}, val_{ij}) est traitée indépendamment par la fonction *Map*, comme définie par l'équation (5.2).

$$Map\left(k_{ij}, val_{ij}\right) \rightarrow \left\{\left(k'_{ij1}, val'_{ij1}\right), \left(k'_{ij2}, val'_{ij2}\right), \ldots, \left(k'_{ijn}, val'_{ijn}\right)\right\} \quad (5.2)$$

Où (k_{ij}, val_{ij}) représente une paire clé-valeur d'entrée du split S_i et (k'_{ijp}, val'_{ijp}) est une paire (clé –valeur) intermédiaire est produite par la fonction *Map* pour le split S_i.

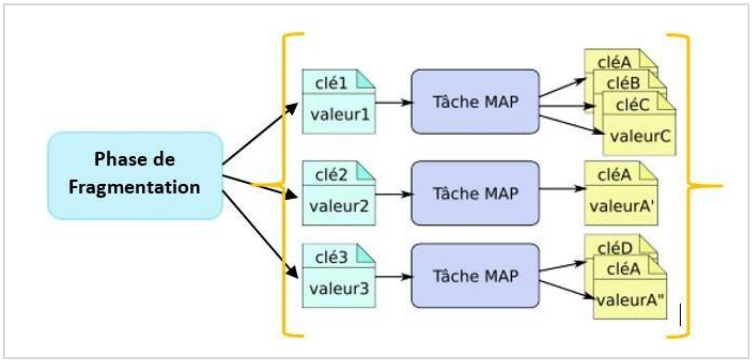

Figure 5.3 Tâche de la fonction *Map*

- ***Phase de mélange et tri***

Cette phase permet de collecter et de regrouper toutes les paires intermédiaires générées par la fonction *Map*, puis les regroupées par la clé intermédiaire k'_{ijp} afin que toutes les valeurs val'_{ijp} associées à cette clé soient rassemblées ensemble sur la même machine. Les paires regroupées (k'_{ijp}, val'_{ijp}) sont ensuite triées par clé intermédiaire k'_{ijp} et envoyées aux nœuds de calcul du cluster pour être utilisées par la fonction *Reduce*.

- ***Phase de réduction***

Une fois le mélange et le tri des paires intermédiaires sont terminés, une fonction *Reduce* est appliquée pour agréger toutes les paires intermédiaires ayant la même clé k'_{ijp}, comme illustré dans la Figure 5.4.

Pour chaque clé unique, la fonction Reduce est appliquée à toutes les valeurs associées à cette clé pour produire le résultat final. Cette fonction est définie par l'équation (5.3).

$$Reduce\left(k'_{ijp}, [val'_{ijp1}, val'_{ijp1}, .., val'_{ijpm}]\right) \rightarrow \left(k'_{ijp}, score\right) \qquad (5.3)$$

Où :

- ✓ $(k'_{ijp}, [val'_{ijp1}, val'_{ijp2}, .., val'_{ijpm}])$ est une paire regroupée et triée par la clé intermédiaire k'_{ijp}.
- ✓ $[val'_{ijp1}, val'_{ijp2}, .., val'_{ijpm}]$ représente une liste ordonnée des valeurs intermédiaires associées à la clé k'_{ijp}.
- ✓ *score* est un résultat agrégé, obtenu après l'application de la fonction *Reduce* sur toutes les valeurs intermédiaires ordonnées $[val'_{ijp1}, val'_{ijp2}, .., val'_{ijpm}]$.

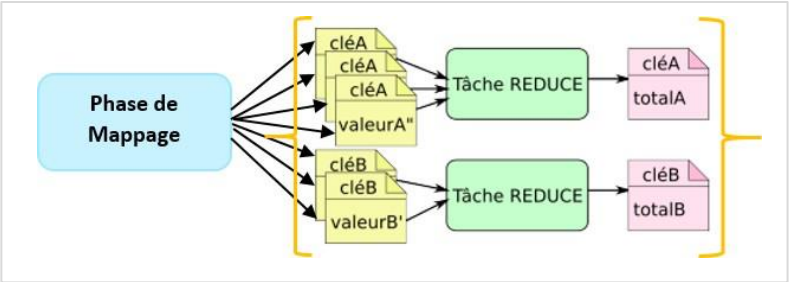

Figure 5.4 Tâche de la fonction *Reduce*

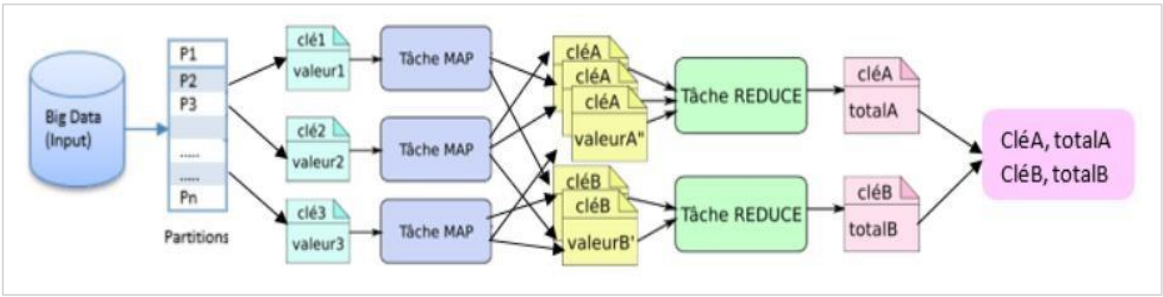

Figure 5.5 Architecture générale du modèle *MapReduce*

5.4 Modèle DisEMLC

L'approche *DisEMLC* (Distributed Ensemble Multi-Label Classifiers) repose sur une architecture parallèle et distribuée, basée sur le framework *MapReduce*. Cette approche intègre diverse CEMLs au niveau de la phase de *Mappage* et un mécanisme de pondération des labels avec une fonction de seuillage appliquée après la phase de *Réduction*.

La combination de la puissance du traitement parallèle et distribué offerts par *MapReduce* avec l'efficacité de l'approche ensembliste pour la CML, permet de viser les objectifs suivant :

En assignant divers CEMLs, tels qu'ECC [29], EPS [30], RAKEL [31], et RFPCT [32] à chaque mapper, l'approche proposée permet de réduire les biais associés à l'utilisation d'un modèle unique et améliore la performance prédictive en capturant une plus grande variété de caractéristiques des DML. Elle permet également d'accélérer le processus de classification et de gérer efficacement de grands volumes de données grâce au traitement parallèle.

Ensuite, l'intégration d'une pondération des labels après la phase de réduction permet d'ajuster l'importance relative de chaque label en fonction de leur fréquence dans les données. Ainsi, une fonction de seuillage est appliquée pour conserver les labels les plus pertinents tout en éliminant ceux les moins fréquents, ce qui améliore la précision des résultats finaux.

Enfin, l'intégration des CEML au niveau des mappers maximise l'utilisation du parallélisme et la diversité des classifieurs pour une performance accrue, tandis que la pondération des labels après réduction affine les résultats en focalisant l'attention sur les labels les plus significatifs et informatifs, conduisant à des prédictions finales plus robustes.

Comme illustré dans la Figure 5.6, notre approche *DisEMLC* comporte les cinq modules principaux suivants :

- Répartition des données
- Mappage avec prédiction multi-label.
- Exécution des tâches *Reduce*.
- Pondération et sélection des labels.
- Emission des résultats finaux.

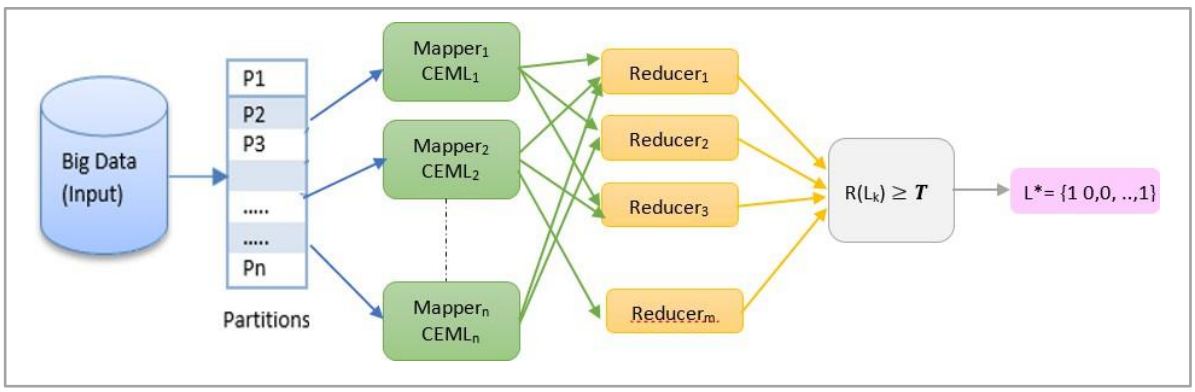

Figure 5.6 Architecture de l'approche DisEMLC

5.4.1 Répartition des données

Après avoir chargé l'ensemble de données d'apprentissage, *D*, il est nécessaire de le divisé en partitions disjointes $P_1, P_2,..., P_n$. Chaque partition P_i contient un sous-ensemble d'échantillons, chacun étant étiqueté par plusieurs labels. L'ensemble des partitions est ensuite transféré dans le système de fichiers distribué Hadoop (HDFS), assurant ainsi la distribution optimale des données sur les nœuds du cluster, ce qui permet un traitement parallèle et distribué.

Pour chaque partition P_i, un mapper Map_i est dédié et applique un classifieur ensemble multi-label C_j (parmi ECC, EPS, RAKEL et RF-PCT). Chaque Cj permet de générer une prédiction multi-label Pij pour chaque échantillon S_{ij} dans sa partition respective Pi.

5.4.2 Mappage avec prédiction multi-label

Cette étape vise à traiter chaque partition Pi de données en parallèle dans un cluster Hadoop. Chaque mapper Map_i applique un classifieur Cj pour générer des prédictions, qu'il associe aux éléments de chaque échantillon S_{ij} avant de les transmettre à la phase Reduce.

Chaque mapper Map_i commence par charger les échantillons S_{ij} d'une partition donné P_i. Ensuite, chaque échantillon S_{ij} est nettoyé, puis segmenter en éléments constitutifs (phrases), tel que $S_{ij} = \{e_{ij1}, e_{ij2},.., e_{ijm}\}$, où chaque e_{ijk} représente le *k-ième* élément extrait de l'échantillon S_{ij} d'une partition P_i. Le mapper Map_i extrait ensuite les caractéristiques des éléments de l'échantillon S_{ij} pour former un vecteur X_{ij} concaténé de caractéristiques, défini par l'équation (5.4) comme suit :

$$X_{ij} = \bigcup_{e_{ijk} \in S_{ij}} ExtractFeatures\left(e_{ijk}\right) \tag{5.4}$$

Le vecteur X_{ij} de l'échantillon S_{ij} est ensuite envoyé au classifieur C_j pour générer la prédiction multi-label P_{ij} de cet échantillon. Ainsi, la liste des labels prédits pour l'échantillon S_{ij} est définie par la fonction (5.5) suivante :

$$L_{pred}(S_{ij}) \leftarrow C_j(X_{ij}) \tag{5.5}$$

Les prédictions multi-label générées par C_j sont renvoyées au mapper Map_i, qui va associer à chaque élément e_{ijk} de S_{ij} la liste des labels prédits $L_{pred}(S_{ij})$ et le nombre d'occurrences O_{ijk} de cet élément dans S_{ij}.

Le mapper Map_i émet ensuite une paire clé-valeur sous la forme $(e_{ijk}, (L_{pred}(S_{ij}), O_{ijk}))$ pour chaque e_{ijk}, destinée à être utilisées lors de la phase Reduce. Il est évident que la clé de cette paire est l'élément e_{ijk}, tandis que la valeur est constituée de la liste des labels prédits de l'échantillon S_{ij} et du nombre d'occurrences de cet élément dans cet échantillon. La sortie du mapper *Mapi* constitue l'ensemble des paires clé-valeur générées pour tous les échantillons S_{ij} dans la partition Pi, comme définie par l'équation (5.6). L'algorithme 5.1, explique en détail le déroulement de la phase de mappage et de prédiction, jusqu'à l'émission des paires Clé-Valeur.

$$Map_i(P_i) = \left\{ \left(e_{ijk}, \left(L_{pred}(S_{ij}), O_{ijk} \right) \right) \forall\, S_{ij} \in P_i, e_{ijk} \in S_{ij} \right\} \tag{5.6}$$

Algorithme 5.1 : Mappage avec Prédiction

1 : Entrée :
Partition *Pi* de l'ensemble de données original (*D);*
{S_{i1}, S_{i2}, ..., S_{in}} ensemble d'échantillons de *Pi* ;
C_j : classifieur ensemble multi-label associé au mapper *;*

2 : **Pour** chaque S_{ij} dans *Pi* **faire**
3 : charger S_{ij};
4 : nettoyer et segmenter S_{ij} en *{e_{ij1}, e_{ij2},.., e_{ijm}}*;
5 : initialiser *Liste_paires* à vide ;
6 : initialiser X_{ij} à vide ;
7 : **Pour** chaque e_{ijk} dans S_{ij} **faire**
8 : X_{ijk} = *ExtractFeature* (e_{ijk});
9 : ajouter X_{ijk} à X_{ij};
10: **Fin pour ;**
11: *Lpred* (e_{ijk}) = C_j (X_{ijk});
12: **Pour** chaque e_{ijk} dans S_{ij} **faire**
13: O_{ijk} = *Nb-Occ*(e_{ijk}, S_{ij});
14: ajouter *(*e_{ijk}, *(L_pred(*S_{ij}*),* O_{ijk}*))* à *Liste_paires ;*
15: **Fin pour ;**
16: Retourner *Liste_paires;*

❖ *Déroulement de la tache de Mappage avec prédiction*

Supposons que nous avons une collection de messages électroniques classés en trois catégories de sujets, tels que *Réunion*, *Evénement* et *Rappel*. Cette collection est répartie en deux partitions, P_1 et P_2, chacune constituée de trois échantillons (emails). Soient S_{11}, S_{12} et S_{13} trois échantillons de P_1, présentés comme suit :

- ✓ S_{11} : " Une réunion est prévu pour demain à 10h, et n'oubliez pas de préparer le rapport du séminaire précédent" avec L_{pred} (S_{11}) = {*"Réunion", "Evénement"}.*
- ✓ S_{12} : " Veuillez transmettre les états des suivis mensuels de ce trimestre avant la fin du mois" avec L_{pred} (S_{12}) = {*"Rappel"}.*
- ✓ S_{13} : " Rappel : La réunion hebdomadaire est prévue pour demain. Préparez également le rapport du séminaire de cette semaine et n'oubliez pas de ramener les suivis mensuels de ce trimestre " avec L_{pred} (S_{13}) = { *"Réunion", "Evénement", "Rappel"}.*

L'ensemble des éléments pour chaque échantillon sont les suivant :

S_{11} = { "Une réunion est prévue pour demain à 10h.", "n'oubliez pas de préparer le rapport du séminaire précédent"}.

