# CHAPITRE VI. Le resultat : Classification multi-label par ensemble pour le suivi médical

## VI.1. Introduction

La classification multi-label fait partie de l’apprentissage automatique où la même instance peut appartenir à plusieurs classes simultanément. Dans ce cas, les classes ne sont pas exclusives. L’une des méthodes de transformation les plus populaires est la méthode "Binary Relevance" (BR) (M.-L. Zhang, 2018), où chaque label génère un problème de classification binaire. En conséquence, un problème de classification multi-label est donc transformé en différentes tâches de classification binaire dont le nombre est égal au nombre de labels. Cependant, un inconvénient majeur de cette méthode est qu’elle gère les labels séparément et qu’elle ignore donc les dépendances entre eux.

Cela peut entraîner certaines conséquences, telles que la dégradation des performances et la perte d’informations importantes. Si la méthode du "Label Powerset" (LP) prend en compte les dépendances existantes entre les labels, son utilisation sur des bases de données caractérisées par un grand nombre de labels conduit à certains inconvénients. En effet, à mesure que le nombre de labels augmente, le nombre de sous-ensembles de labels augmente de façon exponentielle. Afin de surmonter ce problème, la méthode "Radom *k*-label sets" (RAkEL) est introduite (G. Tsoumakas, 2010). Cette méthode considère un certain nombre de sous-ensembles aléatoires de labels de taille *k* et utilise la méthode LP sur chacun de ces sous-ensembles. Avec un nombre approprié de *k*-label sets ou ensemble de labels de taille k, cette méthode a tendance à gérer les corrélations entre les labels.

Dans la littérature, plusieurs études sont présentées sur les méthodes d’ensemble, certains d’entre elles utilisent la corrélation entre les labels, d’autres sont basées sur la sélection de caractéristiques (C. Shi, 2011).

## VI.2. Approche proposée : COLEX

L’apprentissage des relations entre les labels est un problème difficile en classification multi label. En effet, les classes se chevauchent et sont corrélées, comme l’association d’un label a une instance peut fournir des informations sur l’appartenance de cette instance à d’autres labels. En l’absence de relations entre les labels, l’aspect multi-label devient sans intérêt et les données peuvent être traitées par une méthode mono-label sans perte de généralité. Différentes approches ont été proposées dans la littérature pour modéliser la corrélation entre les labels (C. Shi, 2011).

Les relations entre les labels peuvent être considérées sous forme de relations binaires entre chaque paire d’labels, ou bien des relations d’ordre supérieur entre chaque label et les labels restants. Les relations d’ordre supérieur sont plus complexes à représenter que les relations binaires, qui peuvent être mesurées, e.g., par la probabilité conditionnelle d’un label étant donnée l’existence d’une autre.

### VI.2.1. La méthode de RAKEL

On rappelle que la méthode Label Powerset (LP) est une méthode de transformation qui considère chaque sous-ensemble de labels comme une classe unique dans un problème mono-label équivalent. Ainsi, s’il y a *q* labels dans l’espace des labels *Y* du problème multi-label, on peut obtenir un maximum de 2*q* classes dans le problème correspondant en mono-label. Pour une nouvelle instance à classifier, toute méthode de classification mono-label peut être utilisée pour prédire sa classe. La sortie du classifier mono-label est alors remplacée par le sous-ensemble équivalent de labels pour obtenir la sortie du classifier multi-label. La méthode LP offre un moyen simple d’utiliser la méthode mono-label pour résoudre les problèmes de classification multi-label. Cependant, elle présente quelques défauts, en particulier lorsque l’espace des labels contient un nombre important de labels. Dans ce cas, le nombre d’instances associées à chaque classe diminue et rend la méthode mono-label incapable de généraliser correctement pour les nouvelles instances.

