Actes de la 7e conférence nationale sur les Applications Pratiques de l'Intelligence Artificielle

Stephan Brunessaux, Céline Rouveirol

AfIA

Association française
pour l'Intelligence Artificielle

APIA

Conférence Nationale
sur les
Applications Pratiques de l'Intelligence Artificielle

PFIA 2021

Crédit photo : Flicr/xlibber

Table des matières

Éditorial

Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle

L'Intelligence Artificielle poursuit son essor sans précédent dans les laboratoires privés et publics et en entreprise. Les recherches menées ces dernières années ont abouti à des résultats spectaculaires dans certains domaines et des résultats très prometteurs dans d'autres. Aujourd'hui, l'IA se trouve au cœur de nombreuses applications très performantes qui révolutionnent notre vie quotidienne.

Plus que jamais, l'objectif de cette 6ème Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle (APIA 2021) était de donner une tribune dans le cadre de PFIA aux applications concrètes de l'IA qui couronnent de succès l'opérationnalisation de l'IA et des travaux de recherche dans ce domaine. APIA cible des contributions décrivant des applications qui s'appuient sur une ou plusieurs méthodes de l'IA dans tous ses domaines. Cette année, dans les 12 papiers retenus et inclus dans ces actes, les domaines abordés dans les articles retenus sont :

- Agents autonomes et systèmes multi-agents : simulation, planification, décision individuelle ou collective
- Applications de l'Intelligence Artificielle, méthodologie, évaluation
- Apprentissage Automatique (statistique, par renforcement, symbolique, ...)
- Fouille de données, bases de données avancées, web sémantique
- Ingénierie et partage des connaissances
- Intelligence collective, intelligence sociale, réseaux sociaux
- Passage à l'échelle, organisation de systèmes, émergence
- Plates-formes et environnements de développement en IA
- Raisonnement probabiliste et incertain, logique floue
- Raisonnement spatial et temporel, environnements physiques
- Robotique, vision par ordinateur, capteurs intelligents, systèmes physiques
- Systèmes à base de règles, aide à la décision
- Traitement automatique du langage, terminologie, langage naturel contrôlé, explication
- Traitement du signal et de l'image, traitement de la parole

Qu'elles soient industrielles, sociétales, économiques, politiques, environnementales, artistiques ou autres, cette conférence est l'occasion de présenter des applications concrètes et des travaux dont l'objet d'étude adresse des problèmes et/ou des données opérationnelles, Du système de surveillance militaire au système d'aide au diagnostic médical, du climatiseur à l'assistant personnel, du système d'aide à la conduite à l'analyse de données massives, etc., les applications sont nombreuses. Les contributions peuvent illustrer des domaines très divers automobile, robotique, militaire, logistique, télécommunication, finance, domotique, agronomie, réseaux sociaux, risque, grandes masses de données, santé, aide à la personne, jeux vidéo, réalité virtuelle/mixte, musées, EIAH, serious games, récit interactif, et bien d'autres encore...

L'objectif est également de comprendre comment ces applications concrètes font remonter des verrous scientifiques que la communauté des chercheurs en IA doit résoudre pour démocratiser encore davantage son utilisation : l'IA est-elle suffisamment expressive et intelligible pour être utilisée ? Est-elle fiable et robuste ? Est-elle capable de passer à l'échelle ? Quels sont les problèmes éthiques liés à son utilisation ? Comment garantir l'interprétabilité ou l'explicabilité de l'IA ? Ces thèmes sont abordés dans les articles, ainsi que dans la conférence invitée de Stuart Russel, Professeur à l'Université de Californie, Berkeley, intitulée *How not to Destroy the world with AI*.

Afin de favoriser encore davantage l'échange entre chercheurs académiques et industriels et que ces derniers puissent partager leurs expériences et débattre des différents verrous qu'ils rencontrent dans le développement d'applications autour de l'IA, APIA 2021 accueille cette année quatre présentations invitées de membres du Collège Industriel de l'AFIA (ENGIE, JellySmack, Michelin et Société Générale).

Nous tenons à remercier ici tous ceux qui ont participé de près ou de loin au succès de cette conférence, les membres du comité de programme et les auteurs des articles, les conférenciers invités du collège industriel ainsi que Stuart Russel, et plus largement les acteurs de la communauté francophone en IA.

Stéphan BRUNESSAUX, Céline ROUVEIROL

Comité de programme

Président

— Stéphan Brunessaux (Airbus)
— Céline Rouveirol (Université Sorbonne Paris Nord)

Membres

— Florence d'Alché-Buc (Telecom Paris)
— Florence Amardeilh (Elzeard)
— Ghislain Atemezing (Mondeca)
— Alain Berger (Ardans)
— Sandra Bringay (LIRMM)
— Stéphane Canu (INSA Rouen)
— Caroline Chopinaud (Hub France IA)
— Yves Demazeau (LIG)
— Christian de Sainte Marie (IBM)
— Valentina Dragos (Onera)
— Yannick Esteve (LIA)
— Fraçoise Fogelman Soulie (Hub France IA)
— Christophe Guettier (Safran)
— Céline Hudelot (Ecole Centrale Paris)
— Arnault Ioualalen (Numalis)
— Arnault Lallouet (Huawei)
— Christine Largouët (IRISA)
— Vincent Lemaire (Orange Labs)
— Dominique Lenne (Université de Technologie de Compiègne)
— Philippe Leray (Université de Nantes)
— Domitile Lourdeaux (Université de Technologie de Compiègne)
— Sylvain Mahé (EDF Recherche et Développement)
— Juliette Mattioli (Thales)
— Youssed Miloudi (Berger Levrault)
— Philippe Morignot (ASPERTISE)
— Jean Rohmer (ESILV)
— Marie Christine Rousset (Université Grenoble Alpes)
— Frédérique Segond (INRIA)
— Brigitte Trousse (INRIA)

Session 1 – L'IA pour les systèmes critiques ou complexes

Contraintes, objets et ontologies pour la conception de systèmes complexes

Pierre-Alain Yvars[1], Laurent Zimmer[2]

[1] ISAE-Supméca, Laboratoire QUARTZ, EA 7393
[2] Dassault Aviation, Direction de la Prospective

pierre-alain.yvars@supmeca.fr, laurent.zimmer@dassault-aviation.com

Résumé

La plupart des travaux dans le domaine de l'ingénierie des systèmes à base de modèles pour la conception de systèmes techniques consistent à mettre en œuvre des approches orientées solutions. Plusieurs langages de modélisation de systèmes sont disponibles pour représenter des systèmes entièrement définis de plusieurs points de vue. Il est également possible de lier ces descriptions à des outils de simulation ou d'analyse pour évaluer les solutions ainsi décrites. Après avoir étudié les limites de cette façon de concevoir des systèmes, nous proposons dans cet article une approche orientée vers la description du problème de conception à résoudre, à travers un formalisme adapté appelé DEPS. Ce formalisme permet une approche à base de modèles pour la synthèse d'architectures et de systèmes. DEPS (Design Problem Specification) aborde les problèmes de dimensionnement, de configuration, d'allocation de ressources et plus généralement de génération ou de synthèse d'architecture rencontrés dans la conception de systèmes. Les systèmes considérés peuvent être des systèmes physiques, des systèmes à logiciel prépondérant ou des systèmes mixtes (embarqués, mécatroniques, cyber-physiques). Ce langage combine des caractéristiques de modélisation structurelle propres aux langages orientés objet, des capacités de représentation d'ontologie pour l'ingénieur et des caractéristiques de spécification de problème issues de la programmation par contraintes. Nous présentons également une approche intégrée à travers l'environnement DEPS Studio, permettant la modélisation en DEPS, la compilation de modèles et la résolution à l'aide d'un solveur intégré de programmation par contraintes sur des domaines mixtes. Cette intégration permet, entre autres, le développement et le débogage de modèles directement dans DEPS plutôt que dans le langage d'un solveur externe. L'approche est illustrée sur un problème de configuration et de positionnement d'une caméra embarquée.

Mots-clés

Programmation par contraintes, objets, ontologie pour ingénieur, conception de système

Abstract

Most of the work in the field of Model-Based System Engineering for the design of technical systems consists of implementing solution-oriented approaches. Several system modeling languages are available to represent fully defined systems from several points of view. It is also possible to link these descriptions with simulation or analysis tools to evaluate the solutions thus described. After having studied the limits of this way of designing system, we propose in this paper an approach oriented to the description of the design problem to be solved, through an adapted formalism called DEPS. This formalism allows a model-based approach for architecture and system synthesis. DEPS (Design Problem Specification) addresses problems of sizing, configuration, resource allocation and more generally of architecture generation or synthesis encountered in system design. The systems considered can be physical systems, software-intensive systems or mixed systems (embedded, mechatronical, cyber-physical). This language combines structural modeling features specific to object-oriented languages and ontology representation capabilities for engineer, with problem specification features from constraint programming. We also present an integrated approach through the DEPS Studio environment, allowing DEPS modeling, model compilation and solving using an integrated constraint programming solver. This integration allows, among other things, the development and the debugging of models directly in DEPS rather than in the language of an external solver. The approach is illustrated on a problem of configuration and positioning of an embedded stereoscopic camera.

Keywords

Constraint programming, objects, ontology for engineer, system design.

1 Introduction

La majorité des travaux dans le domaine de l'ingénierie des systèmes à base de modèles (MBSE) ont porté sur la représentation de systèmes entièrement définis afin de les vérifier, d'évaluer leurs performances ou de simuler leur comportement. Il s'agit donc le plus souvent d'appliquer une approche analytique à un système connu et non de formaliser le problème d'ingénierie afin de le résoudre [1].

Le développement et la diffusion de langages de modélisation pour l'ingénierie logicielle (UML) puis pour l'ingénierie système (SysML) [2] ont produit des langages de description des systèmes. Les modèles développés sont donc des modèles de définition de la solution envisagée. Ces approches orientées solutions sont nécessaires pour les phases de conception détaillée des systèmes mais sont insuffisantes pour les phases de conception préliminaire dans lesquelles les modèles

détaillés des composants du système sont inutiles puisque le but est d'obtenir le plus rapidement possible une ou plusieurs architectures de système admissibles. Cette limitation avait été soulignée par [3] mais ne semble pas avoir été résolue depuis. [3] suggère que pour faire de réels progrès, il serait nécessaire de disposer d'un véritable langage de modélisation orienté objet pour les problèmes de conception d'ingénierie, qui serait le pendant de ce qu'un langage comme Modelica [4] est pour la simulation. DEPS se propose de combler cette lacune.

1.1 Typologie de problèmes de conception

Les problèmes de conception de systèmes MBSE que nous abordons avec notre approche concernent des systèmes physiques, logiciels ou mixtes (embarqués, mécatroniques, cyber-physiques).

Si nous voulons travailler dans l'espace des problèmes et résoudre les problèmes rencontrés dans la conception de systèmes, nous devrons aborder les types de problèmes suivants :

- Dimensionnement de système : problème pour lequel l'architecture du système est connue mais dont les valeurs de ses paramètres structurels sont inconnues (par exemple, la longueur d'un objet ou sa position). Les inconnues sont des variables souvent continues, parfois discrètes. Les exigences fonctionnelles peuvent être assez complexes et s'exprimer sous forme de relations algébriques linéaires ou non linéaires entre des constantes et des variables.
- Configuration de système : les problèmes de configuration impliquent le choix de composants en fonction d'un ensemble de relations de compatibilité, d'options et de cardinalité. Il s'agit le plus souvent de problèmes à dominante discrète.
- Allocation de ressources : les problèmes d'allocation impliquent l'allocation de ressources physiques aux fonctions du système sur la base d'un ensemble d'exigences fonctionnelles et non fonctionnelles.
- Génération d'architecture : les problèmes d'architecture de système combinent les trois problèmes précédents. Ils sont basés sur une spécification combinant des exigences et des contraintes pour produire des architectures qui répondront à la spécification. On peut également parler de synthèse d'architecture.

1.2 Besoins en représentation et résolution

Pour représenter et résoudre de tels problèmes de conception, nous avons besoin de capacités supplémentaires à celles des CSP pour :

- Exprimer les types manipulés par les ingénieurs.
- Formaliser un système sous-défini ou partiellement défini, c'est-à-dire avec des degrés de liberté tant du point de vue des plages de valeurs possibles (discrètes ou continues) pour les inconnues que du point de vue des parties optionnelles de la structure du système (choix des composants contraints, ...).
- Formaliser les exigences fonctionnelles et non fonctionnelles du système en termes de propriétés déclaratives.
- Résoudre le problème posé en trouvant des valeurs pour les inconnues qui soient compatibles avec les propriétés déclarées.

Il s'agit d'une activité de synthèse complémentaire à l'activité habituelle d'analyse et d'évaluation des performances des MBSE que nous avons appelé MBSS pour Model Based System Synthesis. Le MBSS nécessite de poser des problèmes mixtes portant sur des variables réelles et entières.

1.3 Travaux connexes

Pour modéliser et résoudre des problèmes de conception, le formalisme des problèmes de satisfaction de contraintes (CSP) s'est avéré très utile (voir par exemple [5, 6] en ingénierie mécanique, [7] en ingénierie électrique ou [8] en microélectronique).

Un CSP est défini par un triplet <V, D, C)> tel que [11] :

- $V = \{v_1, v_2, ..., v_n\}$ est un ensemble fini de variables que nous appelons variables de contraintes, n étant le nombre entier de variables du problème à résoudre.
- $D = \{d_1, d_2, ..., d_n\}$ est un ensemble fini de domaines de valeurs de variables tel que :

$$\forall i \in \{1,...,n\}\ v_i \in d_i$$

- $C = \{c_1, c_2, ..., c_p\}$ est un ensemble fini de contraintes, p étant un nombre entier quelconque représentant le nombre de contraintes du problème.

$$\forall i \in \{1,...,p\}, \exists ! V_i \subseteq V / c_i\ (V_i)$$

Une contrainte est tout type de relation mathématique (linéaire, quadratique, non linéaire, booléenne...) impliquant au moins une variable. Elle peut être logique, explicite, etc. Les contraintes peuvent être non seulement des équations et des inégalités algébriques mais aussi des contraintes globales [12].

Représenter un problème de conception sous forme de CSP revient à identifier :

- L'ensemble des variables du problème
- L'ensemble des domaines de valeurs possibles pour chaque variable,
- L'ensemble des contraintes du problème.

Cependant, on peut remarquer qu'il s'agit d'une représentation de bas niveau du problème sans la possibilité de représenter explicitement la structure du système physique sous-défini (c'est-à-dire les quantités physiques de l'ingénieur, les relations entre les sous-systèmes, les variabilités structurelles, ...). Si nous utilisons l'analogie avec la programmation informatique, le modèle <V, D, C> est un modèle de conception comparable à l'assembleur pour la programmation. Pour la conception de systèmes complexes, nous avons besoin d'un langage de représentation des problèmes capable de capturer bien plus que de simples variables, domaines et contraintes. En particulier, ce langage doit être capable de modéliser la structure du problème et de manipuler des quantités cardinales et ordinales pertinentes pour les ingénieurs et les experts.

Pour faciliter le développement et l'utilisation des CSP, la communauté scientifique a développé au fil du temps un certain nombre de bibliothèques disponibles telles que Choco [13], IBEX [14] ou RealPaver [15]. Ces bibliothèques sont construites à l'aide de langages de programmation orientés objet comme C++ ou Java. Cependant, il s'agit de langages de programmation (destinés aux programmeurs) et non de langages de modélisation (destinés aux concepteurs et aux experts en ingénierie). Notre objectif est donc de développer un langage de modélisation de haut niveau et un environnement intégré capable non seulement de modéliser et de résoudre des problèmes de conception mais aussi de s'adresser tout particulièrement aux concepteurs. Pour la partie

résolution, ce langage utilisera la programmation par contraintes sur des domaines mixtes.
Du point de vue des langages de modélisation, il faut distinguer les langages de modélisation de systèmes et les langages de modélisation de problèmes. En ce qui concerne l'analyse et la simulation, les langages de modélisation de systèmes sont bien adaptés car ils permettent la représentation d'un système entièrement défini de plusieurs points de vue (structurel, comportemental, sécurité, analyse du cycle de vie...). En bref, seul un système connu peut être simulé ou analysé. En ce qui concerne la recherche d'une solution, nous avons affaire à un problème de synthèse : le système est sous-défini et nous devons disposer d'un formalisme qui nous permette de l'exprimer. [16] soulignent les difficultés d'utiliser des formalismes tels que SysML (System Modeling Language), initialement conçu pour représenter des systèmes totalement définis pour modéliser des problèmes. Ils proposent dans leur travail d'utiliser le formalisme Clafer [17] associé à la bibliothèque de programmation par contraintes Choco [13] pour modéliser et résoudre un problème d'allocation de calculateurs à des tâches embarquées. Néanmoins, Clafer reste un langage dédié à la configuration de lignes de produits logiciels avec des variables et des contraintes discrètes. Le nombre de variantes que l'on peut exprimer est limité et les problèmes de dimensionnement ne peuvent être traités par Clafer. De son coté, [18] propose d'ajouter un premier niveau de variabilité au langage SysML. L'approche est couplée avec la bibliothèque Choco pour résoudre des problèmes de configuration simples. Cependant, depuis ces premiers travaux de recherche, cette initiative n'a pas progressé de manière significative. Les limites actuelles de l'approche sont le faible niveau de variabilité pouvant être pris en compte, l'utilisation d'un solveur traitant essentiellement des contraintes discrètes et un faible couplage entre le formalisme et le solveur. Ce dernier point signifie que le débogage des modèles s'effectue, en cas de bug ou de problème de modélisation, dans le langage hôte du solveur, en l'occurrence Java et non dans le langage SysML. Ces limites ont été soulignées par [19].
Notre approche vise à surmonter les trois principales limitations identifiées dans l'état de l'art en développant :
- Un langage de modélisation de problèmes pour la synthèse de systèmes à base de modèles.
- Un solveur mixte de programmation par contraintes adapté à la résolution de problèmes de conception.
- Une chaîne intégrée permettant la modélisation, la compilation de modèles et la résolution dans un environnement unique.
Ce papier est organisé de la manière suivante : Dans un premier temps, nous présentons notre approche intégrée basée sur un langage de modélisation de problèmes appelé DEPS. Puis nous décrirons l'environnement intégré de modélisation et de résolution par contraintes appelé DEPS Studio. L'ensemble de l'approche sera illustré par une étude de cas portant sur la configuration et le positionnement d'une caméra stéréoscopique. Enfin, nous évoquerons les travaux en cours ainsi que quelques perspectives d'évolution.

2 Un formalisme pour la représentation de problème de conception

DEPS est un langage de modélisation de problèmes. Le projet de recherche a débuté en 2014 [20] et la distribution et l'évolution du langage sont maintenant soutenues par l'association DEPS Link [21]. Sa grammaire est définie avec une notation BNF « context free ».
DEPS combine certaines caractéristiques de modélisation structurelle propres aux langages orientés objet ainsi que des capacités de description d'ontologie pour l'ingénieur avec des caractéristiques de spécification de problèmes issues de la programmation par contraintes. Aux premiers ont été empruntées les caractéristiques de structuration et d'abstraction qui permettent de représenter les composants et l'architecture (éventuellement partielle) du système étudié. Aux seconds, la possibilité de décrire les quantités ordinales et cardinales utilisées par l'ingénieur (courant, tension, longueur, angle, …). Aux troisièmes ont été empruntés les concepts logico-mathématiques nécessaires à la résolution des problèmes de l'ingénieur. Cette combinaison de paradigmes déclaratifs aboutit à un langage déclaratif qui permet de modéliser l'organisation des systèmes étudiés et les propriétés qui les régissent.

2.1 Types et ontologie pour ingénieur

Les types de données de base utilisés pour le calcul sont :
- Les entiers et les réels,
- Les intervalles entiers, intervalles réels, domaines énumérés entiers et domaines énumérés réels.

En DEPS, les constantes et les variables sont associées à des quantités et des types de quantités physiques ou technologiques appelées *Quantity* et QuantityKind qui sont nécessaires en ingénierie des systèmes. Ainsi :
- Un *QuantityKind* porte un type de base (entier ou réel), une limite minimale, une limite maximale ainsi qu'une dimension au sens de l'analyse dimensionnelle [9] de la quantité. Par exemple, L pour une longueur LT^{-1} pour une vitesse ou U pour une quantité sans dimension ;
- Une *Quantity* est définie à partir d'une *QuantityKind*. Le *QuantityKind* porte la dimension au sens de l'analyse dimensionnelle, la *Quantity* porte l'unité.
- Une *QuantityKind* et une *Quantity* peuvent avoir le même nom.

Plus précisément, une Quantity est définie par:
- Une référence à une *QuantityKind* (*Kind*). Par exemple, Real, Integer, Length ;
- Une borne *Min* (resp. *Max*) qui représente la valeur minimale (resp. maximale) qui peut être prise par toute constante ou variable ayant la quantité définie comme type ;
- Une unité de la quantité.

Le domaine (défini par les champs *Min* et *Max*) d'une *Quantity* doit être inclus ou identique au domaine de sa *QuantitKind* de référence. Ainsi plusieurs *Quantity* peuvent faire référence à la même *QuantityKind*. A l'aide de ces deux traits, il est alors possible de définir un début d'ontologie de domaine et d'application pour ingénieur à la manière de [10].

2.2 Structuration objet et modèle

La caractéristique fondamentale du langage est le modèle . Tout modèle est défini à l'aide du mot-clé *Model* suivi de son nom et de sa liste (éventuellement vide) d'arguments. Il

contient dans l'ordre : une zone de déclaration-définition des constantes du modèle (*Constants*), une zone de déclaration des variables du modèle (*Variables*), une zone de déclaration-définition des éléments du modèle (*Elements*) et une zone de définition des propriétés du modèle (*Properties*). La définition d'un modèle DEPS se termine par le mot-clé *End*.
Un modèle est ainsi écrit avec la syntaxe suivante :

Model <Nom du modèle>**(**<liste d'arguments>**)**
Constants <listOfConstantDeclarationAndOrDefinition>
Variables <listOfVariablesDeclaration>
Elements <ListeDesElémentsDeclarationEtOuDeDéfinition>
Properties <ListOf Properties>
End

Le problème à résoudre est exprimé à l'aide du mot-clé *Problem*. Un problème est un modèle sans arguments. Un problème est la racine de l'arbre des éléments qui sont eux-mêmes des instances de modèles.
Une constante est une quantité numérique dont la valeur ne varie pas pendant la durée de vie d'une copie du modèle dans lequel elle est déclarée et définie. Une constante définie est une constante dont la valeur est donnée.
Une variable est une inconnue du modèle. Elle est caractérisée par sa quantité éventuellement limitée à un sous-ensemble de valeurs possibles. Il est important de souligner que les variables portent le caractère sous-défini des modèles DEPS.
Les modèles DEPS peuvent hériter les uns des autres en utilisant le mot-clé *extends*. Il s'agit d'un héritage public et simple : les constantes, variables, éléments et propriétés sont ainsi hérités des modèles des ancêtres. Certains modèles peuvent être abstraits (*abstract*). Un modèle abstrait ne peut pas être instancié mais il peut faire partie d'une hiérarchie de modèles.
Un élément (*Element*) est une instance de modèle. Il peut être passé en argument à un modèle pour représenter une agrégation et doit alors être déclaré dans la zone de champ *Elements* du modèle. Il peut également être créé dans un modèle pour modéliser une composition et doit alors être déclaré et créé dans la zone des éléments du Modèle.
Tous les éléments d'un problème sont organisés en utilisant des relations d'agrégation et de composition formant une structure arborescente. L'accès aux éléments de cette structure est autorisé par l'utilisation d'une notation pointée. Une constante, une variable ou un élément appartenant à différents niveaux de cette structure arborescente peut être désigné et manipulé en utilisant un chemin.
Chaque modèle possède une signature construite récursivement à partir du nom du modèle et de la signature de ses arguments. Un argument de modèle peut-être une valeur numérique, une constante ou une instance d'un autre modèle. Ce mécanisme permet de lever toute ambiguïté lorsque des Modèles portent le même nom mais ont des arguments différents.

2.3 Contraintes et propriétés

Une propriété (*Properties*) est une relation nécessairement respectée par toute instance du modèle qui la contient. Dans la version actuelle de DEPS, les relations algébriques sont des propriétés (égalité et/ou inégalité entre expressions, définition de la valeur d'une expression déclarée relative aux constantes et variables du modèle ou d'un ou plusieurs éléments du modèle. Tous les opérateurs de la norme IEEE754 pour l'arithmétique à virgule flottante sont disponibles. Une propriété est équivalente à une contrainte au sens de la programmation par contraintes.
Une expression déclarée ou nommée (*expr*) pointe vers une expression algébrique et permet de la référencer et de l'utiliser dans de nombreuses propriétés.
Les équations et les inégalités, linéaires ou non linéaires, sont naturellement des propriétés. Mais les propriétés peuvent aussi être des relations dédiées au domaine de la conception. Ce sera par exemple le cas de la contrainte catalogue qui permet de créer des relations définies en extension par une table de tuples.

3 Environnement de modélisation et de résolution

3.1 Les CSP comme machine virtuelle

La résolution d'un problème de conception nécessite la capacité de prendre en compte :

- Les problèmes sous contraints
- Les équations et inégalités algébriques non linéaires sur des domaines mixtes
- D'autres types de relations telles que les tables de valeurs.

Pour ce faire, nous avons développé un solveur à base de contraintes dédié au calcul sur des modèles DEPS structurés. Les méthodes de calcul sont issues des techniques de résolution de CSP [11]. La structure des modèles DEPS est préservée tout au long de la chaîne de compilation jusqu'aux modèles de calcul.
Le solveur implémente une méthode de propagation *HC4 révisée* [22] sur les équations et les inégalités. Initialement prévue pour les domaines continus, nous avons étendu la méthode à quatre types de domaines : intervalles réels ouverts, intervalles entiers, ensembles énumérés de valeurs flottantes et ensembles énumérés de valeurs entières signées. Les réductions sont effectuées directement sur les domaines typés sans revenir aux intervalles réels. L'algorithme de recherche de solution est une méthode de *branch and prune*. Pour l'instant, seules les stratégies classiques Round-robin et First-fail sont implémentées.
Dans le cas d'un problème sur-contraint, un échec peut survenir lors de la première propagation ou après avoir exploré les parties restantes de l'arbre de recherche. Suivant le paradigme CSP, l'échec est interprété comme la preuve qu'il n'existe pas de solution au problème et non comme un échec de l'algorithme de résolution.
L'architecture orientée objet du solveur a été conçue de manière à pouvoir être étendue à d'autres méthodes de propagation et/ou de résolution existantes (box-consistency, méthodes locales, ...).

3.2 DEPS Studio : Un environnement intégré de modélisation et résolution

L'environnement associé au langage DEPS comprend un éditeur de modèle, un gestionnaire de projet, un compilateur et un solveur. L'expérience montre que la première spécification

d'un problème de conception de système n'est jamais la bonne et que de nombreuses erreurs de modélisation ne sont détectables que par le calcul. Nous avons donc décidé de développer et d'intégrer notre propre solveur dans l'environnement de développement afin que la recherche de solutions contribue efficacement à l'ajustement du processus de modélisation du problème. Il s'agit d'une approche de développement rapide de modèles (analogue à une approche RAD) qui, contrairement à une approche de transformation de modèles, réduit le temps d'exécution de la boucle de développement et de mise au point de modèles. Elle permet également de remonter les erreurs jusqu'au bon niveau d'abstraction. DEPS Studio a été développé en *Delphi*. Les couches inférieures du calcul d'intervalle ont été écrites en C++ et utilisent la bibliothèque open source *gaol* [23].
Un problème à résoudre est organisé en un projet. Un projet est constitué de plusieurs paquets (*Package*). Chaque paquet est enregistré dans un fichier ".deps". Les paquets contiennent des modèles, des types de quantités, des quantités et des tables. L'un des paquets doit contenir un modèle particulier sans argument et déclaré comme étant le problème. Ce modèle représente le problème global à résoudre exprimé sous forme de constantes, variables, éléments et propriétés.
L'environnement dispose:
- d'un éditeur multi-modèles pour charger, modifier et sauvegarder les paquets,
- d'un gestionnaire de projet pour charger, modifier et sauvegarder le projet de modélisation d'un problème composé de tous ses packages.
Un projet est défini comme un ensemble de packages. Chaque package suit la structure suivante :

Package <packageName> ;
Uses <ListOfPackageName> ;
<ListOf DEPS Feature>
Avec
<DEPSFeature> : <QuantityKind>
| <Quantity>
| <Table>
| <Model>
| <Problem>

Le compilateur que nous avons développé transforme directement le modèle source DEPS d'un problème de conception en un réseau de propriétés objet associées au modèle <V, D, C> d'un problème de satisfaction de contraintes. Le niveau CSP est ici considéré comme une machine virtuelle. Il s'agit donc d'un compilateur natif, qui n'est pas une surcouche d'un langage de programmation par contraintes. Le compilateur est de type *ahead-of-time*; l'ensemble du réseau est donc généré avant la résolution.
Le typage statique du langage DEPS est exploité par le compilateur via les quantités pour détecter les erreurs de type sur les constantes, variables, éléments et propriétés avant la résolution.
La compilation se fait en deux passes :
- La première passe de compilation vérifie les paquets utilisés par le projet, analyse lexicalement et syntaxiquement leur contenu et crée la hiérarchie des modèles du projet ;
- La deuxième passe crée l'ensemble des éléments qui définissent le problème à partir de la création de l'élément d'instance unique du modèle déclaré comme problème. Les erreurs sont traitées et signalées à l'utilisateur à toutes les étapes de la compilation : vérification du paquetage, analyse lexicale, analyse syntaxique, création de la hiérarchie du modèle, création des éléments sous-définis. Si l'étape de compilation réussit, l'étape de résolution aura pour tâche d'assigner des valeurs aux variables satisfaisant toutes les propriétés/contraintes du problème.

4 Cas d'étude

4.1 Configuration et positionnement d'une caméra embarquée

Ce problème, décrit dans son intégralité dans [1] est celui de la configuration et du positionnement d'une caméra embarquée. Ce système se compose de capteurs (un ou deux) dont le rôle est d'assurer la stabilisation de l'image ainsi que la mise au point automatique. Ces capteurs sont eux-mêmes reliés à des processeurs (CPUs) (un ou deux) par le biais d'un bus numérique vidéo. Ensuite, les processeurs vont traiter l'image et la compresser pour aboutir à la détection d'obstacles et déclencher la prise de décision automatique du véhicule en lui transférant l'information à travers des « Transceivers » (un ou deux). De surcroît, les transceivers sont reliés aux CPUs grâce à un port numérique parallèle. Capteurs, processeur et transceivers ont leurs caractéristiques de coût et de fiabilité disponibles dans des catalogues. Ainsi la caméra est stéréoscopique lorsqu'elle possède deux chaînes capteur-cpu-transceiver ou bien simple dans le cas d'une unique chaîne.
Afin d'obtenir une caméra à la fois performante et à un prix le plus bas possible, il est nécessaire d'optimiser le coût et la fiabilité qui sont les deux paramètres principaux lors de la conception du système. Pour cela, on procède à la minimisation du coût total du système et à la maximisation de la fiabilité. En plus de l'optimisation du coût et de la fiabilité du système, il est également nécessaire d'optimiser la performance du système en la maximisant. Pour ce faire, le positionnement de la caméra dans le véhicule doit être traité. Correctement positionnée, la caméra doit être être capable de détecter un obstacle à la fois très proche et éloigné (cf Figure 1).

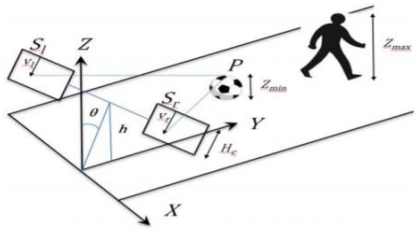


Figure 1 : Configuration géométrique de la caméra

Cette configuration optimale résulte d'une combinaison de paramètres optiques (distance focale des lentilles, position de la caméra) ainsi que du choix des capteurs CMOS. Les capteurs CMOS gauches et droits sont représentés par les deux plans inclinés *Sl* et *Sr*. Le plan horizontal est quant à lui une représentation géométrique de la route et les deux plans verticaux modélisent les obstacles. Le ballon est considéré comme étant l'obstacle proche et le piéton l'obstacle éloigné.

Les capteurs et lentilles sont identiques entre les deux vues. Pour modéliser cette configuration, on se base sur le principe d'optique stéréoscopique permettant de calculer la distance qui nous sépare d'un objet à partir de la comparaison d'image dans les deux vues des caméras. Il s'agit de la disparité *v(y,z)* (cf figure 2)

$$v(Y,Z) = \frac{(v_0 \cos\theta + \alpha \sin\theta)Y + (v_0 \sin\theta - \alpha \cos\theta)(Z-h)}{Y\cos\theta + (Z-h)\sin\theta}$$

Figure 2 : Equation de disparité

Avec :

- v la position de la ligne l dans le plan Sl ou Sr, en pixel
- θ l'angle d'inclinaison de la caméra,
- h la hauteur de la caméra,
- $v0 = H_c/2$ est la projection du centre optique de la caméra, avec Hc la hauteur des capteur identiques en pixel,
- $\alpha = f/tu$ où f est la distance focale des deux lentilles identiques et tu la taille d'un pixel du capteur.

Le système doit être aussi capable de fonctionner dans deux modes : Détection d'un obstacle proche (entre 0 et 2 mètres) et détection d'un obstacle distant (entre 2 et 7 mètres). Pour cela, on statue que l'image d'un obstacle distant sur les capteurs doit avoir une taille minimale qu'on fixe à 50 pixels et l'image de l'obstacle proche doit pouvoir être contenue par les capteurs (*Hc*) Ainsi on a :

- $(obstacle\ distant) > 50\ pixels$
- $(obstacle\ proche) < H_c$

4.2 Description des quantités

Plusieurs quantités vont être manipulées dans ce problème : des références de composants (*Reference*), des longueurs (*Lenght*), hauteurs (*Height*), position (*Position*), angle (*Angle*), des coûts (*Cost*) et des fiabilités (*Reliability*). La figure 3 illustre le cas de la représentation de la quantité *Angle*.

QuantityKind Angle **Type** : real ; **Min** : 0 ; **Max** : 2*Pi; **Dim** : u ; **End**	**Quantity** Angle **Kind** : Angle ; **Min** : 0; **Max** : 2*Pi ; **Unit** : rad ; **End**

Figure 3 : Quantité Angle

4.3 Modélisation des capteurs, processeurs et transceivers

Nous créons tout d'abord un modèle abstrait de Composant qui contient une variable booléenne *isIn* permettant de signifier la présence/absence du composant dans la solution. Tout composant possède une référence (*Ref*), un coût (*Cost*) et une fiabilité (*Rel*) inconnues (cf. Figure 4). Puis nous dérivons des modèles spécialisés de composants pour les capteurs, cpu et transceivers. Chaque modèle dérivé contient une contrainte catalogue permettant d'assujettir les valeurs des variables *Ref*, *Cost* et *Rel* à respecter les triplets d'une table. Ainsi un capteur possède une référence, un coût et une fiabilité définie par une contrainte catalogue sur une table de capteurs (cf Figure 4).

Model Component() **abstract** **Constants** **Variables** isIn : Boolean ; Ref : Reference; Cost : Cost; Rel : Reliability; **Elements** **Properties** **End**	**Model** Sensor() **extends** Component[] **Constants** **Variables** **Elements** **Properties** Catalog([Ref, Cost, Rel], Sensors); **End**

Figure 4 : Modèle de composant et de capteur

Nous définissons également un modèle de paires de composants constitué de un ou deux composants de manière à modéliser les deux types de solutions possibles caméra simple ou caméra stéréoscopique(cf. Figure 5).

Model SensorSet()
Constants
Variables
expr Cost : Cost;
expr Rel : Reliability;
Elements
c1 : Sensor(); c2 : Sensor();
Properties
c1.isIn + c2.isIn >=1; c1.isIn + c2.isIn <= 2;
Cost := c1.isIn*c1.Cost + c2.isIn*c2.Cost;
Rel := max(1, c1.isIn*c1.Rel) * max(1, c2.isIn*c2.Rel);
End

Figure 5 : Modèle de paires de capteurs avec option (un ou deux capteurs)

Enfin, nous définissons un modèle de compatibilité entre capteur, cpu et transceiver (cf. Figure 6). Ce modèle permet de préciser quels sont les triplets (capteur, cpu, transceiver) compatibles ainsi que de préciser qu'au sein d'une même chaîne capteur-cpu-transceiver, si un des composants est présent, les trois le sont et inversement si un des composants est absent, les autres le sont également.

Model Compatibility(s, t, cpu)
Constants
Variables
Elements
s : SensorSet[]; t : TransSet[]; cpu : CPUSet[];
Properties
s.c1.isIn = t.c1.isIn; t.c1.isIn = cpu.c1.isIn;
s.c2.isIn = t.c2.isIn; t.c2.isIn = cpu.c2.isIn;
Catalog([s.c1.Ref, t.c1.Ref, cpu.c1.Ref], CompatibilityTable);
Catalog([s.c2.Ref, t.c2.Ref, cpu.c2.Ref], CompatibilityTable);
End

Figure 6 : Modèle de compatibilité entre deux chaines capteur-cp-transceiver

4.4 Modélisation du système sous défini

```
Model Camera()
Constants
Hc : HeightPix = 480;  v0 : PositionPix = Hc/2 ;
f : Distance = 0.0028;  tu : Height = 0.000015;
alpha : Real = f/tu;
Variables
theta : Angle in [0,1] ;        h : Height in [0.2,1] ;
expr v : PositionPix;
expr TotalCost : Cost;      expr TotalRel : Reliability;
Elements
sSet : SensorSet(); tSet : TransSet(); cpuSet : CPUSet();
compatible : Compatibility(sSet, tSet, cpuSet);
Properties
TotalCost <= 200;
TotalCost := sSet.Cost + tSet.Cost + cpuSet.Cost;
TotalRel :=  sSet.Rel*tSet.Rel*cpuSet.Rel;
End
```

Figure 7 : Modèle de caméra

```
Model Obstacle(Cam,Y)
Constants
Y : Position; (*distance entre l'obstacle et la camera*)
Z : Height = 1.0; default ; (*taille  en hauteur par défaut*)
Variables
Elements
Cam : Camera[];
Properties
Cam.v :=
((Cam.v0*cos(Cam.theta)+Cam.alpha*sin(Cam.theta))*Y+(Ca
m.v0*sin(Cam.theta)-Cam.alpha*cos(Cam.theta))*(Z-
Cam.h))/(Y*cos(Cam.theta)+(Z-Cam.h)*sin(Cam.theta));
End

Model ObstacleProche() extends Obstacle[Camera[], Position]
Constants
Z : Height = 0.22; redefine ; (* ballon de football *)
Variables
Elements
Properties
Cam.v < Cam.Hc;
End

Model ObstacleLoin() extends Obstacle[Camera[], Position]
Constants
Z : Height = 1.8; redefine ; (* piéton *)
Variables
Elements
Properties
Cam.v > 50;
End
```

Figure 8 : Modèles de mode de fonctionnement

A partir de ces modèles il est maintenant possible de définir un modèle de caméra (cf. Figure 7). Le positionnement de la caméra sera déterminé par les variables *theta* et *h*.
Une caméra doit être capable de fonctionner pour détecter des obstacles proches de type ballon de football et lointain de type piéton. Nous définissons pour cela des modes de fonctionnements possibles avec leurs contraintes spécifiques (cf Figure 8).

4.5 Modélisation du problème

```
Problem CameraProblem
Constants
Variables
Elements
Camera : Camera();
Proche1 : ObstacleProche(Camera,0);
Proche2 : ObstacleProche(Camera, 1) ;
Proche3 : ObstacleProche(Camera, 2) ;
Loin1 :  ObstacleLoin(Camera,3);
Loin2 :  ObstacleLoin(Camera,4);
Loin3 :  ObstacleLoin(Camera,5);
Loin4 :  ObstacleLoin(Camera,6);
Loin5 :  ObstacleLoin(Camera,7);
Properties
End
```

Figure 9 : Modèle du problème

Enfin le problème est spécifié en exprimant qu'une caméra doit être configurée et positionnée pour à la fois détecter un obstacle proche de la taille d'un ballon de football ainsi qu'un obstacle lointain de la taille d'un piéton (cf. Figure 9).

4.6 Résultats

Ce problème est résolu quasi instantanément avec le solver de DEPS Studio. Au-delà des performances de résolution, nous nous démarquons de l'approche de [1] sur plusieurs aspects : Là où [1] traite séparément et successivement le problème de configuration et celui du positionnement de la caméra nous proposons un modèle DEPS unique intégrant les deux problèmes en un seul. Là où [1] tente d'utiliser un langage de modélisation de système (SysMl) pour modéliser un problème, nous utilisons un langage de modélisation de problème (DEPS). Là où [1] utilise un solver discret pour résoudre le problème de configuration et un solver continu pour solutionner le problème de positionnement nous utilisons notre solver mixte unique pour les deux problèmes. Le fait de projeter sur un ou deux solveurs externes est une limitation de taille en design. En effet, les résultats sont exprimés à plat dans le formalisme informatique du solveur et non pas dans le langage de modélisation. Enfin là où [1] propose une approche par transformation de modèle plus ou moins automatique pour passer d'un modèle sysml enrichi à un programme pour deux solveurs externes, nous proposons une approche intégrée de modélisation-compilation-résolution. Cette approche permet notamment une mise au point des modèles et une exploitation des résultats de la résolution directement en langage DEPS

5 Conclusion

Dans cet article, nous avons présenté une approche intégrée de modélisation et de résolution de contraintes pour les

problèmes de synthèse de systèmes. Notre proposition permet de surmonter les limitations inhérentes à celle couramment utilisée, qui consiste à coupler des langages de modélisation de systèmes à des solveurs. L'approche est mise en œuvre dans l'environnement DEPS Studio de modélisation et de résolution de problèmes de conception de systèmes exprimés en DEPS. Dans sa version actuelle, cet environnement intègre un gestionnaire de projet, un éditeur multi-modèles, un compilateur natif produisant directement un modèle de résolution et un solveur dédié de programmation par contraintes mixte dont les fonctionnalités répondent aux problèmes rencontrés en ingénierie système. L'ensemble a été illustré sur le cas du problème de configuration et positionnement d'une caméra stéréoscopique embarquée. La combinaison obtenue est actuellement évaluée sur d'autres problèmes de synthèse d'architecture de systèmes [24, 25].
Des évolutions importantes du langage DEPS sont actuellement à l'étude ,notamment l'introduction de collecteurs d'éléments ainsi que la définition de propriétés permettant de formaliser des relations en intension en architecture. Les développements associés portent sur l'ensemble de la chaine de modélisation, de compilation et de résolution. Ces évolutions seront disponibles dans la prochaine version de DEPS Studio.

6 Références

[1]P. Leserf, P. de Saqui-Sannes, and J. Hugues, trade-off analysis for sysml models using decision points and csps, *Software and Systems Modeling*,18(6):3265–3281 2019.

[2] Object Management Group (OMG). *OMG Systems Modeling Language (OMG SysML), Version 1.6*. OMG Document Number formal/19-11-011 (https://www.omg.org/spec/SysML/), 2019

[3] A.A. Shah, *Combining mathematical programming and SysMl for component sizing as applied to hydraulic systems*. Master Thesis, Georgia Tech, 2010.

[4] P. Fritzson, B. Bachmann, K. Moudgalya, F. Casella, B. Lie, J. Kofranek. A. Haumer, C. Nytsch Geusen and L. Vanfretti, *Introduction to Modelica with Examples in Modeling, Technology, and Applications*, First version, June 2017 http://omwebbook.openmodelica.org/

[5] P.A. Yvars, and L. Zimmer, System sizing with a model-based approach: Application to the optimization of a power transmission system. *Mathematical Problems in Engineering*, vol 18, 2018.

[6] Y. Meyer and P.A. Yvars, Optimization of a passive structure for active vibration isolation: an interval computation and constraint propagation based approach. *Engineering Optimization*, 44(12), pp. 1463–1489, 2012.

[7] S. Diampovesa, A. Hubert, P.A. Yvars, Y. Meyer and L. Zimmer, Optimal design for electromagnetic devices: A synthesis approach using intervals and constraint-based methods, *International Journal of Applied Electromagnetics and Mechanics* (IJAEM), 60(1), pp. 35-48, 2019.

[8] K. Kuchcinski, Constraint programming in embedded systems design: Considered helpful, *Microprocessors and Microsystems*, vol 69, pp. 24-34, 2019.

[9] E.S. Taylor, Dimensional Analysis for Engineers, Oxford University Press, 1974

[10]QUDT ontology. https://www.qudt.org

[11]E. Tsang, *Foundations of Constraint Satisfaction*, London and San Diego:Academic Press, 1993.

[12] http://sofdem.github.io/gccat/

[13] X. Lorca, C. Prud'homme, and J.G. Fages, 2014. Choco3 Documentation. TASC, INRIA Rennes, LINA CNRS UMR 6241, COSLING S.A.S., Rennes, France.

[14] http://www.ibex-lib.org/.

[15] L. Granvilliers and F. Benhamou, Algorithm 852: Realpaver: an interval solver using constraint satisfaction techniques, *ACM Transactions on Mathematical Software*, March 2006.

[16] S. Creff, J. Le Noir, E. Lenormand and S. Madelenat, Towards facilities for modeling and synthesis of architectures for resource allocation problem in systems engineering, In Proc of 24th *Systems and Software Product Line Conference*, 2020.

[17] K. Bak, Z. Diskin, M. Antkiewicz, K. Czarnecki, and A. Wasowski, Clafer : Unifying Class and Feature Modeling. *Software and Systems Modeling*, 15(3), July, pp. 811–845, 2016.

[18] P. Leserf, P. de Saqui-Sannes, J. Hugues, and K. Chaaban, Sysml modeling for embedded systems design optimization: A case study. In Proc of 3rd *International Conference on Model-Driven Engineering and Software*, 2015.

[19] A. Shah, C. Paredis, R. Burkhart and D. Schaefer, Combining Mathematical Programming and SysML for Automated Component Sizing of Hydraulic Systems, *JCISE - Journal of Computing and Information Science in Engineering*, 12(4), 2012.

[20] P.A. Yvars and L. Zimmer, DEPS Un langage pour la spécification de problèmes de conception de Systèmes, *proc of the 10th International Conference on Modeling, Optimization & SIMulation (MOSIM)*, France, 2014.

[21] www.depslink.com

[22]F. Benhamou, F. Goualard, L. Granvilliers and J.F. Puget, Revising Hull and Box consistency, *16th International Conference on Logic Programming*, 1993.

[23]F. Goualard, *Gaol 4.2.0 NOT Just Another Interval Arithmetic Library*, 2015, https://frederic.goualard.net/software/gaol-4.2.pdf

[24] L. Zimmer, P.A. Yvars and M. Lafaye, Models of requirements for avionics architecture synthesis: safety, capacity and security, *Proc of the 11th Complex System Design and Management (CSDM) conference, Paris, France, 2020.*

[25]P.A. Yvars, L. Zimmer Synthesis of software architecture for the control of embedded electrical generation and distribution system for aircraft under safety constraints: The case of simple failures, *proc of the 14th International Conference of Industrial Engineering, CIGI-QUALITA 2021, Grenoble, France, 2021*.

Robots Tactiques Polyvalents, vers la réalisation de missions complexes en autonomie

C. Jourdas[1], B. Ricaud[2]

[1] Nexter Robotics
[2] Nexter Systems

c.jourdas@nexter-group.fr, b.ricaud@nexter-group.fr

Résumé

Nexter Robotics et Nexter Systems développent des Robots Tactiques Polyvalents, capables d'assister les combattants sur le terrain dans un ensemble de missions telles que la surveillance de site, l'ouverture d'itinéraire, le transport logistique ou l'extraction de blessés. La réalisation de missions complexes nécessite l'implication de techniques d'intelligence artificielle à différents niveaux, aussi bien dans les modules de perception que pour la prise de décision. Ces développements ont fait l'objet de multiples démonstrations, et de nombreux projets applicatifs, évoqués ici, sont encore à venir. Les travaux réalisés dans le cadre de ces développements sont la base du socle applicatif qui servira pour les robots de tailles équivalentes mais également les robots plus lourds.

Mots-clés

Robotique, autonome, IA décisionnelle, perception.

Abstract

Nexter Robotics and Nexter Systems are currently working on polyvalent tactical robots, able to assist fighters on the field for various missions, such as site surveillance, itinerary opening, logistical transport or the extraction of the wounded. These missions' feasibility depends on the robot's ability to perceive its environment, evolve in it, and decide what to do, thereby requiring AI modules at various levels. These developments have already been applied to numerous demonstrations, and several projects are still to come. These developments will become the base of future products from this weight to heavy robotics platforms.

Keywords

Robotics, autonomy, decisional AI, perception

1 Introduction

Comme l'a indiqué en 2018 le général Charles Beaudouin, alors sous-chef d'état-major chargé des plans et des programmes de l'état-major de l'armée de Terre, l'ambition était de « développer d'ici à 2021 de grands robots, de l'ordre d'une tonne, qui puissent être employés en opération » [1], tandis qu'à l'horizon 2030, les futures générations de robots pourraient être dotées de plus en plus d'intelligence artificielle, pour les rendre les plus autonomes possible.

Les travaux menés sur les Robots Tactiques Polyvalents (RTP) se placent dans ce cadre. Ces plateformes robotisées, allant de 300 kilos à plusieurs tonnes, doivent pouvoir réaliser des missions variées, allant du support logistique à l'extraction de blessés, en passant par l'ouverture d'itinéraire ou la surveillance de site. D'après le cadrage fourni par l'Etat-Major de l'Armée de Terre (EMAT), la finalité de la robotique est en effet d'améliorer la protection du militaire tout en le dégageant des taches pouvant être réalisées par des machines.

Nos travaux visent donc à développer l'autonomie de mobilité et d'observation de ces robots, actuellement téléopérés, pour leur permettre de se rendre rapidement sur le terrain et d'apporter un soutien opérationnel aux troupes.

Les apports potentiels de l'IA dans ce domaine sont nombreux :

- Analyse du terrain : cartographie locale, analyse de texture du sol…
- Mobilité autonome : suivi de convoi, de leader (personne ou véhicule que doit suivre le robot), de chemin…
- Surveillance : détection de menace, d'intention des personnels évoluant autour du robot…

L'introduction d'une autonomie de mobilité sur ces robots évoluant en milieu non structuré apparait en effet comme un besoin incontournable pour pallier les restrictions actuelles de la télé-opération (en particulier au niveau de la communication radio) et pour aboutir à une coopération Homme-Robot efficace, notamment dans le cas de systèmes multi-robots. En effet, l'utilisation de communications radio dans le cadre opérationnel est très cadrée et limitée en termes de transfert de données et de portée.

Afin de pouvoir réaliser ces missions complexes, les RTP sont munis de nombreux capteurs : LIDAR 2D pour la sécurité immédiate des personnels évoluant autour du véhicule, LIDAR 3D 16 ou 32 couches pour une acquisition longue portée autour du véhicule, caméras dans le visible, l'infra-rouge, caméras de profondeur, caméra 360°, etc. La figure 1 présente l'un de ces robots, l'Optio, muni de ces différents capteurs.

Depuis la mise en œuvre des actionneurs jusqu'à l'intégration des robots autonomes dans un peloton, de nombreuses couches algorithmiques intégrant des techniques d'intelligence artificielle interviennent. Au plus bas niveau, nous développons des modules indépendants de mobilité et de perception. Comme l'illustre la figure 2, il convient alors de combiner intelligemment ces briques à plus haut niveau pour pouvoir réaliser des taches complexes à l'échelle d'un robot, voire d'une meute (plateformes hétérogènes) ou d'un essaim (plateformes

homogènes) de robots. Cet article présente ainsi les différentes applications de l'intelligence artificielle à la robotique mobile de défense telles qu'intégrées dans les systèmes de Nexter. Il ne s'agit pas ici d'une revue de l'état de l'art ou d'une synthèse des questions ouvertes mais bien d'un compte rendu des travaux engagés et des problématiques rencontrées dans le cadre de l'automatisation de nos plateformes.

Figure 1 : Vues de l'Optio, sur base de plateforme chenillée Milrem et robotisée par Nexter. On peut distinguer ici les LIDAR 2D et 3D, ainsi que certaines caméras derrière la vitre centrale et sur les côtés.

2 Les modules d'autonomies bas niveau

Les véhicules autonomes font l'objet de très nombreux travaux dans le domaine civil [2], aussi bien au niveau de la perception de l'environnement que pour le suivi de trajectoire. De la même façon, l'ensemble des capteurs disponibles sur les RTP permettent d'envisager à terme une analyse fiable et complète de son environnement et sa mobilité autonome. Nous avons donc choisi dans un premier temps de développer des modules indépendants sur lesquels bâtir une expérience au fur et à mesure des retours des opérationnels suite à nos diverses expérimentations et démonstrations. Néanmoins, si ces modules s'inspirent des travaux effectués dans le civil, les spécificités de notre domaine d'emploi imposent de les adapter.

2.1 Perception et mobilité autonomes

L'objectif de nos travaux n'est pas de faire de la recherche fondamentale sur ces sujets mais d'intégrer et d'adapter les techniques de l'état de l'art aux RTP. Si nous faisons le choix de nous focaliser ici sur les algorithmes d'apprentissage profond (deep learning), les techniques de machine learning ne sont pas pour autant oubliées dans nos travaux. Néanmoins, le domaine du traitement d'image est l'un de ceux à avoir le plus bénéficié des avancées du deep learning ces dernières années. Entre 2011 et 2015, les taux d'erreurs à la compétition de classification d'image ILSVRC ont été divisées par 4 grâce au deep learning, pour atteindre celui d'un humain sur cette même tache [3]. De même, au Caltech Pedestrian Detection Benchmark, les techniques à base d'AdaBoost offrent de bonnes performances mais les meilleurs résultats, sur le jeu de test Caltech [4], sont obtenus par des solutions d'apprentissage profond. Là encore, toutes les pistes sont étudiées en interne mais nous avons choisi de nous intéresser plus particulièrement ici à ces algorithmes.

Dans le domaine de la détection et de la classification d'objets, les réseaux Mask R-CNN [5] et Yolo [6] offrent de très bonnes performances. Si le premier a des temps de traitement moins adaptés aux applications temps réel, le deuxième possède une version embarquée facilement intégrable.

De même, l'estimation de pose a donné lieu à de nombreux travaux tels qu'OpenPose [7] et UniPose [8], susceptibles de servir de base à des algorithmes de détection d'intention ou de commande gestuelle pour les robots. Enfin, le domaine des véhicules autonomes a grandement bénéficié des avancées en segmentation sémantique qui vise à classifier chaque pixel d'une image pour aider, notamment, un véhicule autonome à se déplacer sur une route et à identifier les panneaux de signalisation rencontrés. Dans ce domaine, des réseaux tels que PSPNet [9] et DeepLab [10] ont présenté de très bonnes performances au challenge Pascal VOC 2012.

Bien que quelques solutions académiques évoquées ici nécessiteraient des travaux pour être embarquées sur le calculateur d'un RTP, actuellement de type Jetson TX2 ou Xavier de Nvidia, le problème principal rencontré dans le développement de modules autonomes sur nos RTP n'est pas algorithmique. En effet, la première problématique concerne la donnée et est directement liée aux cas d'usages de ces robots.

Il existe désormais quantité de bases de données civiles annotées (OpenImages de google pour les problématiques de détection, Cityscapes pour la segmentation sémantique et le suivi de route, base de données FLIR pour les traitements Infra-rouge...). Celles-ci ont néanmoins l'inconvénient d'être, pour la plupart, à usage non-commercial uniquement. De plus, ces bases sont rarement adaptées à des cas d'usage militaires. Un soldat en tenue de camouflage est naturellement plus difficile à détecter, et très peu représenté, dans les bases publiques. De même, si l'on prend l'exemple du suivi de chemin : de nombreuses bases de routes urbaines segmentées existent. Néanmoins, les véhicules militaires, et donc les RTP envisagés ici, doivent pouvoir se déplacer dans des environnements non balisés et pour lesquels le chemin à suivre ne sera défini, dans certains cas, que par des ornières faites par le véhicule qui précède. La figure 3 oppose ainsi les données civiles annotées de BDD100K aux environnements d'application cible pour nos robots.

De plus, les robots sont opérés uniquement en extérieur avec un environnement lumineux et météorologique non contrôlé et très variable (fonctionnement sous la pluie, sous la neige, de jour, de nuit…). Aussi, la solution employée doit exploiter au

maximum les données de l'ensemble des capteurs disponibles sur le robot pour assurer la fiabilité du système, et les données publiques acquises par des caméras Infra-Rouge ou de profondeur sont encore plus limitées.

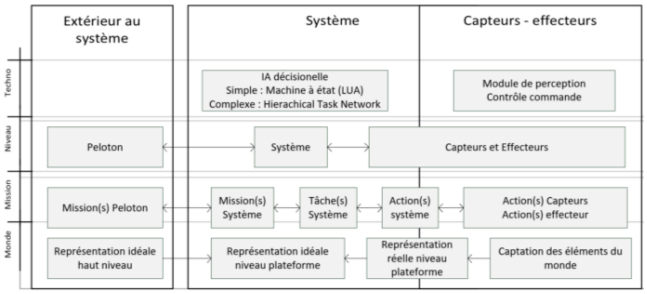


Figure 2 : Différents niveaux d'implication de l'Intelligence Artificielle dans les Robots Tactiques Polyvalents.

Une solution serait bien évidemment d'obtenir des données des théâtres d'opération pour constituer des jeux de données militaires représentatifs. Cependant, la sensibilité de ces données est un frein conséquent, et si la Direction Générale de l'Armement (DGA) s'est muni via le projet Artemis d'une infostructure de stockage et de traitement des données, la source de ces données et leur partage est en cours de définition. Nous avons donc exploré d'autres voies pour réussir à adapter des modules d'autonomie et à les intégrer dans les RTP. Tout ceci dans l'optique de présenter aux opérationnels différents modules lors de démonstrations pour obtenir leurs retours et clarifier leurs besoins.

Figure 3 : A gauche, données civiles annotées de BDD100K, à droite environnements cibles pour nos robots.

2.2 Intérêt de la simulation

Compte-tenu des contraintes militaires, de la diversité des environnements et des cas d'usages, et de la difficulté à obtenir et faire annoter des bases de données réelles, les approches simulées nous semblent indispensables. Pour cette raison, nous avons choisi d'exploiter la simulation en faisant appel à plusieurs outils de simulation externes. Le premier de nos simulateurs, fourni par 4DVirtualiz, offre des environnements que nous pouvons rendre représentatifs de nos cas d'emploi et simule également les données acquises par des capteurs tel que LIDAR, centrale inertielle, GPS. Ce simulateur privilégie la génération de données temps réel à l'identique des interfaces des capteurs réels mais au détriment de la qualité du rendu des images. Néanmoins la qualité est suffisante pour tester les algorithmes de perception sur un robot évoluant dans cet environnement (simulation de type HIL : hardware in the loop). C'est sur ce simulateur que sont testés nos modules de détection d'objet ou de suivi de leader/convoi pour validation et avant intégration sur nos plateformes.

La figure 4 montre ainsi quelques prises de vue de l'environnement de 4DVirtualiz. Ces prises de vue sont faites depuis la vue robot intégrant les résultats d'un module de détection de personne/véhicule/arme et de suivi autonome de véhicule.

La deuxième catégorie de simulateurs utilisés, tels que Unreal Engine et le plugin AirSim, a pour but la création de bases de données d'entrainement pour les algorithmes d'apprentissage. Unreal engine offre l'avantage de pouvoir créer des environnements photo réalistes dans des conditions météo et d'illuminations très divers. Airsim permet de réaliser des enregistrements de bases de données et d'annotation automatique.

Ces bases de données ont notamment été utilisées pour l'entrainement de notre algorithme de suivi de chemin. Ainsi, compte tenu de la multiplicité des environnements ruraux cibles pour nos plateformes, la simulation semble le moyen le plus rapide et efficace d'agrandir nos bases d'apprentissage, bien évidemment en parallèle d'acquisitions et d'étiquetage de données réelles.

La figure 5 présente un exemple de données simulées avec Unreal Engine, nativement annotées, ainsi que les résultats obtenus par le module de segmentation de route sur données réelles.

2.3 Criticité du risque

Les véhicules autonomes civils, objets de dizaines d'années de travaux, sont très encadrés, et les cas d'application de conduite autonome de niveau 3, spécifiquement le maintien sur voie rapide à 60km/h maximum, strictement définis. Dans le cas de véhicules militaires plusieurs problématiques se cumulent, alors que les travaux dans ce domaine ne font que commencer. L'environnement, tout d'abord, qui comme nous l'avons vu est très diversifié et non balisé ; les capteurs, multiples et nécessitant des acquisitions de données spécifiques ; les cas d'usages et les missions visées complexes ; et enfin la criticité du risque.