S_{12} = { "Veuillez transmettre les états des suivis mensuels de ce trimestre avant la fin du mois"}.

S_{13} = { "Rappel : La réunion hebdomadaire est prévue pour demain.", "Préparez le rapport du séminaire de cette semaine", "N'oubliez pas de ramener les suivis mensuels de ce trimestre"}.

La génération des paires Clé-Valeur des trois échantillons sont les suivants :

Pour (S_{11}) :

("Une réunion est prévue pour demain à 10h.", (["Réunion", "Événement"], 1))
("N'oubliez pas de préparer le rapport du séminaire précédent", (["Réunion", "Événement"], 1))

Pour (S_{12}):

("Veuillez transmettre les états des suivis mensuels du trimestre avant la fin du mois", (["Rappel"], 1))

Pour (S_{13}):

("Rappel: La réunion hebdomadaire est prévue pour demain", (["Réunion", "Événement", "Rappel"], 1))
("Préparez le rapport du séminaire de cette semaine", (["Réunion", "Événement", "Rappel"], 1))
("N'oubliez pas de ramener les suivis mensuels du trimestre", (["Réunion", "Événement", "Rappel"], 1))

La sortie finale de la fonction de mappage pour la partition *Pi* sera la suivante :

Map (P_1)= **{**("Une réunion est prévue pour demain à 10h.", (["Réunion", "Événement"], 1)), ("N'oubliez pas de préparer le rapport du séminaire précédent", (["Réunion", "Événement"], 1)), ("Veuillez transmettre les états des suivis mensuels du trimestre avant la fin du mois", (["Rappel"], 1)), ("Rappel: La réunion hebdomadaire est prévue pour demain", (["Réunion", "Événement", "Rappel"], 1)) ("Préparez le rapport du séminaire de cette semaine", (["Réunion", "Événement", "Rappel"], 1)) ("N'oubliez pas de ramener les suivis mensuels du trimestre", (["Réunion", "Événement", "Rappel"], 1))**}.**

La Figure 5.7 résume les différentes étapes du processus du mapper associé à la partition P_1, depuis le chargement des échantillons S_{11}, S_{12} et S_{13} jusqu'à l'émission des paires Clé-Valeur vers le *Reduce*.

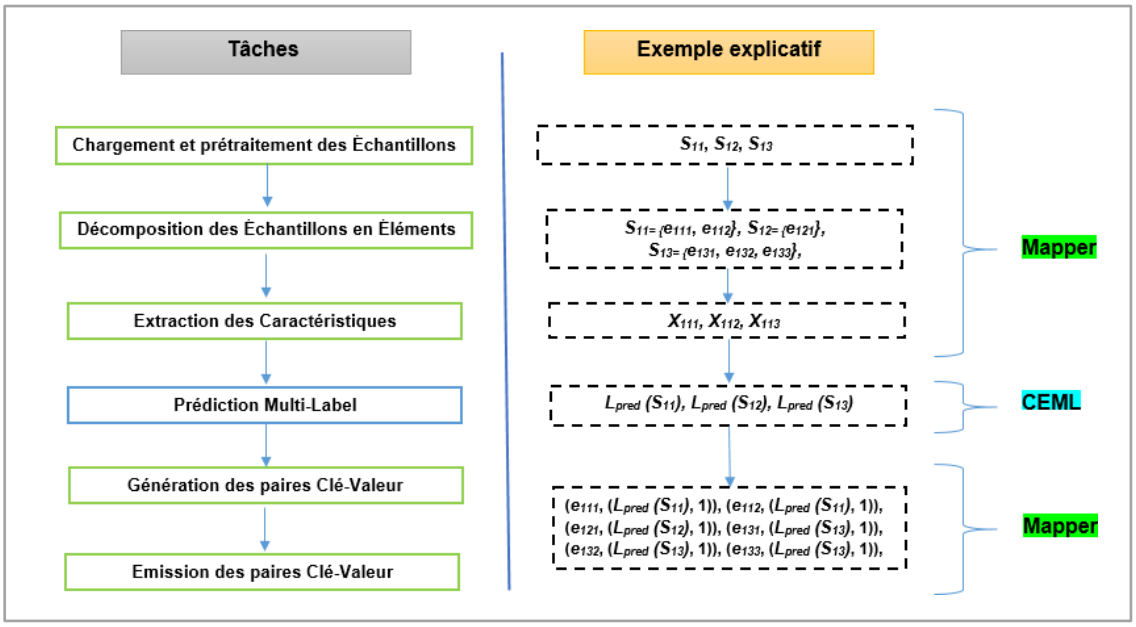

Figure 5.7 Processus d'un traitement d'un Mapper

5.4.3 Exécution de la fonction Reduce

Le *Reducer* reçoit les paires clé-valeur $(e_{ijk}, (L_{pred}(S_{ij}), O_{ijk}))$ émises par les *Mappers* et regroupe toutes les paires partageant la même clé. Pour chaque clé e_{ijk}, les listes complètes de labels prédits $L_{pred}(S_{ijm})$ issues des différents échantillons sont fusionnées selon l'équation (5.7), où N représente le nombre d'échantillons contenant e_{ijk}.

$$L_{pred}(e_{ijk}) = \cup_{m=1}^{N} L_{pred}\left(S_{ij_m}\right) \tag{5.7}$$

Ensuite, le Reducer additionne les comptes O_{ijk} de chaque clé e_{ijk} à travers tous les échantillons pour obtenir le nombre total d'occurrences de l'élément e_{ijk}, tel que défini par l'équation (5.8). Cette somme donne le total d'occurrences de e_{ijk} dans tous les échantillons, ce qui reflète la fréquence de chaque élément dans l'ensemble des données.

$$C(e_{ijk}) = \sum_{m=1}^{N} O_{ijk}\left(S_{ij_m}\right) \tag{5.8}$$

Enfin, le Reducer génère une paire clé-valeur finale pour chaque clé e_{ijk} , incluant les labels prédits fusionnés et le nombre total d'occurrences, tel que défini par l'équation (5.9). Cette sortie représente, pour chaque élément e_{ijk}, l'ensemble des labels associés dans tous les échantillons, ainsi que la somme des occurrences de cet élément.

$$Reduce(e_{ijl}) = \left(e_{ijk}, \left(\bigcup_{m=1}^{N} L_{pred}(S_{ijm})\right), \sum_{m=1}^{N} C(e_{ijk})\right) \tag{5.9}$$

En se référant à l'exemple précédent, la Figure 5.8 résume le traitement effectué par le Reducer, qui reçoit les paires clé-valeur par les mappers. Ces paires sont ensuite regroupées et traitées par le Reducer pour générer des paires clé-valeur agrégées.

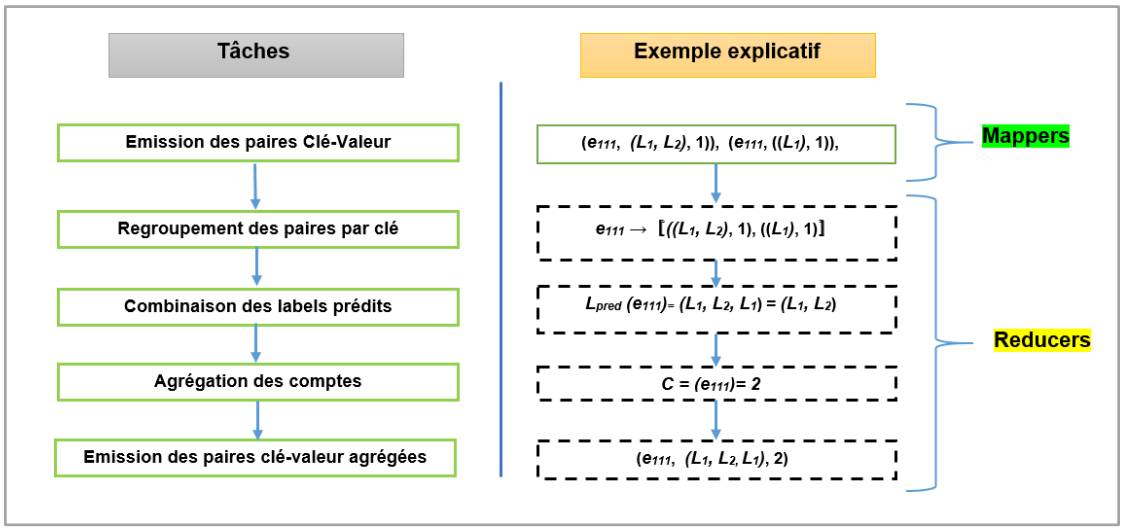

Figure 5.8 Processus d'un traitement d'un Reducer

5.4.4 Pondération et sélection des labels

Pour affiner les résultats produits par les Reducers, une sélectionner des labels les plus représentatifs et pertinents a été appliquée dans chaque paire clé-valeur émise par le Reducer. Pour ce faire, pour chaque paire clé-valeur agrégée reçue du Reducer, nous commençons par évaluer la redondance de chaque label L_k dans l'ensemble des labels prédits $Lpred(S_{ij})$ associés à la clé e_{ijk}.

La redondance $R(L_k)$ d'un label L_k, définie par l'équation (5.10), représente la proportion relative de L_k parmi tous les labels prédits pour un ensemble donné de paires clé-valeur.

$$R(L_k) = \frac{A_{ijk}}{\sum_{L \in L_{pred}(S_{ij})} A_{ijk}(L_{pred})} \tag{5.10}$$

Une fois la redondance $R(L_k)$ calculée pour chaque label, une fonction de seuillage $S(L_k)$ est appliquée pour identifier les labels pertinents. Seuls les labels dont la redondance dépasse un seuil de confiance $\boldsymbol{T}$ sont considérés comme représentatifs pour la clé e_{ijk}, selon l'équation (5.11) suivante :

$$S(L_k) = \begin{cases} L_k & \text{si } R(L_k) \geq \boldsymbol{T} \\ \text{éliminer } L_k & \text{sinon} \end{cases} \tag{5.11}$$

La bipartition de labels créée par la fonction de seuillage, permet de conserver les labels pertinents avec une redondance $R(L_k)$ supérieure ou égale au seuil $\boldsymbol{T}$, tout en éliminant les labels les moins fiables. Cette étape nécessite de considérer toutes les paires clé-valeur produites par les Reducers afin d'obtenir une sélection de labels consolidée et représentative.

5.4.5 Emission des résultats finaux

Après avoir sélectionné les labels les plus pertinents pour chaque clé e_{ijk}, ceux-ci sont alors émis sous une forme clé-valeur finale : $(e_{ijk}, \{L_k : C_{ijk}) / R(L_k) \geq T\})$, contenant uniquement les labels pondérés jugés pertinents. Cela simplifie l'ensemble de données en vue d'un traitement ultérieur.

5.5 Etude expérimentale

Dans cette section, nous offrons un aperçu général de la configuration de l'environnement expérimental, les ensembles DML à grandes échelles utilisés suivis des métriques employées pour évaluer les performances de l'approche proposée, *DisEMLC*.

5.5.1 Environnement expérimental

Notre étude expérimentale a été menée sur une machine équipée d'un processeur Intel(R) Core i7-4600U cadencé à 2,70 GHz, dotée d'une mémoire de 8 Go et un système d'exploitation Ubuntu version 16.04. Cette configuration a servi de base à une configuration Hadoop multi- nœuds, version 3.2.0, afin d'évaluer les performances et la scalabilité de notre approche *DisEMLC* dans un environnement de traitement de données massives distribués. Nous avons exploré l'impact des tailles de cluster en utilisant trois configurations différentes : 1) une petite configuration avec un cluster à 2 esclaves, une configuration intermédiaire avec un cluster à 4 esclaves et une configuration plus robuste avec un cluster à 6 esclaves.