### VI.2.2. Le coefficient de corrélation binaire φ

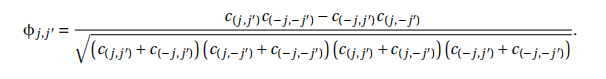
Afin de mieux introduire le coefficient de corrélation binaire, on représente les labels de chaque instance **x***i* de la base de données multi-label par un vecteur binaire **vi** contenant une composante λ*i*,*j* pour chaque label du domaine des labels *Y* qui prend la valeur λ*i*,*j =* 1 si cette instance est labélisée par ` *j* et 0 sinon. Dans le cadre de variables binaires, la mesure de la corrélation entre les variables est généralement effectuée en utilisant le coefficient φ. C’est l’équivalent du coefficient de Pearson appliqué à la référence des variables binaires. Soit *D* une base de données multi-label étiquetée telle que

Équation  : coefficient de corrélation binaire



En utilisant ces notations, le coefficient de corrélation binaire φ entre les labels ` *j* et ` *j0* peut être calculé par :

Équation  : coefficient de corrélation binaire entre label



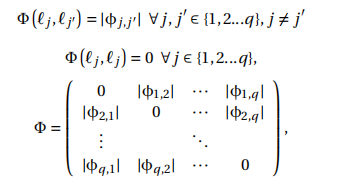
Sous l’hypothèse de variables indépendantes, ce coefficient est égal à zéro. En fait, plus il est proche de zéro, moins les labels sont corrélés. Contrairement au coefficient de Pearson, le coefficient φ n’est pas limité entre -1 et 1. Dans notre travail, nous utilisons ce coefficient comme indicateur de corrélation pour chaque paire de labels possible dans l’espace *Y*, comme il sera montré ci-dessous.

### VI.2.3. Facteur de corrélation globale

La méthode proposée dans ce chapitre consiste à sélectionner un certain nombre de *k* label sets ayant une forte corrélation entre leurs labels. En effet, avec des corrélations élevées, ces labels sont plus susceptibles de tomber ensemble pour les nouvelles instances. En les prenant ensemble, nous évitons d’avoir des prédictions contradictoires par différents prédicteurs. Afin de mesurer la corrélation des labels au sein d’un *k*-label set particulier, nous calculons d’abord la corrélation entre toute les paires de labels dans celui-ci en utilisant le coefficient φ présenté dans la section précédente.

Ce coefficient est égal à zéro lorsqu’il n’y a pas de corrélation entre les deux labels considérés et a une valeur absolue élevée lorsqu’il existe une forte corrélation entre eux. Une matrice de corrélation symétrique Φ est construite à partir de la valeur absolue de ce coefficient de corrélation :

Équation  : Facteur de corrélation globale



## VI.3 Résultats et discussions

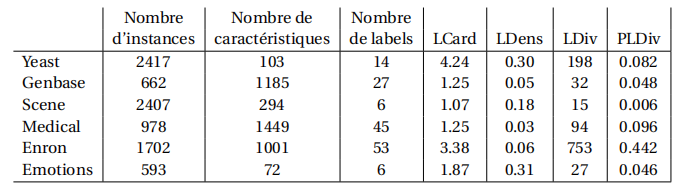
### VI.3.1 Application de la méthode CoLEx à la base de données médicale

Dans le chapitre précédant, une base de données multi-label a été construite en se basant sur les données médicales de MIMIC III. Cette base regroupe un ensemble d’admissions à l’unité de soin intensif d’un hôpital pour lesquels un diagnostic incluant une pathologie cardiaque a été enregistré. En préservant la même notation, cette base de donnée sera notée *H*. Vu que cette base de données a été configurée pour convenir à l’apprentissage du modèle évidentiel, son fort niveau d’indisponibilité de l’ordre de 47% ne présentait pas un inconvénient insurmontable. Cependant, dans une optique d’apprentissage basée sur la méthode SVM, ce n’est plus le cas. En effet, toutes les instances de la base de données doivent avoir toutes leurs caractéristiques disponibles. En éliminant les instances ayant au moins une caractéristique indisponible, la base de donnée se vide littéralement. Alors, pour palier à ce problème, nous avons opté pour une adaptation de la base de donnée qui se fait en deux étapes. La première consiste à sélectionner des caractéristiques selon leurs niveaux de disponibilité respectives et par la suite, la deuxième comprend la suppression de toutes les instances n’ayant pas toutes les caractéristiques disponibles.