En effet, comme toute analyse statistique, les apprentissages profonds sont sujets aux erreurs que ce soit parce que la base d'apprentissage n'est pas suffisamment représentative des cas d'usages, ou parce que le réseau n'est pas assez résistant à tous les leurres envisageables. Il est donc nécessaire de bénéficier sur les véhicules robotisés de « garde fous » susceptibles de corriger ces erreurs (par exemple avec des croisements d'algorithmes), ou de les détecter avant qu'elles ne deviennent dangereuses en rendant la main à l'opérateur. Actuellement, deux opérateurs sont nécessaires pour téléopérer un robot et exploiter la charge utile (senseurs ou effecteurs).

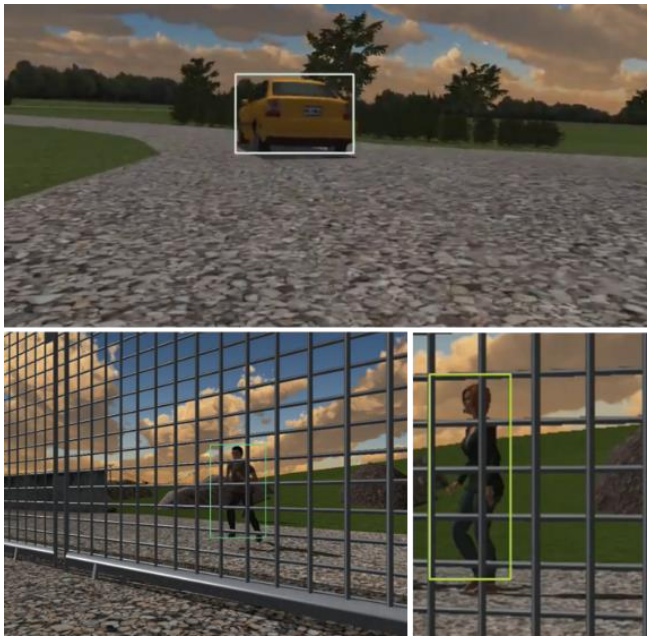


Figure 4 : Exemples de détections et suivi de véhicule depuis le calculateur d'un RTP « nourri » par les capteurs du simulateur de 4DVirtualiz.

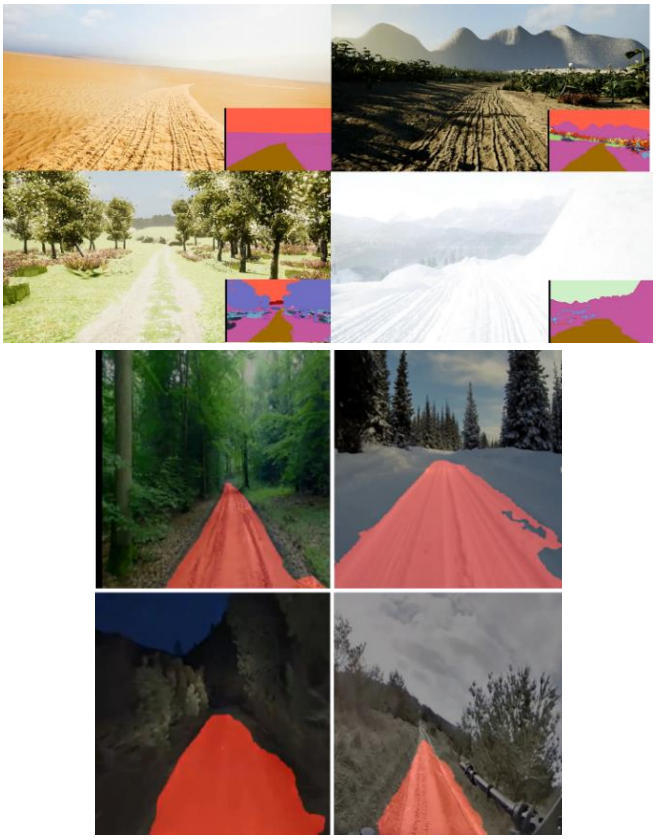


Figure 5 : En haut, des environnements générés avec AirSim et les labels de segmentation sémantique natifs. En bas, les résultats du module de détection de chemin intégrant ces données à l'apprentissage.

L'autonomisation des plateformes doit permettre de réduire ce nombre à un opérateur pour plusieurs fonctions et à terme pour plusieurs robots, de manière à libérer du temps aux combattants pour des tâches à plus forte valeur ajoutée. Néanmoins, même dans le cas d'un opérateur supervisant plusieurs robots, il devra être en mesure de reprendre la main à tout instant en cas de problème.

Disposer d'un indice de confiance pour chaque module autonome est ainsi indispensable pour que le ou les algorithmes indiquent immédiatement à l'opérateur que le robot est dans l'impossibilité de statuer, pour une éventuelle reprise en main. De même, la multiplication de modules sécuritaires bas niveau, notamment de détection d'obstacles, susceptibles de prendre la main sur les fonctions autonomes des couches supérieures, peut permettre de réduire les risques, qu'il s'agisse de risques directs de collision ou de mise en danger des troupes par un mauvais comportement du robot.

On évoquera ici l'exemple du suivi de leader : une première version a été implémentée et testée sur l'une de nos mules (robot de transport de matériel), avec les algorithmes de contrôle associés. Celle-ci suit la trajectoire du chef de groupe en s'adaptant à ses mouvements. Néanmoins, si ce mode de fonctionnement peut suffire dans un cas nominal de déplacement de troupes, cela ne gèrera pas les comportements « tactiques » du leader. Imaginons que celui-ci détecte une menace et se mette à couvert (i.e. se cacher et observer), comment en informer le robot très rapidement et le faire agir en conséquence en se mettant lui aussi en sécurité ? Il serait complexe de devoir reprendre la téléopération dans une situation aussi délicate et des solutions, comme de la détection d'intention et de changement de posture, adaptées au domaine militaire, sont envisagées.

3. Complexification des taches

À terme, l'objectif de l'Armée de Terre est d'employer les robots non pas comme de simples outils à l'instar des robots de déminage, mais comme des robots compagnons évoluant en synergie avec leurs opérateurs humains. En effet, les ressources humaines de l'Armée de Terre sont en flux tendu et la robotique, si l'on souhaite qu'elle soit de masse, ne doit pas accaparer des opérateurs à 100%. Afin de transformer des robots outils actuellement téléopérés 100% du temps en des robots compagnons, il est nécessaire que ces robots gagnent en autonomie. Cette autonomie doit premièrement atteindre un socle de crédibilité exploitant toutes les fonctions de bas niveau présentées ci-avant. Deuxièmement, l'orchestration de ces fonctions élémentaires permettra de réaliser des tâches plus évoluées comme la surveillance d'une zone ou le convoi logistique. In fine, l'objectif est de pouvoir donner des missions à réaliser aux robots comme elles seraient données à des équipages de véhicules blindés ou des groupes d'infanterie. Pour cela les robots doivent avoir une autonomie dite de mission apte à remplir ce type de travail basé sur une interprétation de l'environnement dans lequel ils évoluent. Cette autonomie, en plus de répondre au besoin tactique et s'intégrer au cadre militaire, doit apporter les preuves de fonctionnement pour permettre aux opérateurs et à leurs chefs d'avoir confiance dans les actions entreprises par la machine et d'en garder le contrôle à tout moment pour des questions de

performance d'emploi, mais également d'éthique.

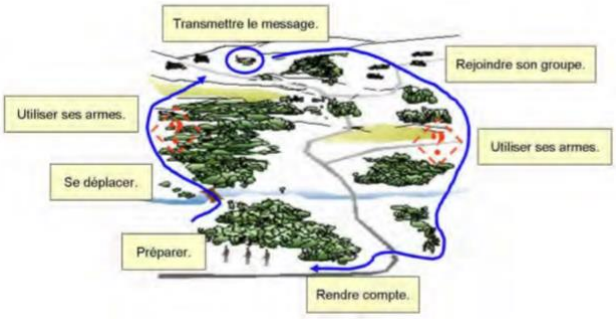


Figure 6 Tâche PORTER UN MESSAGE issue du TTA 150 [11]

Figure 7 Parallèle entre monde simulé et monde réel

3.1 Une autonomie de mission

Comme expliqué précédemment, l'autonomie de mission doit prendre en compte la complexité des missions à réaliser dans ces processus décisionnels tout en permettant aux opérateurs en charge de garder le contrôle dans leur supervision. Pour cela, un des plus gros défis est la représentation du monde exploitée par les robots pour faire ses choix tactiques et évoluer sur le théâtre d'opération de manière sûre et performante.

3.1.1 Processus décisionnels et supervision

A l'instar de la hiérarchie des ressources humaines caractéristiques du monde militaire, les actions réalisées par les soldats, les groupements de soldats et les groupements de groupements de soldats sont elles aussi hiérarchisées.

Ces actions font partie d'un ensemble appelé doctrine, comme le présente le document appelé TTA 150 [11], qui donne la marche à suivre pour réaliser des actions militaires coordonnées et efficaces.

Afin de s'intégrer dans un dispositif militaire, un robot doit rentrer dans le moule de cette doctrine et évoluer sur le théâtre d'opération avec une rigueur et un esprit identique aux soldats qui l'entoureront. Pour cela, les processus décisionnels qu'il doit employer sont nécessairement complexes de par l'ensemble des choix qu'il doit réaliser et les différents horizons de temps qu'il doit considérer, mais doivent rester hiérarchisés.

En effet, cette hiérarchie n'est pas seulement nécessaire au respect de l'esprit de la mission, mais a surtout pour objectif de permettre aux opérateurs et aux chefs de comprendre et de suivre les processus décisionnels des robots en fonctionnement sur les théâtres d'opération avec une logique identique à celle employée par les soldats dans le même espace-temps. Le but est de pourvoir répondre à ces 3 questions : que fait le robot, dans quel but le fait-il et que fera-t-il ensuite ? Sans pouvoir répondre à ces questions, les opérateurs et les chefs qui les emploient ne pourront pas avoir une confiance suffisante dans les actions du robot pour lui déléguer des tâches automatiques.

Dans ce cadre, Nexter travaille avec des opérationnels pour identifier les besoins précis en information qui leur sont nécessaires pour pouvoir déléguer des tâches aux automatismes et les informations essentielles leur permettant de rester toujours conscients de la situation dans laquelle évolue le robot pour pouvoir reprendre la main à n'importe quel moment.

Pour améliorer les performances de l'intelligence artificielle comportementale (« Faire différentes choses sous certaines circonstances » [12]), nous nous intéressons au monde du jeu vidéo qui a une longue expérience et un fort investissement dans ce complexe. En effet des jeux vidéo comme Red Dead Redemption de Rockstar games, ou Cyberpunk 2077 de CD Project ont coûté plusieurs centaines de millions de dollars de développement et ont fait évoluer les méthodes d'intelligence artificielle comportementale du jeu vidéo par les approches qu'ils ont suivies. Comme l'explique K. DILL, l'enjeu est à la fois d'avoir des agents qui soient réactifs, tout en permettant au concepteur d'en garder le control [13]. Un certain nombre d'approches existent pour générer le comportement d'un agent (robot ou personnage), les machines à états, les arbres de décision ou encore les systèmes de planification [12,14]. C'est sur ces derniers que nos travaux se portent. En effet, les systèmes de planification et notamment les réseaux de tâches hiérarchiques (HTN) s'avèrent pertinent pour traduire les processus décisionnels hiérarchisés propres au monde militaire [15], en plus d'être plus évolutifs que d'autres approches [14]. Aussi, Nexter travaille sur l'application de ce type d'algorithmes à des doctrines de cavalerie.

Pour cela la première étape est une phase d'ingénierie des connaissances pour retranscrire numériquement les questions que se posent les soldats afin de réaliser des actions de déplacement ou d'observation. La seconde étape est de retranscrire l'ensemble des actions élémentaires des soldats en tâches systèmes (Figure 2) comme par exemple « PORTER UN MESSAGE » (Figure 6) et de les lier aux questions précédemment identifiées. La troisième étape consiste elle à regrouper et hiérarchiser ces tâches élémentaires pour créer des « missions » dans lesquelles le robot s'intégrera afin de trouver les tâches à réaliser en cohérence avec l'ensemble des tâches réalisées par le groupement dans lequel il s'intègre.

3.1.2 Représentation du monde

Cependant un problème majeur existe : les mondes virtuels dans lesquels évoluent les agents animés par les intelligences comportementales des jeux vidéo sont connus de manière absolue. Le sol est identifié, les obstacles le sont aussi et sont même inventoriés. Ainsi, quand un agent est conçu pour éviter les voitures, mais écraser les buissons, il suffit de lui indiquer que la classe voiture n'est pas franchissable et que la classe buisson l'est. Le moteur de jeu fera le reste [16].

Dans le monde robotique par contre, le premier défi est déjà de réussir à ségréguer les obstacles du sol et le second de les identifier. Ainsi, la représentation du monde robotique ne peut pas, contrairement au jeux vidéo, se baser sur une connaissance absolue du monde, mais doit prendre en compte les biais liés aux algorithmes bas niveau qui interprètent le monde comme ils le perçoivent (et non comme il est) et qui peuvent se tromper.

Figure 8 Expérimentations Magicien d'Oz avec des soldats opérationnels

C'est pour cette raison que les algorithmes à employer dans le cadre de la robotique militaire ne peuvent pas être des applications directes de ceux employés dans les jeux vidéo même s'ils peuvent grandement s'en inspirer. C'est aussi pour cela que le grand défi de l'intelligence comportementale est l'interprétation faite par les algorithmes du monde qui entoure le robot et de la situation tactique qui évolue en permanence.

3.2 Interprétation

L'interprétation est à considérer au sens large et l'intelligence artificielle employée pour celle-ci est à considérer de manière globale sans faire référence à une technologie en particulier.

Différents niveaux d'interprétation seront nécessaires pour avoir une vue éclairée de l'état du monde dans lequel évoluera le robot [17] (Figure 9).

3.2.1 Tâches système de mobilité

La principale problématique de la mobilité pour des systèmes employés au sein d'une force d'intervention militaire est que cette mobilité soit tactiquement pertinente. L'exemple à éviter est qu'à terme le robot emprunte la route la plus simple ou évidente le mettant aux vues de l'ennemis et le rendant vulnérable, ou pire mettant en danger ses opérateurs ou les soldats qui l'entoureront.

Au plus bas niveau de la mobilité, l'interprétation nécessaire est très faible voire nulle pour réaliser du contrôle commande. Par exemple pour éviter un obstacle, seule une nouvelle trajectoire locale va devoir être identifiée avec éventuellement une réduction de vitesse. Cependant, cet obstacle va devoir préalablement être détecté et classifié afin d'identifier s'il faut l'éviter et comment. Pour cela, des briques de perception d'environnement au profit de la mobilité doivent être employées pour différencier un obstacle de la route ou du chemin sur lequel se déplace le robot, et donc d'interpréter les données capteurs (des images par exemple, Figure 9) pour dissocier ces composantes (modules bas niveau). Au niveau supérieur (modules haut niveau), le robot devra interpréter son environnement pour identifier les différentes composantes « terrain » (ce qui est une forêt, une plaine, un terrain boueux) qui serviront de données d'entrée pour la réalisation des choix tactiques pertinents de la mobilité tactique. En effet, comment optimiser mon déplacement en optimisant à la fois ma couverture, ma capacité d'observation, ma vitesse et le potentiel du robot ?

Ainsi, ces différents niveaux ne vont pas utiliser les mêmes méthodes et algorithmes d'intelligence artificielle, mais vont devoir fonctionner ensemble pour que la mobilité soit tactiquement pertinente.

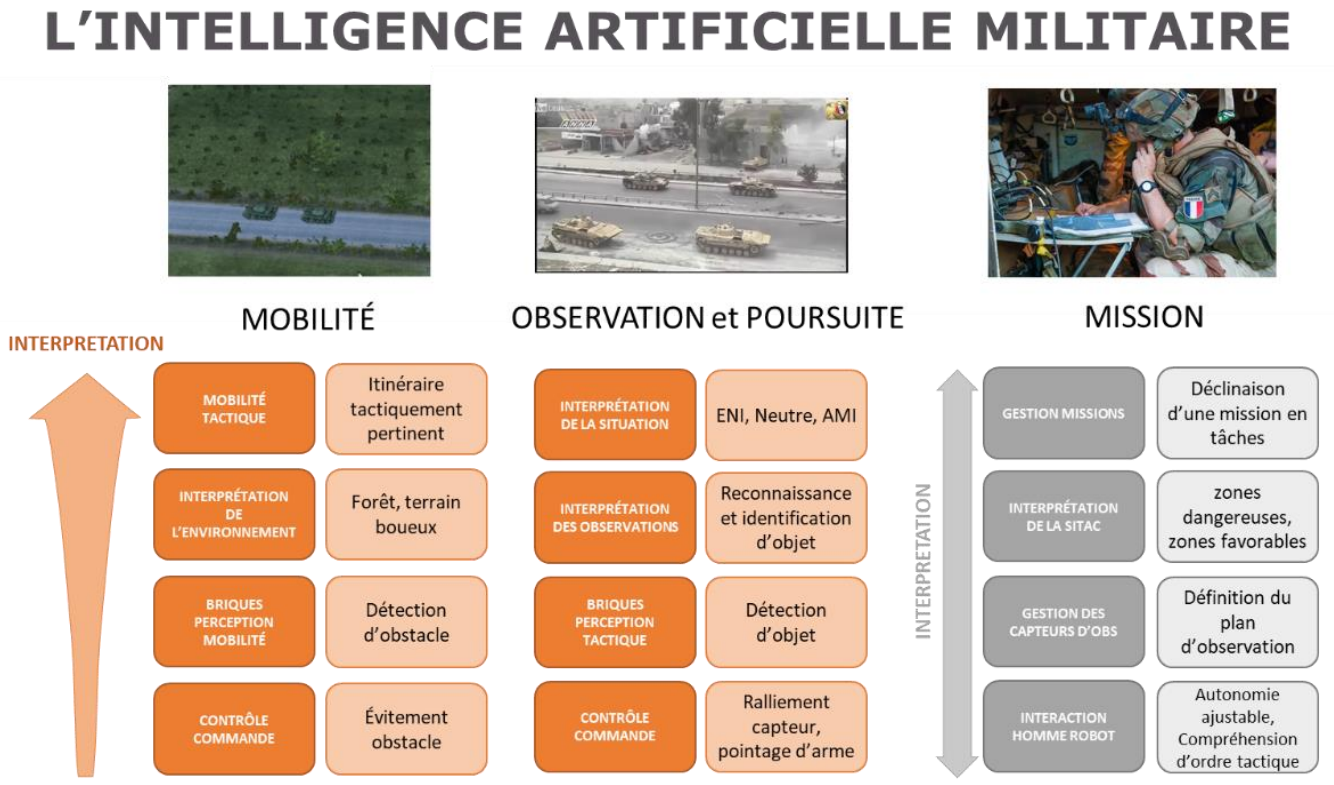


Figure 9 Les différents niveaux d'interprétation d'une Intelligence Artificielle militaire

3.2.1 Tâches système d'observation et de poursuite

Pour ce qui est de l'observation et de la poursuite l'enjeu se trouve dans la prise en compte de situations tactiques très complexes, comme par exemple des civils se déplaçant au milieu d'un dispositif ennemi. Dans cet exemple il ne faut pas que le système remonte à l'opérateur que l'entité civile (par exemple un bus) est une menace à traiter.

Au même titre que pour la mobilité le contrôle commande demandera un niveau d'interprétation bas. Par exemple, pour rallier un capteur ou un système de pointage sur une menace ou sur une cible. En effet, le ralliement se fera toujours sur une position absolue, cependant cette position se basera sur un état du monde permettant de sur des éléments interprétés tel que « l'entrée de l'entrepôt à 9h » ou « au pied de l'arbre entre l'église et le champ de maïs ».

En plus de permettre le ralliement automatique sur des objets interprétés dans le monde, l'interprétation de l'environnement va être nécessaire à l'analyse et au traitement de la menace par les opérateurs supervisant les actions du robot. Premièrement, il est nécessaire que le système puisse détecter des objets dans une scène balayée et extraire (détecter) des éléments pertinents. Comment interpréter qu'un objet est pertinent et doit être remonté à l'opérateur par rapport à un autre ? Par exemple, le garage à 9h n'est peut-être pas aussi pertinent que la grange à 10h et ce car l'ennemi que je cherche n'a pas le gabarit lui permettant de tenir dans le garage.

Deuxièmement, une fois qu'un objet est détecté, la reconnaissance et l'identification de celui-ci vont être primordiales à son traitement afin d'accélérer la prise de décision de l'opérateur. En reprenant l'exemple fournit au début du paragraphe, en plus de détecter un objet en mouvement qui peut être intéressant, le système doit également reconnaitre que c'est un bus et identifier que c'est un bus civil et non un bus militaire.

Enfin, une fois que l'objet a été identifié, le système doit réussir à interpréter l'appartenance et l'état de l'objet. Est-ce que ce bus ou ce char d'assaut est un allié, un ennemi ou un neutre et est-il en état de fonctionnement ou est-ce une carcasse ?

3.2.2 Mission système

Toutes ces briques d'interprétation dans les tâches réalisées par le système permettront d'avoir une interaction Homme-Machine allégée, simplifiée et naturelle. Il est nécessaire que le système et l'opérateur soient en collaboration. Pour ce faire, le système devra répondre comme le ferait un opérateur. Sinon, l'opérateur sera noyé sous les informations de mobilité et d'observation à trier et sa concentration sur les tâches de commandement/décision sera impactée au détriment des chances de réussite de la mission. En effet, dans le domaine militaire terrestre, la durée d'une action de feu est de l'ordre de la seconde ou de la dizaine de secondes, aussi un opérateur surchargé d'informations ne sera plus en mesure de réagir dans les temps si lui ou les robots dont il a la charge sont pris à partie.

Dans ce cadre, l'interprétation dans les missions attribuées au système est à considérer à tous les niveaux de manière équivalente.

Premièrement, pour ce qui est du partage de tâches entre l'opérateur et le robot, le système va devoir gérer une autonomie dite ajustable permettant un changement dynamique du niveau d'intervention de l'opérateur en passant de la téléopération à de l'autonomie complète en quelques secondes.

Deuxièmement, la machine devra comprendre les ordres tactiques en provenance de son opérateur sans que celui-ci ai à utiliser un lexique particulier. A l'inverse, le robot devra générer la synthèse de comptes rendus intelligibles par l'opérateur.

Troisièmement, pour les missions d'observation, il sera nécessaire de gérer les différents capteurs afin d'optimiser les zones d'observation en tenant compte de l'environnement. Pour cela, l'interprétation de la situation tactique sera nécessaire afin d'identifier les zones dangereuses, les zones d'où l'ennemi peut déboucher, etc, et en conclure sur l'attribution des capteurs aux différentes zones.

Enfin, comme détaillé dans le chapitre 3.1.1, la déclinaison d'une mission en un ensemble de tâches de plus bas niveau requerra la prise en compte de tous les niveaux présentés jusqu'alors en plus de l'interprétation de la situation tactique amie et ennemie et en tirer des comportements qui ne soient trop stéréotypés, mais qui restent dans l'esprit de la mission.

4 Mises en application

Comme en attestent les déclarations du Général Beaudouin en 2018 [1], les forces armées françaises n'intègrent pas encore de robots téléopérés, et encore moins de plateformes autonomes. En plus des problématiques d'acceptabilité de ces robots par les troupes évoquées au chapitre précédent, il est nécessaire de cerner le besoin des combattants pour ce type de robot.

Nous avons à cette fin une approche très démonstrative et itérative. Chaque module autonome est intégré et présenté dans un cadre précis de manière à avoir des retours directs des opérationnels et à orienter notre feuille de route en fonction. Cette approche permet aussi de capitaliser sur chacun des modules disponibles et d'augmenter petit à petit l'autonomie de nos plateformes.

Nous participons ainsi à de nombreux projets et expérimentations.

4.1 Développements et expérimentations passées

Nous avons participé en 2019 à une expérimentation de surveillance de site sensible. Pour celle-ci, le robot devait effectuer, en autonomie, des patrouilles sur le site, aller se recharger seul au besoin, et observer le périmètre en remontant des alertes en cas d'intrusion. Dans ce cadre, la relocalisation dans l'environnement et la planification de trajectoire s'effectuaient grâce à des algorithmes de localisation et de cartographie simultanés (SLAM) LIDAR, tandis que les modules de détection et de suivi de personnes/véhicules/armes s'appuyaient sur des techniques d'apprentissage profond, adaptées au cas d'usage via une base de données ciblée. Une machine à états développée en LUA permettait en outre de planifier l'enchainement des modules autonomes « élémentaires » pour réaliser l'ensemble de la mission.

Début 2021, dans le cadre du projet MC^2 avec VEDECOM, ARQUUS et CNIM pour le compte du Battle Lab Terre, s'est déroulée une démonstration de convoi autonome de véhicules militaires en mode « multi-Follow me » [18]. Nexter a ainsi

robotisé deux plateformes intégrant un module de suivi de véhicule basé sur les données LIDAR et celles des caméras dans le visible. Il est en effet important que les informations remontées par les différents capteurs puissent être utilisées indépendamment. Ceci pour le cas où l'un d'eux ne serait pas disponible ou non utilisable, par exemple si la mise en marche du LIDAR rendait le véhicule trop visible de l'ennemi. De plus, ces informations doivent pouvoir être fusionnées de façon à maximiser les performances.

4.2 Projets en cours et futurs

Dans le courant du premier semestre 2021 aura lieu une nouvelle expérimentation de surveillance de site, celle-ci en milieu ouvert non structuré, avec l'adjonction d'un drone embarqué dans le robot. Cette expérimentation s'appuie sur les travaux de 2019 sur la surveillance de site, tout en intégrant les évolutions ayant eu lieu depuis. Des modules de détection, de suivi de leader piéton et véhicule, et de réidentification sont désormais embarqués dans le calculateur du robot pour fonctionner en temps réel.

Nexter participe aussi avec 13 partenaires, au projet iMUGS (integrated Modular Unmanned Ground System) qui a pour but de développer un système terrestre sans pilote normalisé européen. Au cours de ce projet, lancé fin 2020, sera définie une architecture modulaire commune capable d'intégrer, selon les besoins, un ensemble de capteurs et de briques autonomes de perception, d'évolution et/ou de planification.

Un projet lancé fin 2020 dans le cadre du challenge IA, de la DGE et BPI France, vise aussi à développer avec une Start-up, Vidétics, un module de détection de changement de posture, susceptible de poser les prémices d'une détection d'intention adaptée aux comportements des troupes cohabitant avec les RTP.

Un projet RAPID proposé avec Masa Group vise à adapter des techniques d'IA décisionnelles bien établies sur simulateurs à des plateformes réelles, de façon à faire réaliser des missions complexes par des meutes de robots hétérogènes tout en mettant l'accent, au cours des développements, sur l'interprétabilité des comportements observés, et ce afin de faciliter l'acceptabilité de tels systèmes.

Pour finir Nexter, avec d'autres industriels français et allemands conçoit et développera le système Main Ground Combat System (MGCS ou char futur) incluant des véhicules robotisés lourds. Pour préparer ce développement, Nexter est en charge de démonstrateurs technologiques en charge de prouver les capacités d'autonomie notamment sur le plan de la mobilité.

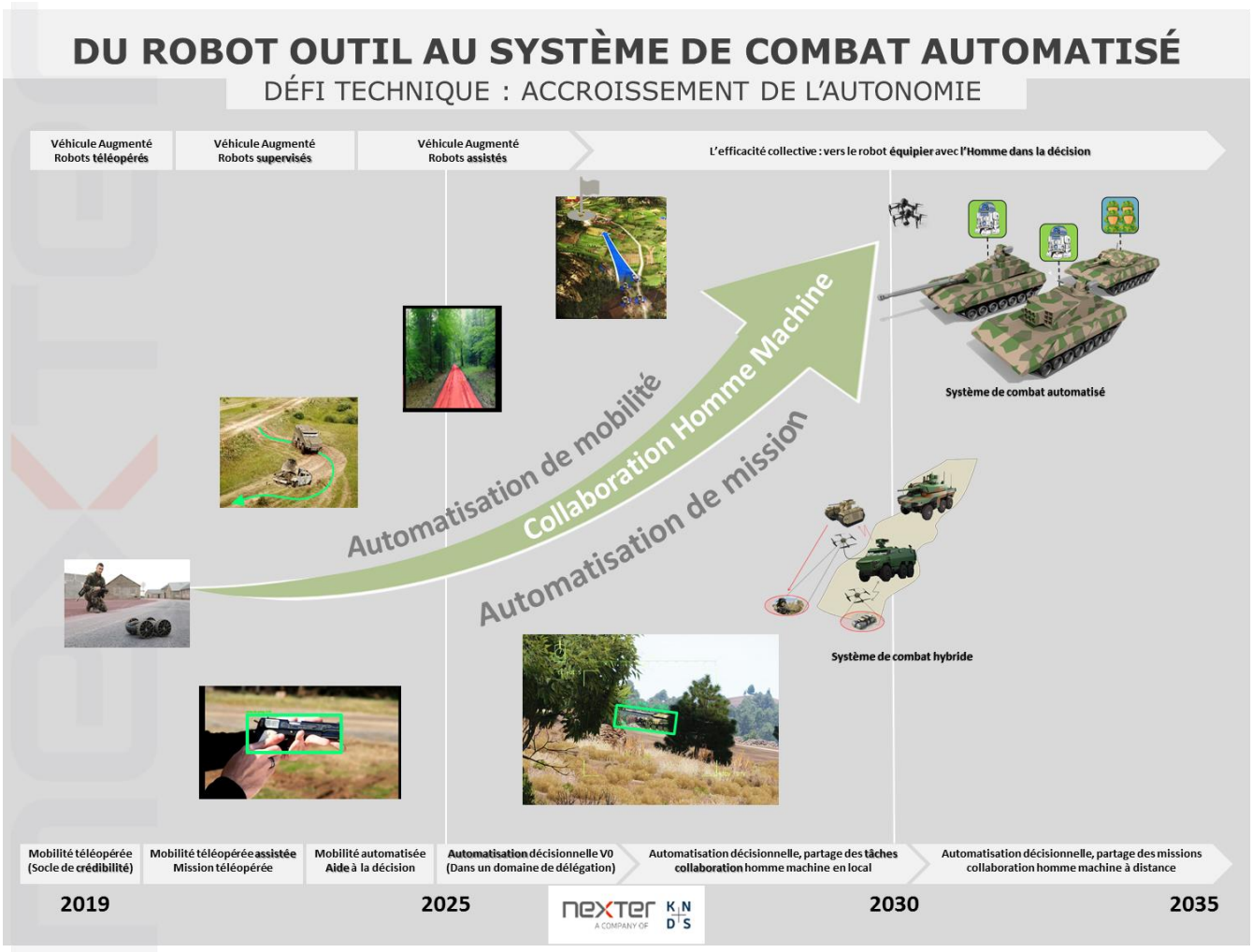


Figure 10 Feuille de route du robot outil au robot compagnon

5 Conclusion

Cet article présente l'application de l'Intelligence Artificielle aux robots tactiques polyvalents. Cette intelligence leur permettra de réaliser des missions plus complexes et d'avoir des capacités à s'adapter aux situations inattendues rencontrées grâce à un socle solide de modules élémentaires. Les développements réalisés dans le cadre de ces robots tactiques polyvalents permettent d'augmenter l'autonomie tactiques des robots de cette gamme mais permettent également de préparer les développements de robots plus lourds tel que le MGCS. En effet, comme le montre la Feuille de route du robot outil au robot compagnon figure 10 les modules développés pour les RTP sont prévus d'être en grande partie réutilisables et/ou transposables à ces robots lourds.

6 Références

[1] C. Beaudouin, *Commission de la défense nationale et des forces armées*, Assemblée Nationale, 16 mai 2018.

[2] F. Rosique, P.J. Navarro, C. Fernandez, A. Padilla, *A Systematic Review of Perception System and Simulators for Autonomous Vehicles Research*, Sensors 2019, 19, 648. https://doi.org/10.3390/s19030648.

[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, *ImageNet Large Scale Visual Recognition Challenge*, Int J Comput Vis 115, 211–252, 2015.

[4] P. Dollar, C. Wojek, B. Schiele, P. Perona, *Pedestrian Detection: A Benchmark*, CVPR 2009.

[5] K. He, G. Gkioxari, P. Dollar, R. Girshick, *Mask R-CNN*, CVPR, arXiv 1703.06870, 2017.

[6] J. Redmon et A. Farhadi, *Yolov3: An Incremental Improvement* , arXiv 1804.02767, 2018.

[7] Z. Cao et al., *OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields*, CVPR, arXiv 1812.08008, 2018.

[8] B. Artacho, A. Savakis, *UniPose: Unified Human Pose Estimation in Single Images and Videos*, CVPR, arXiv 2001.08095, 2020.

[9] H. Zhao, J. Shi, X. Qi, XX. Wang, J. Jia, *Pyramid Scene Parsing network*, CVPR, arXiv 1612.01105, 2017

[10] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation, ECCV, arXiv 1802.02611, 2018

[11] MINArm, DRHAT, *Sous-direction formation écoles, le combat proterre en milieu ouvert*, TTA 150, 2012

[12] Robot Programming: A Practical Guide to Behavior-Based Robotics, Joseph L. Jones, Chapitre 3, 2004

[13] K. Dill, *What Is Game AI?*, 1.2.2 Reactivity, Nondeterminism, and Authorial Control, GAME AI PRO 1, Chap. 1, 2013

[14] Behavior Selection Algorithms: An Overview, Michael Dawe, GAME AI PRO 1, Chap. 1, 2013

[15] Alexandre MENIF, *Planification d'actions hiérarchique pour la simulation tactique*, Chap 4, 2017

[16] Chris Jenner and Sergio Ocio Barriales, *Fast Cars, Big City The AI of Driver San Francisco*, GAME AI PRO 3, Chap. 1, 2017

[17] Robot Programming: A Practical Guide to Behavior-Based Robotics, Joseph L. Jones, Chapitre 1, 2004

[18] *1ère démonstration de suivi en convoi militaire interopérable*, Vedecom, 8 février 2021

Défis ouverts aux systèmes multi-agents dans le cadre des constellations de satellites d'observation de la Terre

Gauthier Picard[1], Clément Caron[2], Jean-Loup Farges[1],
Jonathan Guerra[2], Cédric Pralet[1], Stéphanie Roussel[1]

[1] ONERA DTIS, Université de Toulouse
prenom.nom@onera.fr
[2] Airbus Defence and Space, Toulouse
prenom.nom@airbus.com

Résumé

Dans cet article, nous identifions plusieurs opportunités et défis ouverts aux systèmes multi-agents, suite aux récents développements dans le domaine des constellations de satellites d'observation de la Terre. Nous nous concentrons sur trois catégories de défis relevant de ce domaine : (i) les problèmes de configuration de constellations et de stations au sol utilisées pour les exploiter, potentiellement opérées par différents acteurs, afin d'améliorer les services fournis et leur coordination ; (ii) les problèmes de planification et d'ordonnancement hors ligne, qui consistent à trouver des méthodes de résolution pour planifier les observations et les tâches de chargement de commandes et de déchargement de prises de vue de la constellation ; et (iii) la conception de méthodes opérationnelles en ligne efficaces et réactives pour adapter les plans dans des contextes dynamiques. Étant naturellement distribués et composés de multiples entités et utilisateurs, ces problèmes s'inscrivent clairement dans le paradigme multi-agent, et peuvent représenter des défis scientifiques et techniques pour les chercheurs en intelligence artificielle distribuée et systèmes multi-agents, pour de nombreuses années.

Mots-clés

Satellite, constellation, observation de la Terre, systèmes multi-agents.

Abstract

We identify several challenges and opportunities opened to agent and multiagent systems, following the recent developments in the domain of Earth observation constellations. We focus on three challenge categories that manifest in this field : (i) configuration problems of constellations and ground stations used to operate them, potentially owned by different actors, as to provide better services and coordination ; (ii) offline planning and scheduling problems, which consist in finding solution methods to schedule observation and upload/download tasks over the constellation ; (iii) the design of efficient and reactive online operation methods as to adapt schedules in dynamic settings.Being naturally distributed and composed of multiple entities and users, these problems clearly fit the multiagent paradigm, and may challenge researchers for many years.

Keywords

Satellite, Constellation, Earth observation, Multiagent systems.

1 Introduction

Ces dernières années ont vu une forte augmentation du développement des constellations de satellites. Au lieu de considérer des satellites individuels, l'idée est de tirer parti d'un groupe de satellites, dont certains partagent souvent les mêmes plans orbitaux, pour fournir des services de positionnement, de télécommunications ou d'observation de la Terre [63], plus riches. Avec peu de satellites (*e.g.* deux dans le cas de la constellation PLEIADES [38]), et sur des orbites terrestres basses ou moyennes (altitude inférieure à 35 000 km), aucune région de la Terre n'est couverte à tout moment. Ainsi, la principale motivation pour augmenter la taille de ces constellations est de permettre d'observer n'importe quel point sur Terre à une fréquence plus élevée, comme le fait par exemple la société Planet avec plus de 150 satellites d'observation de la Terre (EOS) [54]. Mais l'exploitation de nombreux EOS nécessite d'améliorer la coopération entre les moyens sol et bord afin d'utiliser au mieux le système, ce qui est une tâche hautement combinatoire. Outre leur taille croissante, la composition des constellations évolue également. Les technologies récentes permettent la production et le déploiement d'EOS agiles capables de changer leur orientation, et de fournir de multiples types de prises de vue avec de multiples capteurs. Tout en fournissant des services plus riches, cela ajoute de nombreux degrés de liberté et des variables de décision pour programmer l'activité de l'EOS, ouvrant ainsi de nombreux défis scientifiques et techniques [3, 65].

La figure 1 montre un système EOS avec ses opérations au sol et dans l'espace. Elle met en évidence la multiplicité et la richesse des acteurs et des composants ayant leurs propres activités et objectifs. Comme les EOS ont une capacité de calcul embarquée limitée, la majeure partie de la mission est déterminée hors ligne et transmise aux EOS à

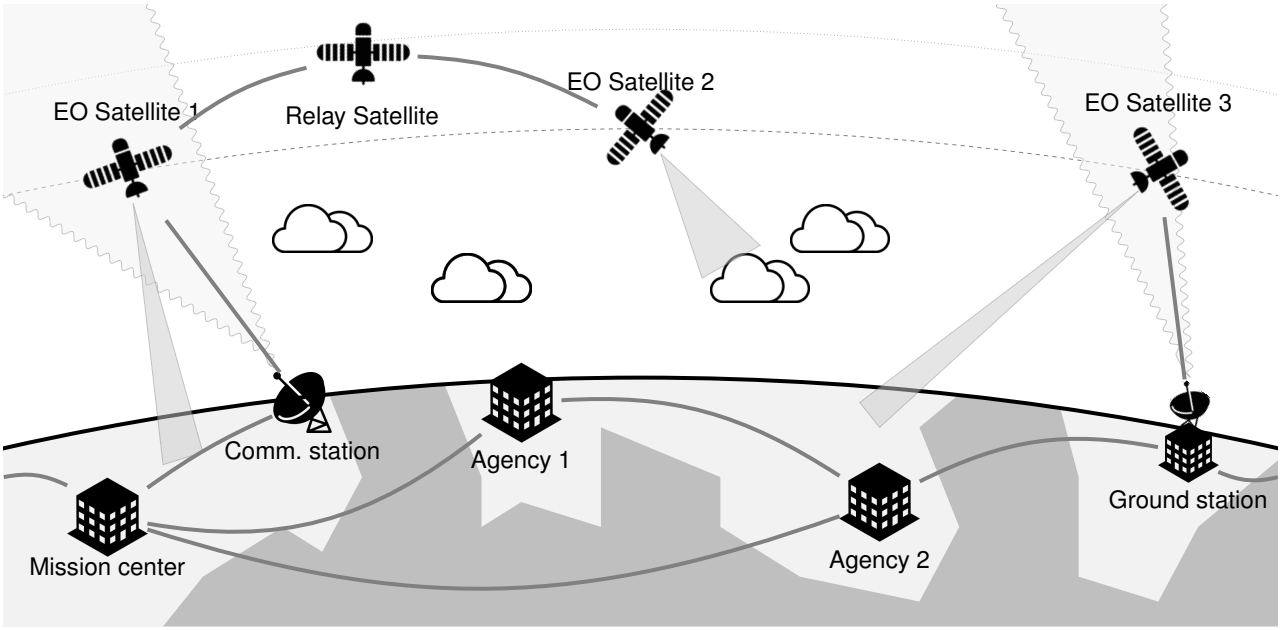


FIGURE 1 – Un système d'observation de la Terre composé d'un centre mission principal et de stations distribuées (ainsi que leurs fenêtres de visibilité), d'agences émettant des requêtes d'observation (au centre mission), de satellites d'observation (et leur empreinte sol), et de communication (reliant les satellites d'observation).

l'aide de stations au sol. En outre, les centres de mission et les agences doivent collaborer pour partager les orbites, programmer les téléchargements de plans, les acquisitions d'images et les données. Si la constellation EOS appartient à différentes parties prenantes, celles-ci peuvent même négocier le partage de certaines ressources embarquées. Les opérations dans l'espace nécessitent également une coopération, notamment entre les EOS qui doivent effectuer de multiples acquisitions, qui sont également souvent composites (*e.g.* plusieurs images requises pour couvrir une large région). Pour les applications optiques, les incertitudes météorologiques doivent être traitées pour éviter de capturer des images inutiles remplies de nuages. Les EOS se partagent également les tâches, de sorte que les observations inutilisables faites par un EOS peuvent être effectuées plus tard par un EOS suivant survolant la région. En ce qui concerne la coopération, les EOS peuvent s'appuyer sur des communications indirectes (via des satellites relais dédiés) ou directes à courte distance pour transférer les tâches de l'un à l'autre, au lieu d'attendre plusieurs minutes pour interagir avec les stations terrestres accessibles.

Ce scénario illustre la nécessité de coopérer, de résoudre et de planifier collectivement, de s'adapter et d'interagir, qui sont les motivations principales des systèmes multi-agents (SMA). Alors que les SMA ont été identifiés très tôt pour modéliser les systèmes de satellites [52, 32], ce scénario ouvre de nouveaux défis à relever par la communauté SMA, regroupés en trois catégories. La section 2 traite des techniques multi-agents qui pourraient aider à concevoir et à dimensionner les constellations de satellites avant leur déploiement. La section 3 passe en revue certains modèles multi-agents pour la gestion au sol des principaux problèmes d'allocation et de planification au sol, de tâches d'acquisition d'images, de communication et de manœuvre, et dont les solutions sont poussées jusqu'à la constellation. Enfin, la section 4 explore comment les systèmes multi-agents, et plus particulièrement les notions d'autonomie et d'interaction, peuvent apporter adaptation et résilience à l'exploitation des satellites, une fois les plans déjà installés à bord. Nous mettons l'accent sur les domaines d'intérêt multi-agents dans lesquels chaque défi se situe, en utilisant la notation ➥ *<domaine>* (tirés des domaines listés dans l'appel des Journées Francophones sur les Systèmes Multi-Agents [1]).

2 Défis dans la conception de constellations

2.1 Modélisation et simulation

La phase de conception consiste à dimensionner la constellation, c'est-à-dire à déterminer le ou les modèles orbitaux, le nombre de satellites sur chaque plan orbital ainsi que leurs éléments, ainsi l'ensemble des stations au sol utilisées pour télécharger les images et les plans de mission. La nature composite, l'hétérogénéité, la dynamique et l'ouverture de la constellation EOS doivent prises en compte dans cette phase. De plus, les systèmes spatiaux sont parmi les plus exigeants en matière d'exigences fonctionnelles et de sûreté.

Par exemple, les constellations EOS contiennent un grand nombre de satellites (d'une dizaine à plusieurs centaines de satellites), qui peuvent être très hétérogènes (plate-forme,

1. https://easychair.org/cfp/jfsma2021

charge utile, orbite) et généralement conçus pour un objectif spécifique à court terme (*e.g.* mission scientifique) [18]. Si toutes les tâches peuvent être effectuées par l'ensemble des satellites, le catalogue des demandes est beaucoup plus vaste que la capacité d'observation réelle de la constellation. Le problème consiste à sélectionner le sous-ensemble de satellites de la constellation pour effectuer les tâches d'observation. Le coût d'une équipe de satellite peut-être défini comme une agrégation du coût des tâches à effectuer, qui est fonction du satellite qui effectue réellement l'observation et du temps d'inactivité de chaque satellite. On vise alors à trouver l'équipe la moins chère pour un ensemble de tâches. En suivant cette approche et en utilisant les concepts de modélisation mutli-agent, il serait possible de prendre en compte un plus large éventail de propriétés lors de la conception des équipes (par exemple, la robustesse ou les objectifs individuels si les satellites appartiennent à des gestionnaires différents) [1].

La *modélisation et la programmation multi-agent* pourraient être d'une grande aide en fournissant des concepts de modélisation (par exemple, rôles, objectifs, organisations, institutions) et des méthodologies pour développer des plateformes permettant de gérer des systèmes multi-satellites et multi-opérateurs [68, 9]. En outre, la *simulation multi-agent* (ou MABS) apparaît comme une approche à grains fins, pertinente pour mieux appréhender le fonctionnement du système, ou pour faire des prédictions sur ses performances [72, 5, 15]. Dans le domaine spatial, les simulateurs existants comme Ptolemy [49] pourraient bénéficier des concepts et des efforts du MABS, d'autant plus que l'on prévoit une plus grande autonomie des systèmes spatiaux. Toutefois, une telle intégration nécessite des efforts de recherche sur le couplage et l'interopérabilité des simulateurs [44, 17]. Enfin, comme les modèles utilisés pour évaluer les performances sont différents des modèles utilisés pour mesurer le respect des exigences et la sûreté [59], une piste s'ouvre pour les chercheurs dans le domaine du génie logiciel orienté agent.

La conception d'une constellation EOS est également un problème *multi-objectif* visant à optimiser simultanément le nombre de passages de satellites au-dessus des régions cibles, le délai de revisite d'un satellite au-dessus de ces mêmes régions et également le coût de la constellation, tout en respectant les contraintes de sécurité. Pour les modèles simples, certaines méthodes analytiques peuvent être utilisées pour optimiser la conception de la constellation [35, 51]. Pour les modèles et simulations complexes, la conception de l'architecture peut être traitée par différentes approches numériques, comme l'optimisation multidisciplinaire [15], les algorithmes génétiques [20] ou l'optimisation par essaim de particules [66]. Mais, ces techniques de type *black box* rendent difficile la compréhension de l'influence des paramètres sur les configurations résultantes. Pouvoir expliquer aux décideurs les différents composants, leur comportement et leurs interactions est un défi majeur, qui pourrait être permis par l'intégration de MABS aux méthodes d'optimisation.

- ➥ *Simulation multi-agent*
- ➥ *Langages de programmation multi-agents*
- ➥ *Méthodes et méthodologies multi-agents*
- ➥ *Organisations, Normes, Coalitions*
- ➥ *Vérification et validation des systèmes multi-agents*

2.2 Allocation et partage de ressources

Lorsqu'une constellation EOS est utilisée par plusieurs parties prenantes, il peut être exigé que son exploitation soit équitable ou juste, *e.g.* en fonction de l'investissement financier de chaque utilisateur. Ce problème relève du domaine de l'*allocation de ressources multi-agents* (ou MARA) [19]. Pour les constellations EOS, les utilisateurs ou clients peuvent partager différents types de biens, tels que des orbites, qui peuvent être considérés comme des biens divisibles, et obtenir l'exclusivité sur les portions d'orbite qui leur sont attribuées. Dans ce cas, les utilisateurs disposent de leur propre centre de mission et peuvent exploiter les EOS sur les portions qui leur sont attribuées. Les utilisateurs peuvent également partager des EOS en demandant une observation d'une zone géographique. Dans ce cas, les demandes sont considérées comme des biens indivisibles et les demandes d'un utilisateur donné qui sont effectivement programmées sur la constellation peuvent être considérées comme le lot de cet utilisateur.

Le cas de la division équitable soulève plusieurs défis [13]. Premièrement, l'*équité* (*fairness*) est étroitement lié aux préférences des utilisateurs car il est nécessaire de comparer les lots qui peuvent être attribués à chaque utilisateur. Représenter les préférences de manière compacte tout en étant capable de raisonner efficacement sur celles-ci est un défi en lui-même (*e.g.* CP-net pour les préférences ordinales [12]). Dans le contexte des EOS, de nombreuses caractéristiques pourraient être utilisées pour définir l'utilité d'une requête, sa priorité, sa zone d'observation et l'incertitude sur la météo [62]. La plupart des formalismes de représentation des préférences supposent que l'utilité d'un ensemble est la somme des utilités des éléments de l'ensemble. Dans le domaine spatial, les utilités des requêtes d'observation peuvent ne pas être indépendantes les unes des autres ou peuvent être assez complexes (*e.g.* requêtes périodiques), et nécessitent d'aller *au-delà de l'additivité* et d'envisager des hypothèses plus réalistes.

En outre, il existe plusieurs concepts d'équité, et peu de travaux tiennent compte de l'équité lors de la planification des activités des EOS. L'équité peut être définie comme proportionnelle à la contribution financière dans le financement de la constellation [37]. L'équité peut également être définie comme une équité *maxmin* [61, 33]. En outre, l'équité n'est généralement pas le seul critère à prendre en compte et un compromis entre plusieurs critères est souvent nécessaire (*e.g.* entre efficacité et équité). Plusieurs procédures caractérisant des allocations efficaces et équitables pour EOS ont été étudiées [37], et alternativement l'équité fait partie d'un critère bi-objectif [61]. Trouver des procédures, centralisées ou décentralisées, qui permettent d'obtenir une allocation optimale ou de bonne qualité est un réel défi technique. Dans le cas d'EOS où les orbites sont partagées entre plu-

sieurs utilisateurs, on pourrait par exemple envisager des mécanismes d'enchères [19, 8] : les utilisateurs soumettent leurs offres (*i.e.* rapporte leurs préférences) publiquement ou en privé, il peut y avoir un ou plusieurs tours et l'allocation est faite par un commissaire-priseur.

➥ *Résolution collective de problèmes*
➥ *Théorie des jeux*

3 Défis dans les opérations hors-ligne

En opération, les centres de mission calculent des plans hors ligne pour chaque EOS, à partir d'un carnet de commandes. Outre le positionnement des acquisitions, ces plans doivent également préciser quand télécharger le résultat de l'activité des EOS vers les stations au sol accessibles, afin de préserver la mémoire embarquée limitée. Ceci résulte en un ensemble de problèmes de large échelle difficiles à résoudre.

3.1 Planification des observations

Ces problèmes sont distribués par la nature, et donc partiellement ou totalement décomposables. Cela ouvre la porte aux techniques multi-agents pour la résolution des problèmes. Par exemple, les différents composants des systèmes peuvent être considérés comme faisant partie d'un marché pour trouver des accords, en utilisant un *Contract Net Protocol* étendu, afin de résoudre un problème de planification de missions multi-satellites [48]. Les techniques d'*optimisation distribuée sous contraintes* (ou DCOP) [42] peuvent également être des méthodes de résolution efficaces pour aborder des problèmes d'allocation des tâches d'observation et de communication, où les variables de décision et les contraintes sont réparties sur un ensemble d'agents. Par exemple, la recherche distribuée à grand voisinage (*Distributed Large Neighborhood Search*) [26] pourrait être appliquée à la planification multi-satellites, comme c'est le cas avec son homologue centralisé [29]. L'ajout de la distribution permettrait d'améliorer l'explicabilité (en identifiant les lieux où des conflits sérieux apparaissent), la réduction du temps de calcul (en divisant le processus de recherche en plusieurs sous-processus simultanés) et la confidentialité (dans le cas où certaines tâches sont secrètes). Cependant, le passage à l'échelle des méthodes de résolution DCOP et la présence de variables de décision à nombres entiers et mixtes sont des défis à relever, comme l'ont récemment montré des études sur les DCOP à variables continues [31].

Plus généralement encore, les problèmes d'ordonnancement des constellations peuvent être modélisés comme des problèmes multi-objectifs (*e.g.* minimisant la consommation d'énergie tout en maximisant les observations réussies) [7, 40, 39, 61, 69] et asymétriques (*e. g.* les utilisateurs peuvent ne pas avoir la même récompense si une certaine observation est effectuée), qui sont encore des modèles pour lesquels l'efficacité des méthodes de résolution distribuées reste à démontrer [21, 28]. Enfin, comme ces problèmes de planification sont à très grande échelle, les approches distribuées *heuristiques et par auto-organisation* basées sur des plans auto-adaptatifs pourraient fournir à tout moment des solutions *anytime* de manière rapide et réactive [11]. Cependant, ces techniques ne fournissent pas (encore) de garanties de qualité, qui sont pourtant des conditions préalables importantes à l'adoption de telles outils par les acteurs aérospatiaux.

➥ *Coordination, Travail en équipe, Planification*
➥ *Négociation multi-agent, consensus*
➥ *Résolution collective de problèmes*

3.2 Planification sous incertitudes

Les systèmes EOS sont soumis à deux principaux types d'incertitudes. Premièrement, certains nuages peuvent être présents lors d'une observation. Si la fraction de couverture nuageuse de l'observation est supérieure à la fraction de couverture nuageuse maximale associée à la requête, l'observation n'est pas valide. De plus, comme le plan est calculé un certain temps avant que l'acquisition ne soit effectivement effectuée, cette incertitude est irréductible. Par exemple, l'espérance de la valeur absolue de la différence entre les fractions de couverture nuageuse prévues et réelles augmente avec l'horizon de prévision atteignant 0,4 pour un horizon d'une heure [71]. Étant donné que les fractions se situent entre 0 et 1, cette valeur reflète une grande incertitude. Ainsi, il existe une incertitude non négligeable concernant le succès de chaque observation prévue.

Deuxièmement, les observations sont stockées dans la mémoire des satellites sous une forme compressée et le taux de compression est spécifique à chaque observation et n'est pas connu à l'avance. Par exemple, des taux de compression variant de 3 à 6 ont été observés sur un petit ensemble d'images [70]. Il existe donc une incertitude sur la quantité de mémoire occupée par chaque observation avant son téléchargement vers une station au sol et sur le temps de téléchargement de chaque observation. En outre, les temps de téléchargement sont également influencés par la variabilité du débit binaire et la récupération après des erreurs de transmission.

Le premier type d'incertitude est directement lié à la récompense, tandis que le second est une caractéristique de la transition d'état.

La *planification multi-agent sous incertitudes* [56] et plus spécifiquement les *processus de décision de Markov décentralisés et partiellement observables* (DEC-POMDP) [6] peut être pertinents dans ce contexte. Néanmoins, les algorithmes fournissant des solutions DEC-POMDP ne passent pas à l'échelle et le défi consiste à concevoir des solutions plus simples. En complément des approches basées sur des processus de Markov, des techniques d'optimisation distribuée traitant des incertitudes ont récemment conduit au développement des *DCOP probabilistes*. Ces techniques étendent les DCOP classiques en augmentant le résultat des fonctions de coût avec des propriétés stochastiques [4, 57, 47] ou en introduisent des variables aléatoires comme entrée des fonctions de coût, pour simuler des propriétés exogènes incontrôlables de l'environnement, et ainsi optimiser le résultat attendu en moyenne [36, 67]. Toutefois, il est également important de noter que la prévision

des mesures d'incertitude associées au succès des observations pose un problème en termes de portée du système multi-agent étudié (si l'agent prédicteur est à l'intérieur ou à l'extérieur du système), et en termes de type de mesure d'incertitude, car en effet presque toutes les techniques de planification des satellites sont basées sur des probabilités qui sont un cas particulier des *fonctions de croyance* [22] [53] pour lequel les éléments focaux sont des singletons. Le cas général des probabilités imprécises permettrait d'accroître la robustesse des décisions mais peut induire une complexité importante, impliquant par exemple le calcul de critères par intégrales de Choquet. Néanmoins, la *théorie des possibilités* [25] pour laquelle les éléments focaux sont emboîtés, pourrait permettre d'une part l'augmentation de la robustesse sans grand impact sur la complexité par un calcul de critère pessimiste relativement simple et d'autre part de maintenir un lien avec les probabilités par des transformations directes et inverses. Enfin, la définition d'une récompense déterministe qui prend en compte des demandes de différents types et priorités et qui peut facilement être combinée à la mesure d'incertitude choisie est un problème en soi.

➥ *Coordination, Travail en équipe, Planification*
➥ *Environnement (modélisation)*
➥ *Résolution collective de problèmes*

3.3 Déconfliction des requêtes utilisateurs

Les constellations de satellites impliquent de nombreux acteurs, comme les propriétaires de satellites, les opérateurs de satellites, les clients de services demandant des observations, les agences gouvernementales ou les opérateurs militaires. Le partage des ressources de la constellation entre des agents ayant des objectifs et des programmes différents implique que certains conflits peuvent survenir, qui ne peuvent être résolus de manière centralisée afin de garantir l'autonomie de décision et la préservation de la confidentialité. Ce dernier point est crucial : les EOS peuvent être utilisés à des fins de défense et de sécurité et la plupart des acteurs ne veulent pas que les autres soient informés de la façon dont ils utilisent les satellites. Par exemple, un opérateur d'un pays peut permettre à un client d'un autre pays d'utiliser son satellite pour effectuer une observation, mais ne peut pas permettre de capturer une image de son propre pays ou de savoir quelles sont les observations prévues avant et après l'observation demandée. Cela signifie que les différents utilisateurs doivent résoudre un problème dont les sous-composantes (variables de décision, contraintes ou paramètres) sont propres et privées. Les *techniques d'optimisation distribuées* comme les DCOP peuvent être considérées à nouveau, lorsque les utilisateurs visent un objectif commun (*par exemple* maximiser le nombre d'observations planifiées) [55]. En cas d'objectifs plus divergents, il est également intéressant d'envisager les approches par *optimisation de consensus* où les utilisateurs établissent des accords sur certaines variables de décision partagées [46, 45, 14]. Là encore, la présence conjointe de variables de décision discrètes et continues rend l'application de telles techniques encore plus difficile [60]. Dans des contextes plus conflictuels et non coopératifs, la *théorie des jeux* peut fournir des schémas de coordination pour résoudre ces situations conflictuelles, comme le propose un travail récent [58], ou pour concevoir des places de marché [23].

➥ *Coordination, Travail en équipe, Planification*
➥ *Négociation multi-agent, consensus*
➥ *Résolution collective de problèmes*
➥ *Théorie des jeux*

4 Défis dans les opérations en ligne

Les constellations EOS sont des systèmes dynamiques déployés dans des environnements également dynamiques. La planification hors ligne ne suffit pas pour assurer un fonctionnement pleinement efficace, lorsque les conditions météorologiques peuvent dégrader la qualité des images ou lorsque des requêtes de dernière minute arrivent. L'autonomie à bord est une dimension à prendre en compte pour doter les satellites de certaines routines d'adaptation à la volée en réponse à des événements imprévus.

4.1 Dynamique et replanification

Comme l'acquisition d'images peut échouer en raison de la présence de nuages, ou parce qu'une requête de dernière minute peut se produire, il est très important de pouvoir reprogrammer certaines observations. La reprogrammation peut être envisagée au sol ou à bord. La réparation du plan au sol est déclenchée une fois que les EOS ont téléchargé les données et que leur mauvaise qualité est identifiée par les centres de validation, qui peuvent alors demander une reprogrammation. Ici, les techniques classiques de réparation centralisée de plans peuvent être envisagées pour ajouter des tâches de manière dynamique, mais elles doivent être suffisamment rapides pour que le plan révisé soit poussé dès que possible vers le prochain EOS capable d'exécuter la tâche. Cette réparation doit être fournie de manière aussi réactive que possible par rapport à la construction d'un plan complet. Dans l'optique d'une prise de décision partiellement ou totalement embarquée, des techniques multi-agents existent pour faire face aux problèmes dynamiques, comme dans les *problèmes dynamique d'optimisation distribuée sous contraintes* (DynDCOP), où les agents coopèrent pour optimiser une série de problèmes au lieu d'optimiser une seule instance à un moment donné [30], ou pour pouvoir résoudre des problèmes qui changent au moment de l'exécution [50]. On peut également envisager, des techniques de *réparation de plans multi-agents* comme celles de la réparation embarquée [34], qui n'envisagent de modifier qu'une partie du plan pour les agents impactés au lieu de tout replanifier. Cependant, tout en fournissant des plans de bonne qualité, ces techniques souffrent encore d'une évolutivité limitée et nécessitent des communications fiables. Dans notre cas, la communication peut ne pas être persistante (*e.g.* les stations au sol ne sont pas accessibles à tout moment ou les satellites peuvent ne pas être en mesure d'interagir directement ou indirectement) [33]. Un schéma d'apprentissage distribué pour réparer les plans

multi-satellites a été proposé, par exemple, en remarquant que les informations historiques de planification des tâches coopératives auront un impact sur les résultats de la dernière planification [64]. Une autre famille de techniques candidates sont celles qui s'appuient sur la détermination de *consensus* [41, 27], où les agents négocient pour s'entendre sur certaines variables de décision (*par exemple* le choix des tâches à effectuer) tout en étant *résilient* aux perturbations de l'environnement et à l'asynchronisme [24, 50]. En outre, les EOS ont une capacité de calcul limitée, ce qui limite la gamme des techniques d'optimisation pouvant être réalisées à bord. Ainsi, les techniques d'auto-organisation, ne reposant que sur une communication limitée et nécessitant des calculs limités, semblent être des pistes pertinentes pour assurer l'adaptation des plans en cours d'exécution [11, 50, 33]. Bien que n'étant pas garantis de fournir des solutions optimales, ces techniques pourraient arbitrer entre les requêtes sur la base de critères simples (priorité) et pourraient transférer les demandes d'EOS à EOS. Cependant, pousser une telle autonomie et une telle décision à bord reste un véritable défi, qui nécessite encore de gros efforts de recherche en matière d'intelligence artificielle et de robotique pour être certifiée puis intégrée dans les systèmes opérationnels.