Il est important de noter qu'un cluster Hadoop est composé de nœuds maîtres et de nœuds esclaves. Les nœuds maîtres sont désignés par un *Name Node* et un *Job Tracker*, tandis que les nœuds esclaves représentent les *Data Nodes* et les *Task Trackers*. La tâche d'un *Name Node* est de gérer le stockage des données (HDFS), tandis que le *Job Tracker* est chargé de gérer le traitement parallèle de toutes les données à l'aide des deux fonctions *Map* et *Reduce*. Quant aux nœuds esclaves, qui constituent la grande partie des machines, exécutent des *Data Nodes* et des *Task Trackers* qui reçoivent des instructions de leurs maitres respectifs, *Name Node* et *Job Tracker*, comme illustré dans la Figure 5.9.

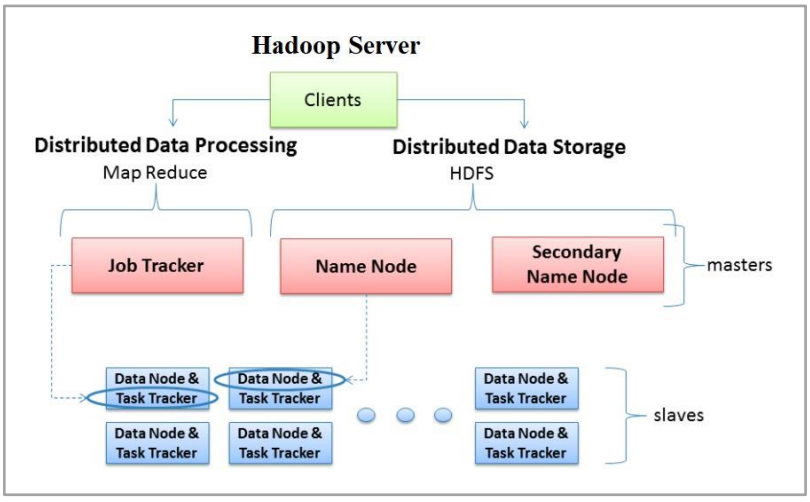

Figure 5.9 Architecture d'un Cluster Hadoop

Pour réaliser nos différentes expériences, nous devons préparer notre environnement expérimental par une série d'installations et configurations. D'abord, pour installer un cluster Hadoop, nous devons décompresser du logiciel sur les machines Maitres, *NameNode* et *JobTracker*.

Cependant, les machines esclaves, *Data Nodes* et *Job Trackers*, sont chargées d'exécuter les instructions des nœuds maitres. Pour cela, nous avons téléchargé Hadoop 3.2.0 depuis le site officiel d'Apache Hadoop, puis extraire le fichier d'installation à la racine de la distribution `HADOOP_HOME`. Toutes les machines du cluster ont généralement le même chemin d'accès à `HADOOP_HOME`. Ensuite, nous installons Java sur tous les nœuds en procédons par l'installation de OpenJDK à l'aide du gestionnaire de package d'Ubuntu (apt). Il est nécessaire de définir la variable d'environnement `JAVA_HOME` dans le même répertoire que celui de `HADOOP_HOME`.

La deuxième étape de la préparation de l'environnement consiste à configurer plusieurs fichiers pour mettre en place l'environnement Hadoop et assurer le bon fonctionnement des différents composants du cluster. Les fichiers à configurer sont principalement : *SHH, Hadoop, Hosts, Master Node, et en fini par* le formatage du fichier *HFDS*. Après avoir effectué toutes ces configurations, il est important de vérifier l'état du cluster. Nous commençons par nous assurer que tous les nœuds du cluster sont en ligne, fonctionnent correctement, et que les ressources sont correctement allouées et utilisées. Pour ce faire, il est utile d'utiliser les interfaces web Hadoop telles que HDFS qui fournit des informations sur l'état de HDFS, telles que l'état des nœuds du cluster, les capacités de stockage et les blocs de données, etc.

5.5.2 Ensemble de données Multi-Label

Pour évaluer notre étude expérimentale, nous avons utilisé cinq ensembles de données multi-label à grande échelle issus du domaine de la catégorisation de textes et répertoriés dans le référentiel MULAN (https://www.uco.es/kdis/mllresources/), à savoir Slashdot [189], Enron [17], Langlog [189], 20NG et Medical [157] qui est déjà utilisé au niveau de l'étude expérimentale du chapitre 4, section 4.3.1.

- Ensemble de données Slashdot représente une collection de titres d'articles et des commentaires partiels constituant les documents, recueillis à partir du site (http://slashdot.org).
- Ensemble de données Enron représente une collection de messages électroniques classés en 53 catégories de sujets, tels que la stratégie de l'entreprise, l'humour et les conseils juridiques.

- Ensemble de données Langlog a été compilé à partir du forum Language Log, abordant divers sujets liés à la langue, avec 75 sujets représentant l'espace des étiquettes.
- Ensemble de données 20NG, il s'agit d'une compilation d'environ 20 000 messages provenant de 20 Newsgroups, avec environ 1 000 messages disponibles pour chaque groupe, disponible sur le site (http://people.csail.mit.edu/jrennie/20Newsgroups/).

Table 5.1. Propriétés des ensembles de données multi-label expérimentaux.

Datasets	m	q	D	Card	Dens
Medical	978	45	1449	1.245	0.028
Slashdot	3782	22	1079	1.181	0.054
Euron	1702	53	1001	3.378	3.378
Langlog	1460	75	1004	1.180	1.180
20NG	19300	20	1006	1.029	1.029

m : nombre d'instances, **Card** : nombre de labels par instance,
q : nombre de labels, **Dens** : ratio entre card et m,
d : nombre d'attribues,

5.1.1 Mesures d'évaluation

Pour évaluer les performances prédictives de l'approche proposée, *DisEMLC*, deux mesures principales ont été prises : *la précision moyenne* du modèle pour évaluer l'exactitude de ses prédictions et *la durée d'exécution* de l'algorithme en fonction de la taille du cluster. De ce fait, ces mesures permettent de mesurer la variation des performances de *DisEMLC* en fonction de la taille du cluster, fournissant ainsi des informations cruciales pour une évaluation complète de l'approche proposée.

Au niveau des mappers, chaque partition *Pi* du dataset est assignée à un mapper *Mapi* qui applique C_j pour générer les prédictions des labels pour les échantillons *Sij* qu'il contient.

Pendant la phase de Reduce, pour chaque échantillon S_{ij} appartenant à la partition P_i, une précision moyenne $Prec_{moy}(S_{ij})$ est déterminée en calculant la proportion de labels correctement prédits pour S_{ij} par rapport au nombre total de prédictions faites pour cet échantillon, tel que défini par l'équation (5.12). Cette mesure qui est réalisée pour tous les échantillons de la partition *Pi*, permet d'évaluer la performance sur un sous-ensemble spécifique des données.

$$Prec_{moy}(S_{ij}) = \frac{nb\ labels\ correctement\ prédits\ pour\ S_{ij}}{nb\ total\ de\ prédictions\ pour\ S_{ij}} \quad (5.12)$$

Une fois toutes les précisions moyennes calculées pour chaque échantillon S_{ij}, elles sont ensuite agrégées pour obtenir la précision moyenne globale de la partition *Pi*, comme défini par l'équation (5.13), où *N* représente le nombre total d'échantillons dans la partition *Pi*.

$$Prec_{moy}(P_i) = \frac{1}{N}\sum_{i=1}^{N}\left(Prec_{moy}(S_{ij})\right) \quad (5.13)$$

Pour obtenir une mesure finale de la performance de l'approche *DisEMLC* sur l'ensemble de données *D*, les précisions moyennes des différentes partitions *Prec_moy (Pi)* sont agrégées selon l'équation (5.14), où *M* est le nombre total des partitions *Pi* dans *D*.

$$Prec(D) = \frac{1}{M}\sum_{j=1}^{N}\left(Prec_{moy}(P_i)\right) \quad (5.14)$$

5.2 Résultats expérimentaux

Pour évaluer les performances de l'approche proposée, *DisEMLC*, sur des ensembles de DML à Grandes Echelles, nous avons réalisé deux scénarios distincts. D'abord, nous avons effectué une comparaison des performances entre l'approche proposée, *DisEMLC*, et notre approche précédente, *ConfBoost*, étant donné que les deux approches partagent les mêmes CEML de base. Ensuite, nous avons analysé le comportement de l'approche *DisEMLC* dans diverses configurations de cluster : 2, 4 et 6 esclaves. A cette fin, pour chaque ensemble de données testé, nous avons évalué la précision moyenne et le temps d'exécution obtenus par chaque approche, comme présenté dans les Tables 5.2 et 5.3 respectivement. Il est évident de mentionner que les ensembles de DML massives sont classés dans les tables des résultats dans un ordre croissant (du moins volumineux au plus volumineux).

5.2.1 Scénario 1 : Comparaison de la précision moyenne entre DisEMLC et ConfBoost

Les résultats obtenus dans la Table 5.2 révèlent que l'approche proposée, *DisEMLC*, surpasse l'approche *ConfBoost* en termes de précision moyenne pour tous les ensembles de données expérimentés, avec la meilleure précision moyenne observée pour l'ensemble de données *Langlog*, atteignant un score de 88.2%, dépassant celle de *ConfBoost* qui est de 79.4%. Cette performance relative des deux approches est due à la manière dont chaque approche traite les données. Par exemple, le framework distribué et parallèle de l'approche *DisEMLC* peut conduire à une meilleure exploitation des ressources disponibles, conduisant ainsi à de meilleures performances pour des ensembles de données volumineux. En revanche, le méta modèle *ConfBoost*, basé sur un framework de Stacking pondéré, peut-être moins adapté aux données massives en raison de sa nature séquentielle et complexe, notamment son mécanisme de pondération, de sa dépendance à l'égard d'un seul nœud de calcul pour le processus de Stacking.

Il est évident de mentionner que l'impact de la taille et des caractéristiques des ensembles de données utilisés peut clairement influencer les différences de performance entre *DisEMLC* et *ConfBoost*, ainsi que les performances de *DisEMLC* en fonction des caractéristiques de l'ensemble de données utilisé. Par exemple, l'ensemble de données *Langlog* a permis une meilleure performance (Prec_moy=88.2%) par rapport au *Euron* (Prec_moy=72.9%), bien qu'il soit moins volumineux que *Langlog*. De même, l'ensemble de données *20NG*, bien que plus volumineux que *Langlog*, a montré une performance moins importante avec une précision moyenne moins significative (Prec_moy=70.8%),

Par conséquent, les ensembles de données à grande échelle bénéficient davantage des capacités de traitement parallèle fournies par *DisEMLC*, ce qui lui permet de gérer efficacement de grands volumes de données et d'élaborer des modèles précis. En outre, la densité des labels peut également jouer un rôle important, car *DisEMLC* peut mieux gérer les tâches de CML à forte densité de labels en distribuant le travail sur plusieurs nœuds de calcul.

Table 5.2 Comparaison de la précision moyenne entre *DisEMLC* et *ConfBoost*.

Dataset	DisEMLC	ConfBoost
Medical	0.784	0.528
Slashdot	0.813	0.662
Enron	0.729	0.520
Langlog	0.882	0.524
20NG	0.708	0.418

5.6.2 Scénario 2 : Temps d'exécution de DisEMLC pour différentes tailles de Clusters

En examinant les résultats de la Table 5.3 pour différents ensembles de DML volumineux et différents nombres de slaves, nous constatons que pour la plupart des cas, le temps d'exécution diminue lorsque le nombre de slaves augmente, ce qui est conforme à l'approche *DisEMLC* qui est conçu sur une architecture distribuée telle que *MapReduce*. Cela permet de paralléliser efficacement le traitement des données, conduisant ainsi à une réduction significative du temps d'exécution vu que les tâches sont réparties sur plusieurs nœuds de calcul.

D'autre part, les temps d'exécution tendent à être plus longs pour des ensembles de données plus volumineux tel que *Langlog* (T=216.746 s) par rapport à l'ensemble de données *Medical* (T=148.978s) qui est moins volumineux que *Langlog*. Cela est cohérent avec l'idée que le traitement d'un ensemble de données ayant une plus grande masse de données nécessite plus de temps.

Un autre point crucial non négligeable est l'impact de la complexité de l'ensemble de données utilisé. Les datasets avec un nombre important de labels, d'attributs, et même instances, tel que le *20NG*, nécessitent plus de temps de calcul en raison de leur complexité accrue. En revanche, les ensembles de données moins volumineux ou moins complexes comme *Medical*, *Slashdot* et *Enron*, la réduction du temps d'exécution en ajoutant plus de slaves est notable mais peut ne pas être aussi significative que pour des ensembles de données plus volumineux ou plus complexes tels que *20NG* et *Langlog*. Pour ces derniers datasets, le temps d'exécution initial est plus élevé, et l'ajout de slaves supplémentaires permet de réduire plus significativement le temps d'exécution, montrant ainsi l'impact important du parallélisme sur des tâches plus exigeantes en ressources.

En conséquence, le temps d'exécution augmente généralement avec la taille de l'ensemble de données testé. En effet, les ensembles de données à grande échelle contenant plus de données à traiter, nécessitent naturellement plus de calculs et, par conséquent, plus de temps. Toutefois, d'autres facteurs tels que l'efficacité de l'algorithme de CML utilisé, la complexité des données et la taille du cluster peuvent influer sur la durée d'exécution.