### VI.3.2 Illustration des performances de la méthode proposée sur des bases de données réelles multi-label

Afin de valider l’efficacité de l’approche CoLEx proposée, nous considérons différentes bases de données multi-label bien connues que sont Yeast, Genbase, Scene, Medical, Enron et Emotions (J. Alcalá-Fdez, 2011). La première base de données Y*east* contient des informations sur un ensemble de cellules de levure, le but étant de déterminer le site de localisation de chaque cellule. Dans la deuxième, appelée G*enbase*, il s’agit de déterminer la famille protéique de chaque protéine. Le troisième, S*cene*, fait référence à des scènes pour lesquelles plusieurs étiquettes peuvent être associées, comme par exemple, une même photographie peut contenir une montagne, la mer et la plaine. Pour la base de données M*ed i cal*, il s’agit d’associer des labels à un texte libre de rapports médicaux

Tableau  : Caractéristiques des bases de données multi-label du monde réel



Dans ce qui suit, nous présentons les performances de notre méthode et ses avantages par rapport à la méthode RAkEL standard et la méthode exacte pour maximiser la corrélation des label sets. Nous donnons les résultats obtenus pour les deux ensembles de données à des valeurs *k* et *n* différentes. Dans le TABLEAU 2, plusieurs métriques de performances sont affichées pour les deux méthodes RAkEL et celle proposée pour la base de données Genbase avec *n =* 9 label sets et avec *k =* 3 labels par label set. Nous pouvons constater une amélioration significative de l’efficacité globale de la tâche de classification dans notre méthode. Cela montre que notre méthode tire parti des corrélations entre labels et réussit à créer un meilleur groupe de label sets. Dans le TABLEAU 3, nous avons utilisé *n =* 18 label sets au lieu de 9 et nous remarquons toujours de meilleurs résultats en utilisant notre méthode. Au contraire, la méthode RAkEL donne de meilleures performances avec ce nombre plus élevé de label sets. Cela peut s’expliquer par le fait que RAkEL a plus de chance d’attraper les corrélations entre labels aléatoirement lorsqu’il y a plus de label sets.

Tableau  : Comparaison entre RAkEL et COLEX pour la base de données Genbase k = 3 et n = 9

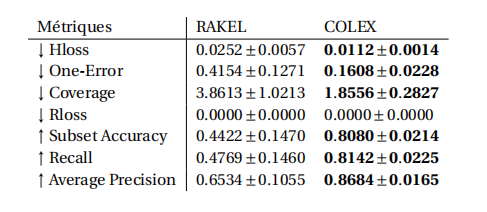
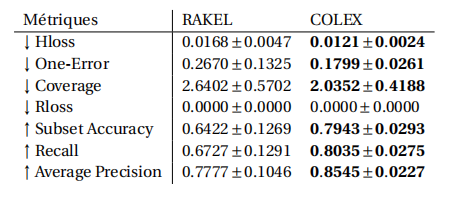


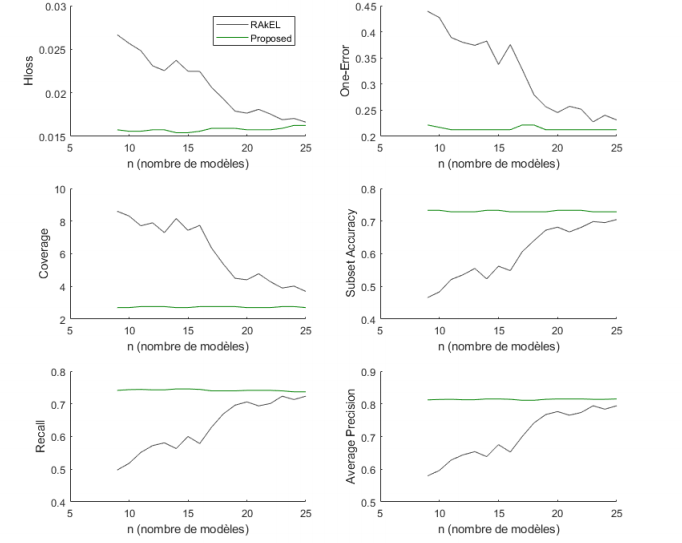
Tableau  : Comparaison entre RAkEL et COLEX pour la base de données Genbase k = 3 et n = 18