➥ *Commande et contrôle de système multi-agent*
➥ *Déploiement de SMA, Résistance aux pannes, Fiabilité*
➥ *Émergence, Auto-organisation, Viabilité*
➥ *Évolution, Adaptation*
➥ *Robotique collective*

4.2 Interactions et protocoles

Lorsqu'il s'agit d'opérations en ligne, l'examen des possibilités de communication est un point essentiel pour améliorer les performances du système. Pour les constellations EOS, les communications directes sont évidemment utilisées entre un centre de mission et les satellites, mais de nombreux autres types de communications peuvent également être utilisés, comme les communications directes entre deux satellites par une liaison intersatellite, les communications directes entre deux centres de mission gérant différentes parties de la constellation, les communications indirectes via des satellites relais géostationnaires ou des drones, et plus généralement les communications indirectes par un réseau de liaisons de communication. L'examen de tous ces liens de communication potentiels pour les futures constellations soulève de nombreux défis pour les opérations en ligne, tels que « quel protocole de communication doit être utilisé », « quand communiquer », « quelles données communiquer et à qui », ou « quelle est la valeur d'une information », pour n'en citer que quelques-unes. Certaines propositions ont déjà été faites pour répondre à ces questions. Par exemple, les *Delay Tolerant Networks* (DTN) peuvent être envisagés pour construire un système dans lequel un satellite donné peut avertir les autres satellites des positions terrestres où un phénomène particulier a été détecté [43]. Alternativement, chaque satellite peut maintenir une estimation de la connaissance des autres satellites en utilisant un protocole de communication épidémique entre satellites [10]. Enfin, la communication directe peut être utilisée pour la négociation et la coordination entre les agents des engins spatiaux [52, 2] et également pour acheminer les données d'observation d'un satellite à une station de réception au sol par le biais de liaisons intersatellites, de manière à réduire le temps pendant lequel les utilisateurs peuvent obtenir leurs images [16]. Ces quelques exemples montrent que les communications peuvent être utilisées à la fois pour des raisons épistémiques (apporter une information qui peut aider à prendre de meilleures décisions ou à mettre en œuvre un protocole de coordination) et pour des raisons de performance (communiquer des données d'observation et obtenir une récompense immédiate des utilisateurs), un objectif commun étant d'obtenir soit une meilleure réactivité soit un meilleur partage des tâches entre les agents.

➥ *Interaction, Communication, Protocoles*
➥ *Robotique collective*

5 Conclusion

Dans ce papier, nous avons identifié plusieurs défis ouverts concernant les constellations de satellites d'observation de la Terre et leurs applications, relevant des domaines de l'intelligence artificielle distribuée et des systèmes multi-agents. En effet, la conception, le déploiement et l'exploitation de tels systèmes composés de plusieurs acteurs et ressources conviennent parfaitement au paradigme multi-agent. Toutefois, la difficulté et la nouveauté de ces problèmes constituent toujours un défi pour les méthodes existantes, ce qui ouvre de nouvelles pistes de recherche pour les années à venir, en particulier dans les domaines d'intérêt identifiés par la communauté multi-agent, allant de l'ingénierie des systèmes multi-agents à la robotique en passant par la représentation des connaissances, le raisonnement et la planification. Finalement, à notre connaissance, du fait de la nouveauté des problèmes soulevés, il n'existe pas encore de benchmark et d'instance complexe leur correspondant. Un effort de formalisation et de publication de benchmarks et instances pourrait permettre non seulement la comparaison des performances de diverses méthodes du domaine des SMA mais aussi la comparaison des approches SMA avec d'autres approches de l'Intelligence Artificielle.

Références

[1] E. Andrejczuk, J. A. Rodriguez-Aguilar, and C. Sierra. A Concise Review on Multiagent Teams : Contributions and Research Opportunities. In Natalia Criado Pacheco, Carlos Carrascosa, Nardine Osman, and Vicente Julián Inglada, editors, *Multi-Agent Systems and Agreement Technologies*, pages 31–39, Cham, 2017. Springer International Publishing.

[2] C. Araguz, A. Alvaro, I. del Portillo, K. Root, E. Alarcón, and E. Bou-Balust. On autonomous software architectures for distributed spacecraft : A local-global policy. In *2015 IEEE Aerospace Conference*, pages 1–9, 2015.

[3] C. Araguz, E. Bou-Balust, and E. Alarcón. Applying autonomy to distributed satellite systems : Trends, challenges, and future prospects. *Systems Engineering*, 21(16) :401–416, 03 2018.

[4] J. Atlas and K. Decker. Coordination for uncertain outcomes using distributed neighbor exchange. In *International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '10, page 1047–1054, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.

[5] G. Bastianelli, D. Salamon, A. Schisano, and A. Iacobacci. Agent-based simulation of collaborative unmanned satellite vehicles. In *2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL)*, pages 1–6, Rome, 2012. IEEE.

[6] D.S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4) :819–840, 2002.

[7] N. Bianchessi, J.-F. Cordeau, J. Desrosiers, G. Laporte, and V. Raymond. A heuristic for the multi-satellite, multi-orbit and multi-user management of earth observation satellites. *European Journal of Operational Research*, 177(2) :750 – 762, 2007.

[8] L. Blumrosen and N. Nisan. Combinatorial auctions. *Algorithmic game theory*, 267 :300, 2007.

[9] O. Boissier, R.H. Bordini, J.F. Hübner, A. Ricci, and A. Santi. Multi-agent oriented programming with jacamo. *Science of Computer Programming*, 78(6) :747 – 761, 2013.

[10] G. Bonnet and C. Tessier. Collaboration among a satellite swarm. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, 2007.

[11] J. Bonnet, M.-P. Gleizes, E. Kaddoum, S. Rainjonneau, and G. Flandin. Multi-satellite mission planning using a self-adaptive multi-agent system. In *2015 IEEE 9th International Conference on Self-Adaptive and Self-Organizing Systems*, pages 11–20, Boston, 2015. IEEE.

[12] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole. Cp-nets : A tool for representing and reasoning withconditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21 :135–191, Feb 2004.

[13] S. Bouveret, Y. Chevaleyre, and N. Maudet. Fair allocation of indivisible goods. In *Handbook of Computational Social Choice*, pages 284–310, Cambridge, UK, 2016. Cambridge University Press.

[14] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1) :1–122, January 2011.

[15] I. A. Budianto and J. R. Olds. Design and deployment of a satellite constellation using collaborative optimization. *Journal of Spacecraft and Rockets*, 41(6) :956–963, 2004.

[16] K.L. Cahoy and A.K. Kennedy. Initial Results from ACCESS : An Autonomous CubeSat Constellation Scheduling System for Earth Observation. In *31st Annual AIAA/USU Conference on Small Satellites*, 2017.

[17] B. Camus, T. Paris, J. Vaubourg, Y. Presse, C. Bourjot, L. Ciarletta, and V. Chevrier. Co-simulation of cyber-physical systems using a devs wrapping strategy in the mecsyco middleware. *SIMULATION*, 94(12) :1099–1127, 2018.

[18] H. Chen, S. Yang, J. Li, and N. Jing. Exact and Heuristic Methods for Observing Task-Oriented Satellite Cluster Agent Team Formation. *Mathematical Problems in Engineering*, 2018 :1–23, August 2018.

[19] Y. Chevaleyre, P.E. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J.A. Padget, S. Phelps, J.A. Rodríguez-Aguilar, and P. Sousa. Issues in multiagent resource allocation. *Informatica (Slovenia)*, 30(1) :3–31, 2006.

[20] C. Dai, G. Zheng, and Q. Chen. Satellite constellation design with multi-objective genetic algorithm for regional terrestrial satellite network. *China Communications*, 15(8) :1–10, 2018.

[21] F. M. Delle Fave, R. Stranders, A. Rogers, and N. R. Jennings. Bounded decentralised coordination over multiple objectives. In *International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '11, page 371–378, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems.

[22] Arthur P Dempster. Upper and lower probabilities induced by a multivalued mapping. In *Classic works of the Dempster-Shafer theory of belief functions*, pages 57–72. Springer, 2008.

[23] G. Denis, A. Claverie, X. Pasco, J.-P. Darnis, B. de Maupeou, M. Lafaye, and E. Morel. Towards disruptions in earth observation ? new earth observation systems and markets evolution : Possible scenarios and impacts. *Acta Astronautica*, 137 :415 – 433, 2017.

[24] S.M. Dibaji and H. Ishii. Resilient multi-agent consensus with asynchrony and delayed information. *IFAC-PapersOnLine*, 48(22) :28 – 33, 2015. 5th IFAC Workshop on Distributed Estimation and Control in Networked Systems NecSys 2015.

[25] D. Dubois and H. Prade. Possibilistic logic — an overview. In Jörg H. Siekmann, editor, *Computational Logic*, volume 9 of *Handbook of the History of Logic*, pages 283 – 342. North-Holland, Amsterdam, Holland, 2014.

[26] F. Fioretto, F. Campeotto, A. Dovier, E. Pontelli, and W. Yeoh. Large neighborhood search with quality guarantees for distributed constraint optimization problems. In *International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, page

1835–1836, Richland, SC, 2015. International Foundation for Autonomous Agents and Multiagent Systems.

[27] P. Franceschelli and P. Frasca. Proportional dynamic consensus in open multi-agent systems. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 900–905, Miami Beach, FL, USA, 2018. IEEE.

[28] T. Grinshpoun, A. Grubshtein, R. Zivan, A. Netzer, and A. Meisels. Asymmetric distributed constraint optimization problems. *J. Artif. Int. Res.*, 47(1) :613–647, May 2013.

[29] L. He, L. Xiaolu, G. Laporte, Y.-W. Chen, and Y. Chen. An improved adaptive large neighborhood search algorithm for multiple agile satellites scheduling. *Computers and Operations Research*, 100 :12–25, 07 2018.

[30] K. D. Hoang, F. Fioretto, P. Hou, M. Yokoo, W. Yeoh, and R. Zivan. Proactive dynamic distributed constraint optimization. In *International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '16, page 597–605, Richland, SC, 2016. International Foundation for Autonomous Agents and Multiagent Systems.

[31] K. D. Hoang, W. Yeoh, M. Yokoo, and Z. Rabinovich. New algorithms for continuous distributed constraint optimization problems. In *International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '20, page 502–510, Richland, SC, 2020. International Foundation for Autonomous Agents and Multiagent Systems.

[32] Hongwei Xia, Guangcheng Ma, Weinan Xie, and Baomin Feng. Multiple satellite formation based on multi-agent. In *2006 1st International Symposium on Systems and Control in Aerospace and Astronautics*, pages 4 pp.–705, Harbin, China, 2006. IEEE.

[33] M. Johnston. Scheduling nasa's deep space network : Priorities, preferences, and optimization. 2020.

[34] A. Komenda, P. Novák, and M. Pěchouček. Domain-independent multi-agent plan repair. *Journal of Network and Computer Applications*, 37 :76 – 88, 2014.

[35] C. L. Korb and A.R. Korb. Methods for optimizing the performance, cost and constellation design of satellites for full and partial earth coverage, May 26 2020. US Patent 10,664,782.

[36] T. Léauté and B. Faltings. Distributed constraint optimization under stochastic uncertainty. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, page 68–73. AAAI Press, 2011.

[37] M. Lemaître, G. Verfaillie, H. Fargier, J. Lang, N. Bataille, and J.-M. Lachiver. Equitable allocation of earth observing satellites resources. In *5th ONERA-DLR Aerospace Symposium (ODAS'03)*, 2003.

[38] M. Lemaître, G. Verfaillie, F. Jouhaud, J.-M. Lachiver, and N. Bataille. Selecting and scheduling observations of agile satellites. *Aerospace Science and Technology*, 6(5) :367 – 381, 2002.

[39] L. Li, H. Chen, J. Li, N. Jing, and M. Emmerich. Preference-Based Evolutionary Many-Objective Optimization for Agile Satellite Mission Planning. *IEEE Access*, 6 :40963–40978, 2018. Conference Name : IEEE Access.

[40] L. Li, F. Yao, N. Jing, and M. Emmerich. Preference incorporation to solve multi-objective mission planning of agile earth observation satellites. In *2017 IEEE Congress on Evolutionary Computation, CEC 2017*, pages 1366–1373. IEEE, 2017.

[41] Z. Li, Z. Duan, and F. L. Lewis. Distributed robust consensus control of multi-agent systems with heterogeneous matching uncertainties. *Automatica*, 50(3) :883 – 889, 2014.

[42] P.J. Modi, W.-M. Shen, M. Tambe, and M. Yokoo. Adopt : Asynchronous distributed constraint optimization with quality guarantees. *Artif. Intell.*, 161(1–2) :149–180, January 2005.

[43] S. Nag, A. Li, V. Ravindra, M. Sanchez Net, K.-M. Cheung, R. Lammers, and B. Bledsoe. Autonomous Scheduling of Agile Spacecraft Constellations with Delay Tolerant Networking for Reactive Imaging. In *International Conference on Automated Planning and Scheduling SPARK Workshop*, 2019.

[44] K. Ndiaye, F. Balbo, J.-P. Jamont, and M. Occello. Simulation coupling limitations with respect to shared entities constraints. In *8th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, pages 338–346. INSTICC, SciTePress, 2018.

[45] A. Nedić, A. Olshevsky, and Wei Shi. *Decentralized Consensus Optimization and Resource Allocation*, pages 247–287. Springer International Publishing, Cham, 2018.

[46] A. Nedić, A. Ozdaglar, and P. A. Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4) :922–938, 2010.

[47] D. T. Nguyen, W. Yeoh, and H.C. Lau. Stochastic dominance in stochastic dcops for risk-sensitive applications. In *International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '12, page 257–264, Richland, SC, 2012. International Foundation for Autonomous Agents and Multiagent Systems.

[48] Peng Feng, Hao Chen, Shuang Peng, Luo Chen, and Longmei Li. A method of distributed multi-satellite mission scheduling based on improved contract net protocol. In *2015 11th International Conference on Natural Computation (ICNC)*, pages 1062–1068, Zhangjiajie, China, 2015. IEEE.

[49] Claudius Ptolemaeus, editor. *System Design, Modeling, and Simulation using Ptolemy II*. Ptolemy.org, Berkeley, California, USA, 2014.

[50] P. Rust, G. Picard, and F. Ramparany. Resilient distributed constraint optimization in physical multi-agent systems. In *European Conference on Artificial Intelligence (ECAI)*, pages 195 – 202, Amsterdam, Holland, 2020. IOS Press.

[51] T. Savitri, Y. Kim, S. Jo, and H. Bang. Satellite constellation orbit design optimization with combined genetic algorithm and semianalytical approach. *International Journal of Aerospace Engineering*, 2017 :1235692, May 2017.

[52] T. Schetter, M. Campbell, and D. Surka. Multiple agent-based autonomy for satellite constellations. *Artificial Intelligence*, 145(1) :147 – 180, 2003.

[53] Glenn Shafer. *A mathematical theory of evidence*, volume 42. Princeton university press, 1976.

[54] V. Shah, V. Vittaldev, L. Stepan, and C. Foster. Scheduling the world's largest earth-observing fleet of medium-resolution imaging satellites. *IWPSS*, 2019.

[55] P. K. Sinha and A. Dutta. Multi-satellite task allocation algorithm for earth observation. In *2016 IEEE Region 10 Conference (TENCON)*, pages 403–408, New York, New York, US, 2016. IEEE.

[56] M. T. J. Spaan and F. S. Melo. Interaction-driven markov games for decentralized multiagent planning under uncertainty. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '08, page 525–532, Richland, SC, 2008. International Foundation for Autonomous Agents and Multiagent Systems.

[57] R. Stranders, F.M. Delle Fave, A. Rogers, and N.R. Jennings. U-gdl : A decentralised algorithm for dcops with uncertainty. Project report, May 2011.

[58] C. Sun, X. Wang, and X. Liu. Distributed satellite mission planning via learning in games. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 4381–4386, New York, New York, US, 2018. IEEE.

[59] A.H. Sánchez, T. Soares, and A. Wolahan. Reliability aspects of mega-constellation satellites and their impact on the space debris environment. In *2017 Annual Reliability and Maintainability Symposium (RAMS)*, pages 1–5, 2017.

[60] R. Takapoui, N. Moehle, S. Boyd, and A. Bemporad. A simple effective heuristic for embedded mixed-integer quadratic programming. *International Journal of Control*, 93(1) :2–12, 2020.

[61] P. Tangpattanakul, N. Jozefowiez, and P. Lopez. A multi-objective local search heuristic for scheduling Earth observations taken by an agile satellite. *European Journal of Operational Research*, 245(2) :542–554, September 2015.

[62] A.E. Vasegaard, M. Picard, F. Hennart, P. Nielsen, and S. Saha. Multi criteria decision making for the multi-satellite image acquisition scheduling problem. *Sensors*, 20(5) :1242, 2020.

[63] J.G. Walker. Satellite Constellations. *Journal of the British Interplanetary Society*, 37 :559, December 1984.

[64] C. Wang, J. Li, N. Jing, J. Wang, and H. Chen. A distributed cooperative dynamic task planning algorithm for multiple satellites based on multi-agent hybrid learning. *Chinese Journal of Aeronautics*, 24(4) :493 – 505, 2011.

[65] X. Wang, G. Wu, L. Xing, and W. Pedrycz. Agile earth observation satellite scheduling over 20 years : formulations, methods and future directions. *CoRR*, abs/2003.06169, 2020.

[66] X. Wang, H. Zhang, S. Bai, and Y. Yue. Design of agile satellite constellation based on hybrid-resampling particle swarm optimization method. *Acta Astronautica*, 178 :595 – 605, 2021.

[67] Y. Wang, K. Sycara, and P. Scerri. Towards an understanding of the value of cooperation in uncertain world. In *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, WI-IAT '11, page 212–215, USA, 2011. IEEE Computer Society.

[68] M. Winikoff and L. Padgham. *Agent Oriented Software Engineering*, chapter 13, pages 695–757. MIT Press, 01 2013.

[69] W. Yang, Y. Chen, R. He, Z. Chang, and Y. Chen. The Bi-objective Active-Scan Agile Earth Observation Satellite Scheduling Problem : Modeling and Solution Approach. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–6, July 2018.

[70] G. Yu, T. Vladimirova, and M.N. Sweeting. Image compression systems on board satellites. *Acta Astronautica*, 64(9-10) :988–1005, 2009.

[71] I. Yucel, J. W Shuttleworth, X Gao, and S Sorooshian. Short-term performance of mm5 with cloud-cover assimilation from satellite observations. *Monthly weather review*, 131(8) :1797–1810, 2003.

[72] C. Zhang, Y. Wang, and Y. Zhao. Agent-based distributed simulation technology of satellite formation flying. In *Proceedings of the 2013 Fourth World Congress on Software Engineering*, WCSE '13, page 13–16, USA, 2013. IEEE Computer Society.

Session 2 – Approches méthodologiques

IA de confiance : condition nécessaire pour le déploiement de l'IA dans les systèmes critiques

J. MATTIOLI[8], F. TERRIER[3], L. CANTAT[4], R. GELIN[5], J. CHIARONI[7], Y. BONHOMME[4], H. AMADOU-BOUBACAR[1], E. ESCORIHUELA[2], S. PICARD[6], C. ALIX[8]

[1] Air Liquide, [2] Airbus, [3] CEA, [4]IRT SystemX, [5] Renault, [6] SAFRAN, [7] Secrétariat général pour l'investissement, [8]Thales

juliette.mattioli@thalesgroup.com

Résumé

Avec le renouveau de l'IA, on assiste aujourd'hui à une croissance de ses usages, sans précédent. Ce qui a changé ces dernières années, c'est que la recherche est passée de connaissances théoriques à de nombreuses applications pratiques. Malgré ces résultats très prometteurs trop peu de preuves de concept (PoC) atteignent un déploiement au niveau de la production des systèmes critiques. L'une des causes est que le déploiement dans des industries telles que l'aéronautique, l'énergie, l'automobile, la défense, la santé, la fabrication, etc. nécessite la conformité à des objectifs de qualité, de sûreté, de sécurité et de fiabilité qui ne sont pas complétés par des systèmes d'IA à l'état de l'art. Ainsi, un système critique à base d'IA doit reposer sur des méthodes de développement bien définies, de sa conception à son déploiement et sa qualification. Cela nécessite une chaîne d'outils de bout en bout garantissant la confiance à toutes les étapes : (1) la spécification, les connaissances et la gestion des données ; (2) conception d'algorithmes et d'architecture de système avec la préoccupation de la relation à l'humain ; (3) caractérisation, vérification et validation des fonctions de l'IA ; (4) déploiement, en particulier sur l'architecture embarquée ; (5) qualification, certification d'un point de vue système.

Mots-clés

IA, confiance, méthodologie, ingénierie algorithmique, ingénierie des données, ingénierie de la connaissance, ingénierie système, méthodes formelles, système critique.

Abstract

With the renewal of AI, we observe an unprecedented growth of its usage. What has changed is that research in recent years turns from theoretical insights into various practical applications. Despite these very promising results, too few Proof of Concept (PoC) are reaching production level deployment within critical systems. One of the causes is that deployment in industries as aeronautics, energy, automotive, defense, health, manufacturing, etc. requires conformity to quality, safety, security, reliability objectives that are from being completed by state-of-the-art AI systems. Thus, an AI based critical system needs to have well defined development methods from its design to its deployment and qualification. This requires a complete tool chain ensuring trust at all stages, as : (1) specification, knowledge and data management ; (2) algorithm and system architecture design taking into account human in the loop ; (3) characterization, verification and validation of AI functions ; (4) deployment, particularly on embedded architecture ; (5) qualification, certification from a system point of view.

Keywords

AI, Trust, Methodology, Algorithm Engineering, Data Engineering, Knowledge Engineering, System Engineering, Formal Methods, Critical Systems

1 Les enjeux de l'IA de confiance

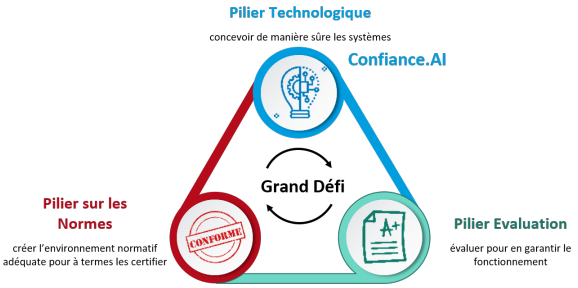


FIGURE 1 – Le Grand Défi de l'IA de confiance repose sur 3 piliers : le pilier technologique avec le programme Confiance.AI, le pilier évaluation et le pilier normalisation

Si l'Intelligence Artificielle (IA) semble promise à un fort développement, de nombreux verrous freinent son adoption, en particulier pour un déploiement dans les systèmes critiques. Ceux-ci doivent par construction garantir des propriétés de sécurité et de sûreté mais aussi suivre des principes de confiance et de responsabilité. En effet, par système critique, nous entendons, tout système pouvant directement ou indirectement engendrer des impacts sur les personnes physiques – impacts de nature corporelle, psychologique, sur la vie privée etc. – mais également sur les personnes morales – typiquement les entreprises avec des impacts de nature industrielle, financière, ou sur l'image au-

près du public, la propriété intellectuelle etc.

En parallèle de ce constat industriel, le conseil de l'innovation [1], a lancé fin 2018 le **Grand Défi National de l'IA de confiance**, avec l'objectif de sortir de l'ère des PoC (preuves de concept) par une réponse appropriée à la question de la qualification (homologation voire certification) des systèmes critiques à base d'IA.

En effet, que les systèmes critiques à base d'IA reposent sur des techniques d'apprentissage ou sur des approches plus symboliques, leur conception n'est pas neutre. Ces systèmes doivent suivre des principes de confiance et de responsabilité, garantir par construction (*by design*) des propriétés de sécurité, de sûreté et de fiabilité, qu'il faut pouvoir démontrer. Ainsi, pour permettre un déploiement de l'IA dans ces systèmes critiques, les pratiques d'ingénierie actuelles sont en défaut de par leur démarche basée sur la constitution de bases de données et de connaissances et leur exploitation via des algorithmes génériques qui masquent, voire abstraient, la logique fine des calculs. Il est alors extrêmement difficile de définir et comprendre leur enchaînement et donc d'établir la conformité des fonctions implantées. Il devient donc nécessaire de repenser ces ingénieries classiques (ingénierie algorithmique, ingénierie logicielle, ingénierie des données, ingénierie des connaissances, ingénierie système et ingénierie des facteurs humains) pour garantir la conformité du système vis à vis des concepts d'emploi, des besoins et des contraintes du client et des utilisateurs, de définir des méthodes et outils pour sécuriser l'ensemble des phases de conception mais aussi pour garantir des propriétés de type fiabilité, sécurité et cybersécurité, et de maintenabilité du système et cela tout au long de son cycle de vie. L'enjeu industriel est alors d'outiller de bout en bout toute cette démarche de « génie de l'IA » prenant en compte les dimensions algorithmiques, logicielles et systèmes. Pour cela, la construction des méthodologies de développement et la mise en place d'un environnement les outillant permettra de répondre aux défis posés par l'intégration de composants ou sous-systèmes d'IA dans des systèmes critiques mais aussi d'en accélérer le déploiement. C'est l'ambition du programme Confiance.ai.

L'objectif de cet article est d'identifier les enjeux induits par l'industrialisation de l'IA pour les systèmes critiques et de présenter les principales composantes d'un environnement de confiance pour les développeurs et intégrateurs, les auditeurs internes ou externes. Cet environnement est un atelier d'ingénieries support à la conception, à la validation et au test en vue d'intégrer de l'IA dans les systèmes critiques tout en répondant aux exigences réglementaires des régulateurs.

2 Un environnement de confiance

Comme mentionné précédemment, cet environnement de confiance est constitué d'un ensemble d'outils opérables et fédérés et de méthodologies qui peuvent être interfacés, notamment au sein d'ateliers d'ingénierie industriels.

Concernant l'état de l'art actuel et à venir à court terme, il existe :

- D'une part, une kyrielle d'initiatives développant des briques technologiques pour l'IA de confiance. Ces briques n'implantent que partiellement les fonctionnalités précédemment identifiées. Elles doivent donc être enrichies (voir fig. 2).
- D'autre part, il existe aujourd'hui un certain nombre d'environnements de développement d'IA existants. Ces environnements sont des environnements outillés mais ne traitant pas des problématiques particulières de la confiance.

Par exemple, le développement de logiciels à base d'IA dans de nombreux secteurs d'activité (automobile, aéronautique, ferroviaire, défense, énergie, naval, santé, etc.) pose des questions de garantie de bon fonctionnement dès lors que ces logiciels prennent des décisions de manière autonome dans un contexte critique. Il faut donc être capable d'assurer la transparence et l'auditabilité de ces systèmes pour pouvoir les comprendre et les corriger le cas échéant, mais aussi parfois aller jusqu'à la preuve formelle de leur bon fonctionnement et ceci de manière industrielle comme mentionné précédemment. Pour cela, il est nécessaire de disposer d'outils d'**ingénierie de la donnée et de la connaissance** afin de pouvoir respecter certains principes comme la loyauté et/ou l'équité lors des étapes de collecte, d'acquisition, d'analyse, de manipulation, de qualification des jeux de données d'apprentissage mais aussi de bases de connaissances. Puis, l'**ingénierie algorithmique** [15] doit être enrichie afin de prendre en compte les spécificités de l'IA de confiance, démontrant que les fonctions implantées sont correctes, prévisibles, stables, reproductibles, explicables, fiables et robustes. Cette ingénierie doit prendre en compte l'incertitude induite par la dynamique de l'environnement dans lequel le système évolue. Enfin, il faut être capable de détecter les erreurs sur un domaine d'emploi défini, et in fine si se pose la question de sa spécification et de sa validation. C'est pourquoi, l'intégralité du processus d'**ingénierie système** doit être outillée.

Les fonctionnalités de l'environnement doivent supporter l'ensemble des usages pressentis (cf. fig. 2) regroupant :

- Un coeur de fonctionnalités pour aider à la conception, à la qualification au déploiement et à la maintenance des composants et des systèmes à base d'IA.
- Un ensemble de fonctionnalités proposant des méthodes outillées.
- Un ensemble de fonctionnalités transverses.
- Un ensemble de fonctionnalités permettant de s'interfacer.

Le coeur de ces fonctionnalités se décompose alors en deux sous-ensembles :

- Le premier pour aider à la conception, à la qualification, au déploiement et à la maintenance des composants d'IA : point de vue de l'ingénierie algorithmique de l'IA.
- Le second pour aider à la conception, à la qualifi-

1. Composé de 6 ministres, des administrations concernées (SGPI, DGE, DGRI), de deux opérateurs (ANR et Bpifrance) ainsi que de 6 personnalités reconnues, ce Conseil fixe les priorités stratégiques de la politique d'innovation française.

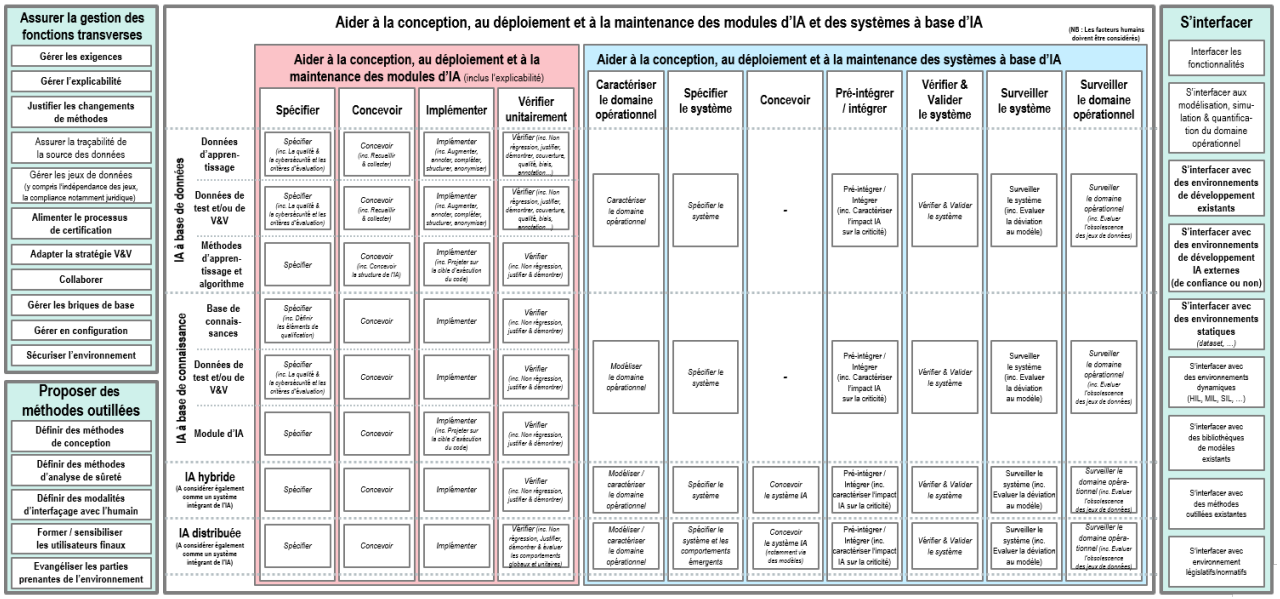


FIGURE 2 – La matrice de fonctionnalités de l'ingénierie de l'IA de confiance pour un déploiement opérationnel [Source Confiance.ai]

cation, au déploiement et à la maintenance des systèmes à base d'IA : point de vue de l'ingénierie système de l'IA.

Chacun de ces sous-ensembles est subdivisé en fonction du cycle de développement, soit :

- Pour la sous partie traitant des composants d'IA, nous trouvons les étapes suivantes : Spécifier, Définir le domaine opérationnel, Concevoir, Implémenter, Vérifier de manière unitaire, incluant bien sûr, en fonction des paradigmes de l'IA choisis, l'ingénierie des données et l'ingénierie des connaissances.
- Pour la sous partie traitant des systèmes à base d'IA (vue ingénierie système) nous trouvons les étapes suivantes : Caractériser le domaine opérationnel, Spécifier le système, Concevoir, Pré-intégrer et intégrer, Vérifier et Valider le système, Surveiller le système, Surveiller le domaine opérationnel, et Garantir son maintien en condition opérationnel.

L'ensemble des fonctionnalités proposant des méthodes outillées est alors constitué des éléments suivants :

- Définir des méthodes de conception,
- Définir des méthodes d'analyse de sureté,
- Définir des modalités d'interfaçage avec l'humain,
- Former / sensibiliser les utilisateurs finaux,
- Evangéliser les parties prenantes de l'environnement.

Il convient de préciser que les 3 premières fonctionnalités sont transverses et seront donc utilisées dans l'ensemble des autres fonctionnalités de l'environnement de confiance, puisqu'elles constituent le sous-jacent méthodologique.

Des fonctionnalités transverses permettent de :

- Gérer les exigences,
- Gérer l'explicabilité,
- Justifier les changements de méthode – Passage à une méthode basée sur de l'IA par opposition à une méthode traditionnelle,
- Assurer la traçabilité de la source de donnée,
- Gérer les jeux de données,
- Alimenter le processus de certification,
- Adapter la stratégie V& V,
- Collaborer – entre plusieurs acteurs via l'environnement,
- Gérer les briques de base – briques de composant d'IA dite de confiance
- Gérer en configuration – toute ou partie de l'environnement et des objets manipulés,
- Sécuriser l'environnement.

Enfin, cet environnement doit permettre de garantir le respect des normes et de la réglementation, comme par exemple, la conformité au RGPD via des approches respectant la vie privée (*Privacy by design*) ou avec la loi pour une République numérique qui induit transparence et loyauté. De plus, dans un contexte de montée de l'autonomie de certaines fonctions, ces ingénieries doivent être repensées pour prendre en compte les contraintes d'**embarquabilité** et la **relation humain-système**.

3 Ingénierie algorithmique de l'IA de confiance

Historiquement, la conception d'algorithmes d'IA émerge dans les années 1950 au travers de deux courants. L'**IA à base de connaissances**, qualifiée aujourd'hui de GOFAI (*Good Old Fashioned AI*) ou d'IA symbolique, se base quasi exclusivement sur le raisonnement symbolique et la logique. Elle se distingue de l'**IA dirigée par les données**, appelée aussi IA statistique et connexionniste, sous les feux de la rampe ces dernières d'années avec la collecte massive des données et l'arrivée de l'IA subsymbolique (et du deep learning), bien qu'aussi ancienne. Ainsi, l'IA symbolique

utilise des connaissances transmises à la machine pour résoudre des problèmes et l'IA dirigée par les données part d'exemples de solutions qu'elle essaie d'extrapoler par des méthodes statistiques. Leurs domaines d'emploi diffèrent. Alors que l'IA connexioniste est l'IA des sens, l'IA symbolique est celle du sens.
Plusieurs travaux cherchent à hybrider ces deux paradigmes, comme le souligne N. Asher [2] : "*L'addition de ces deux courants, IA symbolique et IA connexionniste, constitue le défi d'aujourd'hui*". Par exemple, l'apprentissage par renforcement consiste à récompenser les comportements souhaités et/ou à sanctionner les comportements non désirés avec des stratégies de récompense ou de sanction basées sur des connaissances métiers ou heuristiques issues de l'IA symbolique.

3.1 Conception algorithmique

Pour garantir une conception algorithmique de confiance (robuste, fiable...), l'ingénierie algorithmique (*Algorithm Engineering*) définit par P. Sanders [12, 16], doit intégrer les paradigmes induits par l'IA ainsi que les dimensions de (cyber)-sécurité et l'humain dans la boucle.
De plus, la sûreté et la sécurité des systèmes critiques à base d'IA nécessitent, de démontrer que les algorithmes sont corrects, c'est-à-dire qu'ils font ce qu'on attend d'eux. Il est donc nécessaire de vérifier la conformité entre leurs spécifications et leur comportement, autrement dit l'écart entre ce qu'il est supposé faire et ce qu'il fait réellement. Certaines approches en IA symbolique comme la programmation par contraintes offrent, par construction, cette propriété de correction, mais il reste nécessaire de la démontrer dans les autres cas comme pour l'IA connexionniste.
La robustesse d'un algorithme caractérise son aptitude à fournir des réponses correctes face à des situations inconnues ou à des malveillances. Cependant, cette propriété est plus dure que la précision. En effet, un système non précis ne peut être robuste. Mais surtout, un système précis peut ne pas être robuste. C'est le cas d'un système à base d'apprentissage ayant appris par coeur les données d'apprentissage qui se trompera dans ses décisions futures basées sur de nouvelles données. Ce phénomène est appelé *overfitting* (sur-apprentissage). De plus, l'IA reste vulnérable, et si l'on n'y prend pas garde, particulièrement sensible aux attaques dites "adversarial" (contradictoire), attaques qui tirent parti du fonctionnement des algorithmes sous-jacents pour générer des perturbations de faible amplitude dans les données analysées et et force l'IA à renvoyer un résultat incorrect. Heureusement, l'existence d'attaques 'adversarial' induit l'existence de défenses. De nombreuses défenses ont été proposées ces dernières années par la communauté scientifique [2] mais qui sont parfois réfutées avec de nouvelles attaques les rendant obsolètes. C'est pourquoi, il faut de se doter de méthodes et outils pour concevoir des algorithmes robustes et a minima caractériser leur robustesse.
Il est aussi nécessaire de prouver que les systèmes critiques sont contrôlables, c'est-à-dire qu'ils sont bien-fondés ou cohérents (on emploie aussi l'anglicisme consistant), si l'on peut prouver qu'ils ne font que ce qu'on l'attend d'eux. Les questions relatives aux problèmes de robustesse et de consistance commencent à faire l'objet de travaux liés aux preuves formelles. Ces dernières visent à apporter des garanties a priori sur la sûreté de fonctionnement d'un programme, contrairement aux méthodologies de validation par expérimentations directes qui visent à apporter des garanties a posteriori. Enfin, la compréhension de l'IA et de son raisonnement est nécessaire pour déterminer à quel point nous pouvons lui faire confiance. Un avantage des approches symboliques est de permettre de tracer le raisonnement. Mais même dans ce cas, il est nécessaire pour les usagers d'avoir une explication intelligible (explicabilité) plus que la traçabilité du raisonnement. Par contre, les approches connexionnistes s'apparentent aujourd'hui à des boîtes noires dont la complexité et l'abstraction qui sous-tendent ses décisions nous éloignent davantage de cette compréhension. Il devient alors nécessaire d'offrir des méthodes et outils pour rendre l'IA plus transparente, ouvrir les boîtes noires afin de comprendre comment un résultat a été atteint.

3.2 Vérification, validation et qualification

Lors de la vérification, la validation et la qualification du bon fonctionnement d'un algorithme d'IA, les situations suivantes doivent être abordées [11] :

- Le cas des composants livrés en boîte noire sur lesquels on cherchera principalement à en évaluer la robustesse. Par exemple, des approches ont été proposées dans la littérature [21, 20] présentant des méthodes pour étudier la robustesse de réseaux de neurones sur des problèmes de classification.
- Lorsque le composant est en boîte blanche (accès aux détails de sa structure, configuration, code source), il est alors possible de réaliser une analyse fine à l'aide de méthodes formelles (interprétation abstraite [5], Satisfiabilité modulo théories [8], programmation linéaire, etc.), mathématiques de ses comportements possibles. Un enjeu majeur restant la capacité de formaliser les propriétés de sûreté attendues afin que donner un sens fort aux preuves développées [6]. Cela permet, par exemple, de mettre en place des stratégies de test de robustesse face aux attaques adverses dans le cas d'approches à base d'apprentissage [13]. Il est également possible d'aller plus loin dans la caractérisation en définissant des domaines de stabilité.

Une voie prometteuse d'évaluation de la robustesse consiste à utiliser des approches de randomisation, à l'aide de bruits ajoutés de manière contrôlés à l'entrée du processus de décision, permettant de conduire à des notions de certificats statistiques de robustesse [3].

2. Nicolas Asher chercheur CNRS à l'Institut de recherche en informatique de Toulouse (IRIT) est le directeur scientifique du 3IA ANITI

4 Ingénierie des données et des connaissances

Pour les approches statistiques et connexionnistes, les données sont donc cruciales pour l'apprentissage, le test et la validation. Il ne suffit pas d'avoir beaucoup de données, il faut qu'elles soient de "bonne qualité" et représentatives du domaine d'emploi du système concerné, sans quoi ces approches donnent de mauvais résultats. De même, en IA symbolique, l'exploitation de connaissances de mauvaise qualité conduit à des résultats médiocres voire des erreurs. Il est nécessaire de repenser l'ingénierie des données et l'ingénierie des connaissances au regard de ces exigences.

De nouvelles méthodologies sont à définir pour une meilleure maîtrise des étapes d'acquisition, d'exploration, d'enrichissement, d'annotation et de préparation des données. Par exemple, la décomposition du jeu de données en plusieurs sous-ensembles dédiés aux différentes phase d'apprentissage, de validation et de test, doit respecter la représentativité du jeu de données pour permettre une bonne inférence en adéquation avec le domaine d'emploi.

De plus, comme les performances sont évaluées statistiquement sur un jeu de test préalablement constitué, la fiabilité de l'indice de performance est étroitement liée à la représentativité de ce jeu. La difficulté de cette décomposition réside dans la contrainte de constituer des ensembles distincts tout en garantissant qu'ils préservent des distributions comparables. De plus, il est nécessaire d'identifier automatiquement les situations qui mettent les systèmes en échec critique, et en retrouver le plus grand nombre possible parmi les données déjà acquises est nécessaire et difficile. Les techniques d'apprentissage actif (aussi appelé *machine teaching* [10]) n'y suffisent pas.

L'enrichissement permet de pallier la rareté des données. Cela consiste à ajouter artificiellement certaines données dans le jeu d'apprentissage ou de validation. Allant au-delà de la simple identification ou sélection intelligente d'outils, les techniques suivantes permettent d'augmenter la robustesse des modèles appris, ou de tester la robustesse lors de phase de validation :

- La génération artificielle de cas limites à base de réseaux neuronaux génératifs, pour créer de façon plausible de telles situations. Il sera par exemple possible de produire (et annoter) des situations rarissimes.
- L'utilisation de données réelles peut s'avérer complexe et le recours à des données synthétiques obtenues avec des simulateurs constitue une alternative intéressante.
- La création de nouvelles données à partir des données existantes, en appliquant par exemple, dans le cas de classification d'images, des transformations géométriques sur les images d'origine.

Mais aujourd'hui, les techniques d'apprentissage les plus efficaces sont supervisées reposant donc sur des annotations. La production d'annotations fiables est donc incontournable, puisque l'algorithme va ajuster ses paramètres afin d'associer une donnée d'entrée avec l'annotation cible. Cette phase a fait l'objet de nombreux travaux comme l'apprentissage actif ou l'automatisation de l'annotation par la création de fonctions d'annotation (supervision faible). De plus, caractériser la qualité d'un jeu de données n'est pas aisé. Il existe une pléthore de dimensions [14] qu'il faut choisir au regard d'un contexte décisionnel particulier. Même s'il existe très peu de normes relatives à la qualité des données [3], la question de la qualité de la donnée (*Data Quality* [7, 19]) n'est pas nouvelle : meilleure sera la qualité de la donnée, plus pertinente sera la décision. Dans son programme "*Total Data Quality Management*" (TDQM), le MIT s'attaque à cette question depuis le début des années 1990 [17], identifiant ainsi de nombreuses dimensions telles que la précision, la pertinence, la couverture, la complétude, la crédibilité, la cohérence, la fiabilité...

Enfin, si les données qui nourrissent les algorithmes à base d'apprentissage sont biaisées, les décisions que ceux-ci prendront le seront également. L'identification de ces biais peut poser question car l'une des difficultés consiste à comprendre comment un modèle généralise l'apprentissage qu'il a effectué à partir des données d'entraînement. C'est pourquoi, l'ingénierie des données doit être outillées pour permettre en particulier d'identifier les biais d'échantillonnage, d'enregistrement, de nettoyage, d'exclusion [4], induits par les transformations d'ingénierie des caractéristiques (*Feature Engineering*), voire de confirmation [5].

Les systèmes à base de connaissances, quant à eux peuvent représenter et traiter des principes et des règles de décisions, des taxonomies, des théories, des processus et des méthodes mémorisées dans un système artificiel. Mais pour concevoir un système à base de connaissances ayant un comportement compréhensible et acceptable par l'utilisateur passe par une modélisation à un niveau d'abstraction pertinent qui fait sens pour les différents acteurs impliqués dans sa conception (experts métiers, utilisateurs, etc.). En phase d'utilisation du système, le modèle est rendu opérationnel de manière à ce que l'utilisateur s'approprie le comportement du système et puisse interagir avec lui. L'ingénierie des connaissances (IC) propose des concepts, méthodes et techniques permettant de modéliser et/ou d'acquérir les connaissances dans des domaines ou la formalisation est difficile ou la compréhension des phénomènes partielle. L'IC peut être schématiquement définie par trois étapes : l'acquisition de connaissances disponibles, leur représentation informatique et l'utilisation de celles-ci à des fins de simulation, de prédiction, de validation, d'optimisation pour aider à la décision [18]. Rappelons que l'extraction des connaissances couvre le processus permettant de transformer les connaissances des experts dans un domaine sous forme d'informations organisées, alors que l'acquisi-

3. norme ISO 8000 relative à la qualité des données de référence – Master data

4. le biais d'exclusion provient de données qui sont retirées de manière inappropriée de la source de données.

5. le biais de confirmation est le désir de sélectionner uniquement les informations qui soutiennent ou confirment quelque chose que vous connaissez déjà, plutôt que des données qui pourraient suggérer quelque chose qui va à l'encontre d'idées préconçues.

tion des connaissances est le processus inverse qui consiste à transformer, par l'apprentissage, les informations et les savoirs disponibles en connaissances. Là encore, il est important de se doter d'une démarche méthodologique pour garantir la complétude, à la pertinence et à la qualité des modèles.

5 Evaluation de la qualité de l'algorithme

Évaluer les performances d'une IA dirigée par les données, consiste à évaluer la qualité d'une fonction, apprise selon des principes d'apprentissage statistique, lorsqu'elle sera déployée. Si la théorie donne un cadre clair à l'évaluation du risque théorique, sa mise en pratique implique de définir la notion de risque empirique qui s'appuie sur deux concepts : d'une part la distribution réelle des données n'est pas connue, elle est remplacée par un ensemble de données, ou une distribution approchée ; d'autre part elle repose sur la définition d'une fonction de coût, qui doit au mieux retranscrire l'intention finale. Dans le cadre strict de l'évaluation des performances, les deux problèmes principaux sont donc : 1) comment choisir la bonne métrique d'évaluation ; 2) quelle méthodologie pour l'estimation robuste de cette métrique de performance. Dans ce cadre, un guide d'évaluation a été rédigé par la DGA [11] pour les approches d'apprentissage supervisé. À ces deux problèmes issus de la nature intrinsèque de l'apprentissage statistique, il faut ajouter la question de la reproductibilité des performances rapportées, vis-à-vis de paramètres considérés jusqu'ici comme mineurs. Il faut aussi noter que de nombreux projets s'attellent à la question de l'évaluation ou de l'explication. Citons les travaux issus du programme DEEL (France et Canada) pour le cadre IA des données ou du "GT Explicabilité du GDR IA" pour l'IA symbolique.

Enfin, le changement radical des pratiques de développement des systèmes à base d'IA et la complexité induite pour leur validation, amènent à envisager l'introduction d'approches de qualification et de certification plus souples pour faire face aux différents types d'incertitude que présentent ces systèmes. Outre la définition de référentiels de risques spécifiques liés à l'IA, deux approches de la qualification semblent particulièrement intéressantes : (1) la qualification basée sur des propriétés globales du système [4, 1], offrant plus de souplesse dans la manière de gérer la complexité et l'implantation des pratiques de qualification ("*Assurance Case*" qualification based on "system overarching properties" satisfaction) ; et (2) la qualification modulaire, incrémentale et évolutive, par exemple via des approches par contrat, permettant de prendre en compte l'évolution nécessaire des systèmes liées aux évolutions des données, connaissances et de l'environnement qui risquent d'être beaucoup plus rapides pour l'IA.

6 Conclusion

Sécuriser, certifier et fiabiliser les systèmes qui ont recours à l'intelligence artificielle posent des questions d'ingénierie algorithmique, d'ingénierie des données et des connaissances, d'ingénierie système, de la sûreté et de la (cyber)-sécurité, mais aussi d'ingénierie des facteurs humains, dès lors que ces logiciels prennent des décisions de manière autonome dans un contexte critique. Le programme "Confiance.ai" du Grand Défi national a pour objectif de définir et d'outiller une approche rigoureuse et interdisciplinaire en formalisant l'ensemble du cycle de vie de ces systèmes à base d'IA de confiance [9].

Les besoins principaux auxquels l'environnement devra répondre, in fine, sont les suivantes :

- Disposer de méthodes et d'outils de gestion des données et des connaissances : conception, d'analyse, manipulation, collecte, acquisition, qualification, génération, filtrage des jeux de données d'apprentissage et base de connaissances pour la validation des systèmes cibles.
- Capacité à produire (concevoir, valider, implanter) un algorithme d'intelligence artificielle dit de confiance : correct, prévisible, stable, reproductible, explicable, fiable, robuste, capable de détecter les erreurs sur un domaine d'emploi défini et maîtrisé et donc in fine et si nécessaire certifiable.
- Capacité à définir et outiller l'intégralité du processus de développement, d'intégration et de qualification/certification sur l'ensemble du cycle de vie des systèmes intégrant de l'IA en interopérabilité avec les autres environnements de conception.
- Sortir d'une approche basée uniquement sur les preuves de concepts et passer à l'échelle industrielle en revisitant et repensant la chaîne d'ingénierie de l'algorithme, du logiciel et du système ainsi que la prise en compte du hardware pour le développement de composants à base d'IA.

Pour appuyer la spécification des besoins, valider et caractériser les solutions, le programme s'appuie sur un ensemble de cas d'usages ciblés et partagés par l'ensemble des acteurs. On peut citer notamment :

- Compréhension de scène pour la mobilité autonome à partir d'un capteur caméra 2D ;
- Surveillance et détection de déviation de l'efficacité opérationnelle d'une usine ;
- Contrôle embarqué par réseaux de neurones pour une fonction d'anticollision en vol ;
- Détection de conformité de cordons de soudure par inspection visuelle ;
- Maintenance prédictive des propulseurs de navire.

Des cas d'usage complémentaires seront intégrés au cours du programme afin d'aborder les volets techniques complémentaires, notamment autour de l'IA à base de connaissances et hybride.

Remerciements

Les partenaires de Confiance.ai sont par ordre alphabétique : Airbus, Air Liquide, ATOS, CEA, Inria, IRT SystemX, IRT St Exupéry, Naval Group, Renault, Safran, Sopra Steria, Thales et Valéo

Références

[1] Darpa program assured autonomy, https ://www.darpa.mil/program/assured-autonomy.

[2] A. Araujo, L. Meunier, R. Pinot, and B. Negrevergne. Robust neural networks using randomized adversarial training. *arXiv preprint arXiv :1903.10219*, 2019.

[3] J. Cohen, E. Rosenfeld, and JZ. Kolter. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv :1902.02918*, 2019.

[4] E. Denney, G. Pai, and I. Habli. Dynamic safety cases for through-life safety assurance. In *IEEE/ACM 37th IEEE Int. Conf. on Software Engineering*, volume 2, pages 587–590. IEEE, 2015.

[5] T. Gehr, M. Mirman, D. Drachsler-Cohen, et al. Ai2 : Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2018.

[6] J. Girard-Satabin, G. Charpiat, Z. Chihani, and M. Schoenauer. CAMUS : A framework to build formal specifications for deep perception systems using simulators. In *24th European Conference on Artificial Intelligence (ECAI)*, 2020.

[7] Y. Huh, F. Keller, Thomas C. Redman, and A. Watkins. Data quality. *Information and software technology*, 32(8) :559–565, 1990.

[8] G. Katz, D. Huang, D. Ibeling, K. Julian, et al. The marabou framework for verification and analysis of deep neural networks. In *Int. Conf. on Computer Aided Verification*, pages 443–452. Springer, 2019.

[9] J. Mattioli, F. Terrier, L. Cantat, J. Chiaroni, M. Barreteau, Y. Bonhomme, C. Guettier, and C. Alix. Ia de confiance : condition nécessaire pour le déploiement de l'ia dans les systèmes de défense - hal id : hal-02955575, 2020.

[10] S. Mei and X. Zhu. Using machine teaching to identify optimal training-set attacks on machine learners. In *AAAI Conf. on Artificial Intelligence*, 2015.

[11] DGA (French MoD). Guide méthodologique pour la spécification et la qualification des systèmes intégrant des modules d'intelligence artificielle - version 1.0 b, 2019.

[12] M. Muller-Hannemann and S. Schirra. *Algorithm engineering : bridging the gap between algorithm theory and practice*. Springer-Verlag, 2010.

[13] MI. Nicolae, M. Sinn, MN. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig, et al. Adversarial robustness toolbox v1. 0.0. *arXiv preprint arXiv :1807.01069*, 2018.

[14] L. Pipino, Y. Lee, and R. Wang. Data quality assessment. *Communications of the ACM*, 45(4) :211–218, 2002.

[15] P. Sanders. Algorithm engineering–an attempt at a definition. In *Efficient Algorithms*, pages 321–340. Springer, 2009.

[16] P. Sanders. Algorithm engineering–an attempt at a definition using sorting as an example. In *12th Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 55–61. SIAM, 2010.

[17] F. Sidi, P. Panahy, L. Affendey, M. Jabar, H. Ibrahim, and A. Mustapha. Data quality : A survey of data quality dimensions. In *2012 Int. Conf. on Information Retrieval & Knowledge Management*, pages 300–304. IEEE, 2012.

[18] R. Studer, VR. Benjamins, and D. Fensel. Knowledge engineering : principles and methods. *Data & knowledge engineering*, 25(1-2) :161–197, 1998.

[19] R.Y Wang and D.M Strong. Beyond accuracy : What data quality means to data consumers. *Journal of management information systems*, 12(4) :5–33, 1996.

[20] L. Weng, PY. Chen, L. Nguyen, M. Squillante, A. Boopathy, I. Oseledets, and L. Daniel. Proven : Verifying robustness of neural networks with a probabilistic approach. In *Int. Conf. on Machine Learning*, pages 6727–6736, 2019.

[21] H. Zhang, TW. Weng, PY. Chen, and others. Efficient neural network robustness certification with general activation functions. In *Advances in neural information processing systems*, pages 4939–4948, 2018.

L'Intelligence Artificielle pour l'accompagnement du raisonnement expert, la solution Khresterion

F. Soualah-Alila[1], M. Ben Ellefi[1], D. Pierre[1]

[1] Pôle R&D Khresterion
59 rue des Petits-Champs, 75001 Paris

Résumé

Les assureurs santé font face à une énorme quantité de données mouvantes : le recours à l'automatisation de leur traitement est désormais indispensable. L'objectif de notre travail est de réduire les délais et les coûts d'accès à l'expertise : la gestion des diagnostics, l'accompagnement à la décision et la rédaction des documents conformes aux contraintes métier et juridiques. Notre approche consiste à proposer aux assureurs un outil, non seulement de gestion de leurs traitements, mais aussi de garantie de conformité de ces derniers. Dans cet article, nous présentons nos travaux sur la modélisation des connaissances du domaine de l'assurance santé et notre méthode de raisonnement automatique permettant de reproduire un raisonnement expert basé sur des règles du métier. Un exemple de scénario de raisonnement sur les expressions de garanties dans un tableau d'assurance santé est détaillé dans cet article.

Mots-clés

Graphe de connaissances, Assurance santé, ontologies, règles, raisonnement, validation, SHACL-SPARQL.

Abstract

Health insurers faces a huge amount of moving data : the need to automate their processes is now essential. The objective of our work is to reduce expertise time and costs : diagnostic management, decision support and drafting documents in accordance with business and legal constraints. Our approach consists on proposing to insurers a tool for managing their processing and guaranteeing their compliance. In this article we present our work on knowledge modeling in the domain of health insurance, as well as our automatic reasoning process to reproduce expert reasoning based on facts and buisness rules. An example of a reasoning scenario based on guarantees expressions applied to a health insurance table is detailed in this paper.

Keywords

Knowledge Graph, Health insurance, ontologies, rules, reasoning, validation, SHACL-SPARQL.

1 Introduction

L'Intelligence Articielle (IA) entretient des échanges fructueux avec les sciences cognitives, d'une part elle fournit de nouveaux repères, points de comparaison pour la compréhension de l'intelligence, d'autre part elle peut s'inspirer de ce que l'on sait du fonctionnement du cerveau et de la façon dont l'homme raisonne, même si rien ne dit que l'IA doit copier l'intelligence humaine dans toutes ses manières de procéder [4]. Dans un domaine métier spécifique, la spécification de la conceptualisation de ce domaine consiste à rendre l'expertise métier compréhensible par le système sous forme d'un graphe de connaissance, connu sous le nom de Knowledge Graph (KG), afin d'assurer une IA basée sur les connaissances (IA symbolique).

Khresterion [1] est spécialiste de deux thématiques importantes en Intelligence Artificielle : la représentation des connaissances et le raisonnement automatique.

Le processus de modélisation des connaissances expertes peut être représenté comme un processus assurant la traduction de la connaissance détenue par un expert vers une forme exploitable par un système informatique et appliqué à un domaine particulier. La formulation de cette connaissance consiste en l'élaboration continue d'un modèle qui se réalise au cours d'cycle de cinq étapes principales que sont la collecte, la formalisation, l'implémentation, la validation et la correction des connaissances (figure 1).

Les technologies du Web Sémantique implémentées en graphe de connaissances offrent une solution pour faciliter l'intégration et l'interopérabilité des données. On parle alors de modélisation en schémas ontologiques permettant de mettre en œuvre des mécanismes de raisonnements.

Dans notre cas, le modèle métier sous-jacent exploité est celui de la construction d'un ensemble d'ontologies, dont des ontologies du domaine de l'assurance santé, permettant la définition d'un vocabulaire adaptable et évolutif. Nos ontologies sont complétées par des règles transcrivant les bonnes pratiques du métier en modèle logique. Nous utilisons à cet effet le langage SHACL-SPARQL [2] qui consiste en une collaboration de deux mécanismes : (i) SHACL [3] (Shapes Constraint Language) qui permet de valider des graphes RDF [4] (Resource Description Framework) avec un ensemble de conditions (ii) et SPARQL [5] (SPARQL Protocol and RDF Query Language) un standard du W3C pour le requêtage avec un noyau de graph pattern matching.

1. https://khresterion.com/
2. https://www.w3.org/2014/data-shapes/wiki/Shacl-sparql
3. https://www.w3.org/TR/shacl/
4. https://www.w3.org/RDF/
5. https://www.w3.org/TR/sparql11-overview/

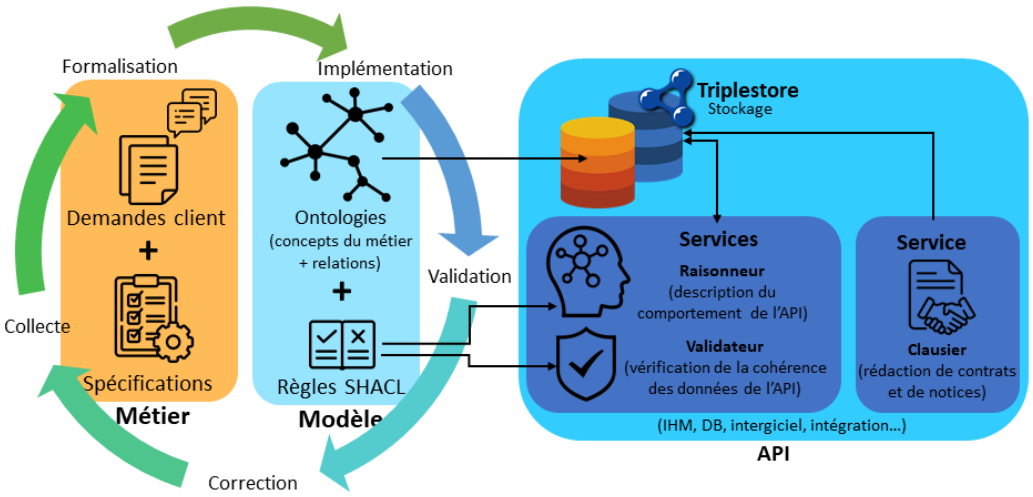


FIGURE 1 – Cycle de vie des modules Khresterion

Le choix de ce mécanisme de raisonnement se justifie par le fait que, parmi nos scénarios de raisonnement, il arrive qu'avec une seule règle le raisonneur doive inférer un gros volume de nouvelles instances (ou individus) à injecter dans le graphe de connaissances. Par exemple, dans notre système, le choix de la date d'effet d'une offre santé déclenche un appel au raisonneur. Le raisonneur infère alors en fonction de cette date de nombreuses instances qui correspondent aux différentes garanties santé de cette offre. La plupart des raisonneurs OWL-DL [6] (Ontology Web Language Description Logics) ne permettent pas la création de nouvelles instances dans la base de connaissances. Une exception est relevée par le langage de règles SWRL [7] (Semantic Web Rule Language) qui permet d'ajouter des fonctions (ou built-ins) dans son formalisme, notamment *swrlx :makeOWLThing*. À notre connaissance aucun raisonneur OWL-DL, à l'exception du raisonneur SWRLTab [8], n'exploite cette fonctionnalité SWRL. Par ailleurs, le raisonneur SWRLTab n'est pas utilisable dans notre cas car il ne permet pas d'ordonner l'exécution des règles, à l'inverse de SHACL avec l'utilisation de *sh :order*.