Table 5.3 Temps d'exécution de *DisEMLC* pour différentes tailles de Clusters.

Dataset	2 Slaves	4 Slaves	8 Slaves
Medical	148.978s	139.228s	126.149s
Slashdot	162.041s	153.336s	146.511s
Enron	186.001s	170.749s	168.173s
Langlog	216.746s	191.115s	179.216s
20NG	222.101s	209.749s	199.561s

Selon l'analyse des résultats obtenus dans cette étude expérimentale, confirmés par les graphes de comparaison des Figures 5.10 et 5.11, nous pouvons conclure que ces expériences soulignent l'importance d'intégrer des méthodes ensembliste pour la CML dans un environnement distribué, afin de traiter efficacement des ensembles de DML à grande échelle. L'approche *DisEMLC* s'avère particulièrement efficace pour le traitement massivement parallèle et l'évolutivité, grâce à l'intégration de CEML diversifiés au niveau des mappers, maximisant ainsi le parallélisme.

De plus, l'application de la pondération des labels après les Reducers affine les résultats en se concentrant sur les labels les plus pertinents. En conséquence, la combinaison du traitement parallèle, des CEML et de la pondération des labels dans *DisEMLC* permet d'améliorer significativement la précision globale du modèle, tout en garantissant une gestion efficace des ensembles de données à grande échelle.

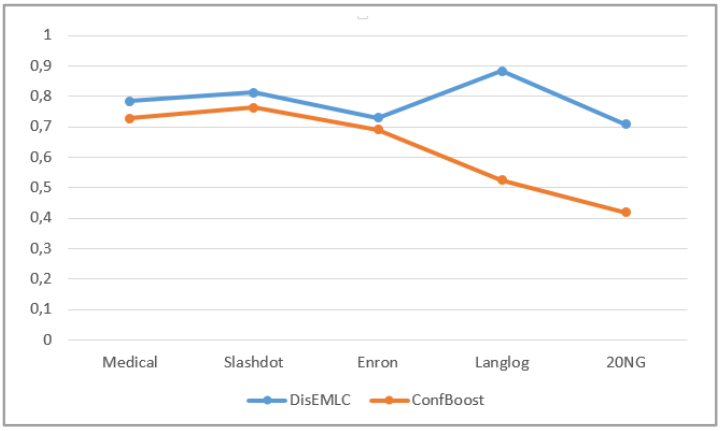

Figure 5.10 Précisions moyennes entre *DisEMLC* et *ConfBoost*

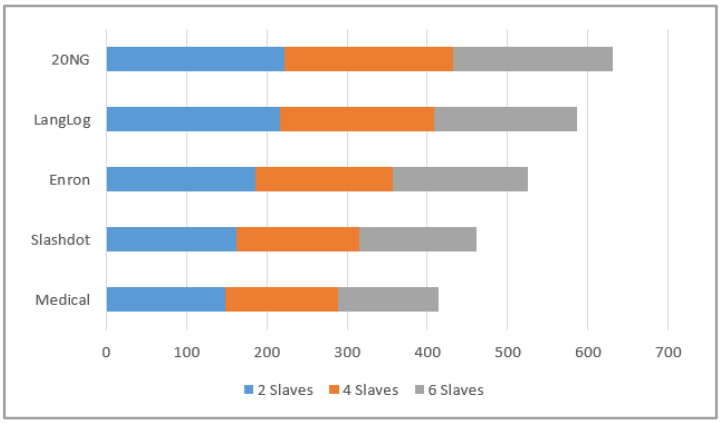

Figure 5.11 Temps d'exécution de *DisEMLC* pour diverses tailles de cluster (2, 4, 6 esclaves)

5.1 Conclusion

Dans ce chapitre, nous avons présenté une nouvelle approche distribuée, *DisEMLC*, fondée sur des CEML pour le traitement de données à grande échelle. Après avoir mettre en lumière l'importance croissante des défis liés à la gestion et à l'analyse des Big Data, nous détaillons l'architecture générale de *MapReduce*, y compris ses composants, son principe de fonctionnement ainsi que son intégration avec les méthodes ensemble Multi- Label. Ensuite, nous avons présenté l'architecture du modèle *DisEMLC* et ses différentes phases, suivies d'une étude expérimentale comportant l'analyse des ensembles de DML utilisés ainsi que l'environnement expérimental.

Les résultats expérimentaux obtenus été analysés et discutés dans deux expériences ; une comparaison de performance entre l'approche distribuée *DisEMLC* et l'approche séquentielle *ConfBoost*, qui partagent les mêmes classifieurs de base. Une seconde expérience est réalisée pour évaluer le temps d'exécution d'une tâche de CML par *DisEMLC* dans différentes configurations de cluster. Enfin, nous avons clôturé ce chapitre par un résumé général des résultats, mettant en évidence les performances et les implications de l'approche proposée.

6

Conclusion Générale et Perspectives

Conclusion générale

L'approche ensembliste, largement sollicitée dans diverses applications du monde réel, est considérée comme une solution de pointe pour de nombreux défis liés à l'apprentissage multi-label, tels que les dépendances entre les labels, le déséquilibre des données , ainsi que la dimensionnalité élevée de l'espace de sortie. Cependant, bien que des études importantes aient été menées ces dernières années sur les performances des méthodes ensemblistes pour la CML, aucune d'entre elles n'a pu résoudre la majorité de ces problèmes simultanément, en tenant compte de la mise à l'échelle des DML utilisées.

Pour cela, cette thése propose deux approches ensemblistes innovantes : une approche séquentielle (*ConfBoost*) et une autre distribuée parallèle (*DisEMLC*). Ces deux approches visent à proposer des systèmes robustes et généralisables, capables de relever les défis cités précédemment, tout en assurant la scalabilité des DML.

Outre la recherche bibliographique, notre travail s'est organisé en trois étapes principales. Tout d'abord, dans la rédaction de l'état de l'art, nous avons évoqué les méthodes de CML basées sur les deux approches TP et AA, ensuite, nous avons examiné les méthodes ensemblistes de pointe, leurs performances et leur robustesse face aux caractéristiques complexes des ensembles de DML provenant de divers domaines. À travers cette investigation, nous avons proposé une nouvelle taxonomie des méthodes en fonction des problèmes traités par chacune. D'autre part, pour tester les performances de ces méthodes, nous avons mené une analyse approfondie des méthodes les plus connues telles que EBR, ECC, EPS, RAKEL, HOMER, MLS, RF-PCT et TREMLC, testées sur sept ensembles de DML de la bioinformatique. L'analyse des résultats obtenus a clairement confirmé qu'aucune méthode ensembliste n'a pu démontrer sa supériorité sur les autres pour relever la plupart des défis posés par le domaine. De plus, nous avons constaté que le choix d'une méthode ensembliste appropriée dépend de plusieurs facteurs clés, tels que les caractéristiques des ensembles DML utilisés, les performances et les faiblesses de la méthode testée ainsi que les exigences spécifiques du domaine traité.

Dans une seconde étape, nous avons cherché a surmenter des defis susmentionnées en proposant un méta-modéle appelée *ConfBoost*, constitué d'une collaboration de classifieurs ensemble multi-label hétérogénes et complémentaires : ECC, EPS, RAKEL et RF-PCT.

L'approche proposée est basé sur un paradigme de Stacking pondéré utilisant une pondération de labels basés sur leur score de confidance, couplés à des seuils ajustés. Pour cela, nous avons mené trois différentes expériences en utilisant diverses mesures d'évaluation sur plusieurs ensembles de DML de référence répertoriés dans le référentiel Mulan.

- ✓ La première expérience s'est concentrée sur la comparaison des performances des méthodes d'ensemble individuelles ECC, EPS, RAKEL et RF-PCT sur des ensembles de données de divers domaines. L'analyse des résultats a montré que, quels que soient les métriques d'évaluation utilisées et les ensembles de données testés, chaque méthode possède des atouts uniques et peut répondre efficacement à un défi spécifique posé par le domaine de la MLC.

- ✓ La deuxième expérience a été réalisée sur la combinaison des méthodes ensemblistes utilisées précédemment, via quatre techniques d'ensemble différentes, à savoir le Vote Majoritaire, le Vote Pondéré, le Stacking et le Stacking Pondéré (*ConfBoost*). L'analyse des résultats a montré, d'une part, que les stratégies de Stacking produisaient des scores très performants par rapport aux stratégies de Vote (Vote Majoritaire et Vote Pondéré) dans la plupart des cas. D'autre part, l'approche *ConfBoost* a surpassé toutes les autres approches combinées, avec des résultats prometteurs sur la plupart des ensembles de données testés. Ce succées est dû à l'intégration de la pondération des labels basée sur leur confiance, couplée à des seuils ajustés, ce qui permet de générer des prédictions plus pertinentes et d'améliorer la précision globale du système.

- ✓ La troisième expérience a permis de comparer les performances de l'approche *ConfBoost* avec cinq approches apparentées : MLS, RFS et ABC-based Stacking, qui sont basées sur le Stacking simple, tandis que MLWSE et WKNNS utilisent des techniques de Stacking pondéré par les classifieurs et par les caractéristiques respectivement. Les résultats obtenus ont montré que les approches basées sur le Stacking pondéré surpassent systématiquement celles basées sur le Stacking simple sur l'ensemble des données utilisées. Par ailleurs, l'approche *ConfBoost* se distingue par des résultats impressionnants, surclassant à la fois les méthodes de Stacking simple et pondéré sur la plupart des mesures d'évaluation employées.

Dans une derniere étape, nous avons examiner la portée de l'approche ensembliste sur des données Multi-Label à grande échelle issues du référentiel Mulan. Pour cela, nous avons proposé une approche distribuée, *DisEMLC*, partageant avec l'approche *ConfBoost* les mêmes classifieurs ensembles diversifiés, intégrés au niveau des Mappers. Après la phase de Réduction, nous avons appliqué un mécanisme de pondération de labels basée sur le nombre de leur redondance, couplé à une fonction de seuilage. A travers deux expériences distinctes, nous avons évalué les performances et la scalabilité de l'approche *DisEMLC* par rapport à l'approche séquentielle *ConfBoost*, ainsi que l'impact des configurations de clusters sur le temps d'exécution des tâches de CML.

- ✓ Une premiere experience s'est menée sur la comparaison des performances entre l'approche distribuée *DisEMLC* et l'approche séquentielle *ConfBoost*. L'analyse des résultats obtenus a montré l'éficacité et la scabilité de *DisEMLC* sur tous les ensembles de données testés. Cela est dû à la combinaison du traitement parallèle, les CEML et l'application de la pondération des labels dans *DisEMLC*. En revanche, le méta-modèle *ConfBoost* s'avère moins efficace pour traiter de grands ensembles de DML en raison de sa dépendance à l'égard d'un seul nœud de calcul pour exécuter le processus de Stacking.

- ✓ La deuxième expérience a évalué le temps d'exécution d'une tâche de CML par *DisEMLC* pour les différentes configurations de cluster (2 esclaves, 4 esclaves et 6 esclaves). L'analyse des résultats obtenus a montré que le temps d'exécution augmente généralement en fonction de la taille de l'ensemble de données testé. Néanmoins, ce temps d'exécution peut également diminuer à mesure que le nombre de nœuds esclaves dans un cluster augmente, ce qui suggère que des clusters plus grands peuvent potentiellement conduire à des temps de traitement plus rapides.

Ainsi, dans un environnement séquentiel, l'approche *ConfBoost* offre des gains en précision et permet une géstion plus efficace des différents défis posés par le domaine, mais elle est limitée par sa scalabilité et son temps d'exécution. En revanche, dans un environnement distribué, l'approche *DisEMLC* prend toute sa portée. Elle devient plus évolutive, plus efficace, et mieux adaptée aux ensembles de DML à grandes échelles, grâce à la parallélisation des calculs et à la bonne gestion des ressources.

➢ Perspectives de la Recherche

Les perspectives de notre thése peuvent s'articuler autour de plusieurs axes de recherche futurs, visant à ouvrir de nombreuses possibilités pour renforcer la robustesse et l'applicabilité des deux approches proposées dans divers domaines pratiques, à savoir :

✓ *Pour ConfBoost*

- Utilisation des techniques de pré-sélection des labels avant leur pondération pour réduire le nombre total de labels à traiter. Cela allège la complexité de traitement et accélére la prédiction des modéles.
- Exploration des techniques d'optimisation, telles que la réduction de la dimensionnalité et l'échantillonnage stratifié, peut contribuer à réduire la complexité computationnelle et à améliorer l'efficacité du modèle.
- Intégration des techniques de rééquilibrage dynamique des classes, tel que le rééquilibrage progressif, qui serait bénéfique pour mieux gérer les labels déséquilibrés. Cela permettrait au modèle de généraliser des prédictions plus équilibrées et pertinentes, notamment pour les labels rares.

✓ *Pour DisEMLC*

- Tester et comparer les performances de l'approche sur diverses plateformes distribuées, telles qu'Apache Spark, Hadoop YARN et autres environnements similaires. Cela permettrait de déterminer la plateforme la plus adaptée en fonction des atous de l'approches pour le traitement DML massives.
- Optimiser le fonctionnement de l'approche en intégrant le Transfert Learning pour pré-entraîner les CEML sur des ensembles de données plus petits ou moins complexes, puis transférer ces modèles vers des tâches plus grandes et plus complexes dans des environnements distribués. Cela permettrait de réduire considérablement le temps d'entraînement, de mieux exploiter les ressources distribuées, tout en rendant la gestion des grands volumes de données plus efficace.