### VI.3.4 Impact des paramètres k et n

Dans la FIGURE 6.2, nous montrons 6 graphiques différents représentant chacun une métrique d’évaluation pour la base de données Genbase et pour *k =* 3 par rapport au nombre de labelsets *n*. Ces graphiques montrent une amélioration significative des performances globales de classification lors de l’utilisation de notre méthode au lieu de RAkEL. Nous remarquons sur cette figure que la méthode RAkEL donne de meilleurs performances lorsque *n* augmente, expliquant un besoin d’un grand nombre de labelsets pour atteindre les mêmes performances que notre méthode. Un autre fait est que notre méthode est insensible aux variations de *n*. Par conséquent, elle conserve les mêmes performances pour tous les nombres de labelsets, ce qui nous permet d’utiliser le nombre minimum de labelsets pour réduire le coût en temps de calcul sans sacrifier la valeur des performances. On peut également noter que même avec deux fois plus de labelsets, notre méthode donne quand même de meilleures performances que la méthode RAkEL.

Figure  : – Comparaison entre RAkEL et COLEX pour différentes valeurs de n avec k = 3 pour la base de données Genbase



### VI.3.5. Discussion

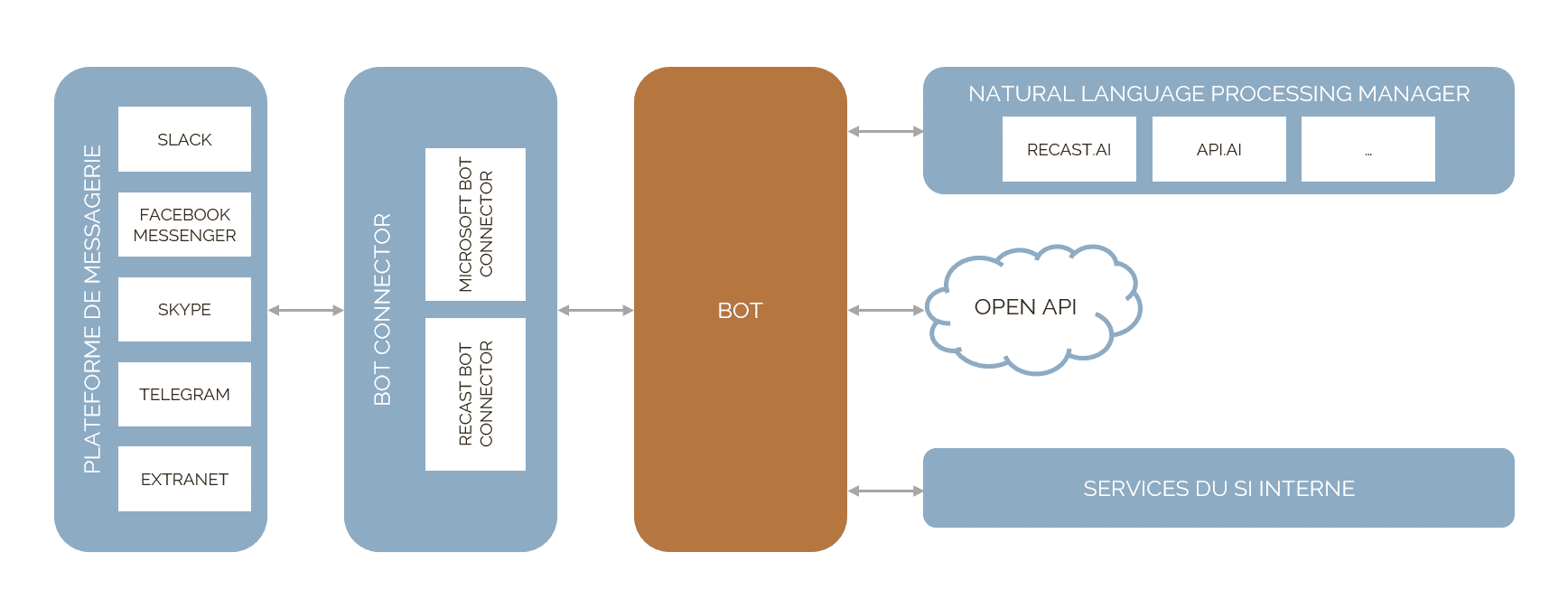
Dans cette premier parti de notre chapitre, une méthode de transformation, appelée CoLEx pour "Correlation based Labelsets by Exchange strategy", et se basant sur l’optimisation du choix des *k*-labelsets en considérant les corrélations existantes entre les labels, a été proposée. Un indicateur de corrélation a été définie, ainsi qu’une procédure d’optimisation qui permet de choisir la meilleure combinaison des *n k*-labelsets. La méthode utilise la stratégie d’échange en maximisant les corrélations entre les labels d’un même *k*-labelsets. Cette stratégie fonctionne en échangeant des labels entre les différents *k*-labelsets. Ensuite, on a entraîné un classifier pour chaque *k*-labelsets en se servant de la méthode LP. On a utilisé tous ces classifieurs entraînés, en plus d’un système de vote pour donner une décision sur chacun des *q* labels. Des simulations sur la base de données médicale du chapitre précédant ainsi que sur des bases de données multi-label du monde réel ont montré l’efficacité de notre méthode par rapport à la méthode RAkEL et à d’autres méthodes de la littérature.