Dans la version actuelle du langage SHACL, la création des instances n'est pas encore implémentée directement, d'où le besoin d'augmenter ce dernier par du SPARQL CONSTRUCT. Les requêtes SPARQL CONSTRUCT permettent ainsi d'ajouter une flexibilité pour la création de nouvelles URIs et les injecter comme nouvelles instances dans le graphe de connaissances.

Le modèle métier est ensuite intégré dans notre moteur de raisonnement. Ce dernier permet d'exploiter nativement un mécanisme logique de subsomption, de gérer des règles générales et des exceptions pour des cas spécifiques, de proposer des choix multiples et d'explorer des alternatives.

Dans cet article, nous présentons le système de raisonnement Khresterion qui assure la **conformité** des garanties santé dans le cadre d'un contrat d'assurance. Notre moteur de raisonnement permet en plus, et à la différence des systèmes de raisonnements classiques, d'exploiter des formes logiques qui tolèrent la présence de contradictions [5]. Alors que les systèmes experts classiques n'interviennent que sur une logique décisionnelle pure, notre moteur de raisonnement demeure ainsi une assistance à la prise de décision et non un outil de raisonnement à contraintes.

L'article est organisé de la manière suivante : la section 2 introduit le graphe de connaissances de Khresterion et les modes de raisonnements, la section 3 présente une démonstration sous forme d'un scénario détaillé de raisonnement sur des expressions de garanties pour une assurance santé, enfin la section 4 présente une démonstration de règles de validation pour assurer la cohérence et la conformité du système avant de conclure.

2 Le graphe de connaissance de Khresterion

Un des atouts de la technologie Khresterion est d'accéder à toutes les sources existantes, de consolider les données pertinentes et de créer une seule base de connaissances référentielle : le graphe de connaissance de Khresterion, *KGK* (Knowledge Graph Khresterion).

Le Knowledge Graph, littéralement graphe de connaissances en français, est un terme officialisé par Google en 2012. La définition de ce terme reste controversée : un certain nombre de définitions générales ou bien techniques, parfois même contradictoires, ont émergé. L'article [1] liste et analyse les différentes définitions de ce terme. Nous adoptons la définition suivante, qui semble la plus adaptée à notre vision : un graphe de connaissances est une représentation de la connaissance relative à un domaine sous une forme facilement exploitable par la machine. Il est constitué d'entités et de relations entre ces entités, formant un graphe de connaissances stocké dans des bases de données de type graphe. Bien que cette représentation graphique de la connaissance ne soit pas récente, elle a gagné en popularité et est aujourd'hui un élément clé pour des applica-

6. https://www.w3.org/TR/owl2-primer/
7. https://www.w3.org/Submission/SWRL/
8. https://github.com/protegeproject/swrltab

tions d'Intelligence Artificielle liées à la recherche rapide et contextuelle d'information ainsi qu'à la prise de décision.
De cette définition se dégagent ces éléments justifiant notre choix pour ce modèle :

- des données facilement accessibles et compréhensibles car organisées selon des vues métiers,
- des ressources directement exploitables par les outils d'Intelligence Artificielle et de raisonnement logique pour inférer de nouvelles connaissances,
- une grande facilité d'enrichissement du graphe de connaissances même avec des données hétérogènes,
- une interopérabilité des données favorisée par la gestion d'identifiants stables et d'alignement avec des ressources externes,
- un environnement et des standards construits nativement sur les standards du Web.

Le modèle KGK se base essentiellement sur deux couches de l'architecture du Web Sémantique qui sont les couches Ontologie et Règles [2]. La construction de notre référentiel, consiste donc à définir un ensemble comportant ontologies, contraintes et actions adéquates à un usage décrit par des faits.

Ontologies Khresterion :
La première couche du référentiel correspond aux connaissances métier conceptualisées sous la forme d'ontologies. Une ontologie permet d'analyser un domaine de connaissances et de réutiliser ces connaissances. Une ontologie comprend un vocabulaire commun aux experts et une représentation des relations entre ces termes, qui définissent cette connaissance. Elle est utilisable par les humains, comme par les machines [7].
La figure 2 illustre un aperçu des différentes ontologies modélisées par Khresterion pour la représentation des connaissances du domaine de l'assurance santé. Toutes ces ontologies sont la propriété intellectuelle de Khresterion et ont été conçues en collaboration avec des clients assureurs, en nous basant sur des documentations internes et des spécifications fonctionnelles.
Nous distinguons trois niveaux d'ontologies :

- **Une ontologie générique :** elle regroupe des concepts, des relations et des données génériques, indépendants du domaine de l'assurance santé. Ces éléments décrivent des notions universelles ou des concepts généraux et abstraits tels que la modélisation du temps et de l'espace. Ces éléments sont applicables à plusieurs domaines, et dans notre cas exploités par les ontologies de domaine ci-dessous.
- **Des ontologies de domaine** : elles décrivent des connaissances bien spécifiques au domaine de l'assurance santé : description des garanties d'assurances, des établissements d'assurances, des bénéficiaires, des différents documents à éditer sous forme de contrats ou notices, etc. Elles peuvent se décliner en versions spécifiques à un pays afin d'intégrer des définitions locales.
- **Des ontologies d'applications :** elles caractérisent une conceptualisation encore plus spécifique que les ontologies de domaine pour un usage ou un client précis. Nous modélisons par exemple dans ces ontologies les coordonnées des directions commerciales d'un client en particulier ou encore des expressions de garanties particulières à un client, etc.

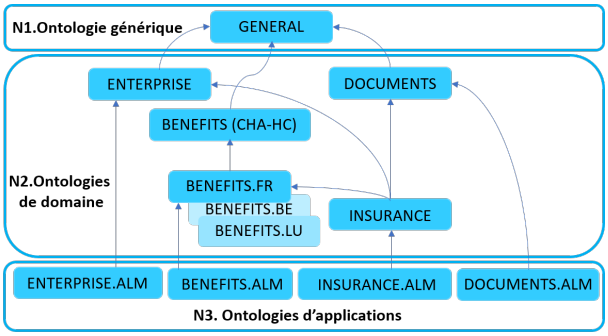


FIGURE 2 – Aperçu des ontologies Khresterion

La gestion de ces multiples ontologies est confiée à un gestionnaire de connaissances, qui regroupe l'ensemble des concepts et relations de ces ontologies sous une même hiérarchie, formant ainsi la TBox (Terminology Box)[3] de notre Knowledge Graph. À la TBox nous associons une population ABox (Assertions Box)[3] décrivant les individus et leurs relations (quel individu appartient à quelle entité, quel individu est lié à quel autre, à travers quelle relation). TBox + ABox sont stockées dans une base de données RDF de type triple-store.

Règles Khresterion :
La deuxième couche du référentiel correspond à un ensemble de règles de cohérence stockées dans un gestionnaire de règles, capable d'exécuter ces règles à la demande afin de gérer la cohérence du domaine.
Les règles capitalisent les connaissances d'un expert, la réglementation et traduisent les contraintes du métier (règles de bonne pratique, de fonctionnement ...). L'une des difficultés consiste à collecter et traduire les textes réglementaires qui n'existent souvent que sous-forme rédigée en un système cohérent et complet de règles formalisées dans un langage compréhensible par la machine.
Nous définissons trois familles de règles :

- **Des règles d'inférences :** elles sont classiquement exprimées sous la forme *Si...Alors...* Ce type de règles de cohérence n'a d'effet que si toutes les prémisses d'une règle sont vraies, c'est à dire si les informations qu'ils représentent sont présentes. Nous utilisons par exemple ces règles pour proposer à l'utilisateur des *expressions de garanties* de manière adaptative et contextuelle.
- **Des règles de calcul :** elles sont utilisées pour calculer certaines informations, qu'elles soient numériques, datées ou bien des chaînes de caractères. Les opérations de calcul sont représentées dans des

règles de cohérence classiques auxquelles sont attachées des formules de calcul. Nous utilisons par exemple ces règles pour calculer *des montants de cotisations sociales*.

— **Des règles de validation :** elles sont utilisées pour vérifier si l'environnement d'exécution contient une information vérifiant une contrainte ou non. Nous utilisons par exemple ces règles pour vérifier si une valeur saisie par l'utilisateur est cohérente ou non dans un certain contexte : vérifier la validité d'un numéro de téléphone, vérifier la cohérence d'une valeur dans une *expression de garanties*, identifier une valeur manquante, etc.

Ces ensembles de règles sont définis en SHACL. Ce nouveau standard recommandé par le W3C, permet de décrire explicitement la forme, *Shape*, du schéma RDF attendue par les machines. SHACL comprend, entre autres, des fonctionnalités permettant d'exprimer des conditions qui limitent le nombre de valeurs qu'une propriété peut avoir, le type de celles-ci, les plages numériques, les modèles de correspondance de chaîne et les combinaisons logiques de certaines contraintes. Seulement, dans la plupart des cas, nos règles expriment des contraintes un peu plus complexes. Nous avons alors rédigé nos règles en SHACL mais en incluant un mécanisme d'extension en SPARQL.
Le gestionnaire de règles est en écoute du système Khresterion lors du changement au niveau du KG. En cas de violation des shapes SHACL-SPARQL, le gestionnaire produit un certain nombre de rapports sous la forme de messages d'informations, de différents niveaux de sévérité, permettant d'accompagner l'utilisateur dans ses prises de décisions. Ces messages, accompagnés de codes couleurs, peuvent être de type conseil (couleur vert), dé-conseil (couleur orange), violation (couleur rouge) ou incomplétude (couleur violet) (figure 3).

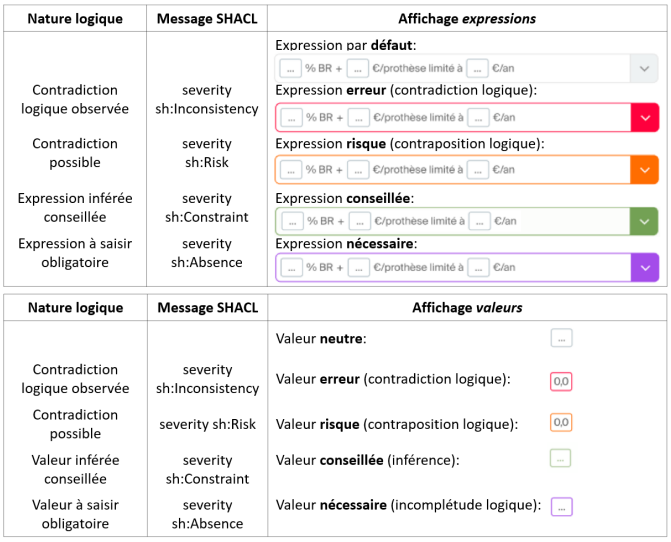


FIGURE 3 – Niveaux de sévérité SHACL-SPARQL

L'utilisation de ces règles contribue efficacement à la qualité des données stockées.
Dans les sections 3 et 4 nous présentons quelques exemples de ces règles appliquées à un scénario de raisonnement sur des expressions de garanties.

3 Scénario de Raisonnement : Cas d'un Tableau de Garantie Santé

Cette section présente une démonstration d'un scénario de raisonnement sur un tableau de garantie santé dans le système Khresterion.
Un *tableau de garanties* d'une assurance santé définit un montant de remboursement pour un poste de soins et un jalon (ou niveau) de remboursement. Par exemple, un poste *Lentilles* est ainsi associé à un certain *remboursement* pour un jalon dit de *base*, mais peut bénéficier d'un niveau de remboursement supérieur pour un niveau *optionnel* ou *surcomplémentaire*.
Les données experts sur des *garanties assurance santé pour les lentilles* considèrent deux types de lentilles : celles qui sont remboursées par la sécurité sociale et celles qui ne le sont pas. Le modèle Khresterion définit dans ce cas : (i) le concept $Lentilles$ pour représenter le type de garantie, et (ii) les deux concepts $Actes\_et\_prestations\_pris\_en\_comptes\_par\_la\_SS$ et $Actes\_et\_prestations\_Non\_pris\_en\_comptes\_par\_la\_SS$ pour qualifier les deux types de remboursement sécurité sociale. Ainsi les lentilles seront représentées dans KGK comme suit :

— Lentilles acceptées par la sécurité sociale (ASS) est une instance RDF de type $Lentilles$ et $Actes\_et\_prestations\_pris\_en\_comptes\_par\_la\_SS$

— Lentilles non acceptées par la sécurité sociale (NASS) est une instance RDF de type $Lentilles$ et $Actes\_et\_prestations\_Non\_pris\_en\_comptes\_par\_la\_SS$

Le modèle Khresterion conceptualise aussi les différents niveaux de garanties (bases, options et surcomplémentaires) et leurs qualifications (responsable, non responsable, facultative, obligatoire, y compris le remboursement sécurité sociale, y compris le régime de base, etc). Par exemple, le niveau de garantie base peut être exprimé en y compris le remboursement de la sécurité sociale (YCRSS), alors que le deuxième niveau de garantie, sa surcomplémentaire, peut être exprimé en y compris le régime de la base (YCRB).
Pour faire le lien entre le poste de garantie (les lentilles dans notre scénario), le niveau de remboursement et l'expression sélectionnée, une instance de type $ReimbursementLevel$ assure la liaison entre ces différents éléments. La figure 4 illustre, par exemple, le cas d'une lentille ASS liée à un paiement. Le paiement la relie à un niveau de garantie de base exprimé en y compris le remboursement sécurité sociale (c.f. l'instance milestone), et la relie aussi à l'expression de remboursement correspondante.
Si un tableau de garanties contient plusieurs niveaux de remboursements (i.e. base + surcomplémentaire), le poste

sera lié à deux instances différentes de paiement dont chacune est liée au niveau de garantie et expression correspondants. Une telle conceptualisation est particulièrement utile pour la modélisation des règles d'inférence sur les expressions possibles par poste et par niveau de remboursement. Il existe, en effet, de nombreuses contraintes, de nature réglementaire ou métier, qui définissent les expressions de remboursement possibles pour un poste et un niveau donnés en fonction de l'expression sélectionnée pour ce même poste et un niveau précédent.

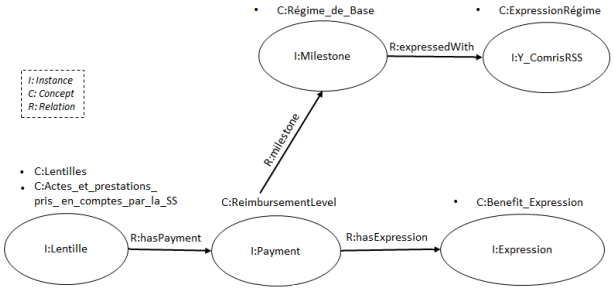


FIGURE 4 – Echantillon du Knowledge Graph Khresterion pour la garantie de base YCRSS au niveau des lentilles ASS

Le système Khresterion propose à l'utilisateur une liste d'expressions conformes selon le poste et le niveau de remboursement : pour les lentilles ASS au niveau de la base YRSS, l'utilisateur peut choisir entre 3 expressions possibles. Le choix de l'expression de base aura une conséquence sur l'inférence du niveau surcomplémentaire correspondant. Par exemple, si l'utilisateur choisit une expression "X % de la BR" dans la base il aura trois expressions proposées par le raisonneur Khresterion dans la surcomplémentaire correspondante : "X % de la BR" , "X % de la BR + un crédit de Y € par année civile" ou "X % de la BR + un crédit de Y % du PMSS par année civile". Ainsi, le système assure par raisonnement la conformité du choix utilisateur par rapport aux règles métiers. Suite au choix de l'utilisateur dans le régime de base, le système rafraîchit son raisonnement pour inférer les expressions possibles dans la surcomplémentaire correspondante tout en respectant la cohérence du KGK. Ce raisonnement est assuré par un ensemble de règles d'inférence SHACL étendu par SPARQL comme expliqué dans la section 2.

4 Scénario de Validation : Cas d'un Tableau de Garanties Santé

En plus des règles d'inférence, pour obtenir la conformité expert, le système Khresterion applique des règles de validation pour vérifier la cohérence des valeurs saisies par l'utilisateur. Par exemple pour les lentilles ASS dans la base YCRSS, le "X % de la BR" doit valider la règle "100 $\leq$ X $\leq$ 1000".
Nous étendons SHACL par du SPARQL pour bénéficier de toute la performance des techniques du graph matching apportée par ce langage de requêtage.

La figure 5 illustre un exemple de la représentation d'une expression dans KGK. Une expression peut avoir un ou plusieurs composants. Chaque composant est constitué d'une unité (exemple Euros), d'une valeur, d'au maximum un opérateur de liaison (exemple, +, -, limité à) et d'au maximum un quantificateur (exemple, crédit, forfait).

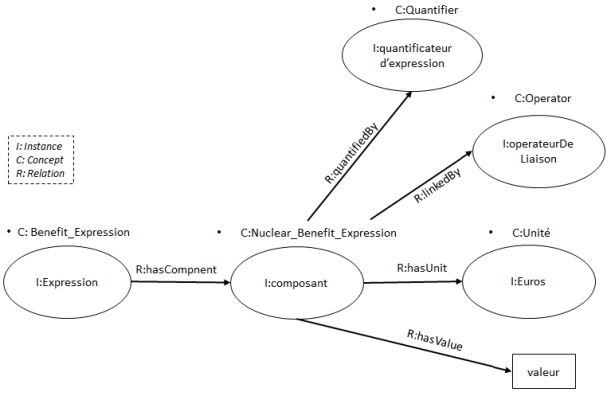


FIGURE 5 – Instance d'une expression dans KGK

Ce niveau de détails dans la modélisation des expressions dans KGK nous permet d'avoir un contrôle total pour assurer la conformité de tous ces composants. La validation SHACL-SPARQL exécutée par le biais du graph matching (par exemple, le parcours du graphe constitué de l'ensemble des noeuds des figure 4 et figure 5) permet de contrôler les valeurs des composants d'une expression d'un poste pour un niveau de garantie spécifique. Par exemple, pour l'expression "X % de la BR" (Base $\leq$ X) des lentilles ASS dans la surcomplémentaire YCRB, si pour le même poste dans le régime de base nous avons "100 % de la BR", alors le raisonneur applique la règle de validation pour vérifir si la valeur du régime surcomplèmentaire est bien supérieure ou égale à la valeur de base.
En plus du contrôle sur les valeurs, le langage SHACL permet de programmer dans les Shapes des messages à afficher à l'utilisateur en exploitant la propriété $sh : severity$. Différents codes couleurs sont attribués selon la sévérité modélisée dans la règle. Exemple, l'expression "80 % de la BR" pour les lentilles ASS dans la base YCRSS va donner une alerte sur l'expression de couleur orange signifiant que l'état de cette expression est déconseillé par rapport à la conformité expert exigeant "100 $\leq$ X". La liste des niveaux de sévérité et des couleurs correspondant est détaillée dans la Section 2.
À la fin de la saisie, l'utilisateur a la possibilité d'éditer son contrat d'assurance. La figure 6 illustre un aperçu de la représentation des expressions de remboursement du poste lentilles dans un contrat. Notons ici que le raisonneur applique des règles de calcul de libellés pour composer les textes des expressions et les textes des contrat d'assurance. Par exemple, des règles de calcul sont appliquées pour la construction du libellé "120 % de la BR + crédit de 400 € par année civile" à partir des différents composants présents dans la figure 5.

Autres dispositifs médicaux optique		
Lentilles acceptées par la SS	100 % de la BR	120 % de la BR + crédit de 400 € par année civile
Lentilles refusées par la SS (y compris lentilles jetables)	Crédit de 300 € par année civile	Crédit de 400 € par année civile
Chirurgie réfractive (Myopie, hypermétropie, astigmatisme, presbytie)	Crédit de 400 € par oeil par année civile	Crédit de 80 % du PMSS par oeil par année civile

FIGURE 6 – Exemple des clauses pour les lentilles dans un contrat

5 Conclusion

Les Knowledge Graphs servent de substrat commun de connaissances au sein d'une organisation ou d'une communauté, permettant la représentation, l'accumulation, la conservation et la diffusion des connaissances au fil du temps [6]. Les Knowledge Graphs ont été appliqués dans une grande variété de cas d'utilisation, allant des applications commerciales - impliquant la recherche sémantique, les systèmes de recommandations, la publicité ciblée, l'automatisation du transport, etc. Dans ce papier, Khresterion spécialiste de solutions IA, présente une démonstration de son système de raisonnement à base de Knowledge Graphs. Un scénario spécifique pour l'accompagnement du raisonnement expert sur les expressions de garantie assurance santé est présenté.
Dans des travaux futurs, nous envisageons d'investiguer l'identification des incohérences entre différents ensembles de règles (pratiques métier spécifiques, pratiques commerciales et réglementations), et de travailler sur un outil de rédaction de règles qui permettra aux experts métiers de définir eux-même les règles avec un langage naturel.

Abréviations

— ASS : accepté par la sécurité sociale
— KGK : Knowledge Graph Khresterion
— NASS : non accepté par la sécurité sociale
— YCRB : y compris le régime de base
— YCRSS : y compris remboursement sécurité sociale

Références

[1] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, J.E. Labra Gayo, S. Kirrane, S. Neumaier, A. Polleres, R. Navigli, A.C. Ngonga Ngomo, S.M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge Graphs, *arXiv :2003.02320v1*, cs.AI, 2020.

[2] G. Antoniou, C.V. Damásio, B. Grosof, I. Horrocks, M. Kifer, J. Maluszynski, P.F. Patel-Schneider, Combining Rules and Ontologies : A survey, *Technical Report, IST506779/Linköping/I3-D3/D/PU/a1, Linköping University*, Linköping University, 2005.

[3] G. De Giacomo, M. Lenzerini, TBox and ABox Reasoning in Expressive Description Logics, *Proc. of the 5th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'96)*, Morgan Kaufmann, 1996.

[4] H. Prade, L'intelligence artificielle, mais enfin de quoi s'agit il ? *Les Livrets du Service culture UPS* , Université Paul Sabatier, 2001.

[5] N. da Costa, D. Krause, O. Bueno, Paraconsistent Logics and Paraconsistency, *Philosophy of Logic*, Elsevier, 2007.

[6] N. F. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, and J. Taylor. Industry-scale Knowledge Graphs : Lessons and Challenges . *Communications of the ACM, 62(8), 36-43.*,(2019).

[7] N. Guarino, D. Oberle, S. Staab, What is an Ontology ?, *Handbook on Ontologies*, Springer, 1-17, 2009.

[8] T. Eiter, G. Ianni, T. Krennwallner, A. Polleres, Rules and Ontologies for the Semantic Web, *Reasoning Web : 4th International Summer School 2008, Venice, Italy, September 7-11, 2008, Tutorial Lectures*, Springer Berlin Heidelberg, 2008.

Session 3 – L'IA pour l'analyse d'images et de vidéos

SATELLITE IMAGE QUALITY ASSESSMENT USING DEEP LEARNING

Bouchra Harnoufi, Ségolène Bourrienne, Mathias Ortner, Renaud Fraisse

Airbus Defence and Space, Toulouse (France)

ABSTRACT

The Modulation Transfer Function (MTF) is one of the key indicators regarding Image Quality of Earth Observation systems. It characterizes the level of contrast that can be maintained by the optical system and is monitored during the whole life of the satellite. Due to the strong acquisition constraints as well as the gradually complexity of its estimation with future systems, it becomes necessary to increase the reactivity with a method free of acquisition constraints. In this paper, we present a model able to estimate the absolute MTF (or blur) level as well as its prediction uncertainty without any reference image or user parameter using deep learning techniques.

Index Terms— Satellite Imagery, No-Reference Image Quality, Focus, Defocus, Deep Learning

1. INTRODUCTION

The Modulation Transfer Function (or MTF) of an optical system is a measurement of its ability to reproduce various levels of details (spatial frequencies) from the object to the image. More formally, it indicates its capability to transfer contrast at a particular resolution: it shows how well frequency information is transferred from object to image. As a function of spatial frequency, its unit is the ratio of the image contrast over the object contrast (Fig. 1). For high frequencies it is limited by the optical instrument and its analog / digital chain. Hence, it is an important part of the system requirements during the satellite design phase and is the subject of a close monitoring during the Maintenance in Operational Condition (MCO) phase where its level is checked and corrected with refocusing. It is possible to refocus the instrument in orbit by modifying the temperature of the telescope. The range of variation depends on the instrument. At each temperature change, the MTF level is measured: when the highest value of MTF is reached at Nyquist frequency (0.5 when the frequencies are normalized) $MTF_{Nyquist}$, the defocus is corrected (Fig. 2).

Currently, the MTF estimation for a given temperature is notably achieved via dedicated acquisitions on specific areas containing well-known ground patterns (Fig. 3). There are a few patterns around the earth. Viallefont-Robinet et al. [1] compares the MTF measurements using the edge method. This method is widely used. However, for on-orbit MTF assessment, it involves strong constraints.

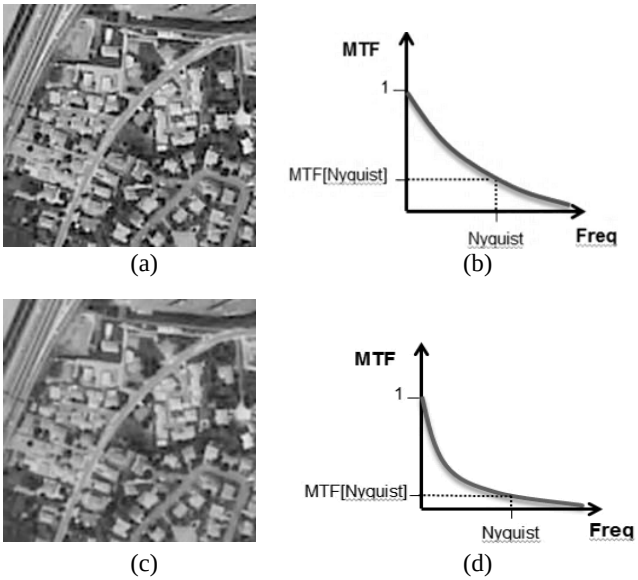


Fig. 1 (a) and (c) represent the same image with the same resolution but with different levels of MTF (or blur). (b) and (d) correspond to the MTF for image (a) and (c) respectively. The lower the MTF is, the less contrast is transferred, and thus the blurrier the image is (in this case (c) is blurrier than (a)).

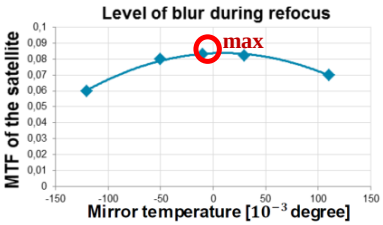


Fig. 2 Evolution of the MTF at Nyquist frequency during thermal refocus. At each temperature variation, the MTF is computed and in particular at Nyquist frequency. The highest MTF value corresponds to the defocus correction.

First of all, a few acquisitions are needed for each focus point and can take a long time depending on the "revisit period" of the satellite. These patterns must be visible which implies cloud-free acquisitions. Consequently, the mission is

Fig. 3 Ground pattern examples (Baoutou). These patterns are currently used for MTF estimation. There are a few around the earth.

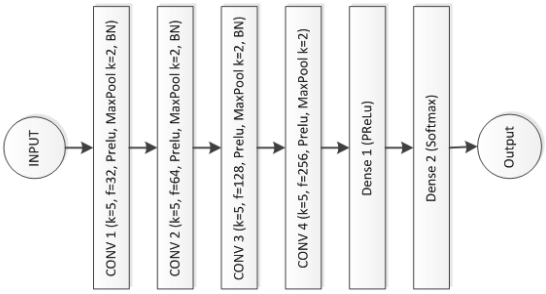


Fig. 4 Neural network model architecture. The convolutional blocks are defined by k: kernel size, f: feature maps, the activation function, the max pooling layer and BN: batch normalization, A probability distribution over the discrete defocus classes is predicted for each input image.

interrupted in order to achieve such specific acquisitions. Besides, with new satellite generations, instruments tend to defocus due to their sensitive materials to heat. Monitoring the MTF is thus challenging while the need is increasing.

The main aims are thus to reduce the mission time spent for refocus operations increasing the reactivity and to avoid mission interruption. Instead of manually estimating the MTF using dedicated acquisitions, a new deep learning neural network model is proposed to estimate the MTF using nominal acquisitions in natural scenes. Recent advances in deep learning approaches are tackling the problem of image quality assessment mainly through comparing images notably leading to a relative measure or using reference images (full references or reduced references). The applications of such techniques still require adjustments to satellite imagery. Instead, we used a classification approach in order to get the level an image belongs to without any reference. The next part details more specifically the methodology followed

2. DEEP LEARNING FOR NO-REFERENCE SATELLITE IMAGE QUALITY ASSESSMENT

Given a set of images each associated with its own defocus MTF level, the objective is to assess the blur level of each image. Comparing two images in order to rank them using notably siamese networks [2] was our first approach. However, it requires a threshold selection which may need user parameters. We chose to use a discrete representation of the defocus MTF levels and to implement a classifier in order to predict the blur level instead of using a regression more suited for continuous variables. Classifiers have demonstrated very powerful results especially in images classification using Convolutional Neural Networks. Our approach is similar to Yang et al. method [3] which assesses microscope image focus quality. However, instead of using a ranked probability score loss function, we used the cross entropy loss function. The model uses this categorical cross entropy to learn to give a high probability to the correct blur level and a low probability to the other levels. We wanted to evaluate if the model is able to learn itself an order without prior knowledge: indeed, the human eye cannot distinguish very close blur levels. Additionally, we used batch normalization regularization technique which significantly improves the results. The network consists of four convolutional blocks followed by two dense blocks. A convolutional block is composed of a convolution layer with a Parametric Rectified Linear Unit (PReLu) activation function, a max-pooling layer and a batch normalization layer except for the last one. The output feature maps are then fed to fully connected layers after having been flattened to finally produce the probability distribution over the classes considered. The softmax activation function ends the neural network since it highlights the target level and normalizes the outputs so that they sum to 1. Consequently, they can be directly treated as probabilities over the output (Fig. 4).

3. EXPERIMENTS

3.1. Data

Data needed are the different MTF levels and the images as described in the previous section. A first simulator produces realistic defocus MTF levels equidistant at Nyquist frequency meaning that from a level to the very next level the amount of “blur” added is constant (from the top higher curve to the lower one). In Fig. 5 each curve represents a MTF level: the lower the point at Nyquist is, the less contrast is transferred and consequently the blurrier is the image. The MTF range considered at Nyquist frequency is [05.25, 12.75]. Two separation powers have been considered: 10% (or 0.5 point MTF) with 16 defocus levels (equidistant at Nyquist frequency) generated and 5% (or 0.25 point MTF) with 31 defocus levels simulated. In both cases, the human eye struggles to distinguish an image with a given defocus MTF level and the same image with the very next defocus MTF level.

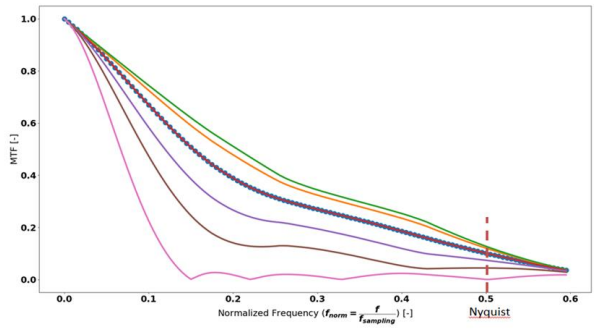


Fig. 5 Different MTF level curves representation generated by the MTF simulator. Each curve represents the level of contrast restored and corresponds to a temperature variation. At Nyquist frequency, the lower the point is, the blurrier the image is (less contrast restored).

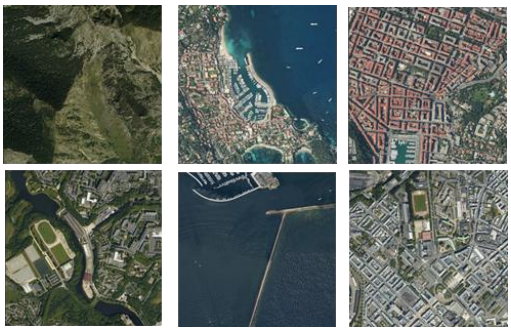


Fig. 6 Examples of images used during the training and the test phase from France. These images contain various frequencies: higher in downtown areas and lower in seas areas for instance.

The available images are aerial images with a resolution of 10 cm. A radiometric simulator has also been developed in order to:

- apply the MTF (blur level) to the images
- resample images to 50 cm: the target resolution
- perform corrections and conversions: remove gamma correction, convert to 12-bit images …

The model has been trained using simulated images from France. These images contain various land cover types such as: some rural regions, homogeneous structures, seas, city centers, buildings (Fig. 6). The diversity of land covers helps evaluating the MTF on a wide range of spatial distributions. To evaluate the performances we used some other simulated images not used in the training data set. Each image is divided into adjacent tiles of size 128 x 128 pixels from which each defocus MTF level is applied to. The dataset is balanced: there are as many examples for each defocus MTF level. Given training examples of 12-bit 128 x 128 pixels input images patches and the corresponding defocus MTF level, the model predicts the

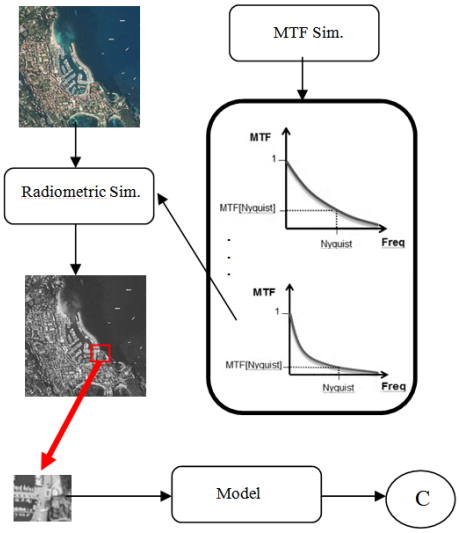


Fig. 7 A MTF simulator generates defocus MTF levels given a MTF range at Nyquist frequency and a separation power. These defocus MTF levels are applied to each image. The images are then divided into tiles to be fed to the neural network. The model output "C" corresponds to the predicted defocus MTF level.

probability distribution over these classes (Fig. 7). The optimizer is Adam to update the network weights with a learning rate value of 5e-6. The model has been trained during 300 epochs and the framework used is PyTorch, in a cloud environment, specifically in a virtual machine containing one GPU NVIDIA Tesla P100.

3.2. Results

The most probable class and the prediction certainty given by the trained model can be visualized for each image patch as a colored border, with a color indicating the predicted class (defocus MTF level) and the transparency denoting the certainty as Yang et al [3] representation (Fig. 8). Tests have been performed using 16 and 31 classes.

Even if the model has been trained not using a specific ordered loss function, it learnt the order (Fig. 9). Indeed, when it fails predicting the right class, it predicts the neighbor class. Moreover, the model is able to produce interesting results even in areas it has not seen before and seems to be able to generalize (no Lyon city image was in the training data set Fig. 8). Another behavior we wanted to check is regarding homogenous areas like seas or rivers where we expect low certainty. Without pre-processing techniques including filtering some areas, the model predicts a defocus MTF level with low probability in areas where frequencies are very low: in Fig. 8 the river area has a very high transparency traducing low certainty. Finally, the separation power obtained using this classifier is 0.4 point

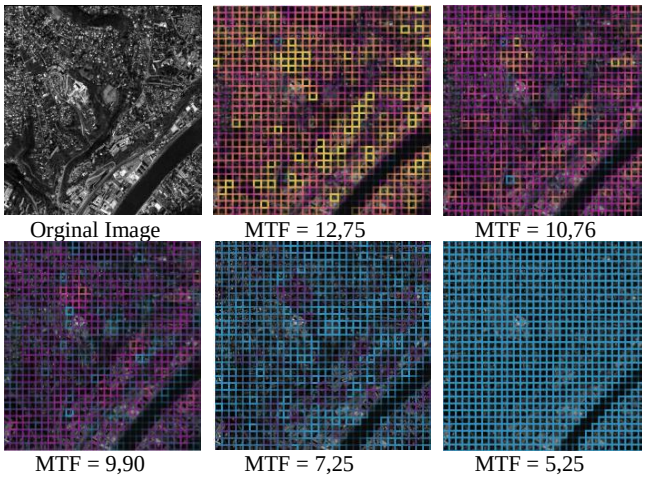


Fig. 8. Examples of result in Lyon city, (a city not contained in the training dataset). Each patch has a color and a transparency representing the MTF level and certainty respectively. The certainty is very low in the river area.

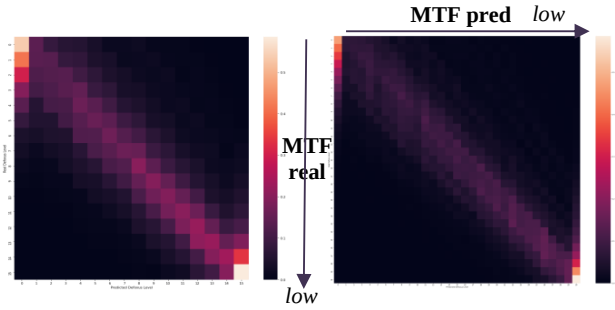


Fig. 9 Covariance matrix: the left one corresponds to 16 ordered blur levels and the right one, to 31. Without using a specific order loss function, the model is able to learn the rank. Indeed, it tends to predict either the true blur level or the previous or next one.

MTF which is equivalent to 8% when $MTF_{Nyquist}$=0.05 and 3% when $MTF_{Nyquist}$=0.12. Given that it is very close and sometimes even better than manual methods, these results are very promising.

4. DISCUSSION

One of the main advantages of this approach is the simulators developed. Indeed, both the defocus MTF level simulator and the radiometric simulator enable us to generate data with their ground truth (i.e. their defocus MTF level) without having to label them manually. It offers also the opportunity to simulate a large number of data which is a requirement when training a deep learning neural network. Moreover, future work includes noise robustness (necessary to real data applications) and can also be simulated. Another improvement is data augmentation. Yang et al. [3] demonstrates the importance in using this technique to train a model. It ensures the diversity of examples learnt by the model. In addition, as our model performs better in low MTF levels, we can also add more examples of images with high defocus MTF levels in the training dataset to help the model distinguish them.

Regarding the neural network model architecture, several optimizations can improve the performances. First of all, we can also use a ranked probability score loss to help the model to focus on learning the MTF level order. Furthermore, as batch normalization has improved our result (without it, the model does not converge), we could use new classification architectures [4] that do not include batch normalizations and have been designed for models sensitive to data statistics. Finally, the requirement of using a defined size patch (here 128 x 128) may be not necessary if the architecture model is fully convolutional.

An extension of this application is to apply such model while performing other image enhancements so as to know how modifying another image parameter affects the blur level of the image. Some algorithms should not alter the blur level such as noise reduction. A measure using our method can confirm if it respects this property

5. CONCLUSION

We have proposed a deep learning model able to assess the MTF or blur level of any satellite image without needing dedicated acquisitions. To train and validate such method, we have also developed a simulator capable of simulating images at different defocus MTF levels. Experiments conducted in this frame showed promising results in terms of accuracy of MTF levels as well as generalization to unseen images. It opens up a wide range of new applications needing a MTF measure and several improvements have been identified to improve performances before being challenged to real data.

6. REFERENCES

[1] Viallefont-Robinet, Françoise, et al. "Comparison of MTF measurements using edge method: towards reference data set." *Optics express* 26.26 (2018): 33625-33648.

[2] Niu, Yuzhen, et al. "Siamese-network-based learning to rank for no-reference 2D and 3D image quality assessment." *IEEE Access* 7 (2019): 101583-101595.

[3] Yang, Samuel J., et al. "Assessing microscope image focus quality with deep learning." *BMC bioinformatics* 19.1 (2018): 1-9.

[4] Brock, Andrew, et al. "High-Performance Large-Scale Image Recognition Without Normalization." *arXiv preprint arXiv:2102.06171* (2021).

Utilisation de techniques d'intelligence artificielle pour la détection d'animaux en milieu forestier

B. Benet[1], A. Marell[2], Y. Boscardin[2]

[1]Université Clermont-Auvergne, INRAE, UR TSCF, 63178 Aubière, France
[2] INRAE, UR EFNO, 45290, Nogent-sur-Vernisson, France

bernard.benet@inrae.fr

Résumé

Un logiciel de vision artificielle qui comprend des opérations de traitement d'images et des méthodes d'intelligence artificielle (deux réseaux de neurones intégrés développés avec les outils Tensorflow et Caffe), a été développé pour détecter et identifier des espèces animales en milieu forestier sur des vidéos enregistrées, en prenant en compte des fortes contraintes au niveau de l'environnement, telles que les conditions d'éclairage très variables dans le sous-bois, les variations de colorimétrie et de morphologie des animaux, qui peuvent être partiellement cachés par des objets naturels (arbres, herbes) et des conditions météorologiques variables (pluie, neige, vent).

Mots-clés

Traitement d'image, intelligence artificielle, forêt, biodiversité, animaux

Abstract

An artificial vision software which includes image processing operations and deep learning methods (two integrated neural networks developed with Tensorflow and Caffe tools), has been developed to detect and identify animal species in forest environments on recorded videos, taking into account strong environmental constraints, such as highly variable lighting conditions in the undergrowth, variations in colorimetry and animal morphology, which may be partially hidden by natural objects (trees, grasses) and variable weather conditions (rain, snow, wind).

Keywords

Image processing, deep learning, forest, biodiversity, animals

1 Introduction

Les scientifiques et les naturalistes sont nombreux à utiliser des dispositifs de capture d'images, appelés aussi « pièges photographiques » pour inventorier et suivre les populations animales et la biodiversité de la faune sauvage [1][2][3][4]. Ces systèmes sont en particulier utilisés dans les régions éloignées et dans les milieux où il est difficile d'observer les animaux comme par exemple dans les vastes étendues de la forêt tropicale. Ces pièges permettent également d'étudier des animaux rares et cryptiques tels que les grands prédateurs (tigre, panthère). En France, le suivi par les pièges photographiques est employé par les agents de l'OFB (Office Français de la Biodiversité) pour étudier et suivre les populations de loup, de lynx et d'ours brun. Ils sont également utilisés par les chasseurs pour suivre les populations de grand gibier (chevreuil, cerf élaphe, sanglier), en milieu forestier. Dans le cadre du projet de recherche OPTMix, mené par l'INRAE sur le site de Nogent-sur-Vernisson, des systèmes de capture d'image ont été disposés dans les parcelles de la forêt domaniale d'Orléans, pour enregistrer des vidéos ou des photos, tout au long de l'année, pour détecter la présence d'animaux et étudier leurs impacts sur la forêt.

Ces dispositifs de capture d'images produisent de grande quantité d'images. Elles ont été compilées dans des vidéos dont l'exploitation était réalisée, il y a quelque temps, manuellement, par des agents de l'INRAE qui étaient amenés à parcourir visuellement ces vidéos, dans leur intégralité, en analysant les différentes images, et à enregistrer manuellement des informations telles que les numéros des parcelles forestières sur lesquelles sont effectuée la détection d'animaux, la date de chaque vidéo, les plages temporelles de présence des animaux, l'espèce et la catégorie d'animal, leurs comportements dans une parcelle donnée (position fixe ou mobile). Il s'agissait d'un travail fastidieux et contraignant à réaliser, pouvant conduire à des erreurs dans la détection des animaux. Ces contraintes de traitement manuel limitaient les possibilités de suivi à des fins scientifiques mais aussi son application à grande échelle par les gestionnaires.

L'objectif était donc de développer un système de traitement automatique des vidéos afin de compter les animaux et identifier les espèces dans des programmes de suivi de la biodiversité pour faciliter et améliorer l'exploitation de ces données et rendre la méthodologie accessible à un plus grand nombre d'utilisateurs. Un logiciel de vision artificielle, qui comprend des opérations de traitement d'images (fonctions de la librairie OpenCV et fonctions personnalisées) qui ont été complétées par des méthodes d'intelligence artificielle [5] en utilisant des outils tels que TensorFlow et Caffe, a été développé pour détecter et identifier automatiquement des

animaux sur les vidéos enregistrées. L'innovation technologique réside dans la combinaison de ces deux techniques pour pouvoir développer un système de détection des animaux, en temps réel, en prenant en compte des contraintes qui ont une influence importante, aussi bien pour la partie traitement d'image que pour les opérations d'apprentissage et de prédiction de la partie intelligence artificielle du logiciel : les conditions de luminosité sont très variables dans le sous-bois selon les lieux au cours de la journée, avec des problèmes d'ombres relativement important, les conditions météorologiques sont variables (pluie, neige, vent), les variations de colorimétrie et de morphologie des objets naturels (arbres, herbes, animaux) sont importantes, les animaux recherchés peuvent apparaître dans les différents plans de l'image, et peuvent être cachés partiellement par des objets naturels de type arbres, végétation, dans l'environnement forestier. Les intérêts de ce logiciel sont multiples et évitent des opérations manuelles et visuelles fastidieuses : la détection des animaux est réalisée de façon automatique, le traitement des vidéos permet de conserver en mémoire uniquement des images contenant des animaux, au lieu de garder beaucoup de fichiers volumineux en mémoire, et l'enregistrement des résultats de détection d'animaux est effectué automatiquement dans un fichier pour un ensemble de vidéos traitées.

2 Matériel et méthode

2.1 Le dispositif d'acquisition vidéo

Des dispositifs d'acquisition d'image ont été positionnés sur des arbres dans différentes parcelles forestières (Figures 1 et 2), dans la forêt domaniale d'Orléans. Ces systèmes sont des appareils photo numérique nommés « GameSpy modèle M-80XT » de la marque Moultrie.

Figure 1: Le dispositif d'acquisition d'image

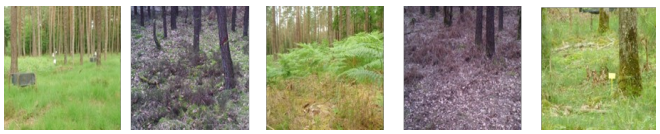


Figure 2: Les parcelles forestières

Dans les travaux menés, le suivi est réalisé sur 12 placettes de forme rectangulaire d'une superficie de 0.5 ha au cours de trois périodes de l'année : février-mars, mai-juin et novembre-décembre. Pendant ces périodes, les données enregistrées par les caméras sur une carte mémoire sont relevées par un opérateur une fois par semaine. L'acquisition automatisée des données a lieu pendant six jours sur les sept jours de la semaine, le 1er jour étant nécessaire à l'appareil pour s'étalonner au lever et coucher du soleil. Pour chaque placette, le dispositif de mesure consiste à utiliser quatre caméras vidéo positionnées de façon à ce que la prise de vue soit réalisée depuis ses quatre coins en direction de son centre. Pour cela, chacune des caméras est fixée sur l'arbre le plus proche de chacun des coins, et orientée après avoir matérialisé les diagonales de la parcelle. Elles sont placées à 2 mètres de hauteur, avec une orientation vers le bas de quelques degrés, qui permet d'obtenir une bonne visualisation des parcelles pour la détection des animaux. Cette hauteur "inaccessible" par les animaux, permet d'éviter que ces derniers détériorent les caméras. La caméra acquiert des images toutes les minutes pendant 4 heures à partir du lever et 4 heures jusqu'au coucher du soleil, soit 240 photographies par vidéo. Ces photos sont compilées bout à bout dans une séquence vidéo.

Les images traitées par le logiciel développé extraites des vidéos, sont dans l'espace couleur visible classique (Rouge, Vert, Bleu) et ont une résolution de 1280 x 720 pixels.

2.2 Le logiciel de vision et d'intelligence artificielle

Deux étapes sont utilisées dans le logiciel développé en langage Python. Des fonctions de traitement d'images (fonctions de la librairie OpenCV et fonctions personnalisées) et l'utilisation de techniques d'intelligence artificielle, avec le développement de réseaux de neurones avec des outils tels que TensorFlow et Caffe, composent ce logiciel. Le premier objectif du travail réalisé, était de détecter et d'identifier les animaux forestiers, sur les images extraites des vidéos, en travaillant, avec une fréquence de 1Hz. A terme, l'objectif final sera d'implémenter le logiciel développé sur des systèmes embarqués temps réel, en milieu forestier, disposés sur différents arbres, qui pourront être composés d'ordinateur monocarte de type Raspberry pi ou Jetson Nano, et de caméras de type webcam.

La partie traitement d'image qui comprend diverses fonctions de filtrage, de seuillage, de morphologie mathématique, dans l'espace couleur RGB, permet de comparer sur le plan colorimétrique, des images successives acquises aux instants t et t+1 sur les vidéos et ainsi de détecter des variations de couleurs sur tous les pixels des images. Ces différences de couleurs peuvent être soit des éléments qui apparaissent dans la scène (détection d'un animal qui apparaît ou qui est en mouvement), soit des mouvements d'herbes, soit des changements de couleurs des objets dues aux variations de luminosité. Le logiciel cherchant à détecter des animaux sur l'image acquise à l'instant t+1, on va appliquer, en complément de cette phase de différenciation d'images, un seuillage colorimétrique, dans l'espace couleur HSV (teinte, saturation, intensité lumineuse) sur l'image à l'instant t+1, pour éliminer les pixels qui ne vérifient pas des conditions colorimétriques désirées pour les animaux à détecter. Par exemple, les pixels de teinte verte, jaune, rouge sont éliminés. On obtient alors, à l'issue de ces opérations de traitements d'image (différenciation entre deux images successives (t et t+1) et élimination des pixels ne vérifiant pas la couleur recherchée des animaux sur l'image à l'instant t+1), des groupements de points séparés (labélisés) sur les images.

Pour chacun d'entre eux, une zone rectangulaire va être définie. On obtient alors des images de tailles réduites, de différentes dimensions appelées "imagettes", sur lesquelles on va travailler dans une phase d'intelligence artificielle pour la détection des animaux. A l'intérieur de celles-ci, on peut trouver soit un animal complet (ou une partie physique d'un animal (la tête, le corps, les pattes,...)), soit un élément de l'environnement forestier (arbre, branche, herbe,...) qui correspond à du bruit, que l'on appelle "faux positif". La quantité d'imagettes obtenues sur une image donnée, qui dépend de l'environnement (quantité d'herbe) et des conditions climatiques (vent, variation de luminosité, ombres portés), varie approximativement entre 5 et 80. La partie intelligence artificielle va permettre d'identifier chacune des imagettes, en indiquant si un animal y est présent ou non, pour détecter les animaux et d'éliminer les "faux positifs". On obtient, pour chaque imagette, un résultat de type: objet détecté (animal ou autre type) avec un score (probabilité d'appartenance). La figure 3 ci-dessous présente un exemple avec l'obtention d'imagettes, de différentes tailles, obtenues avec les opérations de traitement d'image, et la figure 4 montre le résultat de détection d'un animal obtenu, avec les opérations d'intelligence artificielle.

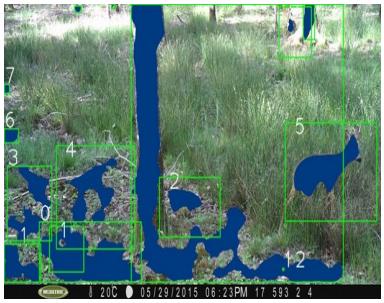


Figure 3: Détection d'imagettes par traitement d'image

Figure 4: Détection d'un animal

2.3 Les opérations d'intelligence artificielle

La figure 5 ci-dessous présente un exemple de type de réseau de neurones utilisé, qui est constitué d'une couche d'entrée, dans laquelle on va disposer nos données (imagettes couleurs RGB de différentes dimensions, pouvant contenir un animal (ou partie animalière) ou du bruit), d'un ensemble de couches cachées et d'une couche de sortie, qui présente le résultat de prédiction obtenu : type d'animal détecté (ou bruit), avec un score d'appartenance entre 0 et 1.

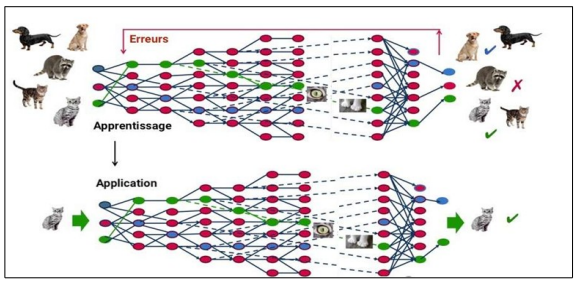


Figure 5 : Type de réseau de neurones utilisé

Une méthode de rétro-propagation de gradient est effectuée pour créer le réseau, dans la phase d'apprentissage, c'est-à-dire pour déterminer les valeurs des poids des liens entre les différents neurones du réseau qui communiquent entre eux. A partir du réseau créé, l'opération de prédiction va consister à envoyer en entrée du réseau une « imagette » et à obtenir à la sortie des types d'objets détectés, avec des scores d'appartenance. Par exemple dans notre application, le réseau trouvera pour une image d'entrée, différents objets résultats (en fonction des classes d'apprentissage) : Chevreuil avec un score de 0.51, Arbre avec un score de 0,2, Sanglier avec un score de 0.09,...

Deux opérations d'intelligence artificielle, avec deux réseaux de neurones utilisés, sont appliquées pour obtenir les résultats de détection/identification d'animaux, avec l'objectif de prendre en compte la qualité de détection mais aussi l'aspect temporel, pour pouvoir à terme, développer un système de détection en temps réel et transmettre les informations de détection en « live ».

Les imagettes obtenues par traitement d'image, dans la première étape du logiciel, sont envoyées dans un premier temps, sur un réseau de neurone de type GoogleNet (réseau de neurones à convolution avec 22 couches qui est une variante du réseau Inception), entrainé sur une base d'image d'apprentissage contenant une importante base d'objets de différentes natures dont 500 classes d'animaux), utilisé avec l'outil Caffe, pour effectuer un filtrage rapide des imagettes. Cette opération permet d'éliminer une grande partie des imagettes (environ 60%), qui contiennent du bruit (arbres, végétation,…) et de garder en mémoire des imagettes pouvant contenir un animal. On ne recherche pas, à ce stade à identifier les animaux trouvés. Ces imagettes restantes sont alors envoyées sur un deuxième réseau de neurone de type Inception V3, développé et utilisé avec l'outil tensorflow, qui a été réentrainé sur la dernière couche, avec une base d'images d'apprentissage personnalisée contenant des animaux forestiers (chevreuil, biche, pigeon, cerf elaphe, sanglier,...), et des éléments de l'environnement forestier (arbres, feuilles, branches, herbes,…). Les figures 6 et 7 présentent des exemples d'images d'apprentissage. 3000 images d'animaux forestiers et de bruit, ont été introduites dans la base d'apprentissage pour obtenir ce deuxième réseau de neurones.

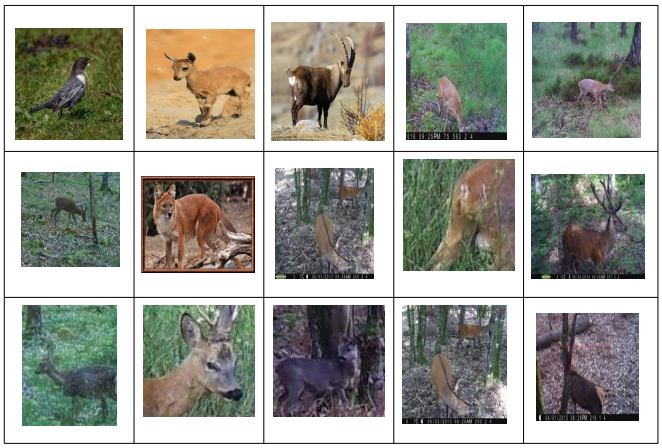


Figure 6: Exemples d'images d'animaux

Figure 7: Exemples d'images de l'environnement forestier

Etant donné que des animaux peuvent être visibles soit en totalité soit partiellement sur les images, car il peuvent être masqués par des arbres, par des branches, ou par de la végétation, il était nécessaire de considérer pour la base d'apprentissage animalière, des animaux complets, en considérant différentes faces, orientations, postures et dimensions car les animaux peuvent être présents dans les différents plans de l'image, mais aussi des parties d'animaux (tête, pieds, corps,…). Différents types de réseaux de neurones ont été testés tels que Inception V1, V2 et V3, et des réseaux Mobilenet, pour obtenir un réseau « fiable », au niveau de la qualité des résultats obtenus, en considérant différents jeux de données pour les phases de validation et de test des réseaux, et en faisant varier le nombre d'étapes d'apprentissage.

3 Résultats

Le logiciel développé a été appliqué sur 5000 vidéos acquises entre les années 2013 et 2018, dans des parcelles de la forêt domaniale d'Orléans, acquises à différents moments de la journée, dans différentes périodes de l'année, pour prendre en compte le coté saisonnier dans le suivi de la grande faune. 240 images ont été extraites de chaque vidéo. La figure 8 présente un exemple de résultats des opérations de traitement d'image du logiciel : dans un premier temps, comparaison d'images successives dans l'espace couleur RGB, acquises aux instant t et t+1, pour détecter des différences au niveau de tous les pixels des deux images, prise en compte des couleurs des pixels dans l'image à l'instant t+1, opérations de filtrage et de morphologie pour éliminer du bruit, labellisation pour regrouper des pixels sur l'image et obtention des imagettes rectangulaires qui encadrent chaque groupe de points.

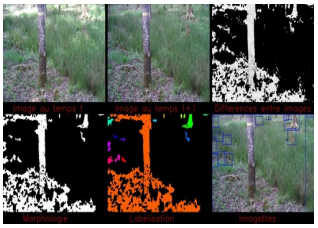


Figure 8: Opérations de traitement d'image

La figure 9 présente le résultat de détection d'un animal (gazelle), obtenue avec les deux réseaux de neurones, dans la phase d'intelligence artificielle.

Figure 9: Détection d'un animal

La figure 10 montre des exemples de résultats de détection obtenus.

Figure 10: Résultats de détection

Concernant la première opération d'intelligence artificielle, menée avec la base d'images GoogleNet, on notera que les animaux que nous rencontrons dans la forêt de Nogent et qui ont été détectés par le logiciel, ont été classés, dans la phase de prédiction, dans des animaux de type sanglier et renard, qui sont présents sur le site, mais aussi de type bouquetin, poule d'eau, dhole qui ne sont pas présents sur le site. Le deuxième réseau de neurones utilisé qui est le réseau Inception V3 qui a été pré-entraîné sur la dernière couche, à partir d'une base d'images d'animaux et de bruits dans l'environnement forestier, a permis de répondre à notre besoin, en matière de qualité de détection et d'identification des animaux forestiers.

Un travail portant sur 1140 vidéos (Nombre d'images traitées = 273600) a été réalisé, pour comparer les résultats obtenus par une analyse visuelle (expertise) avec ceux obtenus avec le logiciel développé et ainsi évaluer la qualité de détection des animaux. Sur le plan général, le logiciel détecte correctement les animaux, les différents bruits dans l'environnement forestier (mouvements d'herbes, variations de luminosité,...) sont bien éliminés grâce aux deux opérations d'intelligence artificielle.

Une matrice de confusion a été établie avec le comptage des quatre éléments (Vrai Positif, Faux Positif, Vrai Négatif et Faux Négatif) sur l'ensemble des vidéos traitées, et a permis de déterminer les valeurs de sensibilité, de spécificité et de précision du logiciel (traitement d'image et intelligence artificielle). La définition de ces différents éléments est présentée ci-dessous :

- TP ('True positives' - Vrais positifs) : le nombre total d'images avec animaux (ongulés sauvages, cerf élaphe, chevreuil, sanglier) détectés par l'expertise et par le logiciel

- FP ('False positives' - Faux positifs) : le nombre total d'images, sur lesquelles le logiciel a détecté un animal mais qui ne contiennent aucun animal

- FN ('False negatives' - Faux négatifs) : le nombre total d'images contenant des animaux mais qui n'ont pas été détectés par le logiciel

- TN ('True negatives' - Vrais négatifs) : le nombre total d'images sans animaux (images identifiées sans animaux à la fois par l'expertise et par le logiciel).

Les paramètres de sensibilité, de spécificité et de précision sont définis par les formules suivantes :

-Sensibilité = TP / (TP + FN)

-Spécificité = TN / (TN + FP)

-Précision = (TP + TN) / N (N est le nombre d'image traitées)

Le tableau 1 ci-dessous présente les données obtenus de la matrice de confusion et les valeurs caractérisant la qualité du logiciel.

