Actes des 21es Rencontres des Jeunes Chercheurs en Intelligence Artificielle

Brian Ravenet

AfIA

Association française
pour l'Intelligence Artificielle

RJCIA

Rencontres des Jeunes Chercheurs
en Intelligence Artificielle

PFIA 2023

Table des matières

Brian Ravenet
Éditorial 5

Comité de programme 6

P. Quentel, Y. Kermarrec, L. Grivault, P. Le Berre, L. Savy
Approche multi-agent pour la collaboration multi-plateforme dans un contexte de défense navale 7

Arthur Baudet, Oum-El-Kheir Aktouf, Annabelle Mercier, Philippe Elbaz-Vincent
MAKI : Une infrastructure à clefs publiques pour systèmes muti-agents 16

J. Ferry, U. Aïvodji, S. Gambs, M.-J. Huguet, M. Siala
Exploiter l'équité d'un modèle d'apprentissage pour reconstruire les attributs sensibles de son ensemble d'entraînement 24

N. Boudegzdame, K. Sedki, R. Tsopra, and JB. Lamy
SMOTE : Apprenons-nous à classifier ou à prédire la nature synthétique des données ? 34

Q. Mazouni, H. Spieker, A. Gotlieb, M. Acher
A Review of Validation and Verification of Neural Network-based Policies for Sequential Decision Making 43

N. Mokhtari, A. Nédélec, M. Gilles, P. De Loor
Amélioration de la recherche d'architecture neuronale en combinant un algorithme FireFly avec une évaluation sans apprentissage 53

F. Hamdani, D. Monticolo, V. Boly
Étude de l'apport de l'Intelligence Artificielle pour l'innovation de produit 63

Théo Duchatelle
Encodage Logique et Explications Visuelles pour l'Argumentation 72

J. Gonzalez, F. Dama
Génération de données synthétiques à partir d'une forêt aléatoire 75

A. Godinot, E. Le Merrer, C. Penzo, F. Taïani, G. Tredan
Change-Relaxed Active Fairness Auditing 78

Maya Medjad, Mathieu Buonomo, Raphaël Szymocha, Frédéric Armetta
Un modèle pour la généricité des agents conversationnels vocaux multi-domaines 84

J. Michel, P. Parrend
Metrics for community dynamics applied to unsupervised attacks detection 87

Majed Jaber, Nicolas Boutry, Pierre Parrend
Structural and spectral analysis of dynamic graphs for attack detection 90

X. Wang, F. Meyer, P. Kuntz
Formalisation de la classification multi-Labels en flux et son application en cybersécurité : une étude approfondie 93

Julien Soulé, Jean-Paul Jamont, Michel Occello, Paul Théron, Louis-Marie Traonouez
De l'Organisation des Systèmes Multi-Agents de Cyber-defense 96

J-B. Ly, Q. Reynaud, C. Le Bail, M. Schumann, V. Boccara, N. Sabouret
Dans quelle mesure les simulations informatiques de l'activité humaine sont-elles réalistes ? 99

K. Khoussa, Y.-A. Chapuis, N. Lachiche
Optimisation de matériaux et dispositifs pour l'énergie à partir de concepts d'intelligence artificielle pour small data 108

Sébastien Bertrand, Silvia Ciappelloni, Pierre-Alexandre Favier, Jean-Marc André
Replication and Extension of Schnappinger's Study on Human-level Ordinal

Maintainability Prediction Based on Static Code Metrics 117

Éditorial

Rencontres des Jeunes Chercheurs en Intelligence Artificielle

Les RJCIA sont destinées aux jeunes chercheurs en IA, doctorants ou titulaires d'un doctorat depuis moins d'un an. L'objectif de cette manifestation est double :

permettre aux jeunes chercheurs préparant une thèse en Intelligence Artificielle, ou l'ayant soutenue depuis peu, de se rencontrer et de présenter leurs travaux, et d'ainsi former des contacts avec d'autres jeunes chercheurs et d'élargir leurs perspectives en échangeant avec des spécialistes d'autres domaines de l'intelligence artificielle, et former les jeunes chercheurs à la préparation d'un article, à sa révision pour tenir compte des observations du comité de programme, et à sa présentation devant un auditoire de spécialistes, leur permettant ainsi d'obtenir des retours de chercheurs de leur domaine ou de domaines connexes. Thèmes de la conférence Toute contribution relevant de l'Intelligence Artificielle est la bienvenue. Les contributions pourront s'inscrire dans la liste (indicative et non exhaustive) de thématiques suivante :

— recherche heuristique et résolution de problèmes,
— incertitude et intelligence artificielle,
— logique, satisfiabilité et satisfaction de contraintes,
— apprentissage automatique,
— extraction, ingénierie et gestion des connaissances,
— représentation des connaissances et raisonnement,
— planification, contrôle,
— aide à la décision,
— agents autonomes et systèmes multi-agents,
— reconnaissance des formes et vision par ordinateur,
— traitement automatique des langues naturelles,
— interaction avec l'humain,
— robotique,
— IA et web,
— environnements Informatiques d'Apprentissage Humain et apprentissage à distance,
— IA responsable,
— explicabilité,
— certification, éthique et IA.

Brian Ravenet

Comité de programme

Présidence

— Brian Ravenet (LISN-CNRS, Université Paris-Saclay, France).