- Etendre l'application de *DisEMLC* à divers types de données. Cette extension permettrait d'évaluer la flexibilité et l'adaptabilité de *DisEMLC* sur des ensembles de données de nature hétérogène.

Productions Scientifiques

Publication Internationale

1. S. Guehria, H. Belleili, N. Azizi, et D. Zenakhra, « Boosting Multi-Label Classification Performance Through Meta-Model », International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)., vol. 38, no 01, p. 2350033, jan 2024, ISSN : 0218-0014. https://doi.org/10.1142/S0218001423500337

Conférences Internationales

1. S. Guehria, H. Belleili, et N. Azizi, « Distributed Approach for Large-Scale Ensemble Multi-Label Classification », *in the 2nd International Conference on Scientific and Innovative Studies (ICSIS 24)*, Konya, Turkey, Ed. All Sciences Academy, ISBN: 978-625-6314-01-6, p. 834-840, April, 2024, https://as-proceeding.com/index.php/icsis/2ndicsis2024

Chapitres de Livres

1. S. Guehria, H. Belleili, et N. Azizi, « A Comparative Analysis of Ensemble Learning Methods for Multi-Label Classification on Bioinformatics», *Proceedings of the 14th International Conference of Innovations in Bio-Inspired Computing and Applications (IBICA'23)*, Washington, 2023 (publication en Mars 2025).

2. S. Guehria, H. Belleili, et N. Azizi, « A Survey on Ensemble Multi-label Classifiers », *Proceedings of the 14th International Conference on Soft Computing and Pattern Recognition (SoCPaR'22)*, Washington, USA, vol. 648, p. 100-109, Cham: Springer Nature Switzerland, 2022. https://link.springer.com/chapter/10.1007/978-3-031-27524-1_11

3. S. Guehria, H. Belleili, N. Azizi, et S. B. Belhaouari, « "One vs All" Classifier Analysis for Multi-label Movie Genre Classification Using Document Embedding », *Proceedings of the 20th International Conference on Intelligent Systems Design and Applications (ISDA'20)*, India, vol. 1351, p. 478-487, Cham: Springer International Publishing, 2020. https://link.springer.com/chapter/10.1007/978-3-030-71187-0_44

Références Bibliographiques

[1] T. Joachims, « Text categorization with Support Vector Machines: Learning with many relevant features », in Machine Learning: ECML-98, Éd, in Lecture Notes in Computer Science, vol. 1398, Berlin, 1998, p. 137-142. doi: 10.1007/BFb0026683.

[2] B. Klimt et Y. Yang, « The Enron Corpus: A New Dataset for Email Classification Research », in Machine Learning: ECML 2004, Éd., in Lecture Notes in Computer Science, vol. 3201, Berlin, 2004, p. 217-226. doi: 10.1007/978-3-540-30115-8_22.

[3] I. Katakis, G. Tsoumakas, et I. Vlahavas, « Multilabel Text Classification for Automated Tag Suggestion », Proceedings of ECML PKDD'08, vol. vol 18, p. 75-83, 2008.

[4] A. Clare et R. D. King, « Knowledge Discovery in Multi-label Phenotype Data », in Principles of Data Mining and Knowledge Discovery, Éd, in Lecture Notes in Computer Science, vol. 2168. , Berlin, 2001, p. 42-53. doi: 10.1007/3-540-44794-6_4.

[5] R. M. M. Vallim, « The Multi-label OCS with a Genetic Algorithm for Rule Discovery: Implementation and First Results », Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation (GECCO '09), p. 1323 - 1330, July, 2009. doi:10.1145/1569901.1570078,

[6] M. R. Boutell, J. Luo, X. Shen, et C. M. Brown, « Learning multi-label scene classification », Pattern Recognition, vol. 37, no 9, p. 1757-1771, sept. 2004, doi: 10.1016/j.patcog.2004.03.009.

[7] G. Tsoumakas, I. Katakis, et I. Vlahavas, « Mining Multi-label Data », in Data Mining and Knowledge Discovery Handbook, Éd, Boston, Springer US, 2009, p. 667-685. doi: 10.1007/978-0-387-09823-4_34.

[8] W. Qu, Y. Zhang, J. Zhu, et Q. Qiu, « Mining Multi-label Concept-Drifting Data Streams Using Dynamic Classifier Ensemble », in Advances in Machine Learning, Éd, in Lecture Notes in Computer Science, vol. 5828. Berlin, 2009, p. 308-321. doi: 10.1007/978-3-642-05224-8_24.

[9] S. Guehria, H. Belleili, N. Azizi, et S. B. Belhaouari, « "One vs All" Classifier Analysis for Multi-label Movie Genre Classification Using Document Embedding », in Intelligent Systems Design and Applications, Éd, in Advances in Intelligent Systems and Computing, vol. 1351. , Cham: Springer, 2021, p. 478-487. doi: 10.1007/978-3-030-71187-0_44.

[10] Hung-Yi Lo, Ju-Chiang Wang, Hsin-Min Wang, et Shou-De Lin, « Cost-Sensitive Multi-Label Learning for Audio Tag Annotation and Retrieval », IEEE Trans. Multimedia, vol. 13, no 3, p. 518-529, juin 2011, doi: 10.1109/TMM.2011.2129498.

[11] K. Ozonat et D. Young, Towards a universal marketplace over the web: statistical multi-label classification of service provider forms with simulated annealing. 2009, p. 1304. doi: 10.1145/1557019.1557158.

[12] R. Rak, L. Kurgan, et M. Reformat, « A tree-projection-based algorithm for multi-label recurrent-item associative-classification rule generation », Data & Knowledge Engineering, vol. 64, no 1, p. 171-197, janv. 2008, doi: 10.1016/j.datak.2007.05.006.

[13] B. Zhu et C. K. Poon, « Efficient Approximation Algorithms for Multi-label Map Labeling », in Algorithms and Computation, in Lecture Notes in Computer Science, vol. 1741. , Berlin, 1999, p. 143-152. doi: 10.1007/3-540-46632-0_15.

[14] A. Rivolli, L. Parker, et A. de Carvalho, Food Truck Recommendation Using Multi-label Classification. 2017, p. 596. doi: 10.1007/978-3-319-65340-2_48.

[15] J. W. Kasubi et M. D. Huchaiah, « Human Activity Recognition for Multi-label Classification in Smart Homes Using Ensemble Methods », in Artificial Intelligence and Sustainable Computing for Smart City, Éd., in Communications in Computer and Information Science, vol. 1434. , Cham: Springer, 2021, p. 282-294. doi: 10.1007/978-3-030-82322-1_21.

[16] G. Tsoumakas et I. Katakis, « Multi-Label Classification: An Overview », International Journal of Data Warehousing and Mining, vol. 3(3), p.1-13, 2007.

[17] J. Read, B. Pfahringer, et G. Holmes, « Multi-label Classification Using Ensembles of Pruned Sets », in 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy: IEEE, déc. 2008, p. 995-1000. doi: 10.1109/ICDM.2008.74.

[18] G. Tsoumakas et I. Vlahavas, « Random k-Labelsets: An Ensemble Method for Multilabel Classification », in Machine Learning: ECML 2007, Éd., in Lecture Notes in Computer Science, vol. 4701. Berlin, 2007, p. 406-417. doi: 10.1007/978-3-540-74958-5_38.

[19] M. A. Tahir, J. Kittler, K. Mikolajczyk, et F. Yan, « Improving Multilabel Classification Performance by Using Ensemble of Multi-label Classifiers », in Multiple Classifier Systems, Éd., in Lecture Notes in Computer Science, vol. 5997. Berlin, 2010, p. 11-21. doi: 10.1007/978-3-642-12127-2_2.

[20] G. Tsoumakas, I. Katakis, et I. Vlahavas, « Effective and Efficient Multilabel Classification in Domains with Large Number of Labels », In Proceedings of the ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08), p. 53-59, 2008.

[21] E. Gibaja et S. Ventura, « A Tutorial on Multilabel Learning », ACM Comput. Survey, vol. 47, no 3, p. 1-38, avr. 2015, doi: 10.1145/2716262.

[22] D. Ganda et R. Buch, « A Survey on Multi Label Classification », Multi Label Classification Conference: STM journals, vol. 5, no 1, 2018,

[23] P. Prajapati, A. Thakkar, et A. Ganatra, « A Survey and Current Research Challenges in Multi-Label Classification Methods », International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, vol. 2, no 1, p. 6, March 2012.

[24] Tidake et Sane, « Multi-label Classification: a survey », International Journal of Engineering and Technology, vol. 7, no 4.19, p. 1045, nov. 2018, doi: 10.14419/ijet.v7i4.19.28284.

[25] G. Tsoumakas et I. Katakis, « Multi-Label Classification: An Overview », International Journal of Data Warehousing and Mining. vol 3, no 3, p.1–13, 2007, doi:10.4018/jdwm.2007070101.

[26] J. Read, B. Pfahringer, G. Holmes, et E. Frank, « Classifier Chains for Multi-label Classification », in Machine Learning and Knowledge Discovery in Databases, Éd, in Lecture Notes in Computer Science, vol. 5782. , Berlin, 2009, p. 254-269. doi: 10.1007/978-3-642-04174-7_17.

[27] J. Read, « A Pruned Problem Transformation Method for Multi-label Classification », in 2008 Eighth IEEE International Conference on Data Mining, 2008, pp. 235–243

[28] H. Blockeel, L. De Raedt, et J. Ramon, « Top-down induction of clustering trees », in Proceeding of 15th International Conference on Machine Learning (ICML), Morgan Kaufmann Publishers, 1998, pp. 55–63.

[29] Min-Ling Zhang et Zhi-Hua Zhou, « Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization », IEEE Transactions on Knowledge and Data Engineering, vol. 18, no 10, p. 1338-1351, oct. 2006, doi: 10.1109/TKDE.2006.162.

[30] M.-L. Zhang et Z.-H. Zhou, « ML-KNN: A lazy learning approach to multi-label learning », Pattern Recognition, vol. 40, no 7, p. 2038-2048, juill. 2007, doi: 10.1016/j.patcog.2006.12.019.

[31] S. L. Salzberg, « C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993 », Machine Learning, vol. 16, no 3, p. 235-240, sept. 1994, doi: 10.1007/BF00993309.

[32] X. Zheng, P. Li, Z. Chu, et X. Hu, « A Survey on Multi-Label Data Stream Classification », IEEE Access, vol. 8, p. 1249-1275, 2020, doi: 10.1109/ACCESS.2019.2962059.

[33] W. Liu, H. Wang, X. Shen, et I. W. Tsang, « The Emerging Trends of Multi-Label Learning », IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no 11, p. 7955-7974, nov. 2022, doi: 10.1109/TPAMI.2021.3119334.

[34] N. Aljedani, R. Alotaibi, et M. Taileb, « Multi-Label Arabic Text Classification: An Overview », International Journal of Advanced Computer Science and Applications (IJACSA), vol. 11, no 10, 2020, doi: 10.14569/IJACSA.2020.0111086.

[35] M.-L. Zhang et Z.-H. Zhou, « A Review on Multi-Label Learning Algorithms », IEEE Transaction on Knowledge and Data Engineering, vol. 26, no 8, p. 1819-1837, août 2014, doi: 10.1109/TKDE.2013.39.

[36] M. Chevalier, T. I. Diop, I. Megdiche-Bousarsar, N. Souf, et O. Teste, « Classification multi-labels de données de santé médico-économiques », 5e Seminaire de Veille Strategique Scientifique et Technologique (VSST 2018), p. 12, 2018.

[37] S. Sharma et D. Mehrotra, « Comparative Analysis of Multi-label Classification Algorithms », in 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), India: IEEE, déc. 2018, p. 35-38. doi: 10.1109/ICSCCC.2018.8703285.

[38] B. Lauser et A. Hotho, « Automatic Multi-label Subject Indexing in a Multilingual Environment », in Research and Advanced Technology for Digital Libraries, Éd, in Lecture Notes in Computer Science, vol. 2769, Springer Berlin Heidelberg, 2003, p. 140-151. doi: 10.1007/978-3-540-45175-4_14.

[39] Y. Song, L. Zhang, et C. L. Giles, « Automatic tag recommendation algorithms for social recommender systems », ACM Transactions Web, vol. 5, no 1, p. 1-31, févr. 2011, doi: 10.1145/1921591.1921595.

[40] Y. Yan, G. Fung, J. G. Dy, et R. Rosales, « Medical coding classification by leveraging inter-code relationships », in Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, Washington DC USA: ACM, juill. 2010, p. 193-202. doi: 10.1145/1835804.1835831.

[41] J. Yearwood, M. Mammadov, et A. Banerjee, « Profiling Phishing Emails Based on Hyperlink Information », in 2010 International Conference on Advances in Social Networks Analysis and Mining, Denmark: IEEE, août 2010, p. 120-127. doi: 10.1109/ASONAM.2010.56.

[42] S. Vogrincic et Z. Bosnic, « Ontology-based multi-label classification of economic articles », Computer Science and Information, vol. 8, no 1, p. 101-119, 2011, doi: 10.2298/CSIS100420034V.