		Détection manuelle	
		Positif	Négatif
Détection avec le logiciel	Positif	452	4367
	Négatif	216	272615
Sensibilité	67 %		
Spécificité	98 %		
Précision	99 %		

Tableau 1. Matrice de confusion et qualité du logiciel

La spécificité du logiciel, qui mesure la capacité de l'analyse à donner un résultat négatif lorsque l'image est vide, s'avère élevée avec une valeur de 98 %. Ce qui veut dire que l'analyse permet d'éliminer les images sans animaux. La précision est aussi élevée 0.99. C'est à dire que l'analyse arrive à identifier les images avec des animaux et exclure les images sans animaux.

En revanche, la sensibilité, qui mesure la capacité de l'analyse à donner un résultat positif lorsque l'image contient des animaux, est de 67 %. Ce résultat de non détection de certains animaux sur des images peut s'expliquer par les points suivants :

- des animaux de petites tailles peuvent apparaître dans la scène, avec des couleurs sombres devant des objets naturels de couleurs sombres, par exemple des arbres, ce qui ne permet pas de les distinguer facilement dans l'environnement

- des animaux peuvent être partiellement cachés dans l'environnement forestier, par exemple derrière des arbres, des branches ou de la végétation, et leur visibilité est ainsi limitée sur les images

- des animaux peuvent présenter un profil/une face semblable à un élément de type arbre ou branche, par exemple. Dans ce cas, le logiciel, dans la partie intelligence artificielle, donne des résultats de prédiction de mauvaise qualité : détection d'un objet de type animal avec une probabilité inférieure à 5 %.

Concernant les faux positifs détectés, ils correspondent en général à des arbres ou à des branches, qui ont des couleurs (marron/noir) comme celle de la majorité des animaux forestiers, qui peuvent apparaître dans les imagettes obtenues dans la partie traitement d'image du logiciel, à cause des variations de luminosité entre des images successives.

Sur le plan temporel, la durée moyenne de traitement des 240 images pour une vidéo donnée est de 4mn, sur un PC portable avec 4 cœurs, avec un processeur intel core i7 de 3,8 Ghz, et une carte graphique classique (non CUDA). Ce logiciel enregistre dans un dossier toutes les images détectées, sur un ensemble de vidéos, ce qui permet alors de réaliser des comparaisons entre les parcelles, entre les périodes de l'année, pour étudier notamment, de façon approfondie, l'influence des animaux sur la dégradation des jeunes pousses en milieu forestier, et d'étudier la biodiversité concernant la grande faune forestière.

4 Conclusion et Perspectives

Une méthode de vision artificielle a été développée pour détecter des animaux qui apparaissent sur des vidéos. Compte tenu du bruit plus ou moins important sur les images (variation de luminosité, mouvements d'herbes, environnement complexe), l'algorithme de traitement d'image développé, basé sur la différenciation d'images successives, n'était pas suffisant pour détecter les animaux (beaucoup de faux positifs étaient détectés). Une méthode d'intelligence artificielle a donc dû être ajoutée, en complément de la méthode de traitement d'image, pour éliminer une quantité importante de faux positifs et détecter les animaux sur les images extraites des vidéos. Les résultats de détection d'animaux obtenus avec ce logiciel de vision vont permettre aux agents de l'INRAE de Nogent de l'utiliser de façon permanente, pour les animaux forestiers et d'approfondir aussi le coté identification, pour des

animaux détectés qui apparaissent sur les images de façon complète (une grande partie de leurs corps est visible).

Des modifications devront être apportés dans le logiciel développé, pour améliorer la sensibilité de détection, pour pouvoir détecter, le mieux possible, des animaux non détecté, actuellement, en jouant sur des paramètres et des seuils de détection au niveau des deux réseaux de neurones utilisés, tout en limitant la détection de faux positifs.

Le principe général du logiciel développé qui combine la vision artificielle et l'intelligence artificielle, qui peut fonctionner soit en utilisant des réseaux de neurones préentrainés sur des bases de type ImageNet ou GoogleNet, soit en créant des réseaux de neurones personnalisés, entraînés sur des bases d'apprentissage personnalisées contenant des images d'objets que l'on cherche à détecter et identifier, soit en combinant ces deux méthodes, pourrait être appliqué pour détecter d'autres types d'animaux (oiseaux, poissons, insectes,...) pour différentes applications de détection ou de suivi d'animaux, dans des milieux forestiers, agricoles, aquatiques.

Dans l'application présentée dans cet article, le logiciel travaille en mode différé, en traitant des vidéos enregistrées, en milieu forestier, mais l'objectif est de développer un système de traitement en temps réel, en disposant des caméras et des ordinateurs de type monocartes, sur des arbres, dans différentes parcelles forestières et de communiquer les résultats obtenus, sur un ordinateur central, pour enregistrer les détections des animaux et prendre éventuellement des décisions en temps réel, de type alerte.

5 Biblio

[1] Norouzzadeh, M.S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M.S., Packer, C., Clune, J. 2018. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. Proceedings of the National Academy of Sciences 115, E5716-E5725. https://doi.org/10.1073/pnas.1719367115

[2] Tabak, M.A., Norouzzadeh, M.S., Wolfson, D.W., Sweeney, S.J., Vercauteren, K.C., Snow, N.P., Halseth, J.M., Di Salvo, P.A., Lewis, J.S., White, M.D., Teton, B., Beasley, J.C., Schlichting, P.E., Boughton, R.K., Wight, B., Newkirk, E.S.,Ivan, J.S., Odell, E.A., Brook, R.K., Lukacs, P.M., Moeller, A.K., Mandeville, E.G., Clune, J., Miller, R.S., 2019. Machine learning to classify animal species in camera trap images: Applications in ecology. Methods Ecol. Evol.10, 585-590. https://doi.org/10.1111/2041-210x.13120

[3] Yousif, H., Yuan, J., Kays, R., He, Z., 2019. Animal Scanner: Software for classifying humans, animals, and empty frames in camera trap images. Ecology and Evolution 9, 1578-1589. https://doi.org/10.1002/ece3.4747

[4] Yu, X., Wang, J., Kays, R., Jansen, P.A., Wang, T., Huang, T., 2013. Automated identification of animal species in camera trap images. EURASIP Journal on Image and Video Processing 2013, 52. https://doi.org/10.1186/1687-5281-2013-52

[5] Wäldchen, J., Mäder, P., 2018. Machine learning for image based species identification. Methods Ecol. Evol. 9, 2216-2225. https://doi.org/10.1111/2041-210x.13075

ZGP : une alternative aux réseaux de neurones pour la segmentation sémantique de nuages dans les images satellites multi-spectrales

I. Grenet[1,2], Y. Bobichon[1], A. Girard[1], F. Férésin[1],

[1] Institut de Recherche Technologique Saint Exupéry - Sophia Antipolis, France
[2] MyDataModels - Sophia Antipolis, France

ig@mydatamodels.com
{yves.bobichon, adrien.girard, frederic.feresin}@irt-saintexupery.com

Résumé

Cet article présente les résultats de l'algorithme évolutionnaire ZGP appliqué à la segmentation de nuages dans les images satellites multi-spectrales. De nombreuses méthodes ont été développées pour automatiser la détection de nuages mais peuvent parfois s'avérer difficiles à implémenter à bord des satellites. Ici nous comparons les performances de ZGP à celles d'un réseau de neurones (UNET). Les résultats obtenus sur des images du satellite Landsat-8 et les avantages de ZGP en font une bonne alternative aux réseaux de neurones pour le deploiement d'applications de segmentation d'images sur des systèmes embarqués.

Mots-clés

Segmentation de nuages, images satellites multi-spectrales, algorithmes évolutionnaires, ZGP, UNET

Abstract

This paper presents the results of the evolutionary algorithm ZGP applied to cloud segmentation in remote sensing multispectral images. Many methods have been developed in order to automate cloud detection but can sometimes be difficult to implement on-board satellites. Here we compare ZGP to a deep learning algorithm (UNET). The results obtained on the Landsat-8 satellite images and the advantages of ZGP make it a good alternative to neural networks for the deployment of image segmentation applications on embedded systems.

Keywords

Cloud segmentation, multispectral remote sensing images, evolutionary algorithms, ZGP, UNET

1 Introduction

La segmentation des nuages dans les images satellites est un cas d'utilisation bien connu de l'imagerie spatiale. On estime en effet que les nuages recouvrent plus de 54% des terres émergées et 68% de la surface des océans [21] réduisant d'autant l'utilité des images acquises par les satellites d'observation de la Terre pour des utilisateurs généralement intéressés par des phénomènes se produisant à la surface du globe. Pour faire face à la quantité toujours croissante d'images fournies quotidiennement par les systèmes d'observation de la Terre, de nombreuses méthodes ont été développées ces dernières années pour automatiser la tâche de détection des nuages dans les segments sol des satellites [7, 10, 19, 35]. Ces méthodes ont été testées et validées sur de nombreuses bases de données labellisées dont certaines, comme SPARCS [25] et BIOME [15] développées pour le segment sol du satellite Landsat-8, tendent à s'imposer comme des références sur lesquelles de nombreux algorithmes de détection de nuages ont été entrainés, validés et comparés.

Dans une image satellite les nuages se présentent sous la forme de régions à forte radiométrie parfois délimitées par des contours diffus. De tailles très variables et sans forme géométrique caractéristique, leurs positions changent au cours du temps. Ainsi les méthodes de détection de changements ont été les premières à être utilisées pour détecter efficacement des nuages [17, 36]. Cependant ces approches temporelles sont incompatibles avec la détection en temps réel des nuages sur des acquisitions individuelles pour lesquelles d'autres approches (SVM, arbres de décision, forêts aléatoires) ont été proposées [34]. L'objectif n'est pas ici de fournir un état de l'art exhaustif des méthodes de détection de nuages mais de se concentrer sur quelques-unes reconnues comme étant les plus performantes dans le but de définir une référence de performances pour notre algorithme génétique.

Récemment les réseaux de neurones convolutifs tels que UNET [30], FCN[25], SegNet [2] ou DeepLabV3 [6] ont fait progresser notablement les performances des méthodes classiques de segmentation sémantique d'images dans de nombreux domaines d'applications. Pour la détection de nuages dans les images satellites, les architectures de type UNET comme Rs-Net [20], CS-CNN [9] ou Cloud-Net [26] se sont montrées particulièrement performantes.

Les réseaux de neurones sont connus pour être généralement complexes à mettre en œuvre dans des systèmes embarqués tels que des satellites, dont les capacités de calculs

et de stockage mémoire sont limitées. Dans cette optique, plusieurs approches ont été proposées afin de simplifier les architectures de réseaux de neurones en vue de leur implémentation à bord de satellites pour la détection de nuages en temps réel ou la compression d'image sélective. MobU-Net [33] a été développé pour détecter les nuages dans des images à basse résolution compressées en Jpeg2000-à bord d'un CubeSat ARTSU. Bahl *et al*. [3] proposent une version compacte C-UNET pour la détection de nuages. Feresin *et al*. [14] ont démontré la faisabilité d'une implémentation sur FPGA d'un réseau de neurones compact pour la détection de nuages dans les images du satellite OPS-SAT. Plus récemment, un réseau de neurones pour la détection de nuages dans des images hyperspectrales générées par la camera HySCOUT-2 du nano-sat Phi-Sat a été implémenté dans un processeur ARM.

Toutes ces récentes applications montrent l'intérêt de développer des approches de segmentation d'image adaptées aux contraintes des systèmes embarqués, notamment des satellites, tout en conservant un niveau élevé de performance compatible avec les exigences de fiabilité, de robustesse et de faible complexité algorithmique requises par ce type de systèmes.

Dans cet article nous présentons une application d'un algorithme évolutionnaire propriétaire, le Zoetrope Genetic Programming (ZGP), à la segmentation sémantique. Nous montrons la performance atteinte par cet algorithme dans le cas de la détection de nuages dans les images satellite multi-spectrales. Les résultats sont comparés à ceux obtenus avec les méthodes de deep learning qui s'avèrent être les plus performantes actuellement, basé sur l'état de l'art des réseaux de neurones, le UNET. La section 2 introduit des généralités sur les algorithmes évolutionnaires et présente en particulier l'algorithme ZGP. La section 3 décrit les données et méthodes utilisées dans ces travaux. La section 4 détaille les résultats d'inférence des modèles sur les données SPARCS et BIOME. La section 5 présente les conclusions et les perspectives pour de futurs travaux.

2 Les Algorithmes Evolutionnaires

2.1 Principe

Les algorithmes évolutionnaires (AE) sont un type d'algorithmes utilisés en intelligence artificielle inspirés des mécanismes biologiques d'évolution tels que la reproduction, la mutation, la recombinaison et la sélection naturelle [11]. Ils permettent de trouver des solutions à des problèmes particulièrement difficiles à optimiser via la génération de nombreuses solutions candidates (les « individus »), au sein d'une population. Leur qualité est évaluée grâce à une fonction d'adaptation aux données (*fitness function*). La population évolue ensuite au cours des générations grâce à l'application d'opérateurs génétiques (mutation, recombinaison) aux individus préalablement sélectionnés, appelés parents, ce qui permet la création de nouveaux individus, les descendants. Les AE sont des algorithmes stochastiques puisque leur apprentissage implique de nombreux processus aléatoires. Le processus général d'un AE, illustré en Figure 1, est le suivant [8] :

- Initialisation d'une population : création d'un ensemble aléatoire d'individus correspondant aux solutions candidates. Ces individus peuvent être représentés de différentes façons (alphabet fini, valeurs binaires, vecteurs de valeurs réelles, arbres, programmes, etc). La population possède un nombre fini d'individus qui est le plus souvent constant.
- Evaluation : mesure de la qualité de chaque individu par rapport à la meilleure solution. Cette mesure est calculée grâce à une fonction mathématique appelée fonction d'évaluation (ou *fitness function*) et reflète à quel point l'individu correspond à la solution.
- Opérations génétiques : génération de nouveaux individus grâce à deux types d'opérateurs :
  - la mutation : elle applique une transformation aléatoire de certaines caractéristiques d'un individu, indépendamment des autres. La mutation crée de la diversité et permet ainsi l'exploration de l'espace des solutions.
  - la recombinaison (*cross-over*) : mime la reproduction biologique en échangeant de l'information entre deux parents de façon aléatoire, générant ainsi un nouvel individu. La recombinaison a pour but de créer des descendants qui portent les « bonnes » caractéristiques des parents et permet ainsi l'exploitation.
- Sélection : sélection d'individus à partir de l'ensemble des parents et descendants en fonction de leur qualité (*fitness*). Le nouvel ensemble constitue la nouvelle génération.

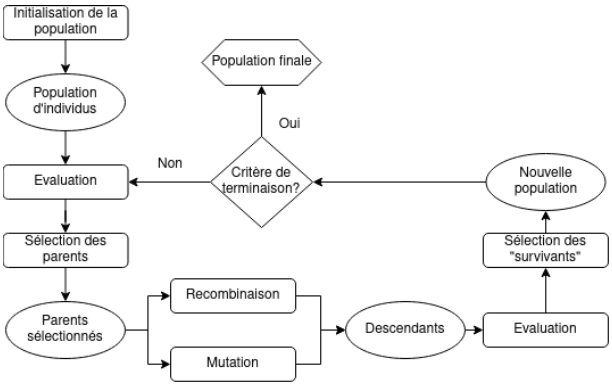


FIGURE 1 – Principe général d'un algorithme évolutionnaire

L'ensemble du processus, depuis la génération d'individus jusqu'à la sélection, est répété un certain nombre d'itérations jusqu'à atteindre une condition de terminaison (*e.g.* : la *fitness* ne s'améliore plus depuis un certain nombre de générations, une limite de temps de calcul est atteinte, etc). Les AE ont plusieurs avantages parmi lesquels leur bonne adaptabilité à une large variété de problèmes. En effet, puisque aucune hypothèse n'est faite sur la distribution des données et la fonction sous-jacente liée à celles-ci,

les AE sont libres d'apprendre n'importe quelle forme de fonction et peuvent adopter différentes stratégies pour trouver de meilleures solutions [16]. Ceci n'est pas le cas d'autres techniques d'optimisation. De plus, ils permettent de trouver des solutions à des problèmes qui ne peuvent être résolus par d'autres types d'algorithmes, en terme de difficulté de calcul, indépendamment de la taille des données. Aussi, contrairement aux réseaux de neurones, les AE sont capables de gérer des fonctions discontinues ou non-différentiables [24].

2.2 Applications

Les AE ont été utilisés dans divers domaines et pour une grande variété de problèmes, allant des sciences naturelles et de l'informatique à l'industrie et au management, en passant par la finance, l'économie ou encore les sciences sociales.
Concernant la vision par ordinateur, les AE ont été appliqués avec succès à différentes tâches telles que la classification d'image, la reconnaissance faciale, l'extraction de caractéristiques, la reconnaissance d'objets ou encore la segmentation d'images. En particulier, on distingue deux types d'applications pour la segmentation d'images [13]. La première vise à améliorer les techniques de segmentation existantes en recherchant leurs meilleurs paramètres. La seconde est quant à elle une réelle classification individuelle de chaque pixel. Ces deux types de segmentation utilisant des AE ont surtout été appliquées à des images médicales [4, 22, 12, 31] mais certaines ont aussi été utilisées pour les images satellites. Par exemple, Yiqiang *et al*. ont proposé le « *Chaos Genetic Algorithm* » qui vise à optimiser les clusters initiaux pour la classification non supervisée de pixels d'images Landsat-5 de la ville de Huainan en Chine [32]. Sur le même principe, Awad a développé une méthode non supervisée basée sur un algorithme génétique multi-objectif afin de trouver les pixels à utiliser à l'initialisation d'une méthode de clustering. Pour cela, l'algorithme génétique maximise à la fois le nombre de pixels dans chaque cluster ainsi que leur homogénéité, en fonction de leur signature spectrale [1]. De plus, Mylonas *et al*. ont développé l'algorithme GeneSIS qui combine les résultats de classification de pixels faite par un SVM avec un algorithme génétique [29]. Les auteurs ont récemment amélioré leur méthode en proposant une version basée sur la recherche locale [28]. Enfin, Hinojosa *et al*. ont également utilisé un algorithme multi-objectif pour la segmentation d'images satellites multi-spectrales en recherchant les meilleurs seuils de valeurs spectrales pour classifier les pixels [18].
L'ensemble de ces travaux permettent la segmentation par l'utilisation d'AE associés à d'autres méthodes. Dans ce travail, nous montrons qu'un AE, utilisé seul, est capable de générer des modèles de segmentation d'images satellites basée uniquement sur l'information spectrale des pixels avec une bonne précision.

2.3 Zoetrope Genetic Programming

Nous utilisons ici un algorithme évolutionnaire propriétaire et original appelé, Zoetrope Genetic Programming (ZGP) [5]. Cet algorithme est au coeur de l'application web de machine learning automatique appelée TADA, développée par la startup française MyDataModels [1]. ZGP est considéré comme un algorithme de régression symbolique, dans lequel les individus de la population pourraient être assimilés à des arbres d'expressions. En effet, même si la façon de générer les individus est unique et non standard, au final ceux-ci sont représentés par une combinaison mathématique de constantes et de variables qui évolue durant la phase d'apprentissage afin de trouver celle qui s'adapte le mieux aux données. Plus précisément, le mécanisme de construction des formules est le suivant : pour commencer, m_e éléments $(E_1, \ldots, E_{m_e})$ sont sélectionnés aléatoirement parmi les variables d'entrée (avec une probabilité de 90%) ou des constantes aléatoires (avec une probabilité de 10%). Ces éléments subissent ensuite m_m étapes de maturation qui consistent à appliquer l'opération de fusion représentée par l'équation (1) aux couples d'éléments E_i et E_j :

$$f(E_i, E_j) = r \cdot \mathrm{op}_1(E_i, E_j) + (1 - r) \cdot \mathrm{op}_2(E_i, E_j) \quad (1)$$

Où op_i, $i = 1, 2$ sont des opérateurs mathématiques uniformément choisis dans un ensemble prédéfini $\mathcal{O}$, et $r = U[0, 1]$; le résultat de $f(E_i, E_j)$ remplace soit E_i soit E_j.
A la fin des étapes de maturation, les éléments matures sont appelés *"zootropes"* et sont donc le résultat de plusieurs opérations de fusions, représentés par une formule mathématique composée de variables et constantes. Ils sont ensuite combinés et pondérés de façon linéaire grâce à une fonction mathématique qui vise à minimiser les erreurs de prédiction sur une partie des données. Cette fonction diffère avec la tâche apprise par l'algorithme (régression, classification binaire ou multi-classes).
Finalement, bien que les zootropes aient été construits de façon naïve par un processus de sélection aléatoire de varibales, ceux sélectionnés par le modèle final sont ceux qui possèdent les variables explicatives les plus pertinentes.
Ensuite, l'algorithme de programmation génétique considère les zoetropes en tant que modèles et les fait évoluer grâce aux opérateurs génétiques. Là encore, ZGP possède des méthodes de mutation et de recombinaison non standards. En effet, lors de la mutation, l'algorithme sélectionne des couples de modèles et remplace le plus mauvais par un "mutant" du meilleur (évalués grâce à la *fitness function*). Durant la recombinaison, l'algorithme sélectionne le meilleur et le plus mauvais modèle parmi un ensemble de modèles ; il propage ensuite un certain nombre d'éléments et de fusions du meilleur modèle vers le plus mauvais. La mutation et la recombinaison sont répétées plusieurs fois au cours d'une même génération. A la fin de chaque génération, tous les modèles sont évalués grâce à la *fitness function* en utilisant un ensemble de données de validation et le meilleur modèle est gardé en mémoire. Une nouvelle

1. www.mydatamodels.com

génération débute alors à partir de tous les modèles obtenus à la précédente génération. Au terme de l'apprentissage, le meilleur modèle, toutes générations confondues, sera sélectionné comme modèle final.
A l'issue de la phase d'apprentissage, l'algorithme sélectionne finalement le sous-ensemble de variables les plus pertinentes pour expliquer le phénomène considéré. Cela se traduit par une expression mathématique relativement simple qui permet de prédire (régression) ou de classifier (classification) de nouvelles données. Grâce à son format compréhensible par l'utilisateur, cette formule est facilement interprétable et permet d'identifier les variables les plus importantes pour prédire le phénomène sous-jacent. De plus, ce type d'expression peut être programmé dans différents langages (Python, JavaScript, C++) et être aisément implémenté sur des microcontrôleurs dans des systèmes embarqués. En effet, le modèle généré possède une très faible taille mémoire (de l'ordre de quelques kilo-octets).
Par rapport aux autres types d'AE, ZGP possède plusieurs avantages. D'abord, la façon spécifique de représenter les individus évite l'effet de congestion, c'est-à-dire la construction de longues formules, alors que la plupart des AE doivent d'abord être paramétrés pour cela. En effet, alors que pour les autres AE, des arbres à expressions finis sont générés aléatoirement, ZGP construit d'abord des formules aléatoires de façon naïve correspondant aux branches d'un arbre de taille finie et seulement les plus informatives sont gardées pour donner l'arbre final. Cela permet une recherche plus rapide des solutions. De plus, la formule est ainsi plus facilement interprétable et compréhensible par un expert métier puisque les dépendances entre les variables explicatives et la variable d'intérêt sont explicites et peu nombreuses. Aussi, tel un générateur de variables, ZGP a une capacité intrinsèque à faire de la "*feature engineering*", contrairement aux algorithmes classiques de machine learning pour lesquels ce travail doit être fait en amont durant la phase de préparation des données (pré-processing). Cette capacité est commune aux algorithmes de régression symbolique.

3 Méthode

3.1 Données

Dans cette étude nous avons considéré deux jeux de données labélisées, SPARCS et BIOME, par ailleurs déjà largement utilisés dans des travaux similaires, notamment [20], pour établir la performance d'algorithmes de détection de nuages. Ces données sont constituées d'images multi-spectrales Landsat-8 avec 10 bandes spectrales allant du spectre visible (bandes de l'instrument OLI) à l'infra-rouge thermique (Bandes TIRS) (Table 1).
La base de données SPARCS [25] contient 80 images de 1000x1000 pixels calibrées radiométriquement dont les valeurs représentent pour chaque bande spectrale la réflectance codée sur une échelle de 16-bits. Les images sont labélisées et fournies avec un masque pour 7 classes de pixels : *cloud shadow*, *cloud*, *shadow over water*, *water*, *snow*, *land*, *flooded*.

Bande	**Longueur d'onde (µm)**
Coastal	0.43-0.45
Blue	0.45-0.51
Green	0.53-0.59
Red	0.64-0.67
NIR	0.85-0.88
SWIR1	1.57-1.65
SWIR2	2.11-2.29
Cirrus	1.36-1.38
TIRS1	10.6-11.19
TIRS2	11.50-12.51

TABLE 1 – Bandes spectrales des images du satellite Landsat-8 utilisées dans les jeux de données SPARCS et BIOME

La base de données BIOME [15] est constituée de 96 images Landsat-8 réparties en 8 types de terrains différents appelés biomes : *barren*, *forest*, *grass crops*, *shrubland*, *urban*, *water*, and *wetlands*. Un masque de pixels est fourni pour les 4 classes suivantes : *cloud shadow*, *clear*, *thin cloud*, *cloud*. Comme pour les données SPARCS, la radiométrie des images BIOME représente la réflectance spectrale sur une échelle de 16-bits.
Cette étude se focalise sur la détection de nuages et la distinction entre nuages et neige. Dans cet objectif, tous les pixels ne représentant pas des nuages sont regroupés dans une même méta-classe *no cloud*, sauf lors de l'utilisation de modèles multi-classes pour lesquels les pixels labélisés *snow* sont considérés séparément des pixels non nuageux dans une classe appelée *snow*.
Les données SPARCS ont été choisies pour l'entrainement des modèles en raison notamment de la présence d'annotations de pixels de la classe *snow* qui permet de réaliser l'entrainement de modèles sur des pixels de neige pour les modèles ZGP multi-binaire et multi-classes. Les images BIOME sont utilisées comme images de test pour évaluer et comparer la capacité de généralisation des différents modèles sur de nouvelles images jamais vues lors de l'entrainement.

3.2 Algorithme de référence

Le modèle utilisé pour établir les performances de référence est un UNET tel que décrit dans [30] et représenté Figure 2. Ce type d'architecture n'est pas parmi les plus récentes développées pour la segmentation (SegNet [2] ou DeepLabV3 [6] sont par exemple plus récents). Cependant, UNET a été sélectionné ici comme méthode de référence car cette approche a été très utilisée pour la segmentation de nuages dans les images satellite dans plusieurs applications récentes [9, 26, 20], notamment sur les bases de données SPARCS et BIOME qui font référence dans ce domaine de la télédétection.
L'architecture d'un réseau UNET standard est constituée d'un encodeur (partie gauche) et d'un décodeur (partie droite). L'encodeur est un réseau de neurones convolutif formé d'une alternance de couches de convolution

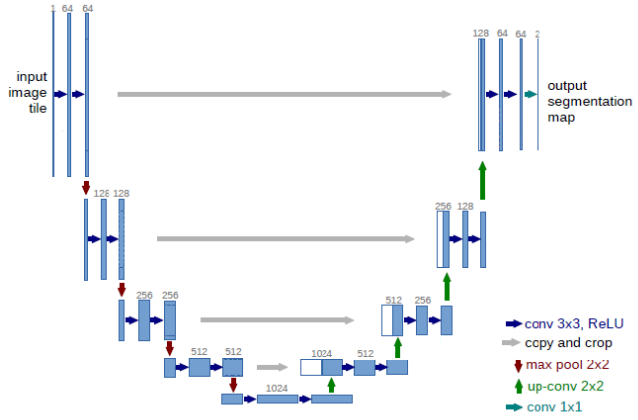


FIGURE 2 – Architecture UNET d'après [30]

par des filtres 3x3 entraînables, des fonctions d'activation ReLU et des couches de max-pooling 2x2 assurant le sous échantillonnage d'un facteur 2 à chaque niveau. Le nombre de filtres de convolution de la couche d'entrée est de 64 et double à chaque niveau pour atteindre 512 au dernier niveau. Chaque étape du décodeur réalise le sur-échantillonnage d'un facteur 2 des couches précédentes dont le nombre est réduit d'un facteur 2 à la sortie de chaque niveau, la concaténation avec les couches de l'encodeur correspondant au même niveau, et la convolution par des filtres 2x2 (noté up-conv sur la Figure 2) suivie par des fonctions d'activation ReLU. A la sortie du réseau, une convolution 1x1 est utilisée pour projeter chacune des composantes du tenseur sur le nombre de classes. Une fonction softmax détermine ensuite la classe des pixels. Pour éviter le sur-apprentissage, un dropout de 50% a été ajouté en sortie de l'encodeur et une *batch normalization* (non représentée sur la Figure 2) après chaque couche de convolution comme pour le Rs-Net dans [20].

L'apprentissage a été réalisé sur des patchs de taille 256x256 extraits des données SPARCS après division par 65 535 de chaque bande spectrale de manière à réduire la dynamique des entrées dans l'intervalle [0,1]. Les paramètres d'apprentissage utilisés sont les suivants :

- Learning rate : $0.5 * 10^{-3}$
- Taille des sous-ensembles : 90% apprentissage (soit 72 images SPARCS) et 10% validation (8 images SPARCS)
- 150 époques d'apprentissage, une validation par époque

Les données ont été traitées par batchs de 96 patchs 256x256 représentant plus de 75 millions d'échantillons d'entrainement (*i.e.* pixels spectraux). La taille de ces batchs a été optimisée en fonction du nombre et de la capacité mémoire des 6 GPU Geforce GTX 1080 utilisés en parallèle lors de l'apprentissage. Il en résulte, malgré la complexité d'UNET, une durée d'apprentissage relativement faible d'environ 90 minutes pour 150 époques sur la totalité des 10 bandes spectrales et des 80 images de la base de données SPARCS.

Le modèle de référence a été entrainé sur deux combinaisons spectrales différentes. Une première combinaison multi-spectrale (MS) comprenant la totalité des 10 bandes spectrales disponibles dans les données Landsat-8 et une combinaison avec uniquement les bandes RGB classiquement utilisées sur des nano-satellites comme dans l'expérience OPS-SAT [14]. Ces mêmes combinaisons de bandes ont également été utilisées pour entraîner les modèles ZGP MS et RGB décrits ci-après.

3.3 Algorithme ZGP

Dans ces travaux nous entrainons ZGP pour générer des modèles de classification permettant de prédire la classe de chaque pixel d'une image à partir de leur information spectrale uniquement. Plus précisément, nous entrainons 3 modèles basés sur les bandes spectrales RGB (modèles RGB) et 3 modèles basés sur les 10 bandes spectrales des capteurs OLI et TIR du satellite Landsat-8 (modèles multi-spectraux ou MS). Dans les deux cas, les 3 modèles sont les suivants :

- un modèle binaire (*cloud*, *no cloud*)
- un modèle multi-binaire : le modèle binaire (*cloud*, *no cloud*) précédent suivi d'un second modèle binaire (*cloud*, *snow*) appliqué uniquement sur les pixels prédits comme *cloud* par le premier modèle
- un modèle multi-classes (*cloud*, *land*, *snow*) dont la classe *land* contient tous les pixels *no cloud* excepté ceux de la classe *snow*.

Pour l'apprentissage, ZGP utilise uniquement 10 000 observations. Chaque observation correspond à un vecteur de pixels spectraux avec respectivement 3 ou 10 variables pour les modèles RGB ou MS. Les 10 000 pixels sont choisis aléatoirement parmi les 80 scènes de la base de données SPARCS de telle sorte que le jeu de données final soit équilibré (nombre égal de pixel dans chaque classe). Les variables d'entrée sont les valeurs des pixels sur 16 bits entre 0 et 65 535. Concernant la variable de sortie (la classe), nous considérons les méta-classes *no cloud*, *cloud* et *snow* définies précédemment. Pour l'apprentissage, ZGP divise le jeu de données de façon automatique et aléatoire en trois sous-ensembles : un ensemble d'apprentissage, un ensemble de validation et un ensemble de test. Les paramètres utilisés pour entrainer l'algorithme évolutionnaire sont les suivants :

- taille des sous-ensembles : 40% apprentissage, 30% validation, 30% test
- taille de la population : 500 individus
- nombre de générations : 100
- critère de terminaison : fin des générations

3.4 Métriques

Les performances sont évaluées de manière globale sur un ensemble de données par les métriques standard suivantes.

$$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$Rappel = \frac{TP}{TP + FN} \tag{3}$$

$$F - score = \frac{2 \cdot precision \cdot rappel}{precision + rappel} \quad (4)$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (5)$$

Quel que soit le modèle, les métriques sont calculées pour la classe *cloud*. Ainsi, les vrais positifs (TP) correspondent au nombre de pixels *cloud* correctement identifiés en tant que nuage alors que les vrais négatifs (TN) correspondent au nombre de pixels *no cloud* correctement identifiés en tant que tels. Les faux positifs (FP) sont le nombre de pixels *no cloud* mal classifiés en *cloud* et inversement pour les faux négatifs (FN). Par définition, l'*accuracy* globale mesure le pourcentage de bonnes prédictions pour toutes les classes. Pour comparer les différents modèles entre eux, on s'intéressera essentiellement à la valeur du F-score qui représente la moyenne de la précision et du rappel pour une classe donnée et permet ainsi de rendre compte à la fois de la complétude des prédictions (rappel) et de leur exactitude (précision).

4 Résultats

4.1 Modèles générés par ZGP

Pour chacun des 6 modèles entrainés, ZGP sélectionne automatiquement les bandes spectrales les plus pertinentes pour la classification à effectuer et n'utilise pas forcément toutes les bandes disponibles, comme illustré dans la Table 2.

Modèle	Classes	Bandes utilisées
RGB binaire	*cloud* / *no cloud*	Red, Blue
RGB multi-binaire	*cloud* / *no cloud* + *cloud* / *snow*	Red, Blue + Red, Green
RGB multi-classes	*cloud* / *land* / *snow*	Red, Green, Blue
MS binaire	*cloud* / *no cloud*	Coastal, Blue, SWIR1, TIRS2
MS multi-binaire	*cloud* / *no cloud* + *cloud* / *snow*	Coastal, Blue, SWIR1, TIRS2 + Coastal, Red, NIR, SWIR2, TIRS2
MS multi-classes	*cloud* / *land* / *snow*	Coastal, Blue, Red, SWIR2, TIRS1, TIRS2

TABLE 2 – Résumé des 6 modèles ZGP avec les bandes sélectionnées par l'algorithme

La formule du modèle MS binaire générée par l'algorithme est donnée à titre d'exemple en Figure 3. On voit que pour ce modèle, ZGP n'a retenu que 4 bandes, ce qui permet une réduction de plus d'un facteur 2 du nombre de bandes à traiter par rapport au modèle de référence UNET MS qui utilise les 10 bandes. Les valeurs des deux formules correspondant aux deux classes (*cloud*, *no cloud*) sont calculées pour chaque pixel. Ensuite, la classe correspondant à la plus grande valeur est attribuée au pixel.

CLOUD :
$-0.339 \times \text{TIRS2} \times \text{SWIR1} + 0.339 \times \text{SWIR1} \times \text{Coastal} + 0.433 \times \text{SWIR1} \times |\text{Coastal}| + 0.227 \times \lfloor 0.439 \times (-\text{TIRS2} + \text{Coastal}) + 0.5601 \times |\text{Coastal}| \rfloor$

NO CLOUD :
$0.855 \times \text{Blue} - 0.855 \times \text{Coastal} + 0.145 \times \text{Blue}^2$

FIGURE 3 – Formule du modèle binaire ZGP multispectral. $\lfloor ... \rfloor$ correspond à l'arrondi à l'entier inférieur.

En terme de temps de calcul, l'entrainement de ZGP est de respectivement 5 et 15 minutes pour générer les modèles binaires et multi-classes présentés ici.

4.2 Performance des modèles

Une première évaluation des performances des différents modèles ZGP a été réalisée sur les données SPARCS. Seulement 1% des données SPARCS sont utilisées lors de l'apprentissage des modèles ZGP contre 90% pour celui des modèles UNET. Ainsi lors de l'inférence sur l'ensemble des données SPARCS, les modèles ZGP sont en situation de généralisation pour 99% des pixels contre seulement 10% pour les UNET. Les conditions d'inférence sur les données SPARCS sont donc plutôt favorables aux modèles UNET. Cependant cette première évaluation permet de comparer les modèles ZGP entre eux afin de mettre en évidence notamment l'apport des modèles MS par rapport aux modèles RGB.
Les données BIOME sont ensuite utilisées comme données de test afin de pouvoir comparer les performances de généralisation des différents modèles dans des conditions identiques (*i.e.* nouvelles données pour tous les modèles).
La Figure 4 indique les valeurs de F-score mesurées sur les bases de données SPARCS (barres pleines) et BIOME (barres hachurées) des modèles ZGP (en vert) et UNET (en bleu) pour les bandes RGB. La Figure 5 fourni les résultats de F-Score pour les modèles MS.

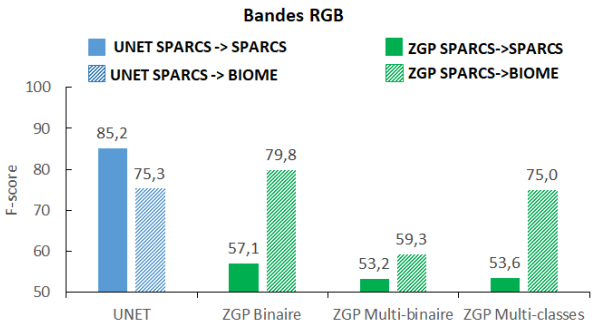


FIGURE 4 – F-score des modèles UNET et ZGP RGB entrainés sur SPARCS et inférés sur SPARCS et BIOME

On observe sur la Figure 4 que les modèles RGB ZGP inférés sur SPARCS sont moins performants que ceux issus de UNET mais que cette tendance s'inverse sur les données BIOME.

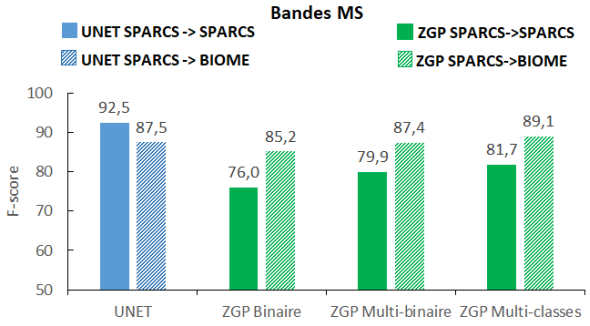


FIGURE 5 – F-score des modèles UNET et ZGP MS entrainés sur SPARCS et inférés sur SPARCS et BIOME

On note une tendance similaire en multi-spectral sur la Figure 5 pour laquelle le F-score UNET MS sur BIOME est seulement de 87,5% alors que celui du modèle ZGP multi-classes MS est de 89,1% (meilleur F-score de tous les modèles UNET et ZGP toutes bandes confondues sur la base de données de test BIOME).
Les Table 3 et Table 4 fournissent le détail des métriques mesurées sur l'ensemble des données SPARCS et BIOME pour les différents modèles UNET et ZGP. Les valeurs en gras indiquent le modèle qui obtient la meilleure métrique.

Métriques sur SPARCS des modèles entrainés sur SPARCS					
Bandes	**Modèles**	**Accuracy**	**Précision**	**Rappel**	**F-score**
RGB	UNET	94.86	**96.19**	76.5	85.22
	ZGP Binaire	76.03	43.69	85.22	57.06
	ZGP multi-binaire	83.43	58.73	48.66	53.22
	ZGP multi-classes	85.22	68.44	44.02	53.57
MS	UNET	**97.12**	93.77	**91.2**	**92.46**
	ZGP Binaire	89.16	66.53	88.68	76.02
	ZGP multi-binaire	91.52	73.88	86.97	79.89
	ZGP multi-classes	93.71	93.4	72.67	81.74

TABLE 3 – Métriques de modèles UNET et ZGP mesurées sur les données SPARCS

Métriques sur BIOME des modèles entrainés sur SPARCS					
Bandes	**Modèles**	**Accuracy**	**Précision**	**Rappel**	**F-score**
RGB	UNET	79.98	**90.88**	64.26	75.29
	ZGP Binaire	77.61	69.83	92.96	79.75
	ZGP multi-binaire	67.04	71.63	50.54	59.26
	ZGP multi-classes	77.53	79.38	71.12	75.02
MS	UNET	87.52	83.27	92.22	87.51
	ZGP Binaire	83.67	74.67	**99.25**	85.22
	ZGP multi-binaire	86.79	79.91	96.4	87.38
	ZGP multi-classes	**89.26**	85.92	92.54	**89.1**

TABLE 4 – Métriques de modèles UNET et ZGP mesurées sur les données BIOME

Concernant l'utilisation des modèles, sur un CPU Xeon multi-coeurs Silver 4114 x86 64 bits à 2.20GHz, le débit d'inférence des modèles ZGP binaires est de 24 Mpixels/s en RGB et 45 Mpixels/s en multi-spectral. Ces débits sont à comparer avec ceux des modèles binaires UNET qui sont de 2,3 Mpixel/s en MS et 2,7 Mpixels/s en RGB. A noter que le débit d'inférence des modèles ZGP dépend essentiellement du nombre et de la complexité des opérateurs mathématiques à appliquer entre les bandes et peu du nombre de bandes spectrales. Ainsi pour effectuer une séparation pertinente des classes, ZGP conduit à un modèle RGB avec plus d'opérations et des opérateurs plus complexes qu'en MS où les bandes sont intrinsèquement plus discriminantes. C'est l'inverse avec les modèles UNET dont le débit d'inférence diminue logiquement lorsque le nombre de bandes à traiter par la couche d'entrée du réseau augmente.

4.3 Interprétation des modèles RGB

Le modèle multi-binaire ZGP n'améliore pas la performance du modèle binaire en RGB. En effet, ce modèle permet de supprimer certains faux positifs dus à des pixels de neige classés *cloud* par le modèle binaire mais il introduit des faux négatifs en déclassant en *snow* des pixels de nuage précédemment bien prédits. Cela est bien visible sur la Figure 6 si l'on compare les faux positifs (en rouge) et les faux négatifs (en bleu) des modèles ZGB binaire et multi-binaire en RGB. Ceci montre que les bandes RGB ne sont pas suffisamment discriminantes pour séparer correctement certains pixels de nuage et de neige malgré un entrainement spécifique sur ces deux classes. Ce phénomène est confirmé par le modèle multi-classes dont les résultats sont très similaires à ceux du multi-binaire. UNET s'en sort en général mieux en RGB que les modèles ZGP RGB. En effet, les filtres de convolution, en réalisant une analyse du contexte autour de chaque pixel, permettent au réseau de neurones de discriminer la neige des nuages en l'absence d'information spectrale peu pertinente. Le UNET échoue parfois à détecter les nuages à partir des seules bandes RGB en produisant des faux négatifs sur les nuages peu texturés ou proches de la saturation. Le modèle ZGP binaire tend plutôt à créer des faux positifs sur les régions à forte radiométrie (neige sur les reliefs de l'image SPARCS ou roches claires sur l'image BIOME).

4.4 Interprétation des modèles MS

Toutes les métriques des modèles ZGP MS sont améliorées par rapport à celles des modèles ZGP RGB, notamment le F-score, comme illustré sur les Figure 4 et Figure 5. Il est cependant intéressant de noter que l'ajout des bandes MS améliore le rappel du UNET mais dégrade fortement la qualité des prédictions (précision du UNET MS inférieure à celle du UNET en RGB). A la vue de ces résultats, on peut s'interroger sur la capacité d'un réseau de neurones comme le UNET à correctement exploiter l'information spectrale lorsque le nombre de bandes devient grand. A l'inverse, on remarque sur les images de la Figure 6, la très bonne capacité des modèles ZGP MS à détecter les nuages dans des conditions difficiles, notamment en présence de neige grâce à une information spectrale beaucoup plus riche qu'en RGB. Le modèle ZGP binaire MS tend cependant à produire encore quelques faux positifs sur les bords des zones enneigées des reliefs de l'image SPARCS. Ces faux positifs sont en partie corrigés par le modèle multi-binaire qui ne rajoute pas de faux négatifs lors de cette correction en exploitant l'information portée par les

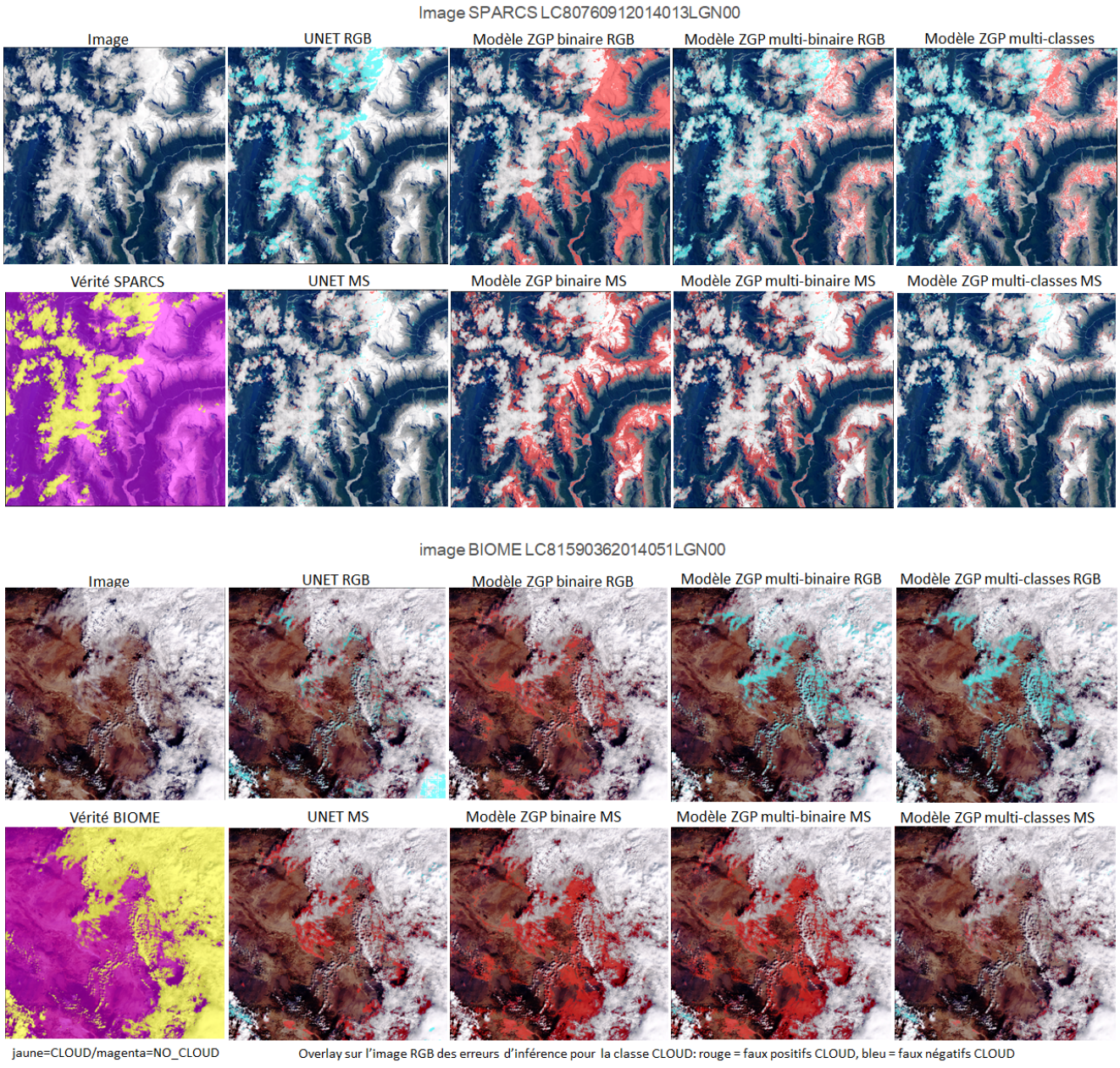


FIGURE 6 – Résultats d'inférence des modèles UNET et ZGP en RGB et MS sur une image extraite de la base de données SPARCS (en haut) et une image extraite de la base de données BIOME (en bas). Pour chaque modèle, les erreurs d'inférence sont superposées à l'image avec en rouge les faux positifs et en bleu les faux négatifs pour la classe *cloud* de la vérité.

bandes infrarouge SWIR et TIR sensibles à la température de surface. Sur l'image BIOME, les faux positifs du modèle ZGP binaire MS sont causés par des roches claires, ils ne sont par conséquent pas corrigés par la post-classification *cloud/snow* du modèle multi-binaire puisqu'aucun de ces faux positifs *cloud* ne correspond en réalité à de la neige. Cet exemple illustre les limites des modèles multi-binaires que n'ont pas les modèles multi-classes MS. Le modèle ZGP multi-classes MS est le plus performant de tous les modèles et rivalise avec les résultats du modèle de référence UNET MS d'un point de vue qualitatif (voir images Figure 6) et quantitatif (voir métriques Table 3 et Table 4), le dépassant notamment sur les images de test BIOME en terme d'*accuracy* globale et de F-score.

Le fait que les résultats des modèles ZGP soient meilleurs sur BIOME que sur SPARCS au contraire des modèles UNET, ne signifie pas pour autant que les modèles ZGP ont une capacité de généralisation supérieure. En effet, comme déjà évoqué précédemment les modèles ZGP sont en situation de généralisation sur 99% des données SPARCS en rai-

son du faible nombre d'échantillons nécessaire à leur entrainement. La différence de performance constatée entre les deux bases de données est donc liée à leur contenu respectif. La mesure des performances hors images contenant de la neige permet d'écarter un possible effet dû à une prépondérance de ce type de zones dans les images SPARCS. L'analyse visuelle des erreurs de prédictions montre que la majorité de ces erreurs est localisée sur les bords de nuages où la labélisation est souvent ambigüe. On observe par ailleurs que les images SPARCS présentent une majorité de nombreux petits nuages fragmentés plus propices à générer ce type d'erreurs de bord que les images BIOME peuplées essentiellement de zones nuageuses étendues.

5 Conclusion

Cette étude présente les performances de l'algorithme évolutionnaire ZGP utilisé dans la WebApp TADA appliqué à un cas d'usage dans le domaine de l'imagerie satellite : la segmentation de nuages. Les performances de différents modèles ZGP ont été comparées à celles de modèles issus d'un réseau de neurones profond de l'état de l'art, sur des images satellite publiques. Nous avons pu montrer que les performances des modèles ZGP pouvaient atteindre voire dépasser celles du réseau UNET, en particulier dans le cas de l'utilisation des images multi-spectrales. Cela démontre ainsi que l'information spectrale à elle seule permet la distinction des différentes classes de pixel, notamment entre les nuages et la neige. ZGP possède plusieurs avantages parmi lesquels sa frugalité en données d'apprentissage, l'interprétabilité des modèles fournis, la rapidité d'entrainement et d'inférence ainsi que sa facilité de déploiement et de portage sur des systèmes embarqués. En fonction des applications et objectifs considérés, ces éléments font de ZGP une bonne alternative aux réseaux de neurones, qui sont à l'inverse gourmands en données, en temps d'apprentissage, et difficiles à porter sur des systèmes embarqués.

Dans de futurs travaux, les performances des modèles générés par ZGP sur des images seront encore améliorées par la mise en oeuvre d'une stratégie permettant de sélectionner un nombre minimum d'échantillons d'apprentissage tout en maximisant la repésentativité de cette selection vis-à-vis des données d'entrainement. Nous envisageons aussi d'ajouter de l'information spatiale à l'aide de filtres de contexte comme ceux utilisés par les réseaux de neurones. Aussi, une technique hybride « réseau de neurones / ZGP » permettrait de tirer profit des avantages des deux types d'approches. Par exemple, la formule mathématique des modèles ZGP pourrait remplacer la fonction de classification utilisée dans la dernière couche des réseaux de neurones. Enfin, ZGP permettrait d'optimiser l'architecture et les hyper-paramètres d'un réseau de neurones, comme dans les travaux de Leung *et al*. [23] ou Mondal [27].

Remerciements

Ces travaux ont été menés dans le cadre du projet CIAR ("Chaîne Image Autonome et Réactive") de l'Institut de Recherche Technologique Saint-Exupéry (www.irt-saintexupery.com). Les auteurs remercient les partenaires industriels et académiques du projet : ActiveEon, Avisto, Elsys Design, GEO4i, Inria, LEAT/CNRS, MyDataModels, Thales Alenia Space et TwinswHeel.

Références

[1] Mohamad Awad. Improving satellite image segmentation using evolutionary computation. *American Journal of Remote Sensing*, 1 :13–20, 01 2013.

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet : A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39 :2481–2495, 2017.

[3] Gaetan Bahl, Lionel Daniel, Matthieu Moretti, and Florent Lafarge. Low-power neural networks for semantic segmentation of satellite images. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 2469–2476, 2019.

[4] Leonardo Bocchi, Lucia Ballerini, and Signe Hässler. A new evolutionary algorithm for image segmentation. In Franz Rothlauf, Jürgen Branke, Stefano Cagnoni, David Wolfe Corne, Rolf Drechsler, Yaochu Jin, Penousal Machado, Elena Marchiori, Juan Romero, George D. Smith, and Giovanni Squillero, editors, *Applications of Evolutionary Computing*, pages 264–273, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[5] Aurélie Boisbunon, Carlo Fanara, Ingrid Grenet, Jonathan Daeden, Alexis Vighi, and Marc Schoenauer. Zoetrope genetic programming for regression. In *2021 Genetic and Evolutionary Computation Conference (GECCO'21)*, 2021.

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv :1706.05587v3*, 2017.

[7] Josef Cihlar and J. Howarth. Detection and removal of cloud contamination from avhrr images. *IEEE Transactions on Geoscience and Remote Sensing*, 32(3) :583–589, 1994.

[8] Kenneth De Jong. *Evolutionary Computation – A Unified Approach*. 01 2006.

[9] Johannes Drőnner, Nikolaus Korfhage, Sebastian Egli, Markus Mühling, Boris Thies, Jörg Bendix, Bernd Freisleben, and Bernhard Seeger. Fast cloud segmentation using convolutional neural networks. *Remote Sensing*, 10(11), 2018.

[10] Matthias Drusch, Umberto Del Bello, Stefane Carlier, Olivier Colin, Valerie Fernandez, Ferran Gascon, Bianca Hoersch, Claudia Isola, Paolo Laberinti, Philippe Martimort, Aimé Meygret, François Spoto, Omar Sy, Franco Marchese, and Pier Bargellini. Sentinel-2 : Esa's optical high-resolution mission for gmes operational services. *Remote Sensing of Environment*, 120 :25 – 36, 2012. The Sentinel Missions - New Opportunities for Science.

[11] Agoston Eiben and Jim Smith. *Introduction To Evolutionary Computing*, volume 45. 01 2003.

[12] Zhun Fan, Jiahong Wei, Guijie Zhu, Jiajie Mo, and Wenji Li. Evolutionary neural architecture search for retinal vessel segmentation. *arXiv preprint arXiv :2001.06678*, 2020.

[13] Michael. E. Farmer and David. Shugars. Application of genetic algorithms for wrapper-based image segmentation and classification. In *2006 IEEE International Conference on Evolutionary Computation*, pages 1300–1307, 2006.

[14] Frederic Feresin, Michael Benguigui, Yves Bobichon, Edgard Lemaire, Matthieu Moretti, and Gaetan Bahl. On board images processing using ia to reduce data transmission : example of opssat cloud detection. In *7th On-Board Payload Data Compression Workshop, OBPDC*, 2020.

[15] Steve Foga, Pat L. Scaramuzza, Song Guo, Zhe Zhu, Ronald D. Dilley, Tim Beckmann, Gail L. Schmidt, John L. Dwyer, M. Joseph Hughes, and Brady Laue. Cloud detection algorithm comparison and validation for operational landsat data products. *Remote Sensing of Environment*, 194 :379 – 390, 2017.

[16] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., USA, 1st edition, 1989.

[17] Olivier Hagolle, Mireille Huc, David Villa Pascual, and Gerard Dedieu. A multi-temporal method for cloud detection, applied to formosat-2, venμs, landsat and sentinel-2 images. *Remote Sensing of Environment*, 114(8) :1747 – 1755, 2010.

[18] Salvador Hinojosa, Omar Avalos, Jorge Gálvez, Diego Oliva, Eric Cuevas, and Marco A. Pérez-Cisneros. Remote sensing imagery segmentation based on multi-objective optimization algorithms. In *2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pages 1–6, 2018.

[19] Ben V. Hollingsworth, Liqiang Chen, Stephen E. Reichenbach, and Richard R. Irish. Automated cloud cover assessment for Landsat TM images. In Michael R. Descour and Jonathan Martin Mooney, editors, *Imaging Spectrometry II*, volume 2819, pages 170 – 179. International Society for Optics and Photonics, SPIE, 1996.

[20] Jacob Høxbroe Jeppesen, Rune Hylsberg Jacobsen, Fadil Inceoglu, and Thomas Skjødeberg Toftegaard. A cloud detection algorithm for satellite imagery based on deep learning. *Remote Sensing of Environment*, 229 :247 – 259, 2019.

[21] Michael. D. King, Steven Platnick, W. Paul Menzel, Steven A. Ackerman, and Paul A. Hubanks. Spatial and temporal distribution of clouds observed by modis onboard the terra and aqua satellites. *IEEE Transactions on Geoscience and Remote Sensing*, 51(7) :3826–3852, 2013.

[22] Chih-Chin Lai and Chuan-Yu Chang. A hierarchical evolutionary algorithm for automatic medical image segmentation. *Expert Systems with Applications*, 36(1) :248 – 259, 2009.

[23] Frank. H. F. Leung, Hak-Keung Lam, Sai-Ho Ling, and Peter. K. S. Tam. Tuning of the structure and parameters of a neural network using an improved genetic algorithm. *IEEE Transactions on Neural Networks*, 14(1) :79–88, 2003.

[24] Bailin Li, Jianguo Lin, and Xiuming Yao. A novel evolutionary algorithm for determining unified creep damage constitutive equations. *International Journal of Mechanical Sciences*, 44(5) :987 – 1002, 2002.

[25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[26] Sorour Mohajerani and Parvaneh Saeedi. Cloud-net : An end-to-end cloud detection algorithm for landsat 8 imagery. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 1029–1032, 2019.

[27] A. S. Mondal. Evolution of convolution neural network architectures using genetic algorithm. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2020.

[28] Stelios K. Mylonas, Dimitri G. Stavrakoudis, John B. Theocharis, Georges C. Zalidis, and Ioannis Z. Gitas. A local search-based genesis algorithm for the segmentation and classification of remote-sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(4) :1470–1492, 2016.

[29] Stelios K. Mylonas, Dimitris G. Stavrakoudis, and John B. Theocharis. Genesis : A ga-based fuzzy segmentation algorithm for remote sensing images. *Knowledge-Based Systems*, 54 :86 – 102, 2013.

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net : Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science*, volume 9351, pages 234–241. Springer, 10 2015.

[31] Christopher Walsh and Nick Taylor. Evolution of convolutional neural networks for lymphoma classification. In *24th International Conference on Image Processing, Computer Vision, and Pattern Recognition 2020, IPCV'20*, 2020.

[32] Guo Yiqiang, Wu Yanbin, Ju Zhengshan, Wang Jun, and Zhao Luyan. Remote sensing image classification by the chaos genetic algorithm in monitoring land use changes. *Mathematical and Computer Modelling*, 51(11) :1408 – 1416, 2010. Mathematical and Computer Modelling in Agriculture.

[33] Zhaoxiang Zhang, Guodong Xu, and Jianing Song. Cubesat cloud detection based on jpeg2000 compression and deep learning. *Advances in Mechanical Engineering*, 10, 2018.

[34] Zhe Zhu, Shixiong Wang, and Curtis E. Woodcock. Improvement and expansion of the fmask algorithm : cloud, cloud shadow, and snow detection for landsats 4–7, 8, and sentinel 2 images. *Remote Sensing of Environment*, 159 :269 – 277, 2015.

[35] Zhe Zhu and Curtis E. Woodcock. Object-based cloud and cloud shadow detection in landsat imagery. *Remote Sensing of Environment*, 118 :83 – 94, 2012.

[36] Zhe Zhu and Curtis E. Woodcock. Automated cloud, cloud shadow, and snow detection in multitemporal landsat data : An algorithm designed specifically for monitoring land cover change. *Remote Sensing of Environment*, 152 :217 – 234, 2014.

Session 4 – L'IA pour la détection d'anomalies

Apprentissage auto-supervisé pour la détection d'actions illégales lors de la surveillance du trafic maritime

P. Bernabé[1,2], A. Gotlieb[1], B. Legeard[2,4], F. Olaf Sem-Jacobsen[3], H. Spieker[1]

[1] Simula Research Laboratory, VIAS - Oslo, Norvège
[2] Université de Bourgogne Franche-Comté, Institut FEMTO-ST - Besançon, France
[3] Statsat AS - Oslo, Norvège
[4] Smartesting - Besançon, France

{pierbernabe, arnaud, helge}@simula.no, frank.Sem-Jacobsen@statsat.no,
bruno.legeard@univ-fcomte.fr

Résumé

La surveillance du trafic maritime est confrontée à des difficultés très importantes dans la détection des activités illégales en mer. Dans cet article, nous présentons les premiers résultats d'une méthode d'apprentissage auto-supervisé qui vise à déceler les déconnexions volontaires du système d'identification des navires. En traitant les données provenant de quatre satellites de surveillance norvégiens, notre modèle d'apprentissage vise l'identification de navires soupçonnés d'activités illégales telles que la pêche dans des zones protégées ou bien le franchissement de zones d'exclusion économique en temps réel. Dans cet article, nous présentons une approche fondée sur des techniques d'apprentissage auto-supervisé, et expérimentée à partir de données réelles.