## VI.4. Implementation : technologie, graphiques, interfaces utilisateurs et dataset view

### VI.4.1. Etude des Technologie

Dans le numérique encore plus que dans les environnements traditionnels, la confiance est impérative. C’est elle qui permet de passer outre un degré d’incertitude élevé renforcé par l’absence totale d’interaction humaine et de communication non verbale. Or la confiance est un concept polymorphe qui mobilise à la fois des dimensions cognitives et des dimensions affectives. De plus la confiance ne se décrète pas, mais se construit. Comment donc influencer favorablement la confiance placée dans un outil de e-santé sur tout si ce dernier intègres des aspects technologiques non rependu ? voilà pourquoi nous avons opéré un choix minutieux de technologies et outils a utilisé pour être fiable, viable et vraie dans notre déploiement.

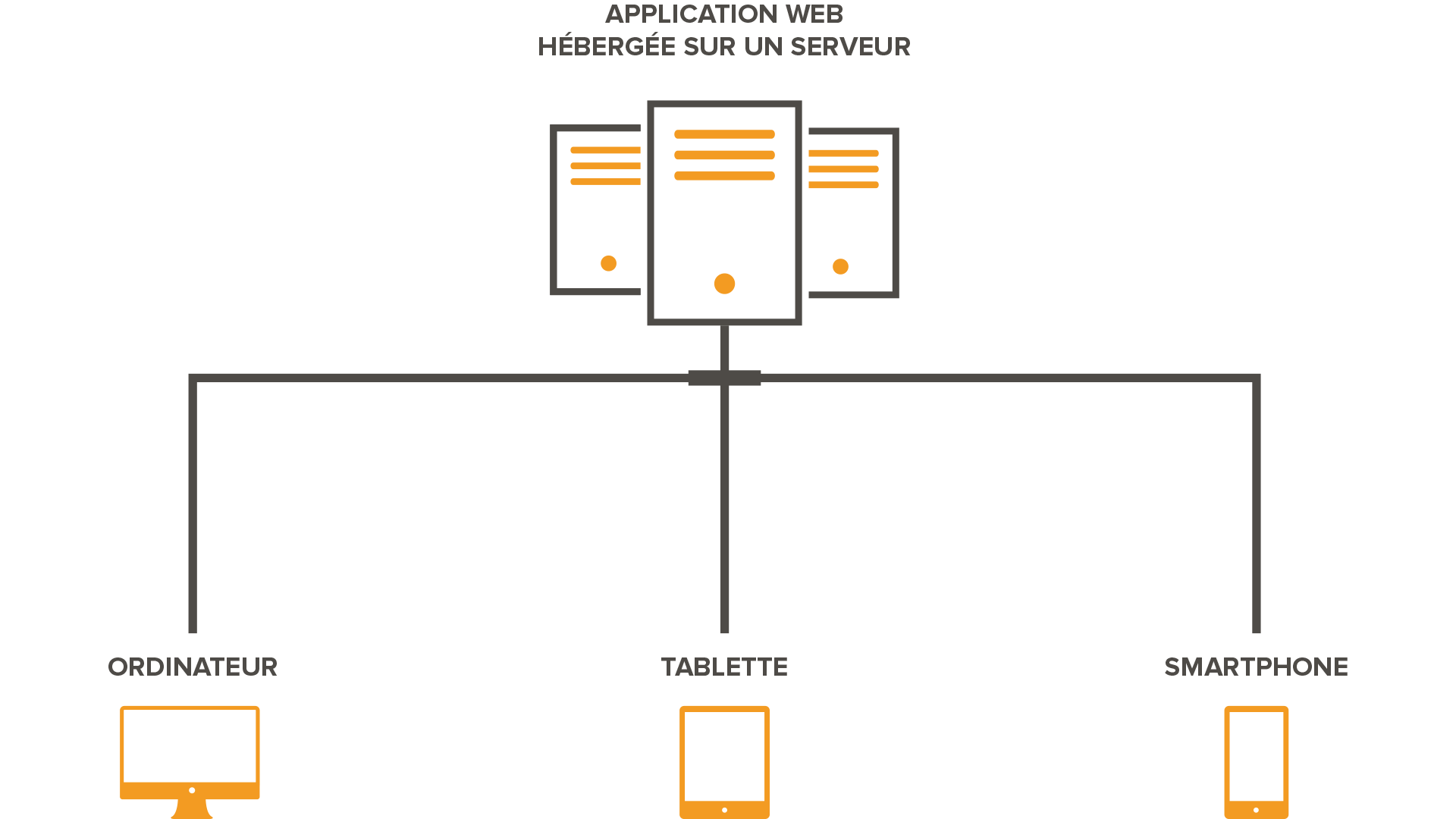
Figure  : Architecture du système AMEN



#### Aspect web

Une **application web** désigne un logiciel applicatif hébergé sur un serveur et accessible via un navigateur web. Contrairement à un logiciel traditionnel, l’utilisateur d’une application web n’a pas besoin de l’installer sur son ordinateur. Il lui suffit de se connecter à l’application à l’aide de son navigateur favori.

Figure  : Application web présentation



##### Outils de développement Front-end :

En développement web, la notion de « front end » fait référence à l’ensemble des éléments visibles et accessibles directement sur un site, en un mot le front-end représente les interfaces utilisateurs. Nous avons utilisé comme outils certain langage descriptifs et de Framework :

* **Framework** :
  + Tailwindcss
  + AlpineJs
  + ViteJs
* **Language:**
  + HTML5
  + CSS

##### Outils de développement back-end :

Cette partie côté serveur fait référence aux fonctions qui s’exécutent sur le serveur, comme l’accès aux données, la gestion des bases de données, les notifications, l’authentification, les fonctions d’application serveur et toutes les autres fonctions. On peut dire que toutes les fonctions auxquelles on accède par le côté client sont exécutées ici.