[43] N. Oza, J. P. Castle, et J. Stutz, « Classification of Aeronautics System Health and Safety Documents », IEEE Transactions Systems, vol. 39, no 6, p. 670-680, nov. 2009, doi: 10.1109/TSMCC.2009.2020788.

[44] E. Loza Mencía et J. Fürnkranz, « Efficient Multilabel Classification Algorithms for Large-Scale Problems in the Legal Domain », in Semantic Processing of Legal Texts, Éd., in Lecture Notes in Computer Science, vol. 6036. Berlin, 2010, p. 192-215. doi: 10.1007/978-3-642-12837-0_11.

[45] E. Mencía et J. Fürnkranz, "An Evaluation of Efficient Multilabel Classification Algorithms for Large-Scale Problems in the Legal Domain", in Computer Science, Ed, Semantic Processing of Legal Texts, p. 2007, p. 192-215, Berlin, 2010, doi:10.1007/978-3-642-12837-0_11.

[46] T. N. Rubin, A. Chambers, P. Smyth, et M. Steyvers, « Statistical topic models for multi-label document classification », Machine Learning, vol. 88, no 1-2, p. 157-208, juill. 2012, doi: 10.1007/s10994-011-5272-5.

[47] H. Cong et L. H. Tong, « Grouping of TRIZ Inventive Principles to facilitate automatic patent classification », Expert Systems with Applications, vol. 34, no 1, p. 788-795, janv. 2008, doi: 10.1016/j.eswa.2006.10.015.

[48] P. Bhowmick, B. Anupam, P. Mitra, et A. Prasad, « Sentence level news emotion analysis in fuzzy multi-label classification framework », Res. Comput. Sci., vol. 46, p. 143-154, janv. 2010.

[49] R. E. Schapire et Y. Singer, « BoosTexter: A Boosting-based System for Text Categorization », Machine Learning, vol. 39, p. 135-168, 2000, doi: 10.1023/A:1007649029923.

[50] Y. Papanikolaou, G. Tsoumakas, M. Laliotis, N. Markantonatos, et I. Vlahavas, « Large-Scale Online Semantic Indexing of Biomedical Articles via an Ensemble of Multi-Label Classification Models », Journal of Biomedical Semantics, vol. 8, sept. 2017, doi: 10.1186/s13326-017-0150-0.

[51] A. F. De Souza et al., « Automated multi-label text categorization with VG-RAM weightless neural networks », Neurocomputing, vol. 72, no 10-12, p. 2209-2217, juin 2009, doi: 10.1016/j.neucom.2008.06.028.

[52] B. Al-Salemi, M. Ayob, G. Kendall, et S. A. M. Noah, « Multi-label Arabic text categorization: A benchmark and baseline comparison of multi-label learning algorithms », Information Processing & Management, vol. 56, no 1, p. 212-227, janv. 2019, doi: 10.1016/j.ipm.2018.09.008.

[53] H. Wu, S. Qin, R. Nie, J. Cao, et S. Gorbachev, « Effective Collaborative Representation Learning for Multilabel Text Categorization », IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no 10, p. 5200-5214, oct. 2022, doi: 10.1109/TNNLS.2021.3069647.

[54] X. Hao, J. Huang, F. Qin, et X. Zheng, « Multi-label learning with missing features and labels and its application to text categorization », Intelligent Systems with Applications, vol. 14, p. 200086, mai 2022, doi: 10.1016/j.iswa.2022.200086.

[55] L. Maltoudoglou, A. Paisios, L. Lenc, J. Martínek, P. Král, et H. Papadopoulos, « Well-calibrated confidence measures for multi-label text classification with a large number of labels », Pattern Recognition, vol. 122, p. 108271, févr. 2022, doi: 10.1016/j.patcog.2021.108271.

[56] A. Elisseeff et J. Weston, « A kernel method for multi-labelled classification », : Proceedings of the 14th International Conference on Neural Information Processing Sys-tems: Natural and Synthetic (NIPS'01), p. 681-687, 2001.

[57] A. Skabar, D. Wollersheim, et T. Whitfort, « Multi-label Classification of Gene Function using MLPs », in The 2006 IEEE International Joint Conference on Neural Network Proceedings, Vancouver, BC, Canada: IEEE, 2006, p. 2234-2240. doi: 10.1109/IJCNN.2006.247019.

[58] S. Diplaris, G. Tsoumakas, P. A. Mitkas, et I. Vlahavas, « Protein Classification with Multiple Algorithms », in Advances in Informatics, Éd., in Lecture Notes in Computer Science, vol. 3746, Springer Berlin, 2005, p. 448-456. doi: 10.1007/11573036_42.

[59] R. Rak, L. Kurgan, et M. Reformat, « Multi-label Associative Classification of Medical Documents from MEDLINE », in Fourth International Conference on Machine Learning and Applications (ICMLA'05), Los Angeles, USA: IEEE, 2005, p. 177-186. doi: 10.1109/ICMLA.2005.47.

[60] Y. Ren et al., « Multi-label classification for multi-drug resistance prediction of Escherichia coli », Computational and Structural Biotechnology Journal, vol. 20, p. 1264-1270, 2022, doi: 10.1016/j.csbj.2022.03.007.

[61] W. Wang, Q. Dai, F. Li, Y. Xiong, et D.-Q. Wei, « MLCDForest: multi-label classification with deep forest in disease prediction for long non-coding RNAs », Briefings in Bioinformatics, vol. 22, no 3, p.104-115, mai 2021, doi: 10.1093/bib/bbaa104.

[62] Y. Guo, F.-L. Chung, G. Li, et L. Zhang, « Multi-Label Bioinformatics Data Classification With Ensemble Embedded Feature Selection », IEEE Access, vol. 7, p. 103863-103875, 2019, doi: 10.1109/ACCESS.2019.2931035.

[63] W. Tang et al., « Identifying multi-functional bioactive peptide functions using multi-label deep learning », Briefings in Bioinformatics, vol. 23, no 1, p. 308-414, janv. 2022, doi: 10.1093/bib/bbab414.

[64] V. Thumuluri, J. J. Almagro Armenteros, A. R. Johansen, H. Nielsen, et O. Winther, « DeepLoc 2.0: multi-label subcellular localization prediction using protein language models », Nucleic Acids Research, vol. 50, no W1, p. W228-W234, juill. 2022, doi: 10.1093/nar/gkac278.

[65] J.-P. Zhou, L. Chen, et Z.-H. Guo, « iATC-NRAKEL: an efficient multi-label classifier for recognizing anatomical therapeutic chemical classes of drugs », Bioinformatics, vol. 36 (5), p. 1391-1396, 2020, doi: 10.1093/bioinformatics/btz757.

[66] A. Bustos, A. Pertusa, J.-M. Salinas, et M. De La Iglesia-Vayá, « PadChest: A large chest x-ray image dataset with multi-label annotated reports », Medical Image Analysis, vol. 66, p. 101-125, déc. 2020, doi: 10.1016/j.media.2020.101797.

[67] J. Xiao, Y. Bai, A. Yuille, et Z. Zhou, « Delving into Masked Autoencoders for Multi-Label Thorax Disease Classification », in 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, janv. 2023, p. 3577-3589. doi: 10.1109/WACV56688.2023.00358.

[68] P. Schwaller et al., « Mapping the space of chemical reactions using attention-based neural networks », Nature Machine Intelligence, vol. 3, no 2, p. 144-152, janv. 2021, doi: 10.1038/s42256-020-00284-w.

[69] E. Ukwatta et J. Samarabandu, « Vision Based Metal Spectral Analysis Using Multi-label Classification », in 2009 Canadian Conference on Computer and Robot Vision, Kelowna, Canada: IEEE, mai 2009, p. 132-139. doi: 10.1109/CRV.2009.42.

[70] J. Du, Q. Chen, Y. Peng, Y. Xiang, C. Tao, et Z. Lu, « ML-Net: multi-label classification of biomedical texts with deep neural networks », Journal of the American Medical Informatics Association, vol. 26, no 11, p. 1279-1285, nov. 2019, doi: 10.1093/jamia/ocz085.

[71] Y. Li, Y. Shen, C. Yao, et D. Guo, « Quality assessment of herbal medicines based on chemical fingerprints combined with chemometrics approach: A review », Journal of Pharmaceutical and Biomedical Analysis, vol. 185, p. 113215, juin 2020, doi: 10.1016/j.jpba.2020.113215.

[72] Zheng-Jun Zha, Xian-Sheng Hua, Tao Mei, Jingdong Wang, Guo-Jun Qi, et Zengfu Wang, « Joint multi-label multi-instance learning for image classification », in 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, juin 2008, p. 1-8. doi: 10.1109/CVPR.2008.4587384.

[73] C. Sanden et J. Z. Zhang, « Enhancing multi-label music genre classification through ensemble techniques », in Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, Beijing China: ACM, juill. 2011, p. 705-714. doi: 10.1145/2009916.2010011.

[74] V. Chauhan, A. Tiwari, B. Venkata, et V. Naik, « Tackling over-smoothing in multi-label image classification using graphical convolution neural network », Evolving Systems, sept. 2022, doi: 10.1007/s12530-022-09463-z.

[75] S. Guehria, H. Belleili, N. Azizi, et S. Brahim Belhaouari, « “One vs All” Classifier Analysis for Multi-label Movie Genre Classification Using Document Embedding », in Intelligent Systems Design and Applications, Vol 1351, p. 478-487, 2021. doi: 10.1007/978-3-030-71187-0_44.

[76] J. Wang, L. Yang, Z. Huo, W. He, et J. Luo, « Multi-Label Classification of Fundus Images With EfficientNet », IEEE Access, vol. 8, p. 212499-212508, 2020, doi: 10.1109/ACCESS.2020.3040275.

[77] S. Coulibaly, B. Kamsu-Foguem, D. Kamissoko, et D. Traore, « Deep Convolution Neural Network sharing for the multi-label images classification », Machine Learning with Applications, vol. 10, p. 100-122, déc. 2022, doi: 10.1016/j.mlwa.2022.100422.

[78] J. Wu et al., « Multi-Label Active Learning Algorithms for Image Classification: Overview and Future Promise », ACM Computing Surveys, vol. 53, no 2, p. 1-35, mars 2021, doi: 10.1145/3379504.

[79] Z. Zeng, X. Wang, et Y. Chen, « Multimedia annotation via semi-supervised shared-subspace feature selection », Journal of Visual Communication and Image Representation, vol. 48, p. 386-395, oct. 2017, doi: 10.1016/j.jvcir.2017.01.030.

[80] A. Schindler et P. Knees, « Multi-Task Music Representation Learning from Multi-Label Embeddings », in 2019 International Conference on Content-Based Multimedia Indexing (CBMI), Dublin, Ireland: IEEE, sept. 2019, p. 1-6. doi: 10.1109/CBMI.2019.8877462.

[81] B. Kostiuk, Y. M. G. Costa, A. S. Britto, X. Hu, et C. N. Silla, « Multi-label Emotion Classification in Music Videos Using Ensembles of Audio and Video Features », in 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), Portland, nov. 2019, p. 517-523. doi: 10.1109/ICTAI.2019.00078.

[82] X. Li, H. Wu, M. Li, et H. Liu, « Multi-label video classification via coupling attentional multiple instance learning with label relation graph », Pattern Recognition Letters, vol. 156, p. 53-59, avr. 2022, doi: 10.1016/j.patrec.2022.01.003.

[83] Y. Cao, C. Tan, et G. Ji, « A Multi-Label Classification Method for Vehicle Video », Journal on Big Data, vol. 2, no 1, p. 19-31, 2020, doi: 10.32604/jbd.2020.01003.

[84] Y. R. Pandeya, J. You, B. Bhattarai, et J. Lee, « Multi-modal, Multi-task and Multi-label for Music Genre Classification and Emotion Regression », in 2021 International Conference on Information and Communication Technology Convergence (ICTC), Korea, oct. 2021, p. 1042-1045. doi: 10.1109/ICTC52510.2021.9620826.

[85] V. F. López, F. De La Prieta, M. Ogihara, et D. D. Wong, « A model for multi-label classification and ranking of learning objects », Expert Systems with Applications, vol. 39, no 10, p. 8878-8884, août 2012, doi: 10.1016/j.eswa.2012.02.021.

[86] M. Amane, K. Aissaoui, et M. Berrada, « Multi-Label Classification of Learning Objects Using Clustering Algorithms Based on Feature Selection », Inteenational Journal of. Emergence Technology and Learning (IJETL)., vol. 17, no 20, p. 248-260, oct. 2022, doi: 10.3991/ijet.v17i20.32323.

[87] L. Tang et H. Liu, « Relational learning via latent social dimensions », in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, Paris France: ACM, juin 2009, p. 817-826. doi: 10.1145/1557019.1557109.

[88] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, et L. Schmidt-Thieme, « Multi-relational matrix factorization using bayesian personalized ranking for social network data », in Proceedings of 5th ACM international conference on Web search and data mining, Washington, 2012, p. 173-182. doi: 10.1145/2124295.2124317.

[89] S. Peters, Y. Jacob, L. Denoyer, et P. Gallinari, « Iterative Multi-label Multi-relational Classification Algorithm for complex social networks », Social Network Analysis and Mining, vol. 2, no 1, p. 17-29, mars 2012, doi: 10.1007/s13278-011-0034-8.