Mots-clés

Apprentissage automatique, Apprentissage auto-supervisé, Système d'identification automatique, Surveillance maritime.

Abstract

The surveillance of maritime traffic is confronted with very important difficulties in detecting illegal activities at sea. In this article, we present the first results of a self-supervised learning method which aims to detect voluntary disconnections of the identification' system of vessels. By processing data from four Norwegian surveillance satellites, our learning model aims to identify vessels suspected of illegal activities such as fishing in protected areas or crossing economic exclusion zones in real time. In this article, we present an approach based on self-supervised learning techniques, and experienced from real data.

Keywords

Machine Learning, Self-Supervised Learning, Automatic Identification System (AIS), Martime Surveillance

1 Introduction

1.1 Surveillance du trafic maritime

La surveillance du trafic maritime est une tâche complexe qui vise à identifier et contrôler l'activité des navires présents dans une zone maritime [12]. La surveillance a plusieurs objectifs, comme par exemple, identifier et guider les navires, aider à la prévention des collisions, lancer des missions de sauvetage en mer ou plus généralement, réguler le trafic maritime. Les systèmes de surveillance modernes visent aussi à détecter des activités illégales telles que les actions de piratage, de pêche illégale, d'intrusion dans les zones d'exclusion économique (EEZ), de transbordement de produits stupéfiants, de dégazage en mer, etc. La plupart du temps, la détection de ces activités illégales repose uniquement sur l'observation des navires, l'analyse visuelle des données et l'expertise des garde-côtes. Pourtant, l'exploitation automatisée des différentes sources de données pourrait s'avérer très utile pour la détection d'actions illégales [1]. En effet, les systèmes de surveillance maritime collectent des données provenant des communication-radio spécifiques émises par les navires, et celles-ci sont captées par différents moyens tels que des balises en mer, les sémaphores côtiers et des satellites dédiés à l'observation du trafic maritime.

L'émission de messages par des transpondeurs VHF [1] embarqués sur les navires d'un certain tonnage (i.e., $\geq$ 300 tonneaux) constitue la source d'information principale de la surveillance maritime. Ces messages s'appuient sur le protocole AIS (*Automatic Identification System*) qui transmet toutes les 2 à 10 secondes des informations telles que l'identifiant unique du navire (MMSI [2]), son statut (amarré, échoué, au mouillage, etc.), son type (cargo, pêche, plaisance, etc.), ses dimensions, sa route, sa vitesse, sa position (latitude, longitude), son cap, etc. Ces messages sont captés non seulement par les autres navires ainsi que les balises et les sémaphores lorsque les récepteurs sont à portée, mais

1. Very High Fréquency
2. Maritime Mobile Service Identity

également par des satellites dédiés à la surveillance du trafic comme indiqué plus haut. Dans ce dernier cas, les données sont nommées S-AIS. Il est important de noter que le capitaine d'un navire a la possibilité de couper volontairement l'émission des messages AIS et ceci est parfois fait dans le but de réaliser des actions illégales en mer. Par exemple, la déconnexion AIS est fréquemment utilisée par certains capitaines de navire de pêche afin de pêcher en toute impunité dans une zone interdite ou fortement réglementée. De part le volume de messages AIS reçus, une analyse manuelle de ces données reste très hasardeuse et la plupart du temps inopérante. Par exemple, en une seule journée (19/5/2019), les satellites de surveillance du trafic maritime opérés par la Norvège ont réceptionné 3 518 649 messages AIS, qui sont à ajouter aux 4 192 381 messages captés par les stations terrestres des côtes norvégiennes.

1.2 Méthodes pour la détection automatique d'actions illégales

L'utilisation de moyens automatisés pour la détection d'actions illégales est donc devenu un enjeu pour la surveillance du trafic maritime [4]. Parmi ces moyens, la recherche de déconnexions AIS volontaires à partir des messages AIS transmis par les navires est apparu comme une première application cruciale [8, 7, 14, 11]. Avec le lancement de satellites de surveillance, la disponibilité de données S-AIS pour des zones habituellement hors de portée a permis de systématiser cette surveillance y compris en pleine mer. Une approche directe pour ce problème consiste à définir des règles qui caractérisent la déconnexion volontaire des transpondeurs VHF mais cette approche se heurte à plusieurs difficultés liées essentiellement à la qualité des données. En effet,

- les messages S-AIS captés par une flotte de satellites de surveillance sont habituellement très irréguliers à cause des conditions météo difficiles, des collisions entre messages, de la position des satellites qui orbitent autour de la terre et qui n'offrent pas une couverture permanente des mers ;
- certains messages S-AIS sont bruités car les données non mises à jour, ou bien émises par des transpondeurs de piètre qualité ;
- certains navires embarquent plusieurs transpondeurs et brouillent volontairement les pistes en changeant d'émission VHF ;
- certaines zones sont saturées en bateaux (e.g., à l'approche des ports ou bien dans des couloirs très empruntés) et conduit à la congestion des modules de réception des satellites ;

Devant l'impossibilité d'identifier un système de règles caractérisant la déconnexion AIS légale et illégale, un courant de recherche a émergé depuis quelques années, qui consiste à entraîner et déployer des modèles d'apprentissage automatique supervisé pour la détection d'actions illégales en mer. La plupart des travaux initiaux se sont concentrés sur la prédiction de trajectoires et la détection d'anomalies pour un navire donné, sans se préoccuper du reste du trafic maritime [12, 4]. De même, des modèles probabilistes du comportement individuel des navires à partir de données historiques AIS ont émergés tels que des modèles de Markov [1, 5] ou bien des réseaux de neurones hiérarchiques [6]. Cependant, ces modèles sont d'une part mal adaptés au traitement de données bruitées telles que les données AIS issues des satellites et d'autre part, en ne prenant en compte que les comportements individuels des navires, ils manquent souvent de pertinence pour détecter des actions illégales dans la multitude des navires traversant une zone maritime donnée. Récemment, l'utilisation de l'apprentissage multi-tâches pour l'entraînement d'un modèle probabiliste de trajectoires typiques de navires à partir de données AIS a donné des résultats très encourageants pour la surveillance du trafic maritime [9]. En particulier, en utilisant une représentation qui régularise la fréquence des messages et complète le jeu de données, cette approche a permis un bien meilleur traitement des données bruitées [10] même si les jeux de données utilisés proviennent, non pas des satellites de surveillance, mais des balises en mer.

1.3 Nos contributions

Les résultats présentés dans cet article s'articulent autour de trois contributions distinctes :

1. À partir de données S-AIS provenant du captage d'une flotte de satellites opéré par notre partenaire Norvégien StatSat AS, nous constatons que la présence d'évènements de déconnexion AIS est très fréquente et seule une très faible proportion correspond à des déconnexions volontaires. Notre approche, basée sur l'apprentissage profond, permet de détecter ces déconnexions de manière fiable. C'est, à notre connaissance, la première fois que les déconnexions volontaires sont recherchées et détectées dans des données S-AIS avec un modèle à base de réseau de neurones multi-couches. Ceci ouvre des perspectives intéressantes pour couvrir des zones maritimes situées en pleine mer, c'est-à-dire situées loin des côtes et des balises, et pour la détection temps-réel des déconnexions permettant une intervention plus rapide des garde-côtes ;
2. Nous utilisons des données non-étiquetées pour entraîner nos modèles à base de réseaux de neurones dans une approche d'apprentissage auto-supervisée. Ces données n'ont pas été annotées par des opérateurs capables de discriminer les déconnexions AIS volontaires des autres, ce qui facilite l'automatisation et la généralisation de notre approche ;
3. À l'inverse d'autres méthodes existantes, notre approche ne s'appuie pas sur la reconstruction de messages AIS manquants. Nous créons une représentation qui s'appuie sur les réseaux d'attention et bénéficions ainsi d'une plus grande précision dans l'analyse des données. C'est, à notre connaissance, la première fois que les réseaux d'attention sont utilisés pour traiter des données AIS. Les résultats expérimentaux que nous avons obtenus démontrent les bénéfices liés à cette représentation.

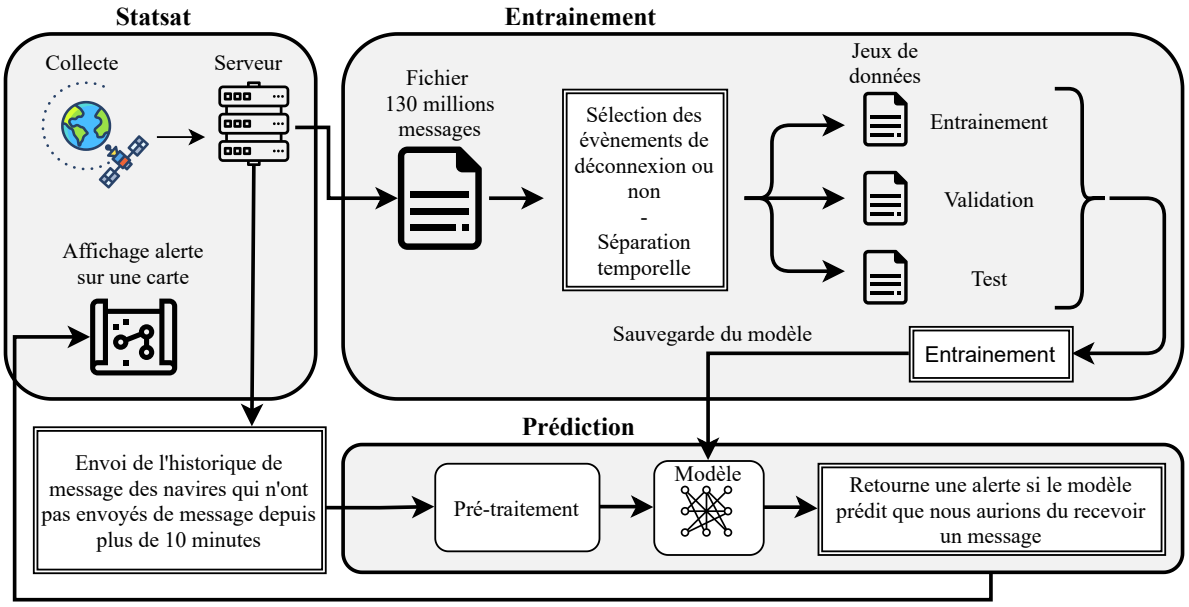


FIGURE 1 – Schéma de fonctionnement général de notre méthode d'apprentissage auto-supervisé pour la détection de déconnexions AIS volontaires.

1.4 Plan du papier

Le reste de cet article est organisé de la manière suivante : la section 2 donne un aperçu général de notre approche et indique les pré-requis nécessaires à sa compréhension avec une présentation succincte de l'apprentissage auto-supervisé et des réseaux d'attention. La section 3 décrit le traitement du flux des données, tandis que la section 4 présente notre modèle d'apprentissage profond et la représentation à base de réseaux d'attention. Dans la section 5, nous interprétons les résultats expérimentaux obtenus. Enfin, la section 6 conclut cet article en évoquant les pistes de travail en cours concernant l'application de l'apprentissage auto-supervisé à d'autres problématiques de la surveillance du trafic maritime.

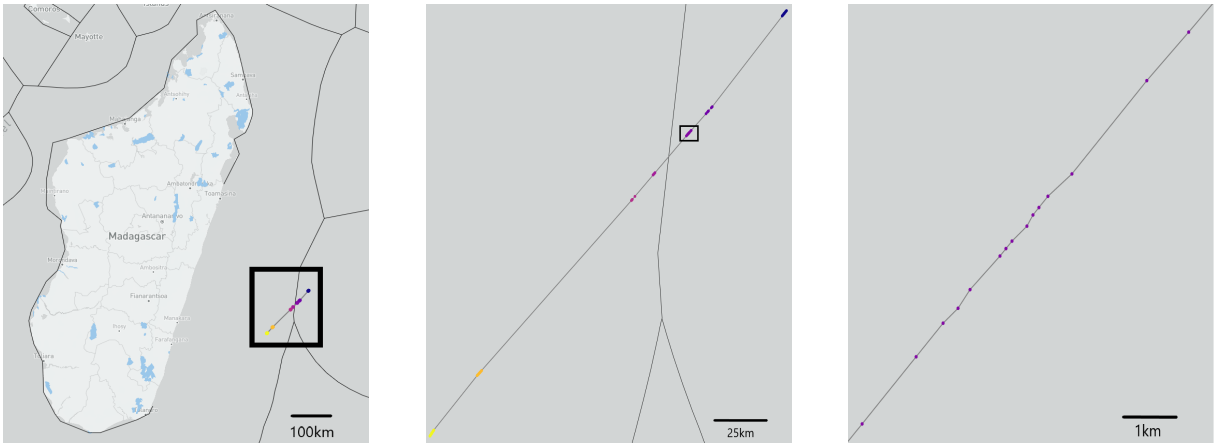
2 Aperçu de notre approche

2.1 Schéma de fonctionnement général

La figure 1 présente le schéma général de notre méthode d'apprentissage auto-supervisé de déconnexions volontaires AIS. Le schéma se décompose en trois blocs distincts, intitulés *Statsat*, *Entraînement* et *Prédiction*. Le bloc Statsat correspond au travail de collecte et stockage des données S-AIS en provenance de satellites d'observation géo-marins gérés par l'opérateur étatique Norvégien Statsat AS. Le bloc Entraînement correspond au travail de sélection et de préparation des données et de calcul des annotations supplémentaires afin de créer un jeu de données propice à l'entraînement d'un modèle d'apprentissage auto-supervisée. Ce jeu de données est composé, comme à l'accoutumée, de trois sous-ensembles distincts : les données d'entraînement, les données de validation et les données de test. Une fois le modèle entraîné, celui-ci est déployé dans le bloc Prédiction qui prend en entrée l'historique en messages AIS de navires qui n'ont pas envoyé de messages depuis plus de 10 minutes. Après une phase de pré-traitement, les trajectoires de ces navires sont classifiées par le modèle entre ceux présentant un risque d'activités illégales (risque de déconnexions AIS volontaires) et les autres. Dans le premier cas, une alerte spécifique peut-être retournée au block StatSat qui se charge d'alerter la garde-côtière si la menace se confirme. À titre d'exemple, la figure 2 montre la trajectoire d'un navire identifié par notre méthode. La figure montre la distance parcourue et l'irrégularité entre les messages. En pleine mer, au franchissement d'une ligne territoriale, le navire cesse d'émettre les messages AIS, ce qui le rend suspect d'activités illégales.

2.2 Apprentissage auto-supervisé

Détecter les déconnexions AIS volontaires se heurte à un obstacle d'importance : il n'existe pas de moyen parfaitement adéquat d'annoter les jeux de données. En effet, comme indiqué plus haut, une solution consisterait à entraîner un modèle avec des données annotées par les garde-côtes, mais un tel travail est peu réaliste en pratique, du fait du temps exigé pour des personnels très qualifiés et très occupés. De plus, dans la mesure où de très nombreuses déconnexions AIS sont involontaires, il s'agirait peu ou prou de rechercher des aiguilles dans une botte de foin et la constitution d'un jeu de données équilibré serait extrêmement fastidieuse. Notre solution pour ce problème a donc consisté à extraire l'annotation depuis les données elles-mêmes dans une approche d'apprentissage auto-supervisé. Du fait que la très grande majorité des exemples où un bateau n'envoie pas de messages AIS pendant un laps de



(a) (b) (c)

FIGURE 2 – Exemple d'une trajectoire d'un navire à différentes échelles (de la plus grande échelle (a) à la plus petite (c)). La trajectoire est ici composée de 100 messages AIS, tandis que le navire parcourt une distance d'environ 250km.

temps suffisant est due à une perte de connexion AIS involontaire, notre modèle peut être entraîné avec l'objectif d'apprendre la tâche qui consiste à déterminer si un message doit être reçu dans ce laps de temps ou non. Si notre modèle prédit qu'un message doit être reçu dans ce laps de temps et qu'il n'arrive pas, alors le navire correspondant peut être classifié comme suspect. Ce laps de temps significatif, à la vue des données S-AIS dont nous disposons, a été fixé à 10 minutes.

Cette approche a l'avantage de permettre la construction d'un large jeu de données annotées à partir de l'historique des messages AIS, sans nécessiter une annotation manuelle. En effet, il suffit d'extraire des exemples de disparition de plus de 10 minutes ainsi que des exemples de réception de messages dans les 10 minutes pour construire un large jeu de données bien équilibré.

2.3 Réseaux d'attention

Dans le domaine de l'apprentissage auto-supervisé, les réseaux d'attention et les *"transformer models"* ont montré des résultats spectaculaires, en particulier dans le traitement automatique du langage naturel [13]. Introduits en 2015 dans [2], l'objectif des réseaux d'attention est d'améliorer l'apprentissage des parties les plus importantes des données d'entrée, en consacrant plus de puissance de calcul à celles-ci. Ces réseaux se sont montrés capables de corriger certains défauts inhérents des *"Long Short Term Memory"*. Une variante des premiers réseaux d'attention, nommée *"self-attention network"* (ou bien *"intra-attention network"*) permet de mettre en relation différents éléments d'une même séquence d'entrée pour en créer ce qui s'appelle *une représentation* [3]. Ces réseaux se sont révélés très utiles dans la création automatique de résumés de texte. Les *"tranformer models"* sont une variante des *"self-attention networks"* construits sans utiliser d'architecture récurrente. Ceci est possible grâce au mécanisme *"multi-head self-attention"* [13].

Dans notre approche, nous utilisons la partie encoder du *"tranformer models"* composé de deux blocs *"tranformer"*. Cet encodeur construit une représentation de la trajectoire du navire basée sur l'attention avec l'objectif de trouver des relations entre les différents messages d'un même navire, plus ou moins proches, et ainsi créer *une représentation*, comme détaillé dans la section 4.

3 Traitement du flux de données

3.1 Messages AIS

Les messages AIS sont composés d'informations statiques et dynamiques. Les champs statiques incluent les identifiants internationaux normalisés du navire, i.e., MMSI [3] and IMO, le nom du navire, le signe d'appel, la longueur, la largeur et le type de navire. Ces éléments statiques, qui sont saisis manuellement par le capitaine du navire dans l'émetteur AIS, sont automatiquement transmis sur un canal de diffusion, toutes les 6 minutes. Les émetteurs AIS envoient également des informations dynamiques toutes les 2 à 10 secondes selon la vitesse du bateau, ou toutes les 3 minutes si le bateau est au mouillage. Les informations dynamiques comprennent l'état de navigation (par exemple, "au mouillage", "pêche", etc.), la position du navire (latitude LAT, longitude LON), la vitesse du navire (SOG) [4], sa direction par rapport au pôle Nord (COG) [5], son cap qui est la direction (par rapport au pôle nord magnétique ou au pôle nord géographique) et les horodatages. Tous les messages AIS ne contiennent pas les mêmes informations et ne sont pas toujours envoyés à des horodatages réguliers. En règle générale, les messages AIS ont une portée d'environ 20 à 40km. La limitation de cette portée est due à la courbure de la terre et à la hauteur à laquelle l'antenne est installée sur les navires.

Depuis une dizaine d'années environ, le corps de garde-côtes utilise des satellites pour capter les messages AIS en

3. Maritime Mobile Service Identity
4. Speed Over Ground
5. Course Over Ground

TABLE 1 – Jeux de données utilisés pour le bloc Entraînement.

Jeux de données	Nombre de messages	Uniques navire	Nombre d'exemples extraits	Date
Entraînement	101 037 023	132 486	80 000	24/04 -> 26/05
Validation	12 497 753	88 489	10 000	26/05 -> 30/05
Test	12 945 336	91 023	10 000	30/05 -> 03/06

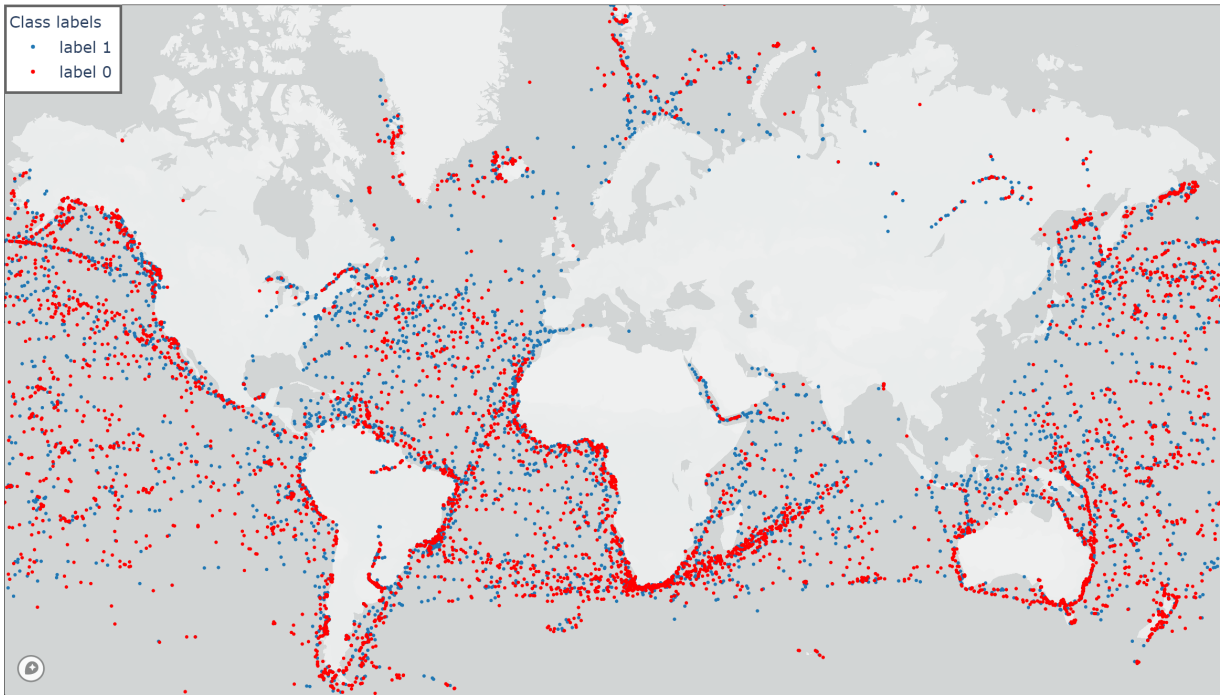


FIGURE 3 – Répartition des échantillons sur la surface du globe

dehors des zones couvertes par les balises. Cependant, il existe un problème inhérent à la norme AIS : le schéma d'accès radio défini dans la norme crée seulement 4500 plages horaires disponibles à chaque minute et les récepteurs peuvent être facilement submergés par de grandes empreintes de réception AIS. De part le nombre croissant d'émetteurs-récepteurs AIS et l'élargissement de la zone de captation des satellites, des collisions de messages peuvent avoir lieu et conduire à la disparition de certains d'entre eux. Certains ports de la mer Méditerranée, de la mer du Nord et des côtes chinoises sont très difficiles à surveiller à cause de cela.

3.2 Collection des données

Statsat est une compagnie norvégienne qui développe des infrastructures spatiales et opère des satellites de surveillance maritime pour le compte du gouvernement norvégien. La compagnie est responsable de la gestion des satellites *AISSAT 1 et 2* et *NORSAT 1 et 2* qui sont dédiés à la supervision des transmissions AIS au nom de l'administration côtière norvégienne et de l'agence spatiale norvégienne. Ces satellites sont positionnés en orbite polaire à une hauteur orbitale de 600-650km, de telle sorte qu'à chaque rotation autour de la terre, ils se décalent pour couvrir toutes les latitudes. Le choix de cette orbite conduit à une bien meilleure couverture des pôles par rapport à l'équateur. Cela conduit à une meilleure efficacité du modèle au niveau des pôles. Le jeu de données S-AIS proposé par Statsat pour le bloc Entraînement est composé de 126 480 112 (environ 126 millions) de messages AIS provenant de l'ensemble du globe, ce qui correspond aux messages collectés entre le 24 avril 2019 et 3 juin 2019 (environ 6 semaines de données).

3.3 Sélection et augmentation de caractéristiques

Notre approche vise la détection des déconnexions AIS volontaires pour les navires en mer (ce qui est illégal). Ainsi, partant du jeu de données S-AIS fourni par le bloc Statsat, pour chaque message AIS, nous sélectionnons les caractéristiques pertinentes que sont les champs de position (lat, lon), l'horodatage (t), et la vitesse (s).

Par ailleurs, nous enrichissons les caractéristiques avec Δt la différence de temps par rapport au message précèdent du même navire, ΔD_V la différence en mètres sur l'axe vertical avec le message précédant du même navire, ΔD_H la différence en mètres sur l'axe horizontal, D_P la distance au port, T_D la seconde du jour (entre 0 et 86 400). Il est à noter que :

— Δt et T_D permettent d'avoir une meilleure précision

sur la dimension temporelle que celle donnée par l'horodatage t ;
— ΔD_V et ΔD_H améliorent la compréhension de la position sur de petites distances ;
— D_P permet de filtrer les échantillons pertinents ;

Ainsi, le vecteur de caractéristiques correspondant à un message prend la forme suivante :

$$x = [t, lat, lon, s, \Delta t, \Delta D_V, \Delta D_H, D_P, T_D]$$

3.4 Trajectoire ou Échantillon

Un échantillon ($\mathcal{E}$) est une séquence temporelle de $\mathcal{T}$ messages successifs d'un navire $\mathcal{E} = [x_1, x_2, \ldots, x_T]$. Nous l'appelons également *trajectoire* car cela correspond à la trajectoire d'un navire dans une fenêtre temporelle. Il est à noter qu'il n'y a aucune condition posée sur les valeurs maximales de Δt et ΔD, entre deux messages successifs au sein d'une trajectoire.

3.5 Séparation des jeux de données

Les jeux de données utilisés pour l'entraînement, la validation et le test du modèle sont séparés en fonction de leur date. Nous aurions pu aussi séparer par navire mais la séparation temporelle a l'avantage de se rapprocher de l'utilisation qui est faite du modèle lors de son déploiement. L'objectif est d'effectuer des prédictions en s'appuyant sur les données passées de la trajectoire d'un navire. La table 1 décrit le volume des jeux de données ainsi que leur division temporelle.

3.6 Sélection et traitement des trajectoires

La figure 3 indique la répartition des messages AIS à la surface du globe. Pour atteindre un bon équilibre du jeu de données à partir de l'ensemble de ces messages, nous extrayons aléatoirement et de manière équitable 1) des trajectoires de navires présentant une continuité de transmission, et 2) des trajectoires de navires présentant une déconnexion AIS. Un seuil arbitraire de continuité a été fixé à 10 minutes pour l'entraînement de notre modèle car nous avons constaté que les déconnexions AIS sont assez courantes dans une période inférieure mais qu'elles deviennent anormales au-delà de 10 minutes. De manière empirique, nous avons constaté que ce seuil de 10 minutes est pertinent pour l'entraînement mais rien n'empêche, lors de la mise en production, de se concentrer sur des déconnexions AIS plus longues (e.g., 1 heure).
Pour la sélection des trajectoires dans le jeu de données, nous posons deux conditions :

1. l'historique des messages AIS du navire concerné doit être composé d'au moins 50 messages. Ceci exclut les bateaux qui viennent de rentrer dans la zone d'observation temporelle de notre jeu de données et qui sont peu pertinents pour l'entraînement du modèle. La limite de 50 messages a été choisie en utilisant les statistiques de la figure 5, on s'aperçoit que de 50 à 100 messages un navire peut parcourir un grande distance et que l'on rencontre régulièrement une longue déconnexion permettant ainsi au modèle de comprendre des situations de déconnexions ;
2. la trajectoire du navire doit être à plus de 5 kilomètres d'un port. Cela permet d'éliminer les exemples de déconnexions AIS volontaires (légales) qui ont lieu dans les ports. De plus, il est relativement facile pour la garde côtière de contrôler les bateaux sans transmission AIS qui se situent dans les ports. Pour déterminer la distance au port le plus proche, nous calculons la distance de l'arc entre deux points sur une sphère [6] avec une base de données de $20\,756$ ports fournie par l'organisation *"global fishing wacth"* [7].

Étant donné R le rayon de la terre (6371m), deux points p_1 (resp. p_2) ayant pour latitude lat_1 (resp. lat_2) et longitude lon_1 (resp. lon_2) (en radians), la formule utilisée pour le calcul de la distance d de l'arc sur un grand cercle est la formule d'Harvesine :

$$d(p_1, p_2) = 2R * \arcsin(\sqrt{\alpha + \beta})$$
$$\alpha = \sin^2(\frac{lat_1 - lat_2}{2})$$
$$\beta = \cos(lat_1) * \cos(lat_2) * \sin^2(\frac{lon_1 - lon_2}{2})$$

4 Modèle d'apprentissage et entraînement

4.1 Architecture du modèle

La figure 4 donne un aperçu de l'architecture générale utilisée pour notre modèle d'apprentissage auto-supervisé. Tout d'abord, la partie pré-traitement divise l'entrée $\mathcal{E}$ en deux vecteurs, $\mathcal{V}_\mathcal{H}$ l'historique des messages AIS qui contient les informations relatives au précédent message et $\mathcal{V}_\mathcal{L}$ la position la plus récente. Ensuite, $\mathcal{V}_\mathcal{H}$ et $\mathcal{V}_\mathcal{L}$ sont normalisés par la couche $\mathcal{N}$. La division de l'entrée a pour objectif d'avoir un encodage qui prend en compte les possibles faibles distances que l'on peut retrouver entre deux messages, tout en conservant une très grande précision sur la position de la déconnexion sur le globe. $\mathcal{V}_\mathcal{H}$ est donné en entré à deux blocs *"transformers"* successifs avec l'objectif de faire des relations entre des messages plus ou moins lointains et d'extraire les informations importantes de la trajectoire. En revanche, $\mathcal{V}_\mathcal{L}$ est mis de coté avant d'être concaténé avec la représentation $\mathcal{R}$ de la trajectoire en sortie du transformer. Un ensemble de couches denses détermine si oui ou non, un message devrait être reçu dans les dix prochaines minutes. Au total, Le modèle $\mathcal{M}$ est composé de $4\,690\,021$ de paramètres entraînables.
Dans la figure 4 , $\mathcal{B}$ représente la taille des batchs (groupe de données) et $\mathcal{W}$ la fenêtre de messages utilisés pour l'entraînement. Les résultats optimaux ont été obtenus avec $\mathcal{B} = 128$, $\mathcal{W} = 100$ et $\mathcal{R} = 64$.

6. La distance de l'arc sur une sphère n'est pas aussi précise que les formules de Vincenty en géodésie puisque la terre n'est pas une sphère parfaite, mais son calcul a l'avantage d'être vectorisable, ce qui est nécessaire dans le cas où le jeu de données contient un très grand nombre de trajectoires.
7. `https://globalfishingwatch.org/datasets-and-code/anchorages/`

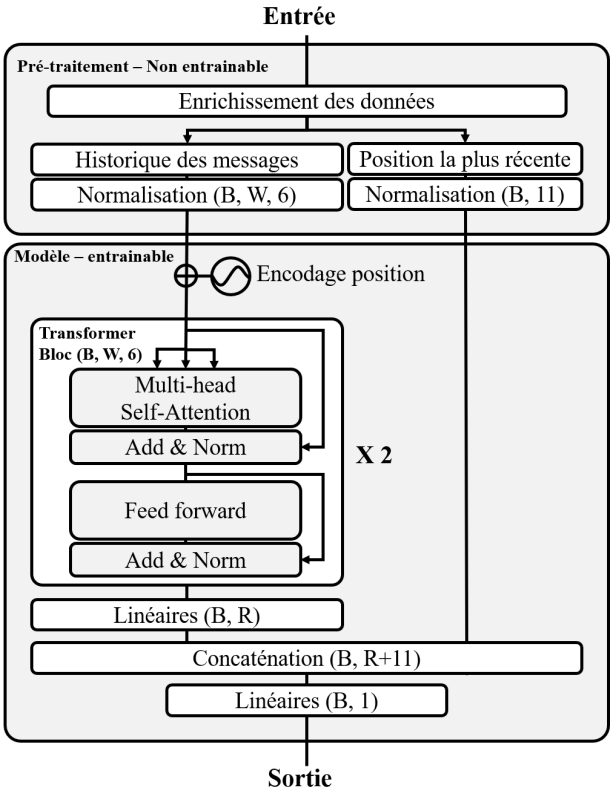


FIGURE 4 – Architecture du modèle

4.2 Préparation des données

Une partie importante de notre travail a été de trouver un encodage et une normalisation, nous permettant de gérer à la fois les grandes distances qui peuvent séparer deux messages tout en gardant une précision suffisante pour les messages espacés seulement de quelques secondes. La figure 5 montre la diversité des trajectoires et des messages. On peut noter par exemple que 10% des trajectoires dure moins de 100 minutes quand 10% des trajectoires durent plus de 40 heures. Dans ce dernier cas, les satellites n'ont été capable de capturer que 100 messages dans les 40 dernières heures. Cette grande durée est principalement due à l'écart entre quelques messages. En effet, on constate que 10% des trajectoires contiennent au moins un coupure de plus de 600 minutes (10h). La même analyse peut être effectuée sur les distances. De plus, nous avons travaillé à rendre le modèle générique pour qu'il puisse être utilisé pour analyser les trajectoires de n'importe quel instant et région du globe.

4.2.1 Historique des messages

Pour rendre le traitement de $\mathcal{V}_\mathcal{H}$ le plus générique possible, $\mathcal{N}$ supprime la position absolue représentée par la latitude et la longitude ainsi que le temps absolu représenté par l'horodatage. Le *"transformer model"* s'appuie sur la différence temporelle et la différence de distance avec le message précédent pour construire la représentation de la trajectoire. De plus, la seconde de la journée $\mathcal{S}_\mathcal{D}$ est ajoutée pour renforcer la détection de motifs temporels. Une normalisation cyclique $\mathcal{N}_\mathcal{C}$ (éq. 1) est appliquée sur les champs cycliques tels que la $\mathcal{S}_\mathcal{D}$, une normalisation linéaire $\mathcal{N}_\mathcal{L}$ (éq. 2) est appliquée sur les valeurs limites telles que la vitesse $\mathcal{V}$. Pour le Δt , ΔD_V et ΔD_H, un logarithme est appliqué (éq. 3).

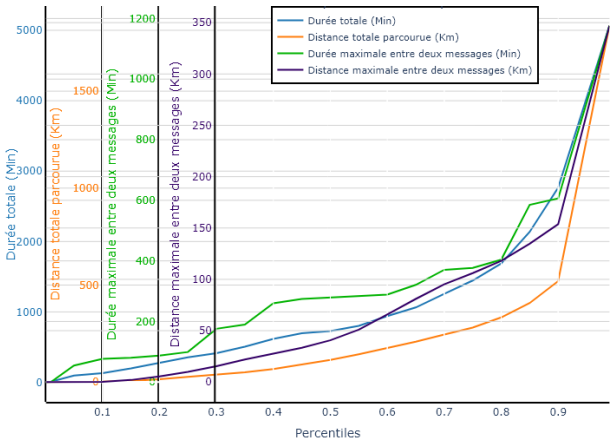


FIGURE 5 – Statistique des échantillons

4.2.2 Position la plus récente

Le vecteur $\mathcal{N}(\mathcal{V}_\mathcal{P})$ est composé de 11 valeurs permettant au modèle d'avoir une précision maximale sur la position du navire lors de la déconnexion. $\mathcal{N}$ décompose la latitude et la longitude en Degré-Minute-Seconde puis les normalise cycliquement pour conserver une continuité (éq. 4), par exemple lorsqu'un navire passe de la longitude 180 à -180. À noter que le degré de la latitude est encodé linéairement puisque un navire ne peut pas passer de la latitude -90 à 90.

$$\mathcal{N}_\mathcal{C} = [\sin \frac{2 * \pi * (x - \min x)}{\max x - \min x}, \cos \frac{2 * \pi * (x - \min x)}{\max x - \min x}] \tag{1}$$

$$\mathcal{N}_\mathcal{L} = \frac{(x - \min x)}{\max x - \min x} \tag{2}$$

$$\mathcal{N}(\mathcal{V}_\mathcal{H}) = [log(\Delta t), log(\Delta D_V), log(\Delta D_H), \mathcal{N}_\mathcal{C}(\mathcal{S}_\mathcal{D}), \mathcal{N}_\mathcal{L}(\mathcal{V})] \tag{3}$$

$$\mathcal{N}(\mathcal{V}_\mathcal{P}) = [\mathcal{N}_\mathcal{L}(lat_{Deg}), \mathcal{N}_\mathcal{C}(lat_{Min}), \mathcal{N}_\mathcal{C}(lat_{Sec}), \mathcal{N}_\mathcal{C}(lon_{Deg}), \mathcal{N}_\mathcal{C}(lon_{Min}), \mathcal{N}_\mathcal{C}(lon_{Sec}), ...] \tag{4}$$

4.3 Détection de déconnexions AIS volontaires

Notre modèle prédit si la réception d'un message AIS du navire est attendue ou non, dans les 10 prochaines minutes. Pour détecter les situations suspectes, nous devons comparer les sorties attendues et les prédictions du modèle, avec deux situations identifiées :

1. Les trajectoires où le navire émet un message AIS dans les 10 minutes sont utiles pour entraîner le modèle mais pas pour la prédiction. En effet, si le modèle prédit qu'un message est attendu, alors le modèle est conforme et si le modèle prédit qu'aucun message n'est attendu, alors une erreur (certaine) du modèle est présente.

2. Les trajectoires où le navire n'émet pas de message AIS dans les 10 minutes sont intéressantes pour la prédiction. En effet, si le modèle prédit qu'aucun message n'est attendu alors le modèle est conforme et la déconnexion AIS est soit involontaire (e.g. panne du transpondeur), soit légale (e.g., navire au mouillage). Par contre, si le modèle prédit qu'un message est attendu alors on a soit une erreur du modèle, soit une déconnexion AIS illégale. Ce sont ces cas-là qui sont les plus pertinents pour la garde-côtière, qui peuvent alors procéder à une analyse détaillée de confirmation ou infirmation. Dans ce dernier cas, l'historique des positions du navire est rapporté.

Dans tous les cas, il est crucial de disposer d'un modèle présentant une très haute précision afin de ne retourner que des trajectoires suspectes intéressantes aux garde-côtes.

5 Résultats et discussion

5.1 Entraînement du modèle

Une chaîne d'intégration continue a été mise en place entre le serveur de pré-traitement, celui d'entraînement et celui de prédictions en utilisant gitlab-ci pour l'orchestration et DVC [8] pour la mise en version des jeux de données. L'infrastructure expérimentale de recherche eX3 [9] nous fournit un environnement de calcul haute performance. L'infrastructure inclut un système DGX-2 [10] composé de 16 cartes graphiques NVIDIA Tesla V100. Pour notre entraînement, une seule de ces cartes nous permet d'entraîner le modèle à une vitesse de une époque toutes les 7s, chaque époque étant composée de 625 pas. Le modèle atteint sa précision maximale après 100 époques mais, afin d'assurer au modèle une convergence garantie, nous l'entraînons sur 1000 époques. Cela amène le temps d'entraînement à 7100s ±150s (soit environ 2h). Aucun sur-apprentissage n'est observé et cela est particulièrement important dans le contexte d'une mise en production avec un apprentissage continu. En effet, l'objectif est d'utiliser les données collectées au jour le jour pour améliorer la précision du modèle.

5.2 Résultats

Les performances obtenues par le modèle pour la prédiction de la réception d'un message dans les dix prochaines minutes sont excellentes. La figure 6 permet d'observer l'évolution de la précision jusqu'à ce que celle-ci atteigne 99% après les 100 premières époques. Pour une meilleure évaluation, le modèle a été entraîné 10 fois sur 1000 époques avec à chaque fois une initialisation aléatoire des paramètres. La figure 7 décrit les précisions finales obtenues sur le jeu d'entraînement, de validations et de test. La précision sur le jeu de test est la plus importante, et nous pouvons

8. Data Version Control
9. https://www.ex3.simula.no/
10. https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/dgx-2/dgx-2-print-datasheet-738070-nvidia-a4-web-uk.pdf

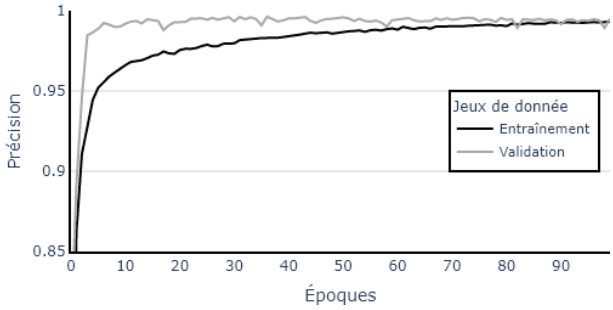


FIGURE 6 – Précision du modèle sur les 100 premières époques d'entraînement

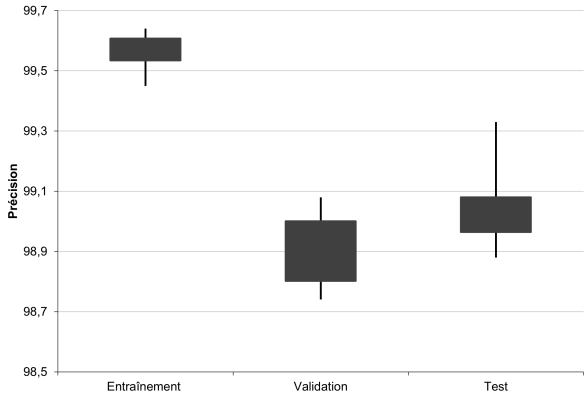


FIGURE 7 – Statistiques de précision sur 10 exécutions

observer une précision de 99% en moyenne avec un maximum atteint de 99.3%. Il est important de rappeler ici que la précision du modèle correspond à la prédiction de réception d'un message dans les 10 minutes et non pas à la prédiction de déconnexion AIS volontaires. Il n'existe pas, à notre connaissance, de jeux de données permettant de tester notre modèle précisément sur cette tâche. Par contre, l'analyse de la matrice de confusion (Table 2) confirme les bonnes performances du modèle. En effet, l'écart entre les faux négatifs et les faux positifs est significatif.

— **Faux négatif (F. nég.)** : le modèle prédit la non-réception d'un message et pourtant un message est reçu ;
— **Faux positif (F. pos.)** : le modèle prédit la réception d'un message mais aucun message n'est reçu.

TABLE 2 – Matrice de confusion, jeu de données de test - Doit-on recevoir un message du navire dans les 10 minutes ?

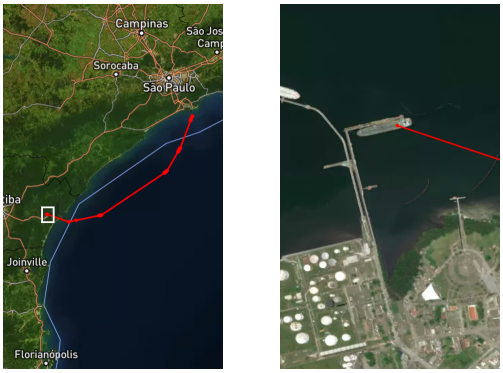
Prédiction \ Annotation	Oui	Non
Oui	4887	104 (F. pos.)
Non	1 (F. nég.)	4992

Sur le jeu de test, un seul faux négatif est détecté. Cette erreur de classification ne peut provenir que d'une erreur du modèle. Par contre, les 104 faux positifs détectés peuvent provenir d'une erreur du modèle mais aussi d'une coupure

anormale du signal. Cet écart consolide les résultats.

5.3 Détection des déconnexions AIS suspectes sur le jeu de données de tests

Sur la courte période de tests dont nous disposons (1 semaine), nous avons extrait tous les cas où la connexion avec un navire est perdue pendant plus de 2h, cela représente 160 574 cas. Le seuil de 2h a été choisi pour limiter le nombre d'exemples à tester. De plus, 2h de coupure sont suffisantes pour effectuer des infractions mais restent un évènement fréquent dans le contexte d'une capture par satellite. Sur ces 160 574 cas, le modèle a permis de classifier 12 navires qui aurait dû envoyer un message dans les 10 premières minutes de la déconnexion. Nous pensons que ce filtrage est une aide précieuse pour les garde-côtes qui peuvent concentrer leur attention sur ces cas suspects. Parmi ces 12 déconnexions suspectes, 10 ont eu lieu après avoir jeté l'ancre dans de petits ports, non répertoriés dans le fichier que nous utilisons pour le filtrage des ports. La figure 8 montre une une de ces dix trajectoires qui sont détectées par notre modèle, mais non illégales. Par contre, les deux autres trajectoires sont suspectes, comme indiquées dans la figure 9. Les deux trajectoires entrent dans une zone économique exclusive avant que l'on perde leur trace. Bien sûr, rien ne nous permet d'affirmer avec certitude que ces déconnexions sont volontaires mais elles nécessitent de lancer des investigations approfondies de la part de la garde-côtière.

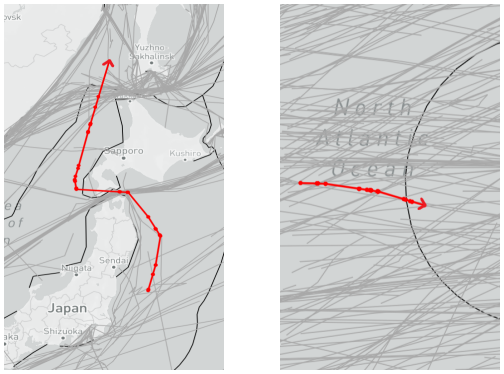


(a) (b)

FIGURE 8 – Un exemple de déconnexion volontaire détectée mais non illégale. Image (b) est le zoom de l'image (a)

5.4 Discussion

Les déconnexions AIS volontaires (illégales) sont des évènements très rares et la quantité de données utilisée pour le test permet de confirmer que le modèle est bien capable de prédire si un navire doit recevoir un message dans les 10 prochaines minutes. Cependant, à ce stade de nos travaux, il ne nous est pas encore possible de fournir un score de précision pour la détection des déconnexions anormales. La matrice de confusion donne une bonne idée de la validité du modèle mais il faut tester le modèle sur un jeu de données plus grand (c'est-à-dire sur une plage de temps plus



(a) (b)

FIGURE 9 – Deux exemples de déconnexion suspecte d'après notre modèle

longue) pour apprendre des motifs de déconnexion AIS. Pour l'instant, le modèle n'est entraîné que sur des données entre avril et juin 2020, c'est-à-dire des données du printemps, ce qui peut créer un biais par rapport à des données d'autres saisons. Nous travaillons à intégrer une année complète de données dans l'expérience. De plus, le modèle ne détecte que les déconnexions AIS qui ont lieu au moment où un des satellites survole la zone. Cela rend le modèle très efficace près des pôles mais moins à l'équateur car les satellites couvrent moins souvent cette zone. Cette faiblesse inhérente à notre jeu de données doit faire l'objet d'une attention particulière.

6 Conclusion et perspectives

Les résultats sur la prédiction de réception de messages sont très encourageants sur nos données de test (99% de précision), ce qui rend cette approche par apprentissage auto-supervisé particulièrement intéressante pour détecter les déconnections AIS volontaires en temps réel. La mise à disposition prochaine d'un jeu de données sur une année entière nous permettra d'expérimenter notre modèle sur d'autres périodes de l'année, ce qui réduira le biais potentiel lié à l'utilisation de données du printemps uniquement. Par ailleurs, l'apprentissage de motifs de déconnexions AIS telles que le franchissement de frontière d'une zone d'exclusion économique sera possible. En parallèle, nous visons également l'entraînement de modèles auto-supervisés pour d'autres tâches de détection détections illégales. En particulier, nous nous intéressons aux rendez-vous en mer pour le transbordement de cargaisons illégales et à l'identification de navires en mer.

Remerciements

Les résultats présentés dans cet article ont été financés par le *"Research Council of Norway" (RCN)* dans le cadre du projet *"AI-driven testing of false data injection attacks against transport infrastructure (TSAR)"* [#287893]. En outre, les résultats ont été obtenus grâce à l'infrastructure expérimentale eX3 (exploration du calcul exascale), qui est également financée par le RCN [# 270053]. Les données S-AIS utili-

sées ont été généreusement mis à disposition par l'administration de la garde-côte Norvégienne.

Références

[1] Bryan Auslander, Kalyan Moy Gupta, and David William Aha. Maritime Threat Detection Using Probabilistic Graphical Models. In *Proc. of the 25th Int. FLAIRS Conf.*, May 2012.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[3] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. In Jian Su, Xavier Carreras, and Kevin Duh, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 551–561. The Association for Computational Linguistics, 2016.

[4] Dominik Filipiak, Milena Stróżyna, Krzysztof Wecel, and Witold Abramowicz. Big data for anomaly detection in maritime surveillance : spatial AIS data analysis for tankers. *Scientific Journal of Polish Naval Academy*, 215(4) :5–28, 2018.

[5] Natalie Fridman, Doron Amir, Yinon Douchan, and Noa Agmon. Satellite Detection of Moving Vessels in Marine Environments. In *Proc. of the AAAI Conf.*, volume 33, pages 9452–9459, July 2019.

[6] Kwang-Il Kim and Keon Myung Lee. Deep Learning-Based Caution Area Traffic Prediction with Automatic Identification System Sensor Data. *Sensors*, 18(9) :3172, September 2018.

[7] Lacey Malarky and Beth Lowell. Avoiding detection : Global case studies of possible AIS avoidance. Technical report, OCEANA, Oct. 2020.

[8] Fabio Mazzarella, Michele Vespe, Dario Tarchi, Giuseppe Aulicino, and Antonio Vollero. AIS reception characterisation for AIS on/off anomaly detection. In *19th Int. Conf. on Information Fusion (FUSION 2016), Heidelberg, Germany, July 5-8, 2016. IEEE 2016*, 2016.

[9] Duong Nguyen, Rodolphe Vadaine, Guillaume Hajduch, René Garello, and Ronan Fablet. A Multitask Deep Learning Architecture for Maritime Surveillance using AIS Data Streams. *IEEE 5th Int. Conf. on Data Science and Advanced Analytics (DSAA)*, pages 331–340, October 2018.

[10] Duong Nguyen, Rodolphe Vadaine, Guillaume Hajduch, René Garello, and Ronan Fablet. GeoTrackNet-A Maritime Anomaly Detector using Probabilistic Neural Network Representation of AIS Tracks and A Contrario Detection. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[11] Jaeyoon Park, Jungsam Lee, Katherine Seto, Timothy Hochberg, Brian A. Wong, Nathan A. Miller, Kenji Takasaki, Hiroshi Kubota, Yoshioki Oozeki, Sejal Doshi, Maya Midzik, Quentin Hanich, Brian Sullivan, Paul Woods, and David A. Kroodsma. Illuminating dark fishing fleets in north korea. 6(30) :eabb1197. Publisher : American Association for the Advancement of Science Section : Research Article.

[12] Lokukaluge P. Perera, Paulo Oliveira, and C. Guedes Soares. Maritime Traffic Monitoring Based on Vessel Detection, Tracking, State Estimation, and Trajectory Prediction. *IEEE Transactions on Intelligent Transportation Systems*, 13(3) :1188–1200, September 2012.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *NeurIPS 2017*, pages 5998–6008, 2017.

[14] Henri Weimerskirch, Julien Collet, Alexandre Corbeau, Adrien Pajot, Floran Hoarau, Cédric Marteau, Dominique Filippi, and Samantha C. Patrick. Ocean sentinel albatrosses locate illegal vessels and provide the first estimate of the extent of nondeclared fishing. 117(6) :3006–3014. Publisher : National Academy of Sciences Section : Biological Sciences.

Session 5 – L'IA pour l'analyse de documents textuels

Improving Patent Mining and Relevance Classification using Transformers

Théo Ding[1,2], Walter Vermeiren[2], Sylvie Ranwez[1], Binbin Xu[1]

[1] EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Alès
[2] Total Research and Technology, Seneffe, Belgium

theo.ding@mines-ales.org, binbin.xu@mines-ales.fr

Résumé

L'analyse et l'exploitation de brevets sont des processus longs et coûteux pour les entreprises, mais néanmoins indispensables si elles veulent rester compétitives. Pour faire face au nombre croissant de brevets à analyser, il est possible de procéder à un filtrage préliminaire automatique, afin de n'en sélectionner qu'un nombre limité, qui est par la suite analysé par des experts. Cet article présente les résultats obtenus sur un cas d'étude industriel. Des modèles d'apprentissage profond pré-entraînés, utilisés en Traitement Automatique de la Langue, ont été fine-tunés et réentraînés pour améliorer la classification de brevets. La solution que nous proposons combine plusieurs traitements de l'état de l'art pour atteindre notre objectif : diminuer la charge de travail des experts tout en préservant les mesures de rappel et de précision.

Mots-clés

Transformeurs, paramétrage, classification de brevets, traitement automatique de la langue naturelle, modèles de langage.

Abstract

Patent analysis and mining are time-consuming and costly processes for companies, but nevertheless essential if they are willing to remain competitive. To face the overload induced by numerous patents, the idea is to automatically filter them, bringing only few to read to experts. This paper reports a successful application of fine-tuning and retraining on pre-trained deep Natural Language Processing models on patent classification. The solution that we propose combines several state-of-the-art treatments to achieve our goal: decrease the workload while preserving recall and precision metrics.

Keywords

Transformers, fine-tuning, patent classification, natural language processing, language models.

1 Introduction

Patents play one of the key roles in business intelligence. Mining state-of-the-art patents can help companies explore new or different ways of innovation to remain competitive. The conventional patent mining activity requires domain experts to manually evaluate all collected patents. A patent is considered relevant when matching the experts' current interests regarding a given subject. When the number of patents increases exponentially, the manual annotation will be too time-consuming and too costly. Tools from Natural Language Processing (NLP) have been used to filter most likely irrelevant patents for years. Unfortunately, the classification rate is quite low in many real cases. In 2018, the biggest change in NLP was the introduction of language models based on Transformer [1]. BERT (Bidirectional Encoder Representations from Transformers) [1] obtained state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5%. It is soon optimized by RoBERTa [2], which beat again the state-of-the-art results. These two have been once again improved in DeBERTa (Decoding-enhanced BERT with disentangled attention) [3]. GPT-2, another transformer model, can not only generate coherent paragraphs of text, but also achieve state-of-the-art performances on many language modeling benchmarks [4]. To better handle longer documents, Longformer was introduced in 2020 [5].

Many successful text classification applications have been reported, including classification of complex documents like patents [6, 7, 8, 9, 10, 11]. However, patent relevance classification remains a challenging problem. The "relevance" itself is context dependent. One will need to retrain or fine-tune the pre-trained models for specific applications.

The main aim of the project is to study the feasibility of building an automatic patent relevance evaluation tool to pre-classify collected patents and reduce the annotation load from experts for further evaluations. All patents classified as relevant by this tool, whether they are truly relevant or not, will then be manually appraised by experts. False positives are eventually rejected during the experts' manual inspection. One critical aspect in patent relevance classification is : misclassifying as few relevant patents as possible is vital as each one holds a potential threat against the business, even though this leads to a slightly larger number of patents left to be manually labeled.

In this work, we report a successful application of fine-tuning on pre-trained deep NLP models on patent relevance classification, by benchmarking on state-of-the-art transfor-

mer models in order to find the most appropriate model(s) for this application.

2 Data and Models

2.1 Data

As aforementioned, the data that we consider are patents in English and, more precisely, their titles combined with the claims. Indeed, claims are the part containing key information about patents and highlight their essential elements [12]. Titles contain more general information, so they could also be used as features.

FIGURE 1 – One claim from patent "Processing a 4D Seismic Signal Based on Noise Model" WO2019053484 [13]

```
1. A computer-implemented method
for processing a 4D seismic signal
relative to a subsoil, the subsoil
including a zone subject to
extraction and/or injection, the
method comprising: providing the 4D
seismic signal; identifying a part of
the 4D seismic signal corresponding
to a zone of the subsoil distinct
from the zone subject to extraction
and/or injection; determining a noise
model of the 4D seismic signal based
on the identified part of the 4D
seismic signal; and processing the
4D seismic signal based on the noise
model.
```

Every year, thousands of patents for one topic only are being evaluated by Total SE. Amongst these patents, in general, only 10% are considered relevant. As imbalanced classes are problematic for machine learning model training, we used an oversampling technique to solve this issue : by randomly duplicating relevant patents until their number is equal to the one of the other class.

2.1.1 Dataset topics

All the datasets in this work have been manually evaluated by experts. Two confidential topics have been studied here. The first topic (Topic A) has been followed by Total SE for more than 20 years, so relevant patents can be easily identified by the experts. This dataset is thus more homogeneous in terms of text diversity. It includes data from two successive years (named A1 and A2) as well as data from the first three quarters of a third year (named AT1, AT2 and AT3). The second topic (Topic B) has been tracked for only a few years and is a more heterogeneous group than topic A. It is composed of three sequential datasets (named B1, B2 and B3) over a period of one and a half years.
Topic A and topic B were only used to evaluate the transformers' performances on different topics. Better results with topic A than with topic B are to be expected due to the nature of the datasets themselves and do not represent the point of interest of this study. Topic B has been studied to know whether newer topics could also benefit from the classification automation.
Any reference to a combination of several datasets of the same topic will be presented by concatenating the corresponding dataset names if no confusion may occur.
In the following table, size refers to the number of patents in each dataset.

TABLE 1 – Approximate number of patents for each dataset.

Topic A	Size	Topic B	Size
A1	3,500	B1	6,500
A2	4,500	B2	1,500
AT1	1,500	B3	3,500
AT2	1,500		
AT3	1,500		

2.1.2 Training and testing strategies

The classification strategy implies to rely partly or entirely on history datasets for model training to predict the relevance of new datasets. For the datasets in Table 1, this led to the following train / test configurations :

TABLE 2 – Training and testing dataset configurations.

Training datasets	Testing datasets
A1	A2
A12	AT123
AT12	AT3
A12T12	AT3
B1	B2
B1	B3
B12	B3

By default, for every model, the maximum sequence length is set to 512 tokens (except for Longformer where it has been set to 1,024 tokens), the batch size to 64, the learning rate to $2e-5$ with a linear scheduler with no warmup steps. Each model has been trained for 20 epochs.
Several pre-processings have been performed on these datasets, although not all of them have been used for each experiment. The datasets have firstly been re-sampled to avoid the imbalanced-class problem. For a given patent, its title and claims have been concatenated into a single string. The maximum sequence length is set to 512 tokens (about 450 words). However, when considering all the claims concatenated for one patent in our dataset, the median sequence length is closer to 1,000 tokens. In consequence, each patent's group of claims longer than 512 tokens have been truncated into chunks up to a maximum of 512 tokens. Then, each chunk has been processed by the transformers as one separate patent with the same label as the other chunks. For a given patent, the mean and the median of the probabilities of each chunk, returned by the transformers, have been calculated, and the resulting probability has been assigned to that patent.

To indicate which strategy has been used for which experiment in Tables 4, 5, 6 and 3, the following naming rules have been used :

— Training and testing datasets are separated by an underscore.
— Suffix **R** : dataset has been re-sampled, its name is followed by a suffix **R**.
— Suffix **T** : titles have been concatenated to claims.
— Suffix **EX** : dataset has been extended – longer patents truncated into shorter chunks of 512 tokens.

If the training dataset has the title and/or the extended modifications, the test dataset will have the same tweaks.
For example, **A1-TR-EX_A2-T-EX** indicates that the training dataset A1 (first dataset of topic A) has the titles of the patents concatenated to the claims and has also been re-sampled and extended. The testing dataset, A2 (second dataset of topic A), has the title tweak as well as its claims transformed into chunks.
In order to maximize the classification rate and evaluate the relevance of different components of patents, many combinations have been tested : experiments with and without titles, re-sampling or extending changes.

2.2 Optimizations, Metric and Models

To better classify the patents, the whole patent should technically be taken into account. However, the maximal patent text lengths can go beyond the transformer models' capacity. For most transformer models used in this work, the maximum token length is 512. GPT-2 supports 1,024 tokens while Longformer can accept up to 4,096 tokens.
Increasing the maximum token length requires more GPU VRAM, which amount is obviously limited. The first and simplest solution is to decrease the batch size. However, this reduction induces learning issues as smaller batches mean causing gradient to be more prone to noise. In heterogeneous datasets, patents are less similar from one batch to another, noise effect will be more important. Gradient accumulation has been applied in order to overcome this limitation. This technique allows accumulating the gradients over a given number of batches. After this, the model is being updated before resetting the gradient.
A possible solution is gradient checkpointing, that loads part of the activation network and recomputes the missing activation before operating the back-propagation, rather than loading the full graph. It trades off time for memory. However, as communications inter-GPUs take considerable time, this technique is less interesting for multi-GPU configurations.
Another way to limit the GPU VRAM consumption is to reduce the vocabulary size and token interaction file from tokenizers. The idea is to build the vocabulary only from the current dataset. For all the benchmarked models except DeBERTa, a unique list of all the tokens and interactions between those tokens encountered in each training and test set couples are kept. This helped to reduce by around 50% the vocabulary size and by 20% the interaction file for all the considered models for topic A, and by 60% and 25% for the same files for topic B. Reducing the vocabulary directly impacts the size of the input tensor fed to the models and thereby the amount of VRAM needed, while not increasing the execution time.