Membres

— Mathieu Chollet (IMT Atlantique, University of Glasgow, Ecosse) ;
— Maxime Devanne (IRIMAS, Université Haute-Alsace, France) ;
— Madeleine El-Zaher (LINEACT, CESI, France) ;
— Mireille Fares (ISIR, Sorbonne Université, France) ;
— Maxime Folschette (CRIStAL, Centrale Lille, France) ;
— Jules Françoise (LISN-CNRS, Université Paris-Saclay, France) ;
— Sahar Ghannay (LISN-CNRS, Université Paris-Saclay, France) ;
— Maxime Guériau (LITIS, INSA Rouen Normandie, France) ;
— Camille Guinaudeau (LISN-CNRS, Université Paris-Saclay, France) ;
— Marie-Jeanne Lesot (LIP6, Sorbonne Université, France) ;
— Guillaume Lozenguez (Center Digital Systems, IMT Lille Douai, France) ;
— Jean-Guy Mailly (LIPADE, Université de Paris, France) ;
— Florian Pecune (SANPSY, Université de Bordeaux, France) ;
— Charlotte Pelletier (IRISA, Université Bretagne Sud, France) ;
— Mohamed-Lamine Messai (ERIC, Université Lumière Lyon 2, France) ;
— Arianna Novaro (Université Paris 1 Panthéon-Sorbonne, France) ;
— Nicolas Verstaevel (IRIT, Université de Toulouse, France) ;
— Genane Youness (LINEACT, CESI, France).

Approche multi-agent pour la collaboration multi-plateforme dans un contexte de défense navale

P. Quentel[1,2], Y. Kermarrec[1], L. Grivault[2], P. Le Berre[2], L. Savy[2]

[1] IMT Atlantique, Lab-STICC, F-29238 Brest, France

[2] Thales Defence Mission Systems, Brest et Elancourt, France

Mai 2023

Résumé

Dans la littérature, l'allocation des tâches à diverses entités a été largement étudiée. Cette problématique apparait également dans le cas des études collaboratives des systèmes de senseurs navals où il est question de pouvoir allouer des tâches à des senseurs en se basant sur des services. Les caractéristiques d'autonomie, de communication, d'intelligence et de distribution des agents font du système multi-agent (SMA) un choix judicieux dans le cadre de l'affectation de tâches senseurs dans un réseau multi-plateforme. Cet article présente en premier lieu le cadre complexe du combat collaboratif naval. Ensuite, nous proposons des modèles d'agents : l'agent plateforme, l'agent service et l'agent tactique. Enfin, nous proposons de faire un point sur l'avancée pratique de nos concepts et sur les méthodes d'évaluation que nous souhaitons appliquer pour valider ces concepts au regard de nos besoins.

Mots-clés

Planification multi-agent, communication, systèmes multi-senseurs, négociations multi-agents

Abstract

In the literature, the allocation of tasks to various entities has been widely studied. This problem also appears in the case of collaborative studies of naval sensor systems, where the question is to be able to allocate tasks to sensors based on services. The characteristics of autonomy, communication, intelligence and distribution of the agents make the Multi-Agent System (MAS) a wise choice in the context of sensors task allocation in a multi-platform network. Firstly, this paper presents the complex framework of naval collaborative combat. Then, we propose agent models : the platform agent, the service agent and the tactical agent. Finally, we propose to show the practical progress of our concepts and the evaluation methods we wish to apply to validate these concepts with respect to our needs.

Keywords

Multi-agent planning, communication, multi-sensor systems, multi-agent negotiations

1 Introduction

Cette introduction permet de comprendre les enjeux liés au contexte d'application des travaux présentés et de saisir la problématique qui en résulte.

1.1 Contexte

L'évolution du contexte de défense aéronaval nécessite une modification majeure des architectures des systèmes de senseurs. Depuis dix ans, la direction générale de l'armement (DGA) a amorcé des recherches et financé des programmes dans le but d'améliorer les capacités opérationnelles des systèmes de combat. De nos jours, les systèmes de senseurs coopèrent, les données des senseurs sont partagées entre les plateformes par le système de gestion de combat (CMS : *Command Mission System*). Le CMS de chaque plateforme permet la gestion des senseurs locaux ainsi que le suivi des pistes et des objets de la zone à surveiller. Chaque plateforme maintient une situation tactique locale à l'aide de ses senseurs, ainsi qu'une situation tactique globale grâce à l'échange d'information avec les autres plateformes au moyen de liaisons de données tactiques (LDT).

Depuis quelques années, les senseurs tendent à devenir des systèmes complexes [20], capables de partager de la donnée, de communiquer et de collaborer. Dans le cadre de notre étude, nous nous intéressons aux senseurs de guerre électronique (GE), mais aussi aux radars et aux capteurs optroniques. Nous détaillerons la notion de *senseur* en section 3. Les besoins technico-opérationnels de notre architecture concernent la détection, la localisation et le pistage de menaces.

De multiples schémas de partage de l'information ont été implémentés dans des systèmes de senseurs aéroportés. Cependant, d'après Beal et al. [1], l'affectation de tâches à des senseurs dans le but de collecter ces informations est difficile et doit, en général, être réalisée manuellement.

1.2 Problématique

L'intelligence artificielle distribuée (IAD) est une branche de l'IA utilisée pour la résolution de problèmes complexes qui, souvent, ne peuvent pas être gérés par un système centralisé [11]. Les algorithmes d'IAD sont classés en trois catégories : l'IA parallèle, la résolution de problèmes distri-

bués (DPS) et les systèmes multi-agents (SMA) [5]. Wooldridge, dans son ouvrage [26] qui introduit les SMA, caractérise l'agent BDI (*Belief, Desire, Intention*), purement communicant, comme un système informatique qui est situé dans un environnement, et qui peut exécuter des actions sur l'environnement avec autonomie, ceci afin d'accomplir un objectif défini. Le système multi-agent comprend des agents intelligents qui peuvent communiquer entre eux à l'aide de protocoles de communication.