[90] A. Omar, T. M. Mahmoud, T. Abd-El-Hafeez, et A. Mahfouz, « Multi-label Arabic text classification in Online Social Networks », Information Systems, vol. 100, p. 101785, sept. 2021, doi: 10.1016/j.is.2021.101785.

[91] X. Zhang, W. Li, H. Ying, F. Li, S. Tang, et S. Lu, « Emotion Detection in Online Social Networks: A Multilabel Learning Approach », IEEE Internet Things Journal., vol. 7, no 9, p. 8133-8143, sept. 2020, doi: 10.1109/JIOT.2020.3004376.

[92] S. Xie, C. Hou, H. Yu, Z. Zhang, X. Luo, et N. Zhu, « Multi-label disaster text classification via supervised contrastive learning for social media data », Computers and Electrical Engineering, vol. 104, p. 201-234, déc. 2022, doi: 10.1016/j.compeleceng.2022.108401.

[93] J. M. Moyano, E. L. Gibaja, et S. Ventura, « MLDA: A tool for analyzing multi-label datasets », Knowledge-Based Systems, vol. 121, p. 1-7, avr. 2017, doi: 10.1016/j.knosys.2017.01.018.

[94] J. Read, « Scalable Multi-label Classification », Ph.D Thesis, Waikato University, Hamilton, New Zealand, sept, 2010.

[95] F. Charte, A. Rivera Rivas, M. J. Del Jesus, et F. Herrera, "A First Approach to Deal with Imbalance in Multi-label Datasets", in 8th International Conference on Hybrid Artificial Intelligent Systems - HAIS 2013, Salamanca, Vol 8073, p. 150-160, 2013. doi: 10.1007/978-3-642-40846-5_16.

[96] J. M. M. Murillo, « Multi-label classification models for heterogeneous data: an ensemble-based approach. », PhD thesis, Virginia Commonwealth University, USA, 2020.

[97] L. Chekina, L. Rokach, et B. Shapira, « Meta-learning for Selecting a Multi-label Classification Algorithm », in 2011 IEEE 11th International Conference on Data Mining Workshops, Vancouver, 2011, p. 220-227. doi: 10.1109/ICDMW.2011.118.

[98] L. Jones et D. Rushton, « Optimising geotechnical correlations using Receiver Operating Characteristic (ROC) analysis », Proceedings of the XVII ECSMGE-2019, no Geotechnical Engineering, foundation of the future, p. 1456-1463, 2019, doi: 10.32075/17ECSMGE-2019-0271.

[99] Y. Zhang, Y. Wang, X.-Y. Liu, S. Mi, et M.-L. Zhang, « Large-scale multi-label classification using unknown streaming images », Pattern Recognition, vol. 99, p. 107100, mars 2020, doi: 10.1016/j.patcog.2019.107100.

[100] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, et I. Androutsopoulos, « Large-Scale Multi-Label Text Classification on EU Legislation », Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Italy, p. 6314–6322, 2019, doi: 10.18653/v1/P19-1636.

[101] H. Amazal, M. Ramdani, et M. Kissi, « Distributed multi-Label classification approach for textual Big Data », Mediterranean Telecommunications Journal, vol. 9, no 2, juill. Casablanca, 2019.

[102] H. Amazal, M. Ramdani, et M. Kissi, « Towards a Feature Selection for Multi-label Text Classification in Big Data », in Smart Applications and Data Analysis, Éd., in Communications in Computer and Information Science, vol. 1207, Springer International Publishing, 2020, p. 187-199. doi: 10.1007/978-3-030-45183-7_14.

[103] W. Indyk, T. Kajdanowicz, et P. Kazienko, « Relational large scale multi-label classification method for video categorization », Multimed Tools Appl, vol. 65, no 1, p. 63-74, juill. 2013, doi: 10.1007/s11042-012-1149-2.

104] J. Gonzalez-Lopez, A. Cano, et S. Ventura, « Large-Scale Multi-label Ensemble Learning on Spark », in 2017 IEEE Trustcom/BigDataSE/ICESS, Sydney, Australia: IEEE, août 2017, p. 893-900. doi: 10.1109/Trustcom/BigDataSE/ICESS.2017.328.

[105] S. Biswas et V. Susheela Devi, « Parallelization of Multi-label classification for large data sets », in 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India: IEEE, nov. 2018, p. 2005-2010. doi: 10.1109/SSCI.2018.8628763.

[106] O. Sagi et L. Rokach, « Ensemble learning: A survey », WIREs Data Mining & Knowledge, vol. 8, no 4, p. e1249, juill. 2018, doi: 10.1002/widm.1249.

[107] M. P. Ponti Jr., « Combining Classifiers: From the Creation of Ensembles to the Decision Fusion », in 24th Conference on Graphics, Patterns, and Images Tutorials (ICGPIT), Alagoas, Brazil: IEEE, 2011, p. 1-10. doi: 10.1109/SIBGRAPI-T.2011.9.

[108] S. Guehria, H. Belleili, et N. Azizi, « A Survey on Ensemble Multi-label Classifiers », in Proceedings of the 14th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2022), Éd., in Lecture Notes in Networks and Systems, vol. 648., Switzerland, 2023, p. 100-109. doi: 10.1007/978-3-031-27524-1_11.

[109] L. Breiman, « Bagging predictors », Machine Learning, vol. 24, no 2, p. 123-140, août 1996, doi: 10.1007/BF00058655.

[110] R. E. Schapire, T. Labs, S. Laboratory, et P. Avenue, « A Brief Introduction to Boosting », in proceedings of the 16th international joint conference on Artificial intelligence (IJCAI'99), vol. 2, p. 1401 - 1406, 1999.

[111] L. Breiman, « Random Forests », Machine Learning, vol. 45, no 1, p. 5-32, oct. 2001, doi: 10.1023/A:1010933404324.

[112] D. H. Wolpert, « Stacked Generalization », Neural Networks, vol. (5), p. 241-259, 1992.

[113] S. González, S. García, J. Del Ser, L. Rokach, et F. Herrera, « A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities », Information Fusion, vol. 64, p. 205-237, déc. 2020, doi: 10.1016/j.inffus.2020.07.007.

[114] C. El Morr, M. Jammal, H. Ali-Hassan, et W. El-Hallak, « Boosting and Stacking », in Machine Learning for Practical Decision Making, vol. 334, Springer International Publishing, 2022, p. 431-448. doi: 10.1007/978-3-031-16990-8_15.

[115] A. K. Dasari, S. Kr. Biswas, D. M. Thounaojam, D. Devi, et B. Purkayastha, « Ensemble Learning Techniques and Their Applications: An Overview », in Advances in Cognitive Science and Communications, Éd., in Cognitive Science and Technology, Singapore, 2023, p. 897-912. doi: 10.1007/978-981-19-8086-2_85.

[116] I. D. Mienye et Y. Sun, « A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects », IEEE Access, vol. 10, p. 99129-99149, 2022, doi: 10.1109/ACCESS.2022.3207287.

[117] G. Madjarov, D. Kocev, D. Gjorgjevikj, et S. Džeroski, « An extensive experimental comparison of methods for multi-label learning », Pattern Recognition, vol. 45, no 9, p. 3084-3104, sept. 2012, doi: 10.1016/j.patcog.2012.03.004.

[118] J. Read, B. Pfahringer, G. Holmes, et E. Frank, « Classifier chains for multi-label classification », Machine Learning, vol. 85, no 3, p. 333-359, déc. 2011, doi: 10.1007/s10994-011-5256-5.

[119] G. Tsoumakas, A. Dimou, E. Spyromitros, I. Kompatsiaris, et I. Vlahavas, « Correlation-Based Pruning of Stacked Binary Relevance Models for Multi-Label Learning », Proceedings of the 1st International Workshop on Learning from Multi-Label Data, p. 17, 2009.

[120] D. Kocev, C. Vens, J. Struyf, et S. Džeroski, « Ensembles of Multi-Objective Decision Trees », in Machine Learning: ECML 2007, Éd, in Lecture Notes in Computer Science, vol. 4701. , Berlin,, 2007, p. 624-631. doi: 10.1007/978-3-540-74958-5_61.

[121] G. Nasierding, A. Z. Kouzani, et G. Tsoumakas, « A Triple-Random Ensemble Classification Method for Mining Multi-label Data », in 2010 IEEE International Conference on Data Mining Workshops, Sydney, déc. 2010, p. 49-56. doi: 10.1109/ICDMW.2010.139.

[122] E. Gibaja et S. Ventura, « Multi-label learning: a review of the state of the art and ongoing research: A review on multi-label learning », WIREs Data Mining Knowl Discov, vol. 4, no 6, p. 411-444, nov. 2014, doi: 10.1002/widm.1139.

[123] M. S. Sorower, « A Literature Survey on Algorithms for Multi-label Learning », Computer Science, Corvallis, vol 18, no 1, p. 26, 2010.

[124] D. Ganda et R. Buch, « A Survey on Multi Label Classification », Recent Trends in Programming Languages, vol 5, no 1, p. 6, 2018.

[125] A. Pakrashi, D. Greene, et B. M. Namee, « Benchmarking Multi-label Classification Algorithms », in 24th Irish Conference on Artificial Intelligence and Cognitive Science (AICS'16), Dublin, Ireland, p. 12, 2016.

[126] N. Aljedani, R. Alotaibi, et M. Taileb, « Multi-Label Arabic Text Classification: An Overview », International Journal of Advanced Computer Science and Applications (IJACSA), vol. 11, no 10, 2020, doi: 10.14569/IJACSA.2020.0111086.

[127] J. M. Moyano, E. L. Gibaja, K. J. Cios, et S. Ventura, « Review of ensembles of multi-label classifiers: Models, experimental study and prospects », Information Fusion, vol. 44, p. 33-45, nov. 2018, doi: 10.1016/j.inffus.2017.12.001.

[128] J. M. Moyano, E. L. Gibaja, K. J. Cios, et S. Ventura, « An evolutionary approach to build ensembles of multi-label classifiers », Information Fusion, vol. 50, p. 168-180, oct. 2019, doi: 10.1016/j.inffus.2018.11.013.

[129] F. Herrera, F. Charte, A. J. Rivera, et M. J. del Jesus, « Ensemble-Based Classifiers », in Multilabel Classification, Springer International Publishing, 2016, p. 101-113. doi: 10.1007/978-3-319-41111-8_6.

[130] A. Büyükçakir, H. Bonab, et F. Can, « A Novel Online Stacked Ensemble for Multi-Label Stream Classification », in Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino Italy: ACM, 2018, p. 1063-1072. doi: 10.1145/3269206.3271774.

[131] V. Freitas Rocha, F. M. Varejão, et M. E. V. Segatto, « Ensemble of classifier chains and decision templates for multi-label classification », Knowledge and Information Systems, vol. 64, no 3, p. 643-663, mars 2022, doi: 10.1007/s10115-021-01647-4.

[132] M. A. Tahir, J. Kittler, et A. Bouridane, « Multilabel classification using heterogeneous ensemble of multi-label classifiers », Pattern Recognition Letters, vol. 33, no 5, p. 513-523, avr. 2012, doi: 10.1016/j.patrec.2011.10.019.

[133] R. Alazaidah et F. Kabir, « Trending Challenges in Multi Label Classification », International Journal of Advanced Computer Science and Applications (IJACSA), vol. 7, no 10, 2016, doi: 10.14569/IJACSA.2016.071017.

[134] S. Guehria, H. Belleili, N. Azizi, et D. Zenakhra, « Boosting Multi-Label Classification Performance Through Meta-Model », International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI), vol. 38, no 01, p. 2350033, janv. 2024, doi: 10.1142/S0218001423500337.

[135] A. Mahdavi-Shahri, M. Houshmand, M. Yaghoobi, et M. Jalali, « Applying an ensemble learning method for improving multi-label classification performance », in 2016 2nd International Conference of Signal Processing and Intelligent Systems (ICSPIS), Tehran, Iran: IEEE, déc. 2016, p. 1-6. doi: 10.1109/ICSPIS.2016.7869900.

[136] M. Jethanandani, A. Sharma, T. Perumal, et J.-R. Chang, « Multi-label classification based ensemble learning for human activity recognition in smart home », Internet of Things, vol. 12, p. 100324, déc. 2020, doi: 10.1016/j.iot.2020.100324.

[137] G. Madjarov, D. Gjorgjevikj, et S. Džeroski, « Dual Layer Voting Method for Efficient Multi-label Classification », in Pattern Recognition and Image Analysis, Éd., in Lecture Notes in Computer Science, vol. 6669, Berlin, 2011, p. 232-239. doi: 10.1007/978-3-642-21257-4_29.

[138] Y. Xia, K. Chen, et Y. Yang, « Multi-label classification with weighted classifier selection and stacked ensemble », Information Sciences, vol. 557, p. 421-442, mai 2021, doi: 10.1016/j.ins.2020.06.017.

[139] Chen, W. Weng, S.-X. Wu, B.-H. Chen, Y.-L. Fan, et J.-H. Liu, « An efficient stacking model with label selection for multi-label classification », Applied Intelligence, vol. 51, no 1, p. 308-325, 2021, doi: 10.1007/s10489-020-01807-z.

[140] H. Liu, Z. Wang, et Y. Sun, « Stacking model of multi-label classification based on pruning strategies », Neural Computing & Application, vol. 32, no 22, p. 16763-16774, nov. 2020, doi: 10.1007/s00521-018-3888-0.