2.3 Introduction to specific metric

Patent relevance evaluation is not a simple text classification problem. Reducing the workload from experts' manual annotations is also crucial. Common metrics only, like recall, precision or even $F1$ score, were not enough. A new metric is proposed here, taking into account the percentage of patents that will not be manually processed, and the proportion of relevant patents properly classified :

$$\mathcal{M} = (4 * \text{recall} + \text{precision} + \text{patents left}) \,/\, 6 \quad (1)$$

The *patents left* changes according to the number of irrelevant patents incorrectly classified as relevant, increasing when this number decreases, and vice-versa. It represents the amount of unnecessary workload that needs to be manually done. The coefficient 4 multiplying the recall is to rank models according to their ability to answer the initial problem : high recall, low number of patents left to be manually handled. Its value has been set to follow the experts' view on the models' performances : by asking the experts which model would be preferred over which other according to their recall, precision and patents left scores.
As a reminder, in the business intelligence context, having a few more patents to be manually labeled (so more false positives) than missing truly relevant patents (false negatives) will always be preferred.

2.4 Models

The transformer models used in our study are state-of-the-art models : BERT, GPT2, RoBERTa, DeBERTa and Longformer.
Since BERT [1] is the first deeply bidirectional, unsupervised language representation (Transformer architecture), it is used here as a baseline model for all the different modifications in the experiments.
GPT2 [4] is the first large scale generative model capable of generating realistic texts. Its token prediction ability can also be used for classification tasks when using specific tokens as class labels [14].
RoBERTa [2] is known to perform better in general compared to BERT, partly thanks to its rigorous training with ten times more data than BERT.
DeBERTa was part of the combination of models giving the second-best results on GLUE and SuperGLUE benchmarks and, as both gather a wide range of general Natural Language Understanding tasks, implementing this model was promising.
Finally, as most of these models can only work with sequences of up to 512 tokens (1,024 for GPT2), and as our dataset has a median length of around 1,000 tokens for the considered patents, trying to implement Longformer with its maximum sequence length of 4,096 was obviously an interesting track. Moreover, Longformer is known to perform better than RoBERTa on tasks involving long documents.

All of the transformers have been adapted from Hugging Face [15].

3 Results

To evaluate the effects of each pre-processing on the datasets as well as the choices of the datasets themselves, 12 experiments have been performed. Furthermore, more combinations, presented hereinbelow, have been tested using BERT only.

3.1 Extensive BERT comparison

3.1.1 BERT results table introduction

To have a baseline for as many experiments as possible and to compare them to one another, running BERT in its base uncased version appeared as the simplest and most obvious solution. The experiment naming rules are those aforementioned in Section 2.1.2. The models' name column has been replaced by the specific features, if any, as a unique model only has been used.

Here, *NV* stands for *No Vocabulary changes*; *Mem* means that the memory has been more thoroughly monitored; *Batch-8* is for the batch size, here only, the batch size has been decreased to 8 and the gradient accumulation step size has been increased to 8 instead of 1 ;$\mathrm{BERT}_{\bar{\mathrm{X}}}$ and $\mathrm{BERT}_{\tilde{\mathrm{X}}}$ respectively indicate that the mean or the median of probabilities, of all the chunks returned by BERT, for a given patent, is taken into account to classify it; *1-GPU* means that the model is run on single-GPU mode, without any parallelization (instead of 4 for the other tests).

All results going forward will, by default, refer to score $\mathcal{M}$. For readability, each training-test combination has been numbered.

3.1.2 BERT results interpretation

To understand these tables, here is an example with the first line of experiment (2). The training dataset is the first one of topic A (A1), which has been re-sampled, the testing dataset is the second dataset of topic A (A2). Line 1 has BERT's vocabulary file not altered, and the memory has been thoroughly monitored during the whole run. The model's recall score is 0.9176 for a precision of 0.6131. The percentage of patents left to be classified manually is 15.1% of the total number of patents. This means that, if we decided to take into account the decision of the model for each patent, 15.1% of all the patents would have been considered relevant by the model and manually classified, the rest would not have been further processed. The $\mathcal{M}$ score for this model is 0.8709 and its $F1$ score is 0.7351. The maximal amount of VRAM consumed is 18.8GB and the total execution time is 82 minutes.

Experiment (1) and line 2 of experiment (2) show that re-sampling the data yields better results (sequentially, Score $\mathcal{M}$=0.8252 against 0.8546, 2.94% better).

Experiments (1) and (4) show that concatenating the titles to the claims yields better results (sequentially, Score $\mathcal{M}$=0.8252 against 0.8436, 1.84% better).

Experiment (1) and line 2 of experiment (5) show that combining re-sampling with concatenation of titles yield better results (sequentially 0.8252 against 0.8780, 5.28% better). Similar results can be observed with topic B.

Lines 1 and 2 of experiment (5) show that using the original vocabulary from the tokenizer yields better results (sequentially 0.8873 against 0.8780, 0.93% better). The execution

TABLE 3 – Extensive BERT experiments. From left to right, the columns are: Specific feature, Recall, Precision, Percentage, Score $\mathcal{M}$, Score $F1$, Maximal Memory usage in GB and Total computing time in minutes.

Feature	R	P	%	$\mathcal{M}$	F1	GB	Time
(1) A1_A2							
	0.8330	0.6618	12.7	0.8252	0.7376	16	60
(2) A1-R_A2							
$\mathrm{NV}_{\mathrm{Mem}}$	0.9176	0.6131	15.1	0.8709	0.7351	18.8	82
	0.9039	0.5783	15.7	0.8546	0.7054	16	79
Mem	0.9039	0.5783	15.7	0.8546	0.7054	18.8	80
Batch-8	0.9497	0.4814	19.9	0.8629	0.6390	18.8	153
(3) A1-R-EX_A2-EX							
$\mathrm{BERT}_{\bar{\mathrm{X}}}$	0.9542	0.4691	20.5	0.8629	0.6290	16.5	106
$\mathrm{BERT}_{\tilde{\mathrm{X}}}$	0.9542	0.4691	20.5	0.8629	0.6290	16.5	106
(4) A1-T_A2-T							
	0.8673	0.6413	13.6	0.8436	0.7374	NA	58
(5) A1-TR_A2-T							
NV	0.9497	0.5912	16.2	0.8873	0.7287	17.6	79
	0.9474	0.5550	17.2	0.8780	0.6999	34	78
1-GPU	0.9451	0.5485	17.3	0.8751	0.6941	25.9	155
(6) A1-TR-EX_A2-T-EX							
$\mathrm{BERT}_{\bar{\mathrm{X}}}$	0.8970	0.6164	14.6	0.8580	0.7307	16.6	110
$\mathrm{BERT}_{\tilde{\mathrm{X}}}$	0.8970	0.6164	14.6	0.8580	0.7307	16.6	110
(7) A12-TR_AT123-T							
	0.9669	0.1535	44.3	0.7743	0.2650	NA	142
(8) A12T12-TR_AT3-T							
	0.8571	0.2426	18.2	0.7555	0.3782	3.2	173
(9) AT12-TR_AT3-T							
	0.8312	0.2092	20.5	0.7287	0.3342	NA	60
(10) B1-TR_B2-T							
	0.6739	0.6392	17.7	0.7118	0.6561	16.5	99
(11) B1-R_B3							
	0.7029	0.4462	23.1	0.6883	0.5459	16.5	105
(12) B1-TR_B3-T							
	0.7695	0.4368	25.8	0.7282	0.5572	16.5	108
(13) B12-TR_B3-T							
	0.7200	0.5362	19.7	0.7208	0.6146	21.8	128
(14) B12-TR-EX_B3-T-EX							
$\mathrm{BERT}_{\bar{\mathrm{X}}}$	0.8895	0.3439	37.9	0.7756	0.4960	16.6	186
$\mathrm{BERT}_{\tilde{\mathrm{X}}}$	0.8895	0.3421	38.1	0.7749	0.4942	16.6	186

time is similar.
Lines 2 and 3 of experiment (2) allow to check the influence of noise on the results as all randomness may not have been properly seeded. Here, the identical results confirmed that noise was not an issue.
Lines 2 and 4 of experiment (2) show that decreasing the batch size to 8 while using a gradient accumulation with a step size of 8 yields better results (sequentially 0.8546 against 0.8629, 0.83% better).
Experiments (3), (6) and (14) show that $\text{BERT}_{\bar{\text{X}}}$ and $\text{BERT}_{\tilde{\text{X}}}$ yield almost identical results. However, experiments (2) and (3), experiments (5) and (6), and experiments (13) and (14) show that extending datasets does not systematically yield better results (respectively 8.3% better, 2.0% worse, 5.41% better with $\text{BERT}_{\bar{\text{X}}}$ or 5.48% better with $\text{BERT}_{\tilde{\text{X}}}$).
Lines 12 and 3 of experiment (5) show that training and testing on only 1 GPU compared to 4 does not change significantly the results (0.29%), but the computing time has been drastically increased, here by two folds.
Experiments (8) and (9) show that using A12 along with AT12, for training, only improved the results by 2.7% but increased the total computing time by three folds (60 minutes when training on AT12, against 173). Depending on the retraining frequency, it might be more interesting to only use the most recent data as train dataset.
Experiments (12) and (13) show that adding B2 to B1 as training dataset and predicting on B3 did not yield way better results than using only B1 as training dataset (sequentially 0.7282 against 0.7208, so 0.74% better).
Similar results can be observed for topic B and its dataset (experiments (10) through (14)). The rest of the results' interpretation will be left to the reader.
The memory usage is the maximal amount used in the whole run. Values are just indicative and may be erroneous ; models may work properly with lower amounts of memory as explained in 3.3.

3.1.3 BERT results conclusion

To conclude on this table, using the titles and re-sampling tweaks consistently led to better results (5.28% better for topic A). For two out of three cases, extending the datasets have yielded better results, even up to 5% for topic B. This may be an interesting track to follow for future studies. Moreover, as memory scarcity is an issue, decreasing the batch size, as well as limiting the vocabulary files, are promising trails that seem to not or only slightly decrease the performances. Besides, using the original vocabulary files yields better results and noise is not an issue for experiment reproducibility when seeds have been properly set.

3.2 Full results

3.2.1 Full results table introduction

Thanks to the powerful computing resources made available by Total SE, it has been possible to massively train different models on different dataset combinations with different strategies.
Hereunder, the detailed metrics for all the experiments are

TABLE 4 – Experiments 1 through 5. (Results ranked by descending metric $\mathcal{M}$ order).

MODEL	R	P	%	$\mathcal{M}$	F1
(1) A1-R_A2					
MEAN	0.9359	0.5603	16.8	**0.8717**	0.7009
DeBERTa	0.9428	0.5372	17.7	0.8711	0.6844
MEDIAN	0.9359	0.5542	17.0	0.8703	0.6962
BERT	0.9039	0.5783	15.7	0.8546	**0.7054**
Longformer	**0.9565**	0.4274	22.5	0.8541	0.5908
GPT2	0.9291	0.4975	18.8	0.8532	0.6480
RoBERTa	0.8787	**0.5818**	**15.2**	0.8388	0.7001
(2) A1-R-EX_A2-EX					
$\text{DeBERTa}_{\bar{\text{X}}}$	0.9519	0.4929	19.4	**0.8670**	0.6495
$\text{DeBERTa}_{\tilde{\text{X}}}$	0.9519	0.4917	19.5	0.8667	0.6485
$\text{BERT}_{\bar{\text{X}}}$	**0.9542**	0.4691	20.5	0.8629	0.6290
$\text{BERT}_{\tilde{\text{X}}}$	**0.9542**	0.4691	20.5	0.8629	0.6290
MEAN	0.8856	0.6627	13.5	0.8600	0.7581
MEDIAN	0.8833	**0.6644**	**13.4**	0.8588	**0.7583**
$\text{GPT2}_{\bar{\text{X}}}$	0.9176	0.5434	17.0	0.8560	0.6826
$\text{GPT2}_{\tilde{\text{X}}}$	0.9176	0.5434	17.0	0.8560	0.6826
$\text{RoBERTa}_{\bar{\text{X}}}$	0.9314	0.4933	19.0	0.8538	0.6450
$\text{RoBERTa}_{\tilde{\text{X}}}$	0.9314	0.4933	19.0	0.8538	0.6450
(3) A1-TR_A2-T					
MEDIAN	0.9451	0.5833	16.3	**0.8826**	0.7214
MEAN	0.9405	0.5855	16.2	0.8801	0.7217
BERT	0.9474	0.5550	17.2	0.8780	0.6999
BERT-L	0.9245	**0.6057**	**15.4**	0.8739	**0.7319**
DeBERTa	0.9451	0.5329	17.8	0.8717	0.6815
RoBERTa-L	**0.9588**	0.4540	21.3	0.8622	0.6162
Longformer	0.9268	0.5407	17.3	0.8614	0.6830
GPT2-M	0.9062	0.5928	**15.4**	0.8592	0.7167
RoBERTa	0.9085	0.5712	16.0	0.8561	0.7014
GPT2	0.9176	0.5161	17.9	0.8500	0.6606
(4) A1-TR-EX_A2-T-EX					
$\text{DeBERTa}_{\bar{\text{X}}}$	**0.9588**	0.5244	18.4	**0.8787**	0.6780
$\text{DeBERTa}_{\tilde{\text{X}}}$	**0.9588**	0.5224	18.5	0.8782	0.6764
$\text{RoBERTa}_{\bar{\text{X}}}$	0.9474	0.5049	18.9	0.8668	0.6587
$\text{RoBERTa}_{\tilde{\text{X}}}$	0.9474	0.5049	18.9	0.8668	0.6587
MEAN	0.8810	0.6912	**12.8**	0.8626	**0.7746**
MEDIAN	0.8787	**0.6919**	**12.8**	0.8612	0.7742
$\text{BERT}_{\bar{\text{X}}}$	0.8970	0.6164	14.6	0.8580	0.7307
$\text{BERT}_{\tilde{\text{X}}}$	0.8970	0.6164	14.6	0.8580	0.7307
$\text{GPT2}_{\bar{\text{X}}}$	0.9039	0.5758	15.8	0.8541	0.7035
$\text{GPT2}_{\tilde{\text{X}}}$	0.9039	0.5750	15.8	0.8539	0.7028
(5) A12T12-TR_AT3-T					
BERT-L	**0.9221**	0.1578	30.1	**0.7654**	0.2694
BERT	0.8571	0.2426	18.2	0.7555	0.3782
DeBERTa	0.8701	0.2030	22.1	0.7512	0.3292
Longformer	0.8571	0.2082	21.2	0.7448	0.3350
RoBERTa	0.8701	0.1791	25.0	0.7424	0.2971
MEDIAN	0.8312	0.2278	18.8	0.7346	0.3575
GPT2	0.8312	0.2169	19.7	0.7312	0.3441
MEAN	0.8182	0.2308	18.3	0.7272	0.3600
RoBERTa-L	0.7792	0.2521	15.9	0.7083	**0.3810**
GPT2-M	0.6883	**0.2611**	**13.6**	0.6523	0.3786

presented in Tables 4, 5 and 6. Only the best results, according to score $\mathcal{M}$ amongst the 20 epochs, have been kept and sorted in the result tables in decreasing order.

In the first row of the table, the *R* stands for Recall, the *P* is Precision, the *%* is the percentage of patents classified as relevant, whether there gold label is relevant, and so left to be manually classified, the $\mathcal{M}$ refers to the proposed score, and the *F1* is the $F1$ score. The next row has the training and testing datasets used and all the dataset modifications that have been applied. Then, each row represents the performances of a model on this training-test combination.

MEAN is the mean of probabilities given by all the models for each patent; MEDIAN is their median. The suffixes $_{\bar{X}}$ or $_{\tilde{X}}$ of certain models follow the same nomenclature as in 3.1.1. Whenever the large version of a transformer was involved, a suffix *L* has been added to the model's name. For GPT2, the *M* stands for the medium-sized version. Finally, the maximum sequence length used has been set to 512 for every model except for Longformer where it was 1,024 tokens.

3.2.2 Full results interpretation

Similarly to the previous table 3, for each experiment, each line corresponds to one model with its values for recall, precision, patents left, score $\mathcal{M}$ and score $F1$.

In 10/12 experiments, DeBERTa is in the top two best models, followed by BERT/BERT large with 8/12. Longformer is in 3/9 top twos, when RoBERTa is in 2/12, and GPT2 reached it only once out of twelve experiments. The MEAN and MEDIAN of all probabilities performed better than individual models respectively only three times and twice out of twelve. Note that these MEAN and MEDIAN came from the best models obtained within the 20 epochs, so they may not be the best combinations that could exist in theory. However, testing all the possible cases is not possible. As an example, there are $2 * (21^8 - 1) \approx 7.6e9$ possible combinations for experiment (9) only.

In 8 experiments out of the 12 ones, MEAN and MEDIAN are amongst the top three models. In 6/12 experiments, the MEAN is better than the MEDIAN of probabilities. However, the mean difference in score $\mathcal{M}$ when the MEAN is above the MEDIAN is only 0.0071, whereas this mean difference is 0.0238 when the MEDIAN is above the MEAN. This shows that, although the MEAN is above the MEDIAN in the same number of experiments, the MEDIAN should be the first choice as voting strategy.

To interpret part of the results in a clearer way, two experiments, designated by their numbers, will be compared to each other. To do so, the mean of score $\mathcal{M}$ of each individual model will be calculated for each couple of experiments compared. In other words, this mean will not include neither MEAN nor MEDIAN. In the form of a listing, the first and the second experiments will be joined by an "&" sign, followed by a coma and two scores, referring sequentially to the first and the second experiments, and joined by an "&". Finally, a colon ": " will lead the interpretations.

Interpretation results:

— (1) & (2), 0.8544 & 0.8599 : extending the datasets

TABLE 5 – Experiments 6 through 10. (Results ranked by descending metric $\mathcal{M}$ order).

MODEL	R	P	%	$\mathcal{M}$	F1
(6) AT12-TR_AT3-T					
BERT	**0.8312**	0.2092	20.5	**0.7287**	0.3342
MEDIAN	0.8052	0.2541	16.3	0.7255	0.3863
GPT2	0.7922	0.2699	15.1	0.7214	0.4026
MEAN	0.7922	0.2629	15.5	0.7196	0.3948
Longformer	0.7662	0.2837	13.9	0.7081	0.4140
RoBERTa	0.7922	0.2089	19.5	0.7039	0.3306
DeBERTa	0.6494	**0.4274**	**7.8**	0.6634	**0.5155**
(7) A12-TR_AT123-T					
MEAN	0.9307	0.2320	28.2	**0.7897**	0.3714
MEDIAN	0.9337	0.2240	29.3	0.7885	0.3613
BERT-L	0.9367	0.2160	30.5	0.7873	0.3510
Longformer	0.9578	0.1797	37.5	0.7839	0.3026
GPT2	0.9307	0.2071	31.6	0.7799	0.3388
BERT	**0.9669**	0.1535	44.3	0.7743	0.2650
RoBERTa-L	0.8735	0.2780	22.1	0.7687	0.4218
DeBERTa	0.8675	**0.2815**	**21.7**	0.7660	**0.4251**
RoBERTa	0.9518	0.1534	43.7	0.7652	0.2642
GPT2-M	0.8404	0.2640	22.4	0.7435	0.4017
(8) B1-TR_B2-T					
DeBERTa	0.7790	0.6324	20.7	**0.7787**	**0.6981**
Longformer	**0.8732**	0.3970	37.0	0.7778	0.5459
MEAN	0.7862	0.5945	22.2	0.7749	0.6771
MEDIAN	0.8007	0.5525	24.4	0.7744	0.6538
RoBERTa	0.7572	0.4988	25.5	0.7333	0.6014
BERT	0.6739	**0.6392**	**17.7**	0.7118	0.6561
GPT2	0.7862	0.3196	41.4	0.6972	0.4545
(9) B12-TR_B3-T					
DeBERTa	0.8476	0.4555	27.3	**0.7829**	0.5925
MEAN	0.8057	0.5146	22.9	0.7710	0.6281
Longformer	**0.9352**	0.2634	52.0	0.7702	0.4111
MEDIAN	0.8114	0.4936	24.1	0.7696	0.6138
RoBERTa	0.8876	0.3117	41.7	0.7625	0.4614
BERT-L	0.7886	0.4272	27.0	0.7378	0.5542
BERT	0.7200	0.5362	19.7	0.7208	0.6146
RoBERTa-L	0.6876	**0.5841**	**17.2**	0.7105	**0.6317**
GPT2-M	0.7562	0.3994	27.7	0.7096	0.5227
GPT2	0.7181	0.4425	23.8	0.6971	0.5476
(10) B12-TR-EX_B3-T-EX					
$\text{BERT}_{\bar{X}}$	**0.8895**	0.3439	37.9	**0.7756**	0.4960
$\text{BERT}_{\tilde{X}}$	**0.8895**	0.3421	38.1	0.7749	0.4942
$\text{DeBERTa}_{\bar{X}}$	0.8343	0.4228	28.9	0.7655	0.5612
$\text{DeBERTa}_{\tilde{X}}$	0.8343	0.4228	28.9	0.7655	0.5612
$\text{GPT2}_{\bar{X}}$	0.7714	0.4540	24.9	0.7340	0.5716
$\text{GPT2}_{\tilde{X}}$	0.7695	0.4524	24.9	0.7323	0.5698
$\text{RoBERTa}_{\bar{X}}$	0.7105	0.5188	20.1	0.7107	0.5997
$\text{RoBERTa}_{\tilde{X}}$	0.7105	0.5188	20.1	0.7107	0.5997
MEDIAN	0.6381	0.6879	13.6	0.6997	0.6621
MEAN	0.6286	**0.7006**	**13.1**	0.6959	**0.6627**

yields better results (0.55%).
- — (1) & (3), 0.8544 & 0.8641 : concatenating the titles to the claims yields better results (0.97%).
- — (3) & (4), 0.8641 & 0.8643 : extending the datasets yields very slightly better results (0.02%).
- — (5) & (6), 0.7314 & 0.7051 : although A1 and A2 are not part of the same year as AT123, using them alongside with AT12 for the training yields better results (2.63%).
- — (5) & (7), 0.7314 & 0.7711 : training on A12 and testing on AT123 yields better results than training on A12T12 and testing on AT3 only (3.97%). Perhaps the patents in A12 are more similar to those in AT12 than in AT3. AT3 has also been published later than AT12 compared to A12.
- — (8) & (12), 0.7398 & 0.7020 : training on B1 and testing on B2 yields better results than training on B1 and testing on B3 (3.78%). Once again, this may be due to the chronological distance being longer between B1 and B3 than between B1 and B2. Topic B being broader, patents considered to be relevant may vary faster with time than with other topics. This can mean that having recent datasets for training may be more critical than having more datasets but less updated.
- — (9) & (12), 0.7364 & 0.7020 : training on B12 and testing on B3 yields better results than training on B1 and testing on B3 (3.44%). The possible interpretation is the same as the previous ones.
- — (11) & (12), 0.6717 & 0.7020 : concatenating the titles to the claims yields better results (3.03%).
- — (9) & (10), 0.7364 & 0.7462 : extending the datasets yields better results (0.98%). Having more data involved in the decision to classify patents, in one or the other class, may yield better results with topic B, as it is broader than topic A.

In extended dataset experiments ((2), (4) and (10)), differences in $\mathcal{M}$ scores between the same models but attributing patents' class with the median or mean of probabilities is minuscule (0.0283% on average).

In experiment (6), the best $F1$ score (with 0.5155) is ranked last according to score $\mathcal{M}$ as its recall is only of 0.6494, whereas the best model's recall is 0.8312. Let us remind that, strategically speaking, it is important to miss as few relevant patents as possible, so a high recall will be preferred over a high precision at equal values. In experiment (7), the same situation occurs.

Scores with topic A are better than those of topic B. This may be due to the fact that patents in this topic are more general, and that topic B has only been tracked for a few years. These data are less refined, thus less representative, which makes the prediction task more challenging. Experiments (3) and (9) had similar configurations. Sequentially, without taking MEAN nor MEDIAN into account, the maximal $\mathcal{M}$ scores are 0.8780 and 0.7829 (9.51%) and the mean of $\mathcal{M}$ scores are 0.8641 and 0.7364 (12.77%).

In 3 out of 4 experiments, BERT large performed better than BERT base (0.895% better on average). In 2/4 experiments, GPT2 performed better than its medium equivalent (2.34%). The same occurred with RoBERTa large being better than its base version (1.91%).

TABLE 6 – Experiments 11 and 12. (Results ranked by descending metric $\mathcal{M}$ order).

MODEL	R	P	%	$\mathcal{M}$	F1
(11) B1-R_B3					
DeBERTa	**0.8133**	0.3509	33.9	**0.7306**	0.4902
MEDIAN	0.7219	0.4572	23.1	0.7032	0.5598
RoBERTa	0.7790	0.3264	35.0	0.7012	0.4601
MEAN	0.7162	0.4619	22.7	0.7008	**0.5616**
BERT	0.7029	0.4462	23.1	0.6883	0.5459
Longformer	0.6381	0.4085	22.9	0.6376	0.4981
GPT2	0.5543	**0.4763**	**17.0**	0.6007	0.5123
(12) B1-TR_B3-T					
DeBERTa	**0.8057**	0.4862	24.3	**0.7641**	0.6065
MEDIAN	0.7543	0.5083	21.7	0.7364	0.6074
MEAN	0.7505	**0.5110**	**21.5**	0.7346	**0.6080**
BERT	0.7695	0.4368	25.8	0.7282	0.5572
Longformer	0.7219	0.4019	26.3	0.6887	0.5163
RoBERTa	0.6838	0.4793	20.9	0.6843	0.5636
GPT2	0.6438	0.4214	22.4	0.6445	0.5094

Though Longformer is capable of taking much longer token sequences, it did not necessarily give better results than the other models. Longformer's large version has also been tested for only one dataset combination by decreasing the maximal sequence length to 512. However, the results were worse than those of the base version, so experiments have not been pursued. Amongst the nine experiments where Longformer has been tested, 6/9 medians of the percentage of left patents of each individual model are lower than the individual score on the same metric that Longformer has. This model may be not decisive enough for our application.

3.2.3 Full results conclusion

Although topic A and B have very different $\mathcal{M}$ scores, the general trend remains the same : concatenating titles to claims, re-sampling the datasets and extending them yield better results. Moreover, datasets chronologically following each other give better results as topics are better depicted in datasets closer to the dataset to be tested. This is particularly true for broad topics as the ones that have not been monitored until recently. DeBERTa performs very consistently the best, while it was the exact contrary for GPT2. DeBERTa and BERT are more likely to be the first choices over other models when training time is limited. Even though neither MEAN nor MEDIAN yielded the best results every time, it is still interesting to take them into account as, for most of the experiments, they were close to the top scores. MEDIAN is usually more secure to be kept.

3.3 GPU Memory usage and training time

3.3.1 GPU Memory usage

Total SE's HPC platform is equipped with multiple GPUs and CPUs. In this work, 4 Nvidia Tesla V100 (32GB

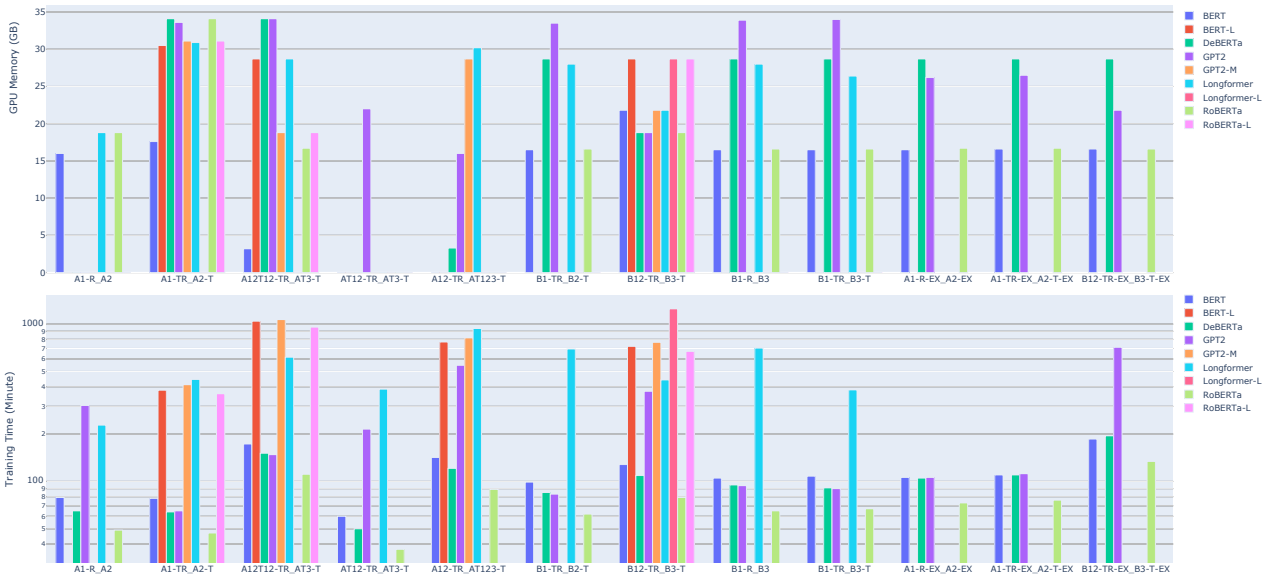


FIGURE 2 – Detailed GPU memory usage and training time for different transformer models in all the experiments.

VRAM) have been used. The peak GPU memory usage for all the tested models can be found in Figure 2. Note that GPU RAM usage reports might not be as accurate as expected. Several factors like GPU caching / RAM fragmentation / initialization etc. could prevent the program from returning the exact RAM usage information [16].

Using gradient accumulation allowed to run larger versions of the models, trading-off time for memory, with very limited performance influences.

For models with same data inputs and close parameter spaces, the maximal memory usage remains similar. These models have similar amounts of network parameters and input tensor dimensions, so the highest memory usages are at the same level. However, once the models have been initialized and are running in stable phase, much less VRAM is required. So, to correctly run these transformer models, one would need a GPU with at least 24GB or ideally 32GB of VRAM, otherwise, the model cannot even be initialized.

3.3.2 Training time

Memory requirements are surely an issue when it comes to training transformer models, but time can also be problematic. Here, the base models have been trained on 4 Nvidia Tesla V100 whereas larger models used only one V100 due to the gradient checkpointing technique. The preliminary benchmarking on gradient checkpointing showed, that there is no gain with multi-GPU training in the current environment. The GPU-GPU communication is much more time-consuming than the potential time-saving from the usage of multiple GPUs. Using a single V100 is faster in this case. The number of training epochs was 20 and the maximum sequence length was 512 tokens. In all tested cases, RoBERTa was the fastest, mostly followed by DeBERTa. If it was not the case, DeBERTa was usually only about 2 minutes slower than BERT or GPT2. As for GPT2, it depends on the datasets : it could perform as fast as the other models in some cases, but way slower for some other tests.

Larger versions of the models tend to need more time to run, due to the large parameter space and the gradient checkpointing application. Longformer, whether it was in its base form or not, took way longer than the other transformers. With a sequence length of 512 tokens on B12-TR_B3-T, base Longformer took 7 hours 20 minutes to complete the training, while large Longformer required 20 hours 50 minutes for the same task.

The Figure 3 shows both training time and GPU VRAM used for each model.

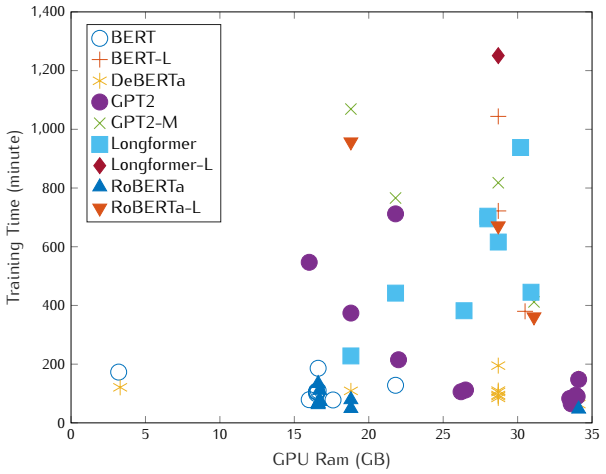


FIGURE 3 – GPU memory usages and training times for different transformer models in all the experiments.

4 Conclusion

Our work aimed at determining a list of promising techniques and models to be used for business intelligence.

Thanks to all the computing resources provided by Total SE, we were able to massively perform experiments with state-of-the-art transformer models and different parameter configurations.

The exhaustive BERT experiments allowed us to evaluate the effects of different features and modifications, both on the models themselves and on the training and test datasets. Amongst the best techniques and models we have been able to test, concatenating titles to claims, re-sampling are essential. Extending the datasets can also be very interesting but may need more work to make it a reliable solution. The best model in terms of performances and resource consumption (both VRAM and time) is DeBERTa, closely followed by BERT. The intuition that a high-performing model on the GLUE and SuperGLUE tasks would also work efficiently for our use-case has paid off. Moreover, the BERT experiments showed, that reducing the batch size, along with using gradient accumulation, gave comparable results to those of models with an unchanged batch size, and achieved this while reducing the amount of VRAM required to run. Neither Longformer nor GPT2, both for their time consumption and performances, can be recommended for a similar task as the one in this study.

For topic A, the best combination of models (the MEDIAN) reached an $\mathcal{M}$ score of 0.8826 with a recall of 0.9451, a precision of 0.5833 and only 16.3% of the total patents left for manual classification by the experts. If originally, 2,000 patents were to be processed, this model would leave only about 300 patents for manual annotation, dividing by almost ten folds the workload of the experts. However, for topic B, results are slightly worse : the best combination of models (the MEAN) reached an $\mathcal{M}$ score of 0.7710 with a recall of 0.8057, a precision of 0.5146 and 22.9% of the total patents left for manual labeling. Results regarding topic B are worse than those of topic A, most certainly because of the broadness of subjects represented in topic B's patents.

5 Discussion

According to the extensive BERT experiments, it has been observed that the original vocabulary produced slightly better results than the customized vocabulary. Note that DeBERTa's vocabulary files have not been modified. Its top performances may partly be explained by this difference. Since modifying the vocabulary means the transformers will be retrained, their weights are being modified, possibly leading to a slight impact on their performances.

BERT experiments also showed better results when used gradient accumulation and reducing the batch size. However, one can argue on the veracity of this result for every model-dataset combination. Gradient accumulation should not yield better performances than using a big enough batch size. Retraining transformer models with customized vocabularies may be usable for domain-specific tasks. However, as confirmed from the experiments, this training would require much more data to build a robust and better model.

Experiments (5) and (6) are distinguished by the non-use of A12 as training datasets in (6). Results are, as expected, better in (5). However, using 11,000 examples as the training dataset against only 3,000 for a mere increase of 2.63% in score $\mathcal{M}$ may not the most interesting time-performance trade-off. So, it is advised to use additional data only when the latter is close enough to the dataset to be predicted. In practice, a small margin, including more patents than the group considered as relevant by the transformers, is expected, and all of them will be manually analyzed. This is to minimize the risk of missing critical patents. The margin will depend on the transformers' performances on a given topic and is to be determined accordingly.

We are aware that not having a low $\mathcal{M}$ score is relatively expected. However, reaching a high $\mathcal{M}$ score may be a very hard achievement (90+%), especially on broad or new topics as topic B. Although score $\mathcal{M}$, given in this study, is specific to the view of Total SE's experts on the ratio of true positives to false positives, this metric may be utilized by other companies to similarly evaluate their models' performances. For example, if missing a few more true positives but decreasing further down the workload is required, assigning a higher weight to the *patents left* score and a lower one to the recall can be considered.

The next steps would involve testing more models, larger versions, training / retraining our own transformer models, including more data from the patents like the descriptions or the abstracts, optimizing the dataset extension technique. A possible improvement might be the attribution of a weight to the probability of each chunk of claims according to their size in terms of tokens.

References

[1] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT : pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018.

[2] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta : A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019.

[3] P. He, X. Liu, J. Gao, and W. Chen, "Deberta : Decoding-enhanced bert with disentangled attention," *arXiv preprint arXiv :2006.03654*, 2020.

[4] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, 2019.

[5] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer : The long-document transformer," *arXiv preprint arXiv :2004.05150*, 2020.

[6] S. Choi, H. Lee, E. L. Park, and S. Choi, "A deep patent landscaping model using transformer and graph convolutional network," *CoRR*, vol. abs/1903.05823, 2019, withdrawn.

[7] S. Li, J. Hu, Y. Cui, and J. Hu, "Deeppatent : patent classification with convolutional neural networks and word embedding," *Scientometrics*, vol. 117, no. 2, pp. 721–744, 2018.

[8] A. H. Roudsari, J. Afshar, C. C. Lee, and W. Lee, "Multi-label patent classification using attention-aware deep learning model," in *2020 IEEE International Conference on*

Big Data and Smart Computing (BigComp). IEEE, Feb. 2020.

[9] A. C. Marco, J. D. Sarnoff, and C. A. deGrazia, "Patent claims and patent scope," *Research Policy*, vol. 48, no. 9, p. 103790, Nov. 2019.

[10] J. Hu, S. Li, Y. Yao, L. Yu, G. Yang, and J. Hu, "Patent keyword extraction algorithm based on distributed representation for patent classification," *Entropy*, vol. 20, no. 2, p. 104, Feb. 2018.

[11] J.-S. Lee and J. Hsiang, "Patent classification by fine-tuning bert language model," *World Patent Information*, vol. 61, p. 101965, 2020.

[12] L. Zhang, L. Li, and T. Li, "Patent mining : A survey," *SIGKDD Explor. Newsl.*, vol. 16, no. 2, p. 119, May 2015.

[13] C. Hubans and N. Shchukina, "Processing a 4d seismic signal based on noise model," FR Patent WO2 019 053 484, 9 18, 2017.

[14] B. Xu, C. Gil-Jardiné, F. Thiessard, E. Tellier, M. Avalos, and E. Lagarde, "Pre-Training a neural language model improves the sample efficiency of an emergency room classification model," in *The 33rd Florida Artificial Intelligence Research Society Conference*, 2020.

[15] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush, "Transformers : State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing : System Demonstrations*. Online : Association for Computational Linguistics, Oct. 2020, pp. 38–45.

[16] fastai, "Working with gpu," in *fastai Documentation*, Mar 13, 2021.

Vers une étude comparative de différentes approches de classification automatique de textes provenant des secteurs métiers

M. B. Billami, M. Kandi, L. Nicolaieff, K. Ducharlet, C. Gosset,
S. Rey, C. Bortolaso, M. Derras

Berger-Levrault, 64 Rue Jean Rostand, 31670 Labège, France

{mb.billami, mohamed.kandi, lina.nicolaieff, kevin.ducharlet,
camille.gosset, stephanie.rey, christophe.bortolaso,
mustapha.derras}@berger-levrault.com

Résumé

De nos jours, la classification automatique de textes est en passe de devenir un domaine de recherche de plus en plus appliqué aux secteurs métiers. De fait, nous assistons à un intérêt croissant des entreprises pour l'exploration des contenus textuels et le traitement automatique du langage naturel. Des techniques de classification ont été proposées impliquant l'entraînement de modèles sur des corpus de données provenant du domaine général. Cependant, ces modèles peuvent se retrouver en difficulté lorsqu'ils sont utilisés pour des métiers spécifiques ayant un vocabulaire spécialisé. Dans cet article, nous proposons plusieurs modèles de classification automatique ayant différents types de représentation de documents pour une application à des secteurs métiers. L'évaluation de nos modèles est effectuée sur un corpus français et validée sur deux corpus de référence pour l'anglais.

Mots-clés

Vocabulaire de spécialité, Types de représentation de documents, Plongements lexicaux, Termes-clés, Classification automatique de textes.

Abstract

Nowadays, automatic text classification is becoming a research field increasingly applied to business sectors. In fact, we foresee a growing interest from companies in exploring text content and automatic natural language processing. Classification techniques have been proposed involving training models on data corpus obtained from the general domain. However, these models tend to run into difficulties when used for specific business domains involving a specialized vocabulary. In this article, we propose different automatic classification models for application to business sectors with different types of document representation. The evaluation of our models is performed on a French corpus and validated on two references corpus for English.

Keywords

Specialty vocabulary, Types of document representation, Word Embeddings, Key-terms, Automatic text classification.

1 Introduction

Face à la concurrence accrue, une entreprise doit rester à la pointe de l'innovation. Cela implique l'identification de potentiels technologiques et d'avancées scientifiques, mais également l'exploration de nouveaux marchés variés et spécifiques. Depuis une vingtaine d'années, nous sommes confrontés à une quantité croissante de données à traiter au quotidien. Tous les secteurs industriels et professionnels, ainsi que toutes les activités scientifiques, sociales, politiques et technologiques génèrent de grandes quantités d'information. Il est ainsi de plus en plus compliqué pour une organisation de se tenir au courant des nouveautés et innovations de ses différents cœurs de marché, que ce soit dans les domaines scientifiques, technologiques ou métiers. Une solution face à ce déferlement d'information est l'automatisation de la veille. Cela implique le recueil et la classification automatique des informations selon les métiers de l'entreprise et l'identification des innovations scientifiques et technologiques à suivre. Ainsi, la classification ou la catégorisation automatique de textes est un domaine de recherche qui peut répondre parfaitement à cette problématique de veille technologique.

La classification automatique de textes peut être considérée comme une procédure permettant d'affecter des documents à un ensemble de catégories prédéfinies. La classification est ainsi utilisée pour organiser, gérer et classer des documents. Dans cet article, nous nous intéressons à la mise au point d'un système automatique de veille innovation et marché. Un tel système implique plusieurs problématiques relatives à la collecte d'information sur le web (*web scraping*) et à la classification des documents collectés. La difficulté principale réside dans le vocabulaire utilisé qui est à la fois : (1) spécifique à des domaines d'activité très pointus et (2) en constante évolution pour cibler pertinemment ce qui relève d'une innovation.

Premièrement, les domaines de spécialités marchés des entreprises peuvent être très pointus avec des vocabulaires métiers très spécifiques, dont les lexiques et/ou les bases de connaissances de spécialité ne sont pas toujours accessibles. De plus, pour une entreprise travaillant dans de multiples secteurs d'activités (comme le *Médico-Social*, les logiciels de *Gestion Financière et de Ressources Humaines*, la *Gestion des*

Ressources Matérielles et Roulantes, et *les Outils de Relation avec les Citoyens*), le sens métier des termes « *pilotage (gouvernance)* », « *facture (dépense)* », voire « *permis de conduire (examen)* » se révèlent souvent plus importants que le sens commun « *pilotage (véhicule)* », «*facture (justificatif)* » et « *permis de conduire (document)* ». Il est ainsi complexe d'utiliser d'une manière directe des modèles de classification entraînés sur des corpus génériques pour permettre une classification de documents provenant de tels domaines métiers, et surtout quand la langue traitée est le français (plus de complexité et peu de corpus métiers fournis pour l'apprentissage). De plus, dans notre problématique, une catégorisation en classes multiples peut être pertinente. Par exemple, un article sur les « *smart city* »[1] peut à la fois concerner la gestion de la relation citoyen à travers les questions de démocratie participative, mais également la gestion des ressources et équipements de la ville comme l'éclairage grâce à l'IoT (*Internet of Things*). Il n'est ainsi pas possible d'utiliser un seul classificateur binaire pour ce genre d'informations multi-spécialités.

Deuxièmement, le vocabulaire spécifique à l'innovation est en constante évolution. Par exemple, plusieurs termes issus des tendances technologiques stratégiques de Gartner[2] ne sont pas présents dans des ressources génériques comme DBpedia [21]. Les termes tels que « *Visualisation de données* », « *3D* », « *réalité mixte* » ou « *Hyperautomation* » ne sont pas actuellement référencés dans DBpedia. De nouvelles avancées technologiques, scientifiques ou en innovations sociales font ainsi émerger chaque jour de nouveaux mots/termes et concepts. Par exemple, des expressions polylexicales telles que « *ville du quart d'heure* », ou « *inclusion numérique* » sont émergées très récemment. Un apprentissage statique devient vite une limite pour une application de veille sur l'innovation dans un cadre métier.

Afin de répondre à ces enjeux, nous proposons dans cet article trois approches de classification/catégorisation automatique de textes : deux approches sont supervisées et une approche est non supervisée. Nous évaluons les méthodes proposées sur un corpus français spécifique à notre cas d'usage de veille métier et innovation. Ensuite, nous validons ces méthodes pour une application sur des corpus de référence anglais et nous proposons une comparaison à plusieurs systèmes état-de-l'art.

Après avoir présenté en section 2 les travaux antérieurs de la classification automatique de textes, nous décrivons en section 3 plus en détail notre problème de classification de documents traitant de thématiques métiers spécifiques. Ensuite, en section 4, nous présentons l'architecture générale de notre système de classification. Par la suite, dans la section 5, nous décrivons notre méthodologie de travail avec la proposition de 3 méthodes de classification automatique de textes. Les différents corpus que nous utilisons sont présentés en section 6. Enfin, nous discutons les résultats d'évaluation en section 7 avant de conclure sur notre travail en section 8.

[1] http://www.envirolex.fr/smart-city-et-ville-du-quart-dheure-paris-veut-se-transformer/

[2] https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2020/

2 État de l'art

Plusieurs travaux de recherche sur la classification de documents textuels ont été proposés. Trois grandes catégories d'approches émergent [5, 7, 9, 18], à savoir : (1) les approches supervisées [1, 2, 8, 10, 16]; (2) les approches non supervisées [6, 14, 23]; et (3) les approches semi-supervisées ou hybrides [12, 13]. Beaucoup de systèmes de classification automatique de documents textuels existent aujourd'hui. Un ensemble de corpus de référence est mis à disposition de la communauté scientifique pour permettre la validation et la comparaison de ces systèmes.

La plupart des systèmes de classification de ces dernières années ont été évalués principalement sur des corpus de référence anglais, tels que : Reuters-21578[3], 20-Newsgroups[4], WebKB[5], voire IMDb[6]. Pour les autres langues, des corpus d'évaluation existent aussi, tels que DEFT'08 [11] pour le français, TanCorp [4] pour le chinois ou Kalimat Corpus[7] pour l'arabe. Ci-après, nous présentons quelques systèmes état-de-l'art ayant été appliqués pour l'anglais et avec lesquels nous effectuons notre étude comparative. Pour un état-de-l'art complet, le lecteur pourra consulter le travail mené par Dhar et al. [7].

Labani et al. [20] ont proposé un modèle, appelé *Critère de discrimination relative multivariée* (MRDC), pour obtenir les caractéristiques de classification de textes. L'importance de leur modèle réside dans la réduction des caractéristiques redondantes basées sur les concepts de pertinence maximale et de redondance minimale. Dans ce modèle, pour chaque token, la fréquence des documents a également été prise en compte. Ce modèle se propose ainsi comme une méthode de filtrage. Son évaluation a été effectuée sur trois corpus anglais, à savoir : (a) Reuters-21578, (b) 20-Newsgroups et (c) WebKB. Une meilleure précision a été obtenue en utilisant le classificateur Multinomial Naïve Bayes (MNB). Par ailleurs, dans les travaux de Jiang et al. [15], une approche hybride de classification a été introduite, basée sur un réseau DBN (*deep belief network*) et une régression *softmax*. Le DBN a été utilisé pour résoudre les problèmes de calcul matriciel à haute dimension et à dispersion.

Kowsari et al. [19] ont proposé la méthode RMDL (*Random Multimodel Deep Learning*). Cette dernière a la capacité de déterminer les frontières entre classes par suite de l'obtention de l'architecture d'apprentissage profond la plus appropriée. Cette méthode permet d'améliorer les performances de classification grâce à des ensembles d'architectures d'apprentissage profond. Wu et al. [24], quant à eux, ont travaillé sur une technique d'équilibre entre la surpondération et la sous-pondération dans un système de pondération supervisée de termes. Ils ont généré quatre règles basées sur les paramètres du modèle qu'ils ont définis. Les documents ont

[3] https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection

[4] https://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups

[5] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/webkb-data.gtar.gz

[6] https://datasets.imdbws.com/

[7] https://sourceforge.net/projects/kalimat/files/latest/download

été classés selon 5 méthodes de défuzzification, à savoir : (a) *Centroid*, (b) *Bisector*, (c) *Smallest of Maxima* (SOM), (d) *Mean of Maxima* (MOM) et (e) *Largest of Maxima* (LOM). Parmi ces méthodes, *Centroid* a obtenu les meilleurs résultats.

Jin et al. [16] ont étudié un modèle de classification conceptuellement simple en exploitant des plongements lexicaux (*word embeddings*) basés sur les classes de textes. Leur hypothèse réside dans le fait que les mots présentent des caractéristiques de distribution différentes selon les classes de textes. Sur la base de cette hypothèse, ils ont entraîné des représentations de mots pour chaque classe. Pour la prédiction de la classe d'un nouveau document, un calcul probabiliste est effectué. La classe sélectionnée est celle qui maximise les probabilités de sélection des vecteurs d'*embeddings* de ses mots.

Jiang et al. [14] ont travaillé sur le concept d'un algorithme à base de similarité pour regrouper les caractéristiques de textes où les tokens sont regroupés en différents groupes en fonction du test de similarité. Des tokens similaires sont associés à une fonction d'appartenance ainsi qu'à une moyenne et un écart en fonction des résultats de leur similarité. Dogan et Uysal [8] ont présenté une étude exhaustive sur l'utilisation de différentes composantes de la fréquence des termes et cela pour 7 méthodes de pondération supervisée de termes parmi lesquelles 6 méthodes existaient déjà et se sont dérivées les unes des autres. L'évaluation des différentes méthodes a été effectuée sur les corpus Reuters-21578 et 20-Newsgroups en utilisant deux algorithmes de classification, à savoir : les Machines à vecteurs de support (SVM) et Rocchio. Ce dernier est l'un des classificateurs basés sur les centroïdes de classes.

Pour le français, la classification automatique de textes a déjà présenté son intérêt par le passé. La 4ème édition des campagnes d'évaluation DEFT (DÉfi Fouille de Textes) datant de 2008 [11] a été concentrée sur cette problématique. La tâche de classification avait pour objectif de catégoriser des textes en 9 classes, à savoir : (a) *Sports*, (b) *International*, (c) *Art*, (d) *Économie*, (e) *Littérature*, (f) *Politique française*, (g) *Sciences*, (h) *Société* et (i) *Télévision*. Le corpus DEFT'2008 a été constitué à partir de deux sources distinctes : *Le Monde* et *la Wikipédia francophone*. Pour chacune de ces sources, un article est identifié s'il apparaît avec l'une des 9 classes. Les résultats obtenus durant cette édition ont montré que les systèmes d'apprentissage supervisé et à base des SVM semblent être très performants. De plus, le prétraitement des textes par lemmatisation de tokens n'était pas indispensable. Cependant, la réduction de l'espace vectoriel par l'utilisation de l'information mutuelle afin de projeter les textes s'est montrée significative. Les meilleures techniques utilisées durant la campagne 2008 pour le français semblent être pertinentes même pour l'anglais puisque des techniques similaires avec des résultats similaires ont été présentés en 2019 par Dogan et Uysal [8].

Dans cet article, nous nous intéressons à une étude comparative de plusieurs méthodes de classification de textes d'actualités écrits en français. Ces textes proviennent d'un corpus de données que nous avons nous-mêmes créé en effectuant une collecte automatique à partir de sources web ciblées. Nous appelons ce corpus par la suite le corpus BL-News. Les méthodes que nous proposons sont indépendantes de la langue. De ce fait, nous présentons une évaluation de nos méthodes sur notre corpus français et une validation sur deux corpus de référence anglais, avec lesquels plusieurs systèmes état-de-l'art ont été validés, à savoir : Reuters-21578 et 20-Newsgroups. Par ailleurs, les méthodes que nous proposons utilisent (a) soit des techniques d'apprentissage supervisé, (b) soit des techniques d'apprentissage non supervisé et à base de similarité sémantique. Pour les systèmes avec un apprentissage supervisé, différentes approches sont proposées.

3 Définition du problème

Comme le démontre l'état de l'art, beaucoup de modèles de classification automatique de textes ont déjà fait leur preuve. Cependant, notre cas d'application a plusieurs particularités pour lesquelles le système de classification automatique doit s'adapter et répondre mieux à ce besoin. Dans cette section, nous présentons trois grandes contraintes sur notre cas d'usage.

3.1 Vocabulaire métier

Les documents que nous souhaitons traiter font référence à des thématiques spécifiques en rapport avec des métiers bien précis. Cette contrainte implique la présence d'un vocabulaire technique qu'il va falloir reconnaître, traiter et pondérer. Nous nous intéressons ainsi à la gestion dans les secteurs métiers suivants : (a) *Éducation*, (b) *Gammes de Gestion* (*financière et ressources humaines*), (c) *Médico-social*, (d) *Relation avec les Citoyens* et (e) *Asset & Fleet* (*ressources matérielles et roulantes*). Le vocabulaire métier de ces thématiques est composé d'une part de mots simples et d'autre part de multi-mots, voire d'expressions. Par exemple, « *préadmission* (*Médico-social*) », « *feuille d'émargement* (*Éducation*) », « *Collectivité territoriale* (*Gammes de Gestion*, *Relation avec les Citoyens*) », « *Internet des Objets* (*Asset & Fleet*) », « *Pacte Civil de Solidarité – PACS* (*Relation Citoyen*) », etc. Nous avons associé manuellement à chaque thématique un ensemble de termes-clés dont ces exemples font partie. Nous appelons chaque ensemble comme étant un lexique d'une thématique donnée.

Afin de quantifier la spécificité de nos lexiques, nous avons mesuré la couverture de la base DBPedia sur les termes-clés de chacun. Le tableau 1 présente le nombre de termes-clés de chaque lexique avec le pourcentage de couverture de la base DBpedia contenant actuellement plus de 4 millions d'entrées.

Lexique	Taille du lexique	Couverture DBpedia (%)
Asset & Fleet	180	29
Éducation	98	16
Gammes de Gestion	122	27
Médico-social	**284**	14
Relation Citoyen	216	30
Innovation	125	**43**
Total	**1 025**	**43**

Table 1 – Couverture des lexiques par DBpedia

En plus des lexiques thématiques, nous avons aussi un lexique « *Innovation* ». Ce dernier fait référence à la catégorie

transversale *Innovation* de tous nos secteurs métiers**.** Cette catégorie tient compte de l'information innovante pouvant se retrouver dans le lot de documents abordant les thématiques métier. En effet, nous nous intéressons ici non seulement à classer l'information relative aux métiers mais aussi à vérifier si elle évoque de l'innovation. Les résultats du tableau 1 montrent que la couverture totale du vocabulaire de nos lexiques représente 43 %, ce qui reste faible. Par exemple, « *5G* », « *devops* », «*générateur de formulaires électroniques*» ne sont pas reconnus actuellement dans DBPedia.

3.2 Classification en classes multiples

En plus de la catégorie transversale « *Innovation* », un document (article) peut être classifié dans une ou plusieurs thématiques. Nous nous retrouvons donc dans le cadre d'une classification multiple. Le nombre de catégories (classes) associées à un document n'est pas fixe. Cette classification se différencie de la classification standard qui attribue une seule étiquette par instance (document).

3.3 Évolution dynamique dans le temps

Nous avons, d'une part, les lexiques métiers qui sont amenés à évoluer dans le temps et, d'autre part, l'aspect *Innovation* que nous voulons faire ressortir du lot des documents à traiter. Cette évolution de l'ensemble des lexiques est importante à prendre en considération pour une meilleure classification.

4 Description du système

Nous avons mis en place un système expérimental pour notre étude comparative des méthodes de classification de textes. L'architecture de ce système est illustrée dans la figure 1.

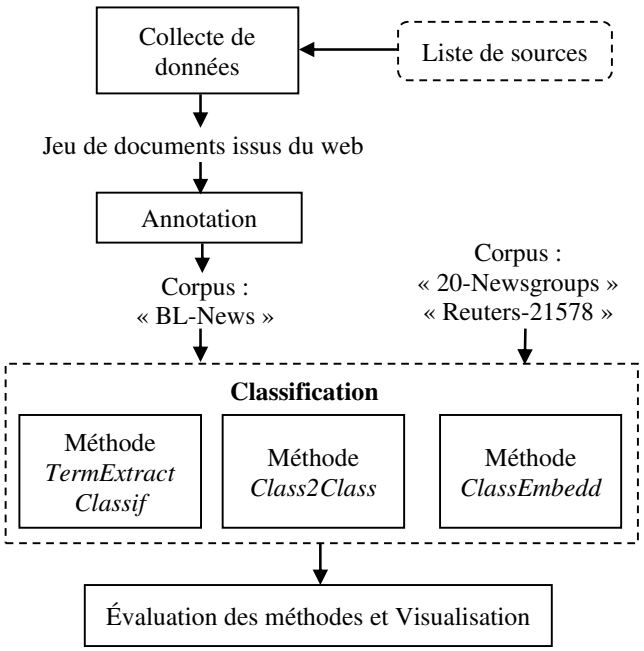


Figure 1 – Architecture du système de classification

Tout d'abord, nous avons développé un module de collecte automatique et périodique de documents issus du web. La liste de sources, constituée de sites d'actualités fréquentés par les experts métiers, est éditée en amont par un expert. Le fonctionnement de la collecte de données est expliqué en détail dans la section 6.1.

Ensuite, nous avons lancé une campagne d'annotation des documents collectés à laquelle neuf annotateurs humains y ont participé. Chacun des annotateurs est un locuteur natif du français et possède la connaissance nécessaire pour juger la pertinence d'associer un document à une thématique donnée. Par ailleurs, chaque document est attribué par un seul annotateur à une ou plusieurs thématiques métiers, voire parfois à aucune. L'annotateur précise aussi si le document traite d'un sujet innovant ou non (cf. *catégorie transversale*). Les résultats de cette campagne sont présentés dans la section 6.2. BL-News représente ainsi le corpus construit. Comme cité en section 2, la validation de nos méthodes de classification est appliquée sur deux corpus anglais : 20-Newsgroups et Reuters-21578.

Les trois corpus sont utilisés pour entraîner (dans le cadre d'une supervision) et évaluer les méthodes de classification automatique. L'implémentation de nos méthodes est effectuée en langage Python et fait appel à l'utilisation de plusieurs librairies *open-source* pour le traitement du langage naturel et l'apprentissage automatique, telles que *spaCy*[8] [13], *PKE* (*Python Keyphrase Extraction*)[9] [2] et *Scikit-learn*[10] [3]. Enfin, une interface web basée sur *Django*[11] a été développée pour faciliter l'édition de la liste de sources et l'annotation des documents. La visualisation et l'analyse des résultats se repose sur la pile *ELK* (*Elasticsearch, Logstash et Kibana*)[12].

5 Méthodologie

Dans cette section, nous présentons trois méthodes de classification automatique de textes. La première méthode repose sur un apprentissage supervisé et utilise des techniques d'extraction de termes-clés pour créer des représentations conceptuelles de documents. La deuxième méthode repose elle-aussi sur un apprentissage supervisé mais vise plutôt à proposer plusieurs classificateurs binaires et utilise des techniques de réduction de dimensionnalité. La troisième méthode, quant à elle, repose sur un apprentissage non supervisé et utilise les plongements lexicaux pour la création d'embeddings représentant les classes et les documents dans un même espace de représentation vectorielle.

5.1 Approche supervisée par extraction de termes-clés

Cette approche se fonde principalement sur une extraction de termes-clés. Nous l'appelons *TermExtractClassif.* Chaque thématique de nos secteurs métiers est caractérisée par des termes-clés. Un document est potentiellement pertinent s'il contient suffisamment de termes-clés dans son titre et/ou son contenu.

[8] https://spacy.io/
[9] https://github.com/boudinfl/pke/
[10] https://scikit-learn.org/stable/
[11] https://www.djangoproject.com/
[12] https://www.elastic.co/fr/what-is/elk-stack/

Dans l'idéal, il serait intéressant d'énumérer en amont tous les termes-clés d'une thématique et leur associer des poids d'importance. Ensuite, ces termes-clés pourraient être utilisés pour mesurer la pertinence des documents vis-à-vis de la thématique. Cependant, il n'est pas raisonnable de construire manuellement une liste suffisamment exhaustive de termes-clés pour des thématiques métiers spécifiques. De plus, si nous prenons en compte l'aspect *innovation*, la liste est amenée à évoluer dans le temps. La méthode *TermExtractClassif* permet de générer une liste de termes-clés pondérés à partir d'un corpus de documents annotés manuellement. Le principe du processus de cette génération est illustré dans la figure 2.