* **Framework :**
  + FastApi
  + Flask
* **Language :**
  + Python

#### Aspect mobile

On parle d’application mobile quand on fait référence à un programme ou à un logiciel applicatif téléchargeable sur un support mobile (tablette ou smartphone). Lorsqu’elle est téléchargée, elle s’exécute grâce à un système d’exploitation : iOS pour l’environnement Apple et Android pour les supports mobiles de chez Samsung, Xiaomi ou Google.

##### Outils de développement font-end : ui/UX

UI est traduit par User Interface, soit, l’interface utilisateur. L’UI est l’élément faisant le lien entre l’humain et la machine, c’est en quelque sorte la partie visible. C’est ce qui permet à l’utilisateur de naviguer de manière fluide et intuitive sur votre application mobile. Cette étape est en quelque sorte l’organisation des éléments graphiques et textuels afin de proposer une expérience attrayante à l’utilisateur. L’UI répond à des normes techniques (couleurs, taille de texte…).

* **Framework** :
  + Flutter
* **Langages** :
  + Dart

##### Outils de développement back-end

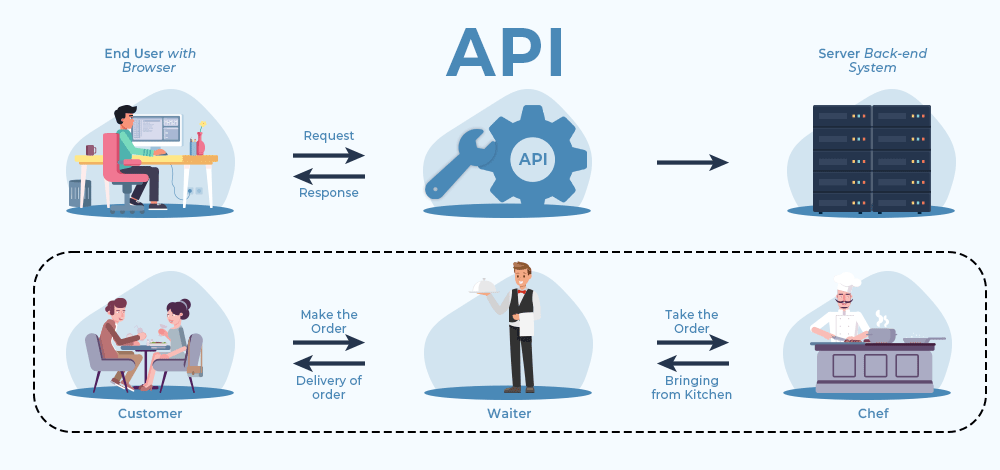
Un Back-end pour le mobile regroupe l’ensemble des supports sur lesquels une appli est présente et coordonne les actions relatives à l’administration et la possibilité de gérer cette dernière, la gestion de droit d’accès et aussi celle du stockage.

* Framework :
  + Flutter
  + Laravel
* Langages :
  + Dart
  + PHP

#### Aspect intercommunication : API

Une API (*application programming interface* ou « interface de programmation d’application ») est une interface logicielle qui permet de « connecter » un logiciel ou un service à un autre logiciel ou service afin d’échanger des données et des fonctionnalités

Figure  : représentation d’un api



* **Stratégie des gestion d’API** :
  + Interne : micro service
  + Partenaire
* **Architecture d’API** :
  + REST
  + Orienté évènement

#### Aspect Traitement et analyse de données

Le traitement des données correspond à l’ensemble des opérations techniques qui permettent d’analyser les informations recueillies pour répondre aux objectifs de l’étude.

* **Framework et Packages** :
  + Tensorflow
  + Pytorch
  + Numpy
  + Opencv
  + Matplotlib
  + Nltk
  + Sklearn
* **Langages** :
  + Python
  + C
* **Gestion de base de données** :
  + MySQL
  + DB Browser for Sqlte

### VI.4.2. Graphique et Dataset