[141] W. Weng, C.-L. Chen, S.-X. Wu, Y.-W. Li, et J. Wen, « An Efficient Stacking Model of Multi-Label Classification Based on Pareto Optimum », IEEE Access, vol. 7, p. 127427-127437, 2019, doi: 10.1109/ACCESS.2019.2931451.

[142] E. Loza Mencía et F. Janssen, « Learning rules for multi-label classification: a stacking and a separate-and-conquer approach », Machine Learning, vol. 105, no 1, p. 77-126, oct. 2016, doi: 10.1007/s10994-016-5552-1.

[143] M. Kirchhof, L. Schmid, et C. Reining, « pRSL: Interpretable Multi-label Stacking by Learning Probabilistic Rules », in 37th Conf. Uncertainty in Artificial Intelligence (UAI '2021), Vol. 161, pp. 461–470, 2021.

[144] S. Burkhardt et S. Kramer, « Multi-label classification using stacked hierarchical Dirichlet processes with reduced sampling complexity », Knowledge and Information Systems, vol. 59, no 1, p. 93-115, avr. 2019, doi: 10.1007/s10115-018-1204-z.

[145] V.-L. Nguyen, E. Hüllermeier, M. Rapp, E. Loza Mencía, et J. Fürnkranz, « On Aggregation in Ensembles of Multilabel Classifiers », in Discovery Science, vol. 12323, Éd., in Lecture Notes in Computer Science, vol. 12323. , Cham: Springer International Publishing, 2020, p. 533-547. doi: 10.1007/978-3-030-61527-7_35.

[146] M. R. Choirulfikri, K. M. Lhaksamana, et S. A. Faraby, « A Multi-Label Classification of Al-Quran Verses Using Ensemble Method and Naïve Bayes », Building of Informatics Technology and Science (BITS), vol. 3, no 4, p. 473-479, mars 2022, doi: 10.47065/bits.v3i4.1287.

[147] R. Kustiawan, A. Adiwijaya, et M. D. Purbolaksono, « A Multi-label Classification on Topic of Hadith Verses in Indonesian Translation using CART and Bagging », Jurnal Media Informatika Budidarma (MIB), vol. 6, no 2, p. 868, avr. 2022, doi: 10.30865/mib.v6i2.3787.

[148] X. Zhu, J. Li, J. Ren, J. Wang, et G. Wang, « Dynamic ensemble learning for multi-label classification », Information Sciences, vol. 623, p. 94-111, avr. 2023, doi: 10.1016/j.ins.2022.12.022.

[149] N. K. Mishra, P. K. Himthani, et P. K. Singh, « StaC: Stacked chaining for multi-label classification », Expert Systems with Applications, vol. 219, p. 119699, juin 2023, doi: 10.1016/j.eswa.2023.119699.

[150] X. Li et al., « Weighted multi-label classification model for sentiment analysis of online news », in 2016 International Conference on Big Data and Smart Computing, janv. 2016, p. 215-222. doi: 10.1109/BIGCOMP.2016.7425916.

[151] L. Wang, H. Shen, et H. Tian, « Weighted Ensemble Classification of Multi-label Data Streams », in Advances in Knowledge Discovery and Data Mining, vol. 10235, Éd., in Lecture Notes in Computer Science, vol. 10235. , Cham: Springer International Publishing, 2017, p. 551-562. doi: 10.1007/978-3-319-57529-2_43.

[152] N. Rastin, M. Taheri, et M. Z. Jahromi, « A stacking weighted k-Nearest neighbour with thresholding », Information Sciences, vol. 571, p. 605-622, sept. 2021, doi: 10.1016/j.ins.2021.05.030.

[153] H. Wu, M. Han, Z. Chen, M. Li, et X. Zhang, « A Weighted Ensemble Classification Algorithm Based on Nearest Neighbors for Multi-label Data Stream », ACM Transaction Knowledge Discovery Data, p. 3570960, nov. 2022, doi: 10.1145/3570960.

[154] M. A. Tahir, J. Kittler, et A. Bouridane, « Multilabel classification using heterogeneous ensemble of multi-label classifiers », Pattern Recognition Letters, vol. 33, no 5, p. 513-523, avr. 2012, doi: 10.1016/j.patrec.2011.10.019.

[155] R. E. Schapire et Y. Singer, « BoosTexter: A Boosting-based System for Text Categorization », in Machine Learning, vol 39, p.135–168, 2000.

[156] X. Wang, G.-Z. Li, J.-M. Liu, et R.-W. Zhao, « Multi-label Learning for Protein Subcellular Location Prediction », in 2011 IEEE International Conference on Bioinformatics and Biomedicine, Atlanta, p. 282-285. doi: 10.1109/BIBM.2011.36.

[157] J. P. Pestian et al., « A shared task involving multi-label classification of clinical free text », in Proceedings of the Workshop on Biological Translational and Clinical Language Processing, Prague, 2007, p. 97. doi: 10.3115/1572392.1572411.

[158] H. Shao, G. Li, G. Liu, et Y. Wang, « Symptom selection for multi-label data of inquiry diagnosis in traditional Chinese medicine », Science China Information Sciences, vol. 56, no 5, p. 1-13, mai 2013, doi: 10.1007/s11432-011-4406-5.

[159] E. A. Tanaka, S. R. Nozawa, A. A. Macedo, et J. A. Baranauskas, « A multi-label approach using binary relevance and decision trees applied to functional genomics », Journal of Biomedical Informatics, vol. 54, p. 85-95, avr. 2015, doi: 10.1016/j.jbi.2014.12.011.

[160] J. Du, Q. Chen, Y. Peng, Y. Xiang, C. Tao, et Z. lu, « ML-Net: multi-label classification of biomedical texts with deep neural networks», in Journal of the American Medical Informatics Association, vol 26, n°11, 2019, p.1279–1285, https://doi.org/10.1093/jamia/ocz085

[161] R. Kaur et J. A. Ginige, « Analysing Effectiveness of Multi-Label Classification in Clinical Coding », in Proceedings of the Australasian Computer Science Week Multiconference, Sydney, 2019, p. 1-9, doi: 10.1145/3290688.3290728.

[162] R. Li, W. Liu, Y. Lin, H. Zhao, et C. Zhang, « An Ensemble Multilabel Classification for Disease Risk Prediction », Journal of Healthcare Engineering, vol. 2017, p. 1-10, 2017, doi: 10.1155/2017/8051673.

[163] J. M. Moyano, E. L. Gibaja, K. J. Cios, et S. Ventura, « Review of ensembles of multi-label classifiers: Models, experimental study and prospects », *Information Fusion*, vol. 44, p. 33-45, nov. 2018, doi: 10.1016/j.inffus.2017.12.001.

[164] O. Gharroudi, H. Elghazel, et A. Aussem, « Ensemble Multi-label Classification: A Comparative Study on Threshold Selection and Voting Methods », *in 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, Italy: IEEE, nov. 2015, p. 377-384. doi: 10.1109/ICTAI.2015.64.

[165] Guehria, Habiba Belleili, et Nabiha Azizi, « A Survey on Ensemble Multi-label Classifiers ». *in 14th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2022)*, Lecture Notes in Networks and Systems, USA, vol. 648, p. 100-109, Cham: Springer Nature Switzerland, 2022. doi: 10.1007/978-3-031-27524-1_11.

[166] J. Xu, J. Liu, J. Yin, et C. Sun, « A multi-label feature extraction algorithm via maximizing feature variance and feature-label dependence simultaneously », *Knowledge-Based Systems*, vol. 98, p. 172-184, avr. 2016, doi: 10.1016/j.knosys.2016.01.032.

[167] F. Briggs et al., « The 9th annual MLSP competition: New methods for acoustic classification of multiple simultaneous bird species in a noisy environment », *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, United Kingdom: IEEE, sept. 2013, p. 1-8. doi: 10.1109/MLSP.2013.6661934.

[168] W. Ding et S. Wu, « ABC-based stacking method for multilabel classification », *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 27, no 6, p. 4231-4245, nov. 2019, doi: 10.3906/elk-1902-188.

[169] J. Gonzalez-Lopez, S. Ventura, et A. Cano, « Distributed nearest neighbor classification for large-scale multi-label data on Spark », *Future Generation Computer Systems*, vol. 87, p. 66-82, oct. 2018, doi: 10.1016/j.future.2018.04.094.

[170] Q. Wu, H. Wang, X. Yan, et X. Liu, « MapReduce-based adaptive random forest algorithm for multi-label classification », *Neural Computing & Applications*, vol. 31, no 12, p. 8239-8252, déc. 2019, doi: 10.1007/s00521-018-3900-8.

[171] J. K U et J. M.David, Issues, « Challenges and Solutions : Big Data Mining», *Computer Science and Information Technology*, vol. 4. 2014, p. 140. doi: 10.5121/csit.2014.41311.

[172] Bifet Albert, « Mining big data in real time », *Informatica*, vol. 37, p. 15-20, 2013.

[173] A. Gandomi et M. Haider, « Beyond the hype: Big data concepts, methods, and analytics », *International Journal of Information Management*, vol. 35, no 2, p. 137-144, avr. 2015, doi: 10.1016/j.ijinfomgt.2014.10.007.

[174] M. Ilifi et H. Belghalem, « The Role of Big Data in Avoiding the Banking Default in Algeria (The Possibility of Upgrading the Preventive Centers of the Bank of Algeria as a Source of Big Data) », *in Big Data Analytics*, 2021, p. 173-186. doi: 10.1201/9781003129660-18.

[175] D. Goldston, « Big data: Data wrangling », *Nature*, vol. 455, no 7209, p. 15-15, sept. 2008, doi: 10.1038/455015a.

[176] R. H. Hariri, E. M. Fredericks, et K. M. Bowers, « Uncertainty in big data analytics: survey, opportunities, and challenges », *Journal of Big Data*, vol. 6, no 1, p. 44, juin 2019, doi: 10.1186/s40537-019-0206-3.

[177] I. Taleb, M. A. Serhani, et R. Dssouli, « Big Data Quality: A Survey », in 2018 IEEE *International Congress on Big Data*, San Francisco, CA, USA: IEEE, juill. 2018, p. 166-173. doi: 10.1109/BigDataCongress.2018.00029.

[178] A. Oguntimilehin et O. Ademola, « A Review of Big Data Management, Benefits and Challenges », *Journal of Emerging Trends in Computing and Information Sciences*, vol. 5, p. 433-438, juill. 2014.

[179] R. Rawat et R. Yadav, « Big Data: Big Data Analysis, Issues and Challenges and Technologies », *IOP Conference Series: Materials Science and Engineering*, vol. 1022, no 1, p. 012014, janv. 2021, doi: 10.1088/1757-899X/1022/1/012014.

[180] L. R. Nair, S. D. Shetty, et S. D. Shetty, « Applying Spark based machine learning model on streaming big data for health status prediction », *Computers and Electrical Engineering*, vol. 65, p. 393-399, 2018, doi: 10.1016/j.compeleceng.2017.03.009.

[181] A. Sapountzi et K. E. Psannis, « Big Data Preprocessing: An Application on Online Social Networks », in *Principles of Data Science*, Éd., in Transactions on Computational Science and Computational Intelligence. Cham: Springer International Publishing, 2020, p. 49-78. doi: 10.1007/978-3-030-43981-1_4.

[182] I. Hashem, I. Yaqoob, N. Anuar, S. Mokhtar, A. Gani, et S. Khan, « The rise of "Big Data" on cloud computing: Review and open research issues », *Information Systems*, vol. 47, p. 98-115, juill. 2014, doi: 10.1016/j.is.2014.07.006.

[183] L. T. Mohammed, A. A. AlHabshy, et K. A. ElDahshan, « Big Data Visualization: A Survey », in *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Turkey, 2022, p. 1-12. doi: 10.1109/HORA55278.2022.9799819.

[184] K. Vassakis, E. Petrakis, et I. Kopanakis, « Big Data Analytics: Applications, Prospects and Challenges », in *Mobile Big Data*, Éd., in Lecture Notes on Data Engineering and Communications Technologies, vol. 10, Cham: Springer International Publishing, 2018, p. 3-20. doi: 10.1007/978-3-319-67925-9_1.

[185] L. Abualigah et B. A. Masri, « Advances in MapReduce Big Data Processing: Platform, Tools, and Algorithms », in *Artificial Intelligence and IoT*, vol. 85, Éd., in Studies in Big Data, vol. 85. , Singapore, 2021, p. 105-128. doi: 10.1007/978-981-33-6400-4_6.

[186] P. Kijsanayothin, G. Chalumporn, et R. Hewett, « On using MapReduce to scale algorithms for Big Data analytics: a case study », Journal of Big Data, vol. 6, no 1, p. 105, déc. 2019, doi: 10.1186/s40537-019-0269-1.

[187] I. A. T. Hashem et al., « MapReduce scheduling algorithms: a review », Journal of Supercomputing, vol. 76, no 7, p. 4915-4945, 2020, doi: 10.1007/s11227-018-2719-5.

[188] M. Saadoon, S. H. Ab. Hamid, H. Sofian, H. H. M. Altarturi, Z. H. Azizul, et N. Nasuha, « Fault tolerance in big data storage and processing systems: A review on challenges and solutions », *Ain Shams Engineering Journal*, vol. 13, no 2, p. 101538, mars 2022, doi: 10.1016/j.asej.2021.06.024.

[189] J. Read, « Scalable Multi-label Classification », PhD Thesis, vol. University of Waikato., 2010.