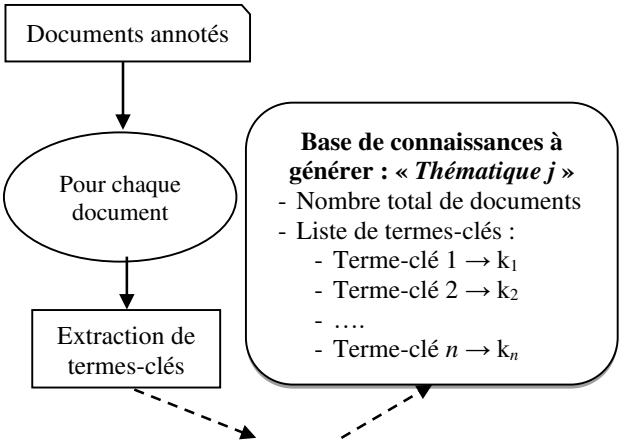


Figure 2 – Construction/enrichissement de la base de connaissances

Pour chaque document annoté en catégorie/thématique par les annotateurs humains, nous effectuons une extraction de termes-clés à partir du titre et du contenu, à l'aide de la bibliothèque *PKE* (*Python Keyphrase Extraction*) [2]. La liste de termes-clés d'une thématique donnée est ainsi constituée de termes extraits automatiquement à partir du corpus. Afin de quantifier l'importance de chaque terme, nous lui attribuons le nombre de documents dans lesquels il apparaît. La base de connaissances d'une thématique donnée est ensuite construite. Nous effectuons cette opération pour toutes les thématiques. Chaque base de connaissances est ainsi utilisée pour mesurer la pertinence de nouveaux documents à classer. La figure 3 illustre ce processus de prédiction. Quand un nouveau document '*d'* arrive, nous effectuons une extraction de termes-clés. Ensuite, nous vérifions si ces termes-clés sont présents dans chaque base de connaissances. Le score du document '*d'* pour la thématique '*j'* est calculé comme suit :

$$score\ (d,j) = -\sum_{i=1}^{n}\left((k_i/T_j) * \log\ (k_i/T_j)\right) \qquad (1)$$

k_i représente le nombre de documents contenant le terme-clé *i* dans le corpus annoté ; T_j représente le nombre total de documents annotés de la thématique *j* ; et *n* représente le nombre de termes-clés dans le document *d*. Nous mesurons un score de pertinence pour le titre et pour le contenu. Un document est considéré comme pertinent si son score de titre et/ou de contenu est supérieur à un seuil fixé au préalable.

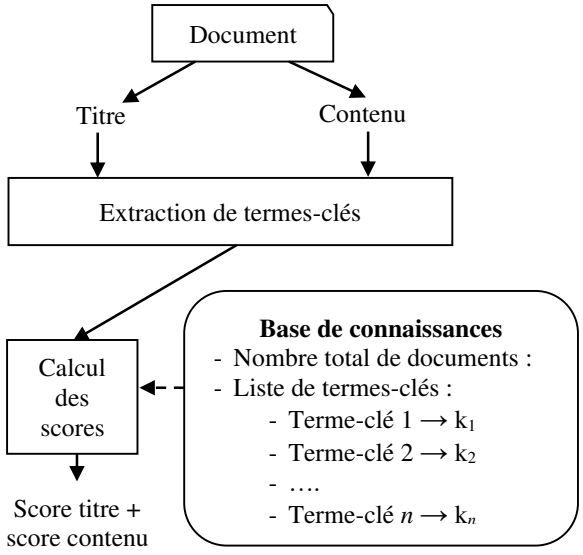


Figure 3 – Approche de classification avec TermExtractClassif

5.2 Approche supervisée : des classificateurs binaires et une analyse discriminante linéaire

La taille des vecteurs caractéristiques permettant de représenter les documents est un élément clé dans la classification de textes. En effet, avoir à disposition un très grand corpus de données, et en appliquant un TF-IDF standard (fréquence du terme-fréquence inverse du document) [17], va engendrer la création de vecteurs avec un très grand nombre de dimensions. L'approche que nous proposons cherche à avoir plus de précision et moins de temps de traitement dans la classification de textes. Pour, cela, nous utilisons des techniques de réduction de dimensionalité avec un top n des caractéristiques provenant du TF-IDF, l'application d'un seuil de variance et une analyse discriminante linéaire (*Linear Discriminant Analysis - LDA*).

Les différentes étapes de cette approche sont présentées dans la figure 4. Nous appelons cette approche par la suite *Class2Class*.

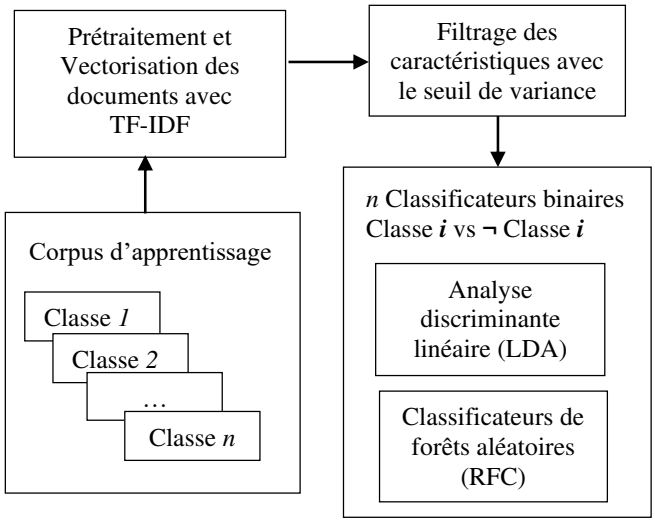


Figure 4 – Approche de classification avec Class2Class

Tout d'abord, nous effectuons un prétraitement sur les données d'apprentissage. Pour cela, nous utilisons *Spacy* [13] pour la lemmatisation. Les modèles « *fr_core_news_lg* » et « *en_core_web_lg* » sont utilisés pour le français et l'anglais, respectivement. Ensuite, nous retirons tous les nombres du corpus et nous gardons en textes seulement les mots portant du sens, à savoir : *noms communs*, *noms propres*, *adjectifs*, *adverbes* et *verbes*.

Par la suite, nous transformons les textes en vecteurs de caractéristiques à l'aide du TF-IDF en utilisant la bibliothèque *Scikit-learn* [3]. Le TF-IDF permet de donner des poids plus élevés aux termes (ou mots) qui apparaissent plus fréquemment dans un texte par rapport aux autres textes du corpus. Par conséquent, nous pouvons dire que le TF-IDF mesure la pertinence et non seulement la fréquence. Nous avons fait le choix de construire un modèle de vectorisation à base de n-grammes avec n ∈ {1, 2, 3}. Cela se justifie par le fait que plusieurs termes-clés de nos lexiques représentent des multi-mots. Nous sélectionnons un nombre maximal de 4 000 meilleures caractéristiques avec TF-IDF. Ensuite, nous utilisons une technique de filtrage, à savoir : le seuil de variance (*Variance Threshold*) qui est proposé dans *Scikit-learn*, pour la suppression des entités constantes. Ces dernières contiennent une seule valeur pour toutes les sorties de l'ensemble du corpus d'entraînement. Ils ne peuvent donc nous donner aucune information précieuse qui pourrait aider la classification de textes.

Après avoir généré le modèle TF-IDF et celui en appliquant le seuil de validation, nous nous intéressons à la création des classificateurs binaires pour chaque thématique traitée. Pour le corpus BL-News, nous créons 12 classificateurs : six avec l'analyse discriminante linéaire et six avec les forêts aléatoires (*Random Forest Classifiers - RFC*). Chacun des six classificateurs traite une thématique donnée, par exemple, un classificateur pour la détection de l'*Innovation* et un classificateur pour détecter si l'article parle d'*Éducation* ou non. Le corpus d'apprentissage d'un classificateur donné représente ainsi deux classes : une classe concerne une thématique bien précise, l'autre classe est construite par complétude du corpus. Ainsi, nous avons des corpus déséquilibrés quelle que soit la thématique à traiter. Par ailleurs, il est à noter que le *RFC* est un méta-estimateur qui ajuste un certain nombre de classificateurs type « *arbres de décision* » sur divers sous-échantillons de l'ensemble de données et utilise la moyenne pour améliorer la précision prédictive et contrôler le surajustement.

5.3 Approche non supervisée : *word embeddings* vers *document embeddings* et *class embeddings*

Nous avons mis en place une méthode non supervisée dans le but qu'elle soit indépendante de toute base d'apprentissage. Cela retire ainsi à la méthode les éventuels biais liées aux données d'apprentissage. Nous appelons cette méthode *ClassEmbedd* (cf. figure 5)**.** Cette méthode fonctionne en deux étapes : (1) tout d'abord, nous créons une représentation vectorielle pour les documents et les thématiques à l'aide de plongements lexicaux ; et (2) nous utilisons un score de similarité pour évaluer la distance entre les documents et les thématiques.

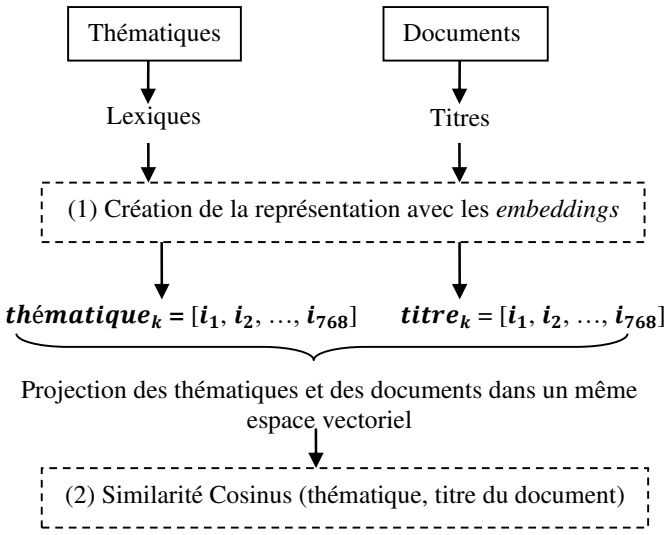


Figure 5 – Approche de classification avec ClassEmbedd

(1) – Représentation sémantique par utilisation de plongements lexicaux

Notre objectif s'est porté sur la recherche d'un modèle de représentation vectorielle permettant la comparaison de nos lexiques métiers avec les documents collectés. Nous avons fait le choix de prendre en considération le modèle *Sentence Transformers* [22], en utilisant la bibliothèque *PyTorch*[13]. Ce modèle permet d'encoder les mots des phrases en prenant en compte le contexte dans lequel les mots apparaissent. Pour ne pas lisser les informations pertinentes dans un document, il faut fournir au modèle un segment résumant l'essentiel du contenu de ce document. Pour cette raison, nous avons fait le choix d'encoder seulement les titres des documents. En effet, le titre est un élément important qui contient de l'information pertinente d'un document. Concrètement, lorsque tout le contenu d'un document est pris en considération, l'information est bruitée par les mots superflus et les résultats peuvent révéler que tous les documents vont tendre vers une représentation similaire. Une fois les documents et les termes-clés de nos lexiques encodés de la même manière, nous pouvons les projeter sur un même espace vectoriel afin de mesurer des similarités sémantiques.

(2) – Similarités sémantiques

Pour la projection, sans avoir besoin d'un corpus de documents annotés manuellement en thématiques, nous avons utilisé l'algorithme des *k*-plus proches voisins (*k-PPV*) pour un entraînement sur les termes de nos lexiques. L'idée derrière l'utilisation du *k-PPV* est de prédire l'emplacement d'un document dans l'espace des lexiques. Les termes-clés des lexiques au voisinage d'un document sont alors retournés. Un voisin proche est défini selon la distance Cosinus la plus petite par rapport à un document donné. Nous avons ensuite récupéré la liste des termes pondérés par leur distance et regroupé par thématique en calculant la distance médiane. La thématique la plus proche correspond donc à celle ayant la distance médiane la plus petite par rapport à un document donné.

[13] https://pytorch.org/

6 Corpus de travail

6.1 Collecte de données

Afin d'évaluer nos méthodes dans le contexte spécifique de la veille métier et innovation, nous avons constitué un jeu de documents textuels issus de sites web d'actualités utilisés par les experts métiers dans notre cas d'utilisation. L'abondance et l'ajout régulier de nouvelles sources rendent impossible le développement d'une méthode spécifique à chaque cas pour l'automatisation de l'extraction du contenu web. Cependant, la qualité de l'extraction à partir d'une méthode générique est dégradée par la variabilité structurelle des sources, ainsi que d'autres problèmes (*restriction du contenu*, *demande de cookies*, etc.). Ce manque de fiabilité dans un cas réel de production peut avoir un impact sur la qualité de la classification, nous avons donc choisi de constituer un jeu de données réaliste pour l'évaluation de nos méthodes. Ce jeu a été généré à partir de 80 sources différentes liées aux différents métiers de notre cas d'étude. L'approche retenue consiste en deux étapes : (1) obtenir, à partir de sources fournies, la liste des derniers articles (documents) publiés ; et (2) extraire, pour chaque article, le contenu textuel d'intérêt.

Nous avons réparti les sources en deux catégories selon qu'elles mettent ou non à disposition un flux RSS (*Really Simple Syndication*). La bibliothèque Python *feedparser*[14] a été utilisée pour collecter les articles depuis les flux RSS des sources (55 sources concernées). Dans le cas où aucun flux RSS n'est fourni (25 sources concernées), nous utilisons les *xpath*, qui permettent de cibler un ensemble de balises dans le code HTML (*HyperText Markup Language*) d'une page web, pour extraire la liste des articles. Pour se faire, nous nous sommes appuyés sur l'implémentation Python de *Selenium*[15]. L'extraction du contenu d'intérêt des articles est réalisée par la bibliothèque Python *Newspaper3k*[16] dont l'approche, similaire à celle employée par les extensions de lisibilité des navigateurs, consiste à isoler le texte d'intérêt des éléments périphériques en s'appuyant sur les types de balises, la longueur de leurs contenus et leur proximité.

6.2 Campagne d'annotation

Nous avons organisé une campagne d'annotation pour un total de 764 documents : 9 annotateurs humains ont participé à la campagne. Nous rappelons que le corps BL-News contient 5 thématiques métier (*Asset & Fleet*, *Éducation*, *Gammes de Gestion*, *Médico-Social*, *Relation avec les Citoyens*), s'ajoute à cela une thématique transversale, à savoir : l'*Innovation*.

Étant donné un document, l'objectif est d'attribuer manuellement une étiquette « *pertinent* » ou « *non pertinent* » pour chacune des thématiques. De plus, il faut préciser si le document traite un sujet *innovant* ou non. Un document peut être associé à une ou plusieurs thématiques, voire aucune. Les résultats de la campagne sont illustrés dans le tableau 2.

Thématique	Nombre de documents	Taille du lexique en termes-clés
Asset & Fleet	103	180
Éducation	30	98
Gammes de Gestion	173	122
Médico-social	128	**284**
Relation Citoyen	174	216
Innovation	**226**	125
Total des documents	**764**	**1 025**

Table 2 – Description du corpus BL-News

6.3 Corpus anglais de référence

Nous utilisons deux corpus état-de-l'art pour la langue anglaise, à savoir : Reuters-21578 et 20-Newsgroups. Le corpus Reuters-21578 est un ensemble d'actualités, pour le domaine de l'économie, tirées par l'agence de presse Reuters. La version originale de ce corpus contient 21 578 documents d'actualité organisés en 135 catégories. Pour ce corpus, nous avons utilisé la version proposée dans la bibliothèque NLTK[17] (*Natural Language Toolkit*) pour laquelle seulement 90 catégories sont prises en considération en respectant le mode de découpage *ModApte* entre l'apprentissage et le test. Le tableau 3 décrit la liste des 10 classes les mieux couvertes en termes d'exemples par le corpus Reuters-21578.

Numéro	Classe	Apprentissage	Test	Total
1	*earn*	**2 877**	**1 087**	**3 964**
2	*acq*	1 650	719	2 369
3	*money-fx*	538	179	717
4	*grain*	433	149	582
5	*crude*	389	189	578
6	*trade*	368	117	485
7	*interest*	347	131	478
8	*ship*	197	89	286
9	*wheat*	212	71	283
10	*corn*	181	56	237
Total	–	**7 192**	**2 787**	**9 979**

Table 3 – Top-10 des classes du corpus Reuters-21578 avec le nombre de documents de chaque classe

Par rapport à l'ensemble des données du corpus Reuters-21578, le top-10 des classes couvre 74,87 % des données (75,04 % pour l'apprentissage et 74,44 % pour le test). Sachant que le corpus Reuters-21578 traité décrit 90 classes.

Le corpus 20-Newsgroups, quant à lui, se compose de près de 20 000 documents ayant été collectés à partir de 20 groupes différents d'actualités constituant ainsi 20 classes différentes. Pour l'application de nos méthodes sur ce corpus, nous avons opté pour l'utilisation de la version proposée dans la bibliothèque *Scikit-learn*[18] [3]. Le tableau 4 décrit la liste des 10 classes les mieux couvertes en termes d'exemples par le corpus 20-Newsgroups.

[14] https://pypi.org/project/feedparser/
[15] https://selenium-python.readthedocs.io/locating-elements.html
[16] https://github.com/codelucas/newspaper/
[17] https://www.nltk.org/book/ch02.html
[18] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_20newsgroups.html

Numéro	Classe	App.	Test	Total
1	*rec.sport.hockey*	**600**	**399**	**999**
2	*soc.religion.christian*	599	398	997
3	*rec.motorcycles*	598	398	996
4	*rec.sport.baseball*	597	397	994
5	*sci.crypt*	595	396	991
6	*rec.autos*	594	396	990
7	*sci.med*	594	396	990
8	*comp.windows.x*	593	395	988
9	*sci.space*	593	394	987
10	*comp.os.ms-windows.misc*	591	394	985
Total	–	**5 954**	**3 963**	**9 917**

Table 4 – Top-10 des classes du corpus 20-Newsgroups avec le nombre de documents de chaque classe

Par rapport à l'ensemble des données du corpus 20-Newsgroups, le top-10 des classes couvre 52,62 % des données (avec une même partition entre l'apprentissage et le test). Il est à noter que le corpus 20-Newsgroups que nous traitons décrit 20 classes.

7 Résultats et discussion

Dans cette section, nous présentons tout d'abord les mesures d'évaluation que nous avons utilisées (cf. sous-section 7.1) avant de présenter les résultats obtenus (cf. sous-section 7.2).

7.1 Mesures d'évaluation

Souvent, la précision (*P*), le rappel (*R*) et la F-Mesure (*F*) sont utilisés pour évaluer les performances des systèmes de classification automatique de textes. Nous avons aussi le taux d'exactitude (Taux), appelé souvent en anglais *accuracy*.

La précision permet de répondre à la question : Quelle proportion d'identifications positives est effectivement correcte ? Le rappel, quant à lui, répond à la question : Quelle proportion de résultats positifs réels est identifiée correctement ? La F-Mesure est une moyenne harmonique de la précision et du rappel. Elle permet de mesurer la capacité d'un système à donner toutes les solutions pertinentes et à refuser les autres. Enfin, le taux d'exactitude se fonde sur la distinction « *correct/incorrect* », toute nuance ou gradation exclues.

Formellement, la description mathématique de ces mesures pour une étiquette de classe donnée i est présentée dans les équations suivantes :

$$P_i = \frac{TP_i}{TP_i + FP_i} \quad (2)$$

$$R_i = \frac{TP_i}{TP_i + FN_i} \quad (3)$$

$$F_i = \frac{2 \times P_i \times R_i}{P_i + R_i} \quad (4)$$

$$Taux_i = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \quad (5)$$

Dans les équations, 4 variables sont à déterminer, à savoir : $\mathrm{TP}_i, \mathrm{FP}_i, \mathrm{FN}_i$ *et* TN_i.

- TP_i représente le nombre de documents correctement classés pour la i-ème classe (vrais positifs)
- FP_i représente le nombre de documents qui sont incorrectement classés en i-ème classe (faux positifs)
- FN_i est le nombre de documents qui appartiennent à la i-ème classe, mais qui sont incorrectement classés dans une classe négative (faux négatifs)
- TN_i est le nombre de documents correctement classés dans une classe négative, non i-ème classe (vrais négatifs)

Pour le calcul des mesures sur toutes les classes, la moyenne est obtenue comme suit :

$$P = \frac{\sum_{i=1}^{k} P_i}{k} \quad (6)$$

$$R = \frac{\sum_{i=1}^{k} R_i}{k} \quad (7)$$

$$F = \frac{2 \times P \times R}{P + R} \quad (8)$$

$$Taux = \frac{\sum_{i=1}^{k} Taux_i}{\mathrm{k}} \quad (9)$$

Dans ces équations, 'k' fait référence au nombre de classes.

7.2 Résultats d'évaluation

Nous avons évalué nos méthodes sur le corpus BL-News. Pour les méthodes *TermExtractClassif* et *Class2Class*, la validation croisée avec 10 *folds* est effectuée. Pour la méthode non supervisée *ClassEmbedd*, le test est effectué sur le corpus en intégralité. Les résultats obtenus sont présentés dans le tableau 5.

Système	Type d'évaluation	F	Taux
TermExtractClassif	Validation croisée	52	71
Class2Class	Validation croisée	98	99
ClassEmbedd	Corpus intégral	57	76

Table 5 – Résultats d'évaluation de nos méthodes sur le corpus BL-News

En comparant *TermExtractClassif* avec *ClassEmbedd*, il s'avère que l'utilisation des plongements lexicaux appliqués sur les termes-clés est plus bénéfique pour la classification que la simple fréquence du nombre de documents contenant les termes-clés. De plus, le test avec *ClassEmbedd* est effectué sur tout le corpus BL-News. Par ailleurs, la méthode *Class2Class* permet d'obtenir les meilleures performances, que ce soit pour la F-Mesure ou le taux d'exactitude. En effet, l'utilisation des forêts aléatoires avec plusieurs techniques de réduction de dimensionnalité a été bénéfique pour la classification. Concrètement, après avoir obtenu un ensemble de 96 489 caractéristiques avec TF-IDF, nous sommes passés à une

sélection des 4 000 meilleures caractéristiques. Ensuite, le seuil de validation nous a permis de sélectionner un ensemble de 1 516 caractéristiques. De plus, l'analyse discriminante linéaire a permis d'obtenir le meilleur temps d'exécution, en termes de rapidité (apprentissage et prédiction sur toute la chaîne de traitement), pour la méthode *Class2Class* par rapport aux autres méthodes. Toutefois, cette méthode est supervisée et exige d'avoir un corpus d'apprentissage représentatif. Nous tenons à préciser que le vocabulaire du corpus BL-News a une nature évolutive et un apprentissage statique n'est pas une bonne solution à long terme. Cela nous emmène à valider l'ensemble de nos méthodes sur des corpus de référence anglais pour lesquels le vocabulaire est bien représenté. Nous utilisons les corpus anglais Reuters-21578 et 20-Newsgroups. Les tableaux 6 et 7 décrivent les résultats obtenus sur ces deux corpus, en faisant une comparaison avec sept systèmes état-de-l'art décrits en section 2.

Système	**P**	**R**	**F**	**Taux**
TermExtractClassif	70	61	65	87
Class2Class	85	84	84	98
ClassEmbedd	37	39	28	19
Labani et al. (2018)	77,0	74,0	75,4	–
Jiang et al. (2018)	–	–	–	86,88
Kowsari et al. (2018)	–	–	**90,69**	–
Wu et al. (2017)	–	–	–	**99,07**
Jin et al. (2016)	–	–	88,60	96,50
Dogan et Uysal (2019) SVM	–	–	87,84	–
Dogan et Uysal (2019) Rocchio	–	–	82,20	–

Table 6 – Comparaison des résultats d'évaluation par l'utilisation du corpus Reuters-21578

Système	**P**	**R**	**F**	**Taux**
TermExtractClassif	60	70	65	91
Class2Class	82	82	82	97
ClassEmbedd	64	57	58	60
Labani et al. (2018)	50,10	66,40	57,10	–
Jiang et al. (2018)	–	–	–	83,33
Kowsari et al. (2018)	–	–	87,91	–
Wu et al. (2017)	–	–	–	94,98
Jin et al. (2016)	–	–	82,70	83,10
Jiang et al. (2011)	91,80	81,70	86,46	**98,72**
Dogan et Uysal (2019) SVM	–	–	**98,54**	–
Dogan et Uysal (2019) Rocchio	–	–	98,26	–

Table 7 – Comparaison des résultats d'évaluation par l'utilisation du corpus 20-Newsgroups

La méthode *ClassEmbedd* a été adaptée pour une application sur l'anglais. En effet, nous n'avons pas de lexiques associés aux classes des deux corpus anglais. Pour cela, l'*embedding* d'une classe est représenté seulement avec l'étiquette de la classe. Cela explique d'une certaine manière la dégradation des performances obtenues. Par ailleurs, les techniques de réduction de dimensionnalité dans la méthode *Class2Class* (TF-IDF + Seuil de validation) nous ont permis de passer de 341 233 caractéristiques à 4 000 puis à 1 349 pour le corpus Reuters-21578. Pour le corpus 20-Newsgroups, nous sommes passés de 72 9451 à 4 000 puis à 1 682. Cela permet de voir que le vocabulaire de base du corpus Reuters-21578 est plus riche que celui du corpus 20-Newsgroups. Les résultats que nous avons obtenus pour l'anglais nous confirment que l'utilisation de la méthode *Class2Class* permet de retourner de meilleures performances.

8 Conclusion

Dans cet article, nous nous sommes intéressés à la problématique de classification automatique de textes et plus particulièrement dans un cadre d'application métier. La finalité de ce travail est d'arriver à distinguer les documents traitant de l'innovation ou non et appartenant à l'un des domaines suivants : É*ducation*, *Gammes de Gestion*, *Médico-Social*, *Relation avec les Citoyens* et *Asset & Fleet*. Nous avons présenté trois méthodes de classification dont chacune prend en considération une représentation différente du contenu des documents. Deux des trois méthodes sont supervisées et reposent sur l'utilisation d'un corpus de documents annoté manuellement par des humains. Nous avons évalué nos méthodes sur un corpus français décrivant des articles provenant du web et collectés à partir d'un ensemble de sources prédéfinies. Nous avons appelé ce corpus BL-News pour faire référence aux articles d'actualités traitant en partie de l'innovation sur les secteurs métiers. Ensuite, dans un but de validation de l'efficacité de nos méthodes, nous avons effectué une étude comparative avec sept systèmes état-de-l'art sur deux corpus anglais de référence, à savoir Reuters-21578 et 20-Newsgroups. Les résultats que nous avons obtenus par l'application de nos méthodes sont significatifs et comparables aux systèmes état-de-l'art. Pour l'anglais, nos méthodes offrent de meilleures performances sur le corpus 20-Newsgroups par rapport aux corpus Reuters-21578. Cela vient du fait que la distribution des exemples d'apprentissage dans 20-Newsgroups est équilibrée sur l'ensemble des classes. Toutefois, notre méthode Class2Class reste performante peu importe le niveau de balancement des classes.

Ce travail nous a permis de proposer de nouvelles approches de classification automatique de textes. Même si nous avons validé ces approches sur les corpus Reuters-21578 et 20-Newsgroups, la nature et la taille du corpus BL-News sont totalement différentes. En effet, le corpus BL-News est beaucoup plus petit que les deux corpus anglais. De plus, même si les méthodes à base d'apprentissage supervisé ont apporté de bonnes performances, elles risquent de se retrouver en difficulté avec l'apparition de nouveaux articles ayant un vocabulaire métier plus poussé. Autrement dit, le vocabulaire de nos secteurs métiers est en constante évolution et utiliser seulement une méthode à base d'apprentissage supervisé ne suffit pas puisque plusieurs termes pertinents seront considérés comme des OOV (*Out of Vocabulary*). Pour faire face à ce problème, nous souhaitons combiner l'utilisation de nos méthodes supervisées avec la méthode non supervisée à base

de plongements lexicaux. Cela nous permettrait d'augmenter la base d'apprentissage, d'effectuer des réentraînements de modèles et de prendre de meilleures décisions dans un contexte évolutif.

Remerciements

Nous tenons à remercier toutes les personnes ayant contribué au projet dans sa globalité, à savoir : la collecte des articles, la campagne d'annotation et le développement des différentes méthodes de classification. Aussi, nous souhaitons remercier toute personne ayant contribué à la rédaction et à la relecture (de près ou de loin) de cet article.

Références

[1] F. Béchet, M. El-Bèze, et J. M. Torres-Moréno, En finir avec la confusion des genres pour mieux séparer les thèmes, *Atelier DEFT (Défi Fouille de Textes) - TALN2008*, p. 161-170, 2008.

[2] F. Boudin, pke: an open source python-based keyphrase extraction toolkit, *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, p. 69-73, 2016.

[3] L. Buitinck *et al.*, API design for machine learning software: experiences from the scikit-learn project, *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, p. 108-122, 2013.

[4] F. Cozman, I. Cohen, et M. Cirelo, Semi-Supervised Learning of Mixture Models, *Proceedings of the International Conference on Machine Learning (ICML)*, 2003.

[5] A. P. Das, K. Nayak, et M. Nayak, A Survey on Machine Learning Based Text Categorization, *The International Organization of Scientific Research (IOSR) Journal*, 2018.

[6] S. Dey Sarkar, S. Goswami, A. Agarwal, et J. Aktar, A Novel Feature Selection Technique for Text Classification using Naïve Bayes, *International Scholarly Research Notices*, p. 1-10, 2014, doi: 10.1155/2014/717092.

[7] A. Dhar, H. Mukherjee, N. S. Dash, et K. Roy, Text categorization: past and present, *Artificial Intelligence Review*, 2020, doi: 10.1007/s10462-020-09919-1.

[8] T. Dogan et A. K. Uysal, On Term Frequency Factor in Supervised Term Weighting Schemes for Text Classification, *Arabian Journal for Science and Engineering*, p. 9545-9560, 2019, doi: 10.1007/s13369-019-03920-9.

[9] S. K. Dwivedi et C. Arya, Automatic Text Classification in Information retrieval: A Survey, *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, 2016, doi: 10.1145/2905055.2905191.

[10] G. Feng, S. Li, T. Sun, et B. Zhang, A probabilistic model derived term weighting scheme for text classification, *Pattern Recognition Letters*, p. 23-29, 2018.

[11] G. Grouin, J. B. Berthelin, S. El Ayari, M. Hurault-Plantet, et S. Loiseau, Présentation de DEFT'08 (DÉfi Fouille de Textes), *Atelier DEFT (Défi Fouille de Textes) - TALN2008*, p. 1-10, 2008.

[12] D. S. Guru, M. Suhil, L. N. Raju, et N. V. Kumar, An alternative framework for univariate filter-based feature selection for text categorization, *Pattern Recognition Letters*, vol. 103, p. 23-31, 2018, doi: 10.1016/j.patrec.2017.12.025.

[13] M. Honnibal, I. Montani, S. Van Landeghem, et A. Boyd, spaCy: Industrial-strength Natural Language Processing in Python, *Zenodo*, 2020, doi: 10.5281/zenodo.1212303.

[14] J. Y. Jiang, R. J. Liou, et S. J. Lee, A Fuzzy Self-Constructing Feature Clustering Algorithm for Text Classification, *IEEE Transactions on Knowledge and Data Engineering*, p. 335-349, 2011, doi: 10.1109/TKDE.2010.122.

[15] M. Jiang *et al.*, Text classification based on deep belief network and softmax regression, *Neural Computing and Applications*, p. 61-70, 2018, doi: 10.1007/s00521-016-2401-x.

[16] P. Jin, Y. Zhang, X. Chen, et Y. Xia, Bag-of-Embeddings for Text Classification, *Proceedings of the International Joint Conference on Artificial Intelligence*, p. 2824-2830, 2016, doi: 10.5555/3060832.3061016.

[17] K. S. Jones, A statistical interpretation of term specificity and its application in retrieval, *Journal of Documentation*, vol. 28, p. 11-21, 1972.

[18] M. Kaur et V. Kumar, Optimization of Text Classification using Supervised and Unsupervised Learning Approach, *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 6, nº 4, p. 3385-3387, 2015.

[19] K. Kowsari, M. Heidarysafa, D. E. Brown, K. J. Meimandi, et L. E. Barnes, RMDL: Random Multimodel Deep Learning for Classification, *Proceedings of the International Conference on Information System and Data Mining*, 2018, doi: 10.1145/3206098.3206111.

[20] M. Labani, P. Moradi, F. Ahmadizar, et M. Jalili, A novel multivariate filter method for feature selection in text classification problems, *Engineering Applications of Artificial Intelligence*, p. 25-37, 2018, doi: 10.1016/j.engappai.2017.12.014.

[21] J. Lehmann *et al.*, DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia, *Semantic Web Journal, IOS Press*, vol. 6, nº 2, p. 167-195, 2015, doi: 10.3233/SW-140134.

[22] R. Nils et G. Iryna, Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

[23] C. Wenliang, C. Xingzhi, W. Huizhen, Z. Jingbo, et Y. Tianshun, Automatic Word Custering for Text Categorization Using Global Information, *Proceedings of the Asia Information Retrieval Symposium*, p. 1-11, 2005, doi: 10.1007/978-3-540-31871-2_1.

[24] H. Wu, X. Gu, et Y. Gu, Balancing between over-weighting and under-weighting in supervised term weighting, *Information Processing & Management*, vol. 53, nº 02, p. 547-557, 2017, doi: 10.1016/j.ipm.2016.10.003.

Session 6 – Approches multimodales

A Benchmarking on Cloud based Speech-To-Text Services for French Speech and Background Noise Effect

Binbin Xu[1*], Chongyang Tao[1+], Zidu Feng[1+], Youssef Raqui[2], Sylvie Ranwez[1*]

[1] EuroMov Digital Health in Motion, Univ Montpellier, IMT Mines Alès
[2] DiappyMed

[1*]firstname.lastname@mines-ales.fr [1+]firstname.lastname@mines-ales.org
[2]youssef.raqui@diappymed.com

Résumé

Alors que les applications de reconnaissance vocale se sont imposées dans notre quotidien, il existe peu d'études à grande échelle pour comparer les performances des solutions de l'état de l'art. Ceci est d'autant plus vrai dans une langue autre que la langue anglaise. Cet article propose une telle analyse comparative basée sur 17 heures d'enregistrement en Français. Quatre systèmes sont analysés : Google Cloud Speech-To-Text, Microsoft Azure Cognitive Services, Amazon Transcribe, et IBM Watson Speech to Text. Chacun ayant été mis à l'épreuve de cinq niveaux de bruit de fond, c'est l'équivalent de 400 heures de discours qui sont analysées. Microsoft Azure Cognitive Services a montré les meilleurs résultats en terme de taux d'erreur et une bonne résistance au bruit, tandis que la sensibilité au bruit d'IBM Watson Speech to Text compromet son usage en situation réelle.

Mots-clés

Reconnaissance vocale, Evaluation comparative, Langue Française, Google Cloud Speech-To-Text, Microsoft Azure Cognitive Services, Amazon Transcribe, IBM Watson Speech to Text

Abstract

This study presents a large scale benchmarking on cloud based Speech-To-Text systems : Google Cloud Speech-To-Text, Microsoft Azure Cognitive Services, Amazon Transcribe, IBM Watson Speech to Text. For each systems, $40\,158$ *clean and noisy speech files about* 101 *hours are tested. Effect of background noise on STT quality is also evaluated with 5 different Signal-to-noise ratios from* $40\,\mathrm{dB}$ *to* $0\,\mathrm{dB}$*. Results showed that Microsoft Azure provided lowest transcription error rate* 9.09% *on clean speech, with high robustness to noisy environment. Google Cloud and Amazon Transcribe gave similar performance, but the latter is very limited for time-constraint usage. Though IBM Watson could work correctly in quiet conditions, it is highly sensible to noisy speech which could strongly limit its application in real life situations.*

Keywords

Speech-To-Text, Benchmarking, French language, Google Cloud, Microsoft Azure Cognitive Services, Amazon Transcribe, IBM Watson

1 Introduction

Lots applications with automated speech recognition (ASR) or Speech-To-Text (STT) over the past few years have been developed to improve our daily life like personal voice assistant, or have been deeply integrated in many of business chains. Thanks to the substantial development of deep neural network (DNN), the performance of STT has been drastically improved. Like other deep neural network applications, today it is not surprising that in some situations, current STT can even outperform humans. The IBM/Appen human transcription study [1] showed that word error rate of human parity is about 5.1%. Microsoft Research is the first team reaching this milestone. However, the outstanding performances in DNN is based on large amount of labeled training data. This is also the case for DNN models on STT. For languages other than English, there's much less high quality audio data like in English. In consequence, the performances of STT on other languages are in general lower than for English, especially for languages featuring rich morphology like French.

Though many public Deep Neural Network models are available for offline use, retraining or regular updating require extensive computing power which prevents individuals or small business from accessing these models or using them in an efficient way. The choice will be the cloud-based API services. Actually, the most powerful STT systems are all cloud-based. Integrating these systems in an application or a product line requires at first a benchmarking on their performance. There exist many benchmarking studies on the performance of cloud-based STT services. However, they are often conducted with very small or small sample size, for example, 20–60 sentences or hundreds of sentences. The benchmarking on English from Picovoice is one of the few large scale tests on STT, which contains 2620 audio files (5h24m) from LibriSpeech dataset [2]. Benchmarking of cloud-based STT on French is even less stu-

died. Another major negative factor on STT performance is the background noise. Very often, only clean speech record is processed. However, for most real-life application, the background noise can hardly be avoided. This should be taken into account in STT benchmarking as well.
The objective of this study is to benchmark four most used Speech-To-Text API (Application Programming Interface) with a large French dataset : 6693 files, about 17hours speech record. Five levels of common background noise are added in the clean speech and evaluated additionally. In total, more than 400 hours speech are transcribed.

2 Speech-To-Text system and data

2.1 Cloud based STT services

Four cloud based Speech-To-Text services are evaluated in this work :

— Amazon Transcribe, is part of Cloud Computing Services from the Amazon Web Services (AWS) which holds currently the largest share in Cloud Computing market. Their recent speech recognition model on English reached State-of-the-Art word error rate at 6.2% [3]. To convert speech files to text, the data needs to be at first uploaded to Amazon Simple Storage Service (Amazon S3). Then Transcribe call the objects from S3 for transcription. Though Transcribe jobs can be treated on batch mode (up to 100 parallel jobs). This S3 requirement adds additional complexity for the transcription tasks. Actually Amazon Transcribe is the only STT requiring storage. The other three services can be feed directly with audio files.

— Google Cloud Speech-to-Text, is integrated in the widely used platform Google Cloud. In 2012, Google Research had achieved word error rate at 15% for English broadcast news transcription. This error rate dropped considerably to 5.6% with updated model trained on over 12 500 hours audio in 2018 [3]. Their STT model is one of the most powerful in the market, and the performance is continuously improving.

— IBM Watson Speech-to-Text. IBM Watson is a conventional top player in speech recognition. In 2015, their speech recognition system beat other models with a word error rate at 8% [4]. Two years later, their system reached 5.5% [1]. It's now among the most popular STT services and provides similar features as other cloud STT.

— Microsoft Azure Cognitive Services. Microsoft's speech recognition is now one of the leading STT service. In 2017, their model reached a historical human parity milestone on conversational telephony speech transcription, with 5.1% word error rate in benchmarked Switchboard task [5]. As all the other STT systems, Microsoft's STT system is also integrated in the Cloud Computing platform.

All the four STT services offer the possibility to customize (like domain-specific) or retrain the Speech-to-Text models. However, since they are all black-boxed APIs, the background models and architectures are unknown, it is difficult to benchmark the customized models with different configurations in a fair way. So, only the basic models (APIs) are called.

2.2 Speech corpus

The basic audio dataset in this work is from WCE-SLT-LIG [6, 7]. This corpus contains 6693 speech utterances recorded by 42 native speakers.

FIGURE 1 – Main topic of WCE-SLT-LIG corpus

They come from French news media with main topic on European economy. The total audio duration is 16h52. The ground-truth transcriptions are also available, which makes the benchmarking possible.
The number of word in this corpus is 22 ± 12.8 (median $\pm$ standard deviation), with audio duration 8.4 ± 4.6 seconds as shown in Figure 2.

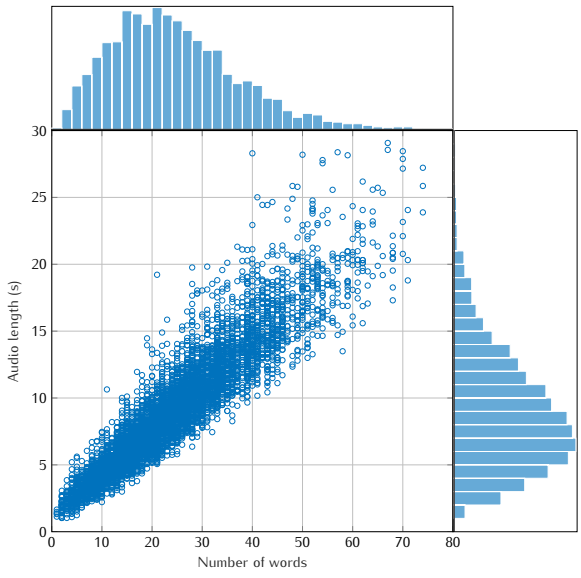


FIGURE 2 – Distribution of WCE-SLT-LIG corpus

2.3 Environmental noise corpus

In real world cases, most speech takes place in noisy environments. This is one of the main challenges in Speech-to-Text applications. To evaluate the effects on the STT quality, we introduce another recently released environmental noise dataset : Microsoft Scalable Noisy Speech Dataset (MS-SNSD) [8]. The dataset provides a variety of common environmental noise, which can be mixed on clean speech data. The signal-to-noise (SNR) in dB can be configured as well.

$$\mathrm{SNR}_{\mathrm{dB}} = 10 \log_{10} \left(\frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}} \right).$$

where P_{signal}, P_{noise} are the power of signal and background noise. We set 5 SNR cases here : 40 dB, 30 dB, 20 dB, 10 dB and 0 dB (1 :1 signal vs. noise).
The raw MS-SNSD contains 181 noise files. However, many of them are recorded with strong conversations in other languages (English, German etc.). Some of noise type are also less common. So, these noises are excluded. We'd like to evaluate the effect of noise type on the performance of STT, to make sure that some types are not overpresented, 96 noise files in 18 types are kept.

AirConditioner	Kitchen	SqueakyChair
AirportAnnouncements	LivingRoom	Station
Babble	Munching	Traffic
Cafe	Restaurant	Typing
CafeTeria	ShuttingDoor	VacuumCleaner
CopyMachine	Square	WasherDryer

TABLE 1 – Types of background noise used in this work. 96 noises in 18 types

2.4 Evaluation metrics

In Speech-to-Text, the most commonly used metric to evaluate the performance is WORD ERROR RATE (WER). Other metrics exist, like MATCH ERROR RATE (MER); WORD INFORMATION LOST (WIL) or WORD INFORMATION PRESERVE (WIP) [9].

$$WER = \frac{S + D + I}{N1 = H + S + D} \quad (1)$$

$$MER = \frac{S + D + I}{N = H + S + D + I} \quad (2)$$

$$WIP = \frac{H}{N_1} \cdot \frac{H}{N_2} \cong \frac{I(X,Y)}{H(Y)}, \quad (3)$$

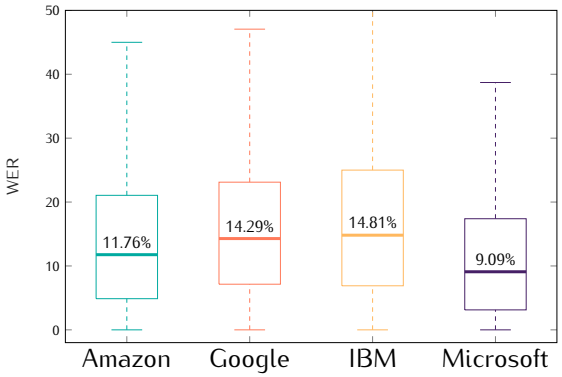
$$WIL = 1 - WIP \quad (4)$$

where H, S, D and I correspond to the total number of word hits, substitutions, deletions and insertions. N_1 and N_2 are respectively the number of words in ground-truth text and the output transcripts. The lower are WER, MER and WIL, the better the performance is.

3 Results

3.1 Clean speech

For clean speech, Microsoft Azure performed quite well, with a WER at 9.09% which is close to the advertised rate. Amazon Transcribe took the second place with WER 11.76%. Google Cloud and IBM Waston gave similar WER (14.29% and 14.81%). These WER are actually very good already. According to the public DeepSpeech model [10] from Mozila, trained with a mixed French dataset "CommonVoice + CssTen + LinguaLibre + Mailabs + Tatoeba + Voxforge", the WER on test dataset is 19.5% (result retrieved on March 10th 2021) [11]. The gain with cloud STT API is between $24\% - 53\%$.



	Amazon	Google	IBM	Microsoft
WER	11.76%	14.29%	14.81%	**9.09%**
MER	11.54%	14.29%	14.29%	**9.09%**
WIL	0.19	0.25	0.24	**0.16**

TABLE 2 – Evaluation on clean audio. Upper, WER distributions; lower, median values. (STTs ACCESSED IN FEBRUARY 2021)

3.2 Noisy speech

After mixing five different levels of environmental noise, Microsoft Azure gave a quite good global WER 11.11% (Tableau 3). Amazon Transcribe and Google Cloud showed the same WER at 20%. But IBM Waston failed at certain point. Its global WER is 29.63%, with a word-information-lost rate at 43% (0.43) which is unfortunately high.

	Amazon	Google	IBM	Microsoft
WER	20.00%	20.00%	29.63%	**11.11%**
MER	19.64%	20.00%	28.57%	**11.11%**
WIL	0.31	0.33	0.43	**0.19**

TABLE 3 – Evaluation on all noisy audio (5 SNR levels combined), median values

At individual SNR level, as shown in Figure 3, Microsoft Azure is the most robust to noise. The variation across different noise levels is quite small. In highly noisy environment, the WER from transcription by IBM Waston can be more than 100%. While other STTs would be at worst less than 50%.
The exceptional STT performance of Microsoft Azure is due to that Microsoft has been working intensively on Artificial Intelligence based noise suppression. This environmental noise dataset MS-SNSD comes from Microsoft. The noise suppression should be already in the pipeline of their

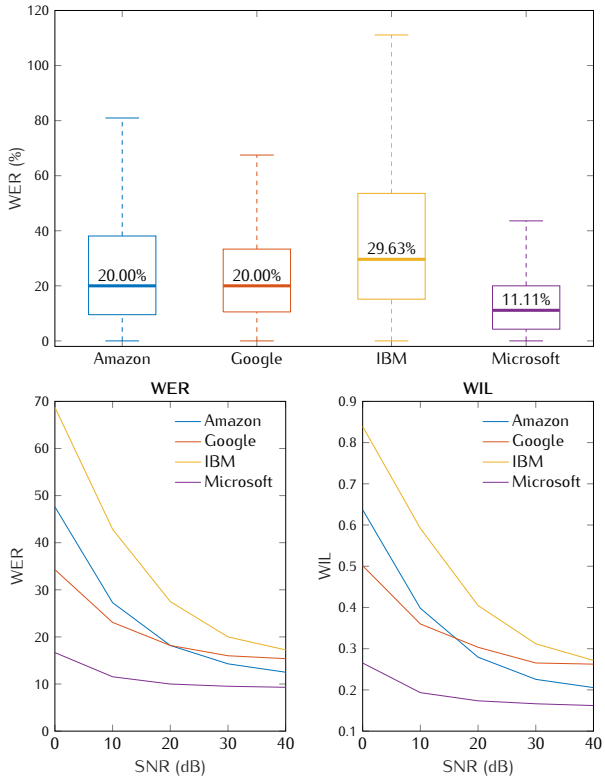


FIGURE 3 – Evaluation on mixed noisy speech by five signal-noise-ratio levels; upper, wer distributions; lower, median value for each level. (STTs ACCESSED IN FEBRUARY 2021)

Speech-to-Text models. Actually, in December 2020 Microsoft introduced background noise suppression functionality in Microsoft Teams meetings [12]. To achieve this, they used 760 hours of clean speech data and 180 hours of noise data. These data are now released for Interspeech 2021 Deep Noise Suppression Challenge [13].

The performance of STT depends also on the noise types. All the STT services are sensible to noise type *Restaurant*. IBM Waston's WER reached 46.51%; Amazon Transcribe had also high WER for this type of noise. Google Cloud and Microsoft Azure dealt it better without shape WER changes. Background noise in environment *Restaurant* could be a mixture of different noises (babble, conversation, munching, traffic etc.) which make it be more difficult for Speech-to-Text tasks. In general, Google Cloud and Microsoft Azure are more robust to environmental noise (variation and standard deviation of the median WER are 6.5% and 2.6% for Google Cloud; 1.4%, 1.2% for Microsoft Azure); Amazon Transcribe can be placed in the second rank with 24% and 4.9%. As for IBM Waston, as shown previously, it can fail in many cases when the background noises are too strong. It suffered also strong performance variation 53.7% and 7.3% of standard deviation of the median WER.

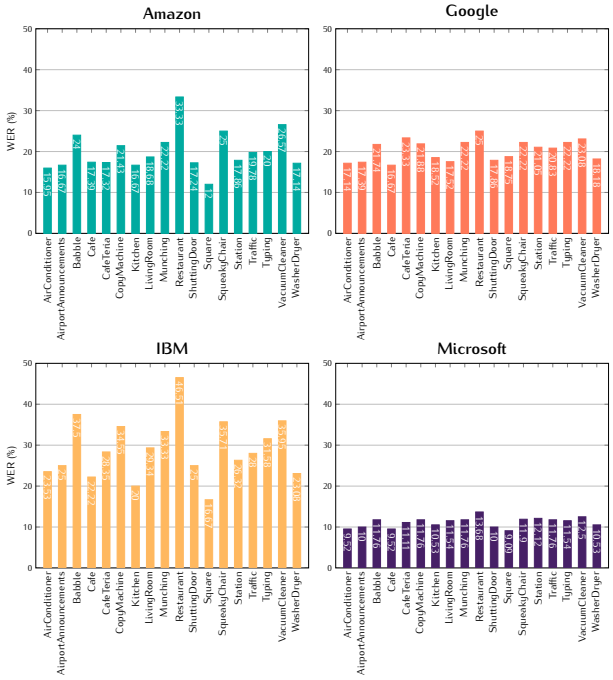


FIGURE 4 – WER by different noise types; median WER values from tests on all the five SNR noisy speech. (STTs ACCESSED IN FEBRUARY 2021)

3.3 Main STT errors

The main source of errors contributed to WER is the substitution. For clean speech, or less noisy speech, the percentage of substitution S is generally much higher than deletion D and insertion I. When speech becomes highly noisy (SNR lower than $10\,\mathrm{dB}$), deletion D percentage increased much more. STT service from Microsoft Azure is quite robust to noisy environment, there's practically no change for SNR from $40\,\mathrm{dB}$ to $10\,\mathrm{dB}$. Only in the tested case when mixing directly noise and speech $0\,\mathrm{dB}$, the deletion D and substitution S increased slightly. However, the changes are much more significant for other three STT services, especially for IBM Waston.

There's also inter-speakers difference of WER. Amount the 42 speakers, all the four STTs had more difficulty to transcribe speech from speaker L23_P08.

3.4 Transcription job time

In a production application, the STT service must be as responsive as possible. Google Cloud is the fastest about the four tested APIs, with a median value at 1.76 second par job. Microsoft Azure is also fast, 3.51 second per transcription job. IBM Waston is slower and require 5.43 second to complete the job. It's not surprising that Amazon Transcribe is the slowest STT service, with 27 second per job. Some transcription jobs can take up to 200 second. Even it's possible to send up to 100 jobs in parallel, single job waiting is not acceptable for any real world application. This time requirement does not include the data transfer time to Amazon S3 storage : with upload speed 100-700 kbps, for a

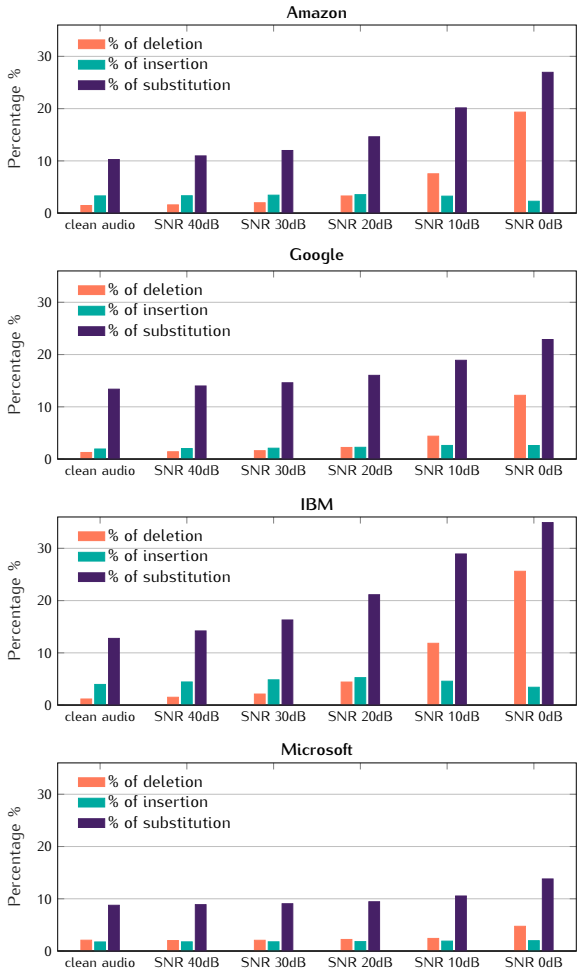


FIGURE 5 – Main transcription errors distribution (mean values. The median percentage values for lower SNR are zero, less meaningful for presentation)

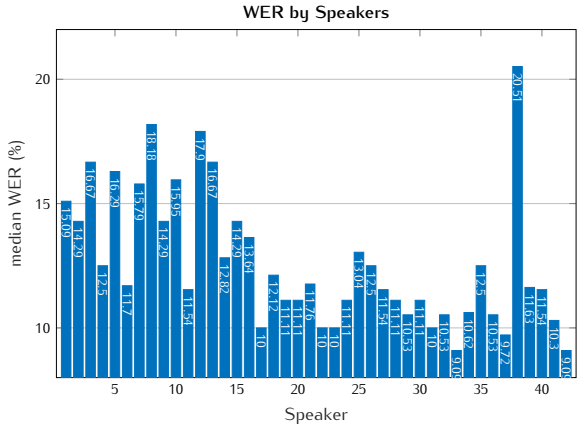


FIGURE 6 – Word Error Rate by speakers (median values) on clean speech for all the four STT systems.

large amount of data, this can take already quite some time to complete. Though it's possible to call Amazon Transcribe for steaming usage, it's not convenient for non-real-time scenario.

One of the potential reasons of the additional seconds from Google Cloud and IBM Waston, could be that Microsoft Azure's returns less complete transcription information than the other three.

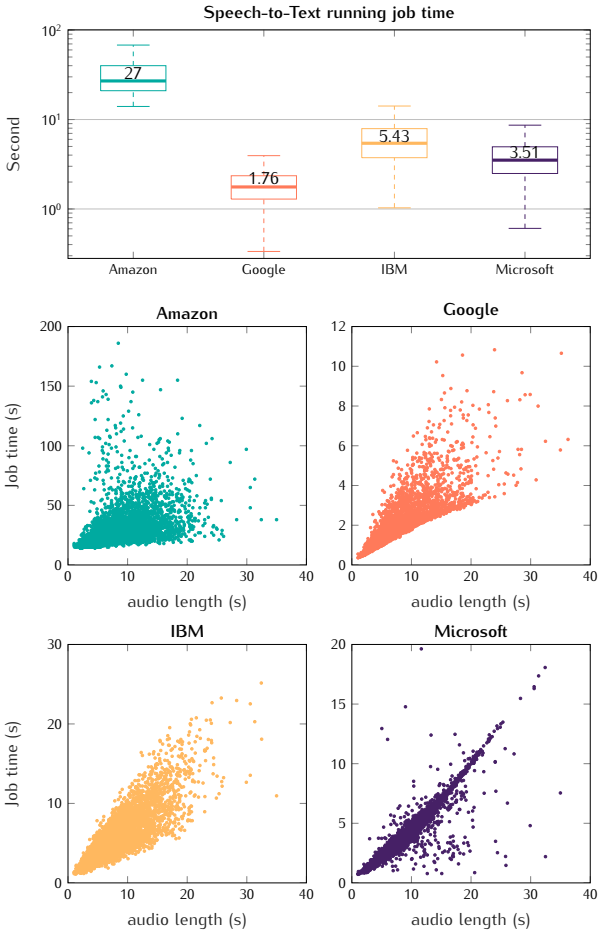


FIGURE 7 – Transcriptio job time in second for all the four STT systems

Another observation is the server responsiveness : job completion time with Microsoft Azure is almost linear to the audio duration. The variation is also very tight. But for other APIs, though the relationship could be regarded as linear, the variation is much larger. Speeches with same length would require 2 to 4 times more execution time to complete the task.

4 Discussion

In this work, we evaluated the four most used Speech-to-Text API on French speech from four Computing Cloud : Amazon Transcribe, Google Cloud, IBM Waston and Microsoft Azure. 5 levels of different environment noises are mixed with 6690 clean speeches (17 hours). 100 hours speech tests per STT API gave 400 hours speech transcription.

The results showed that Microsoft Azure's STT service provided the lowest Word Error Rate (median 9%). It's also very robust to common environment noise, even in strong noise environment, the median WERs are only around 16.67%. STT from Amazon Transcribe and Google Cloud performed well, their WER are respectively at 11.76% and 14%. Amazon Transcribe works better in relatively quiet environment while Google Cloud is better for noisy speech. IBM Waston's STT service can provide reasonable results with a median WER at 14.29%. But when the speech is recorded in noisy environment, the WER can go up to around 70% which is difficult to be used. In general, when the signal-to-noise ratio is higher than $20\,\mathrm{dB}$, the WERs are still acceptable. However, if SNR drops lower than $20\,\mathrm{dB}$, except Microsoft Azure, all the three APIs will have difficulties to recognize correctly the speech. Among the 18 environment noise types, Restaurant type is the most difficult one to deal with for all the four STT APIs.

When the work is time-constraint, Google Cloud will be the first choice with fastest response time and a reasonable word error rate. Amazon Transcribe can be used when the framework of the project is on the platform of Amazon Web Services. The parallel job can help to reduce the total transcription time, however, per job time is too longer than any other STT service. In average, one transcription job on Amazon Transcribe is 15 times longer than the same job on Google Cloud. Otherwise, the general suggestion will be Microsoft Azure, lowest WER and high robustness to noise. It's more suitable for precision-constraint applications.

References

[1] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L.-L. Lim, *et al.*, "English conversational telephone speech recognition by humans and machines," *arXiv preprint arXiv :1703.02136*, 2017. [Online]. Available : https://arxiv.org/abs/1703.02136

[2] Picovoice, "Speech-to-text benchmark," *GitHub*, 2020. [Online]. Available : https://github.com/Picovoice/speech-to-text-benchmark

[3] C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 4774–4778.

[4] G. Saon, H.-K. J. Kuo, S. Rennie, and M. Picheny, "The ibm 2015 english conversational telephone speech recognition system," *arXiv preprint arXiv :1505.05899*, 2015. [Online]. Available : https://arxiv.org/abs/1505.05899

[5] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "The microsoft 2017 conversational speech recognition system," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 5934–5938.

[6] L. Besacier, B. Lecouteux, N.-Q. Luong, K. Hour, and M. Hadj Salah, "Word confidence estimation for speech translation," in *International Workshop on Spoken Language Translation*, Lake Tahoe, United States, Dec. 2014.

[7] N.-T. Le, B. Lecouteux, and L. Besacier, "Joint asr and mt features for quality estimation in spoken language translation," in *International Workshop on Spoken Language Translation*, Seattle, United States, Dec. 2016.

[8] C. K. Reddy, E. Beyrami, J. Pool, R. Cutler, S. Srinivasan, and J. Gehrke, "A scalable noisy speech dataset and online subjective test framework," in *Proc. Interspeech 2019*, 2019, pp. 1816–1820.

[9] A. C. Morris, V. Maier, and P. Green, "From wer and ril to mer and wil : improved evaluation measures for connected speech recognition," in *Eighth International Conference on Spoken Language Processing*, 2004.

[10] A. Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech : Scaling up end-to-end speech recognition," *CoRR*, vol. abs/1412.5567, 2014.

[11] Jaco-Assistant, "Deepspeech-polyglot," *GitLab*, 2021. [Online]. Available : https://gitlab.com/Jaco-Assistant/deepspeech-polyglot

[12] Microsoft, "Reduce background noise in microsoft teams meetings with ai-based noise suppression," 2020. [Online]. Available : https://techcommunity.microsoft.com/t5/microsoft-teams-blog/reduce-background-noise-in-microsoft-teams-meetings-with-ai/ba-p/1992318

[13] C. K. Reddy, H. Dubey, K. Koishida, A. Nair, V. Gopal, R. Cutler, S. Braun, H. Gamper, R. Aichner, and S. Srinivasan, "Interspeech 2021 deep noise suppression challenge," *arXiv preprint arXiv :2101.01902*, 2021. [Online]. Available : https://arxiv.org/abs/2101.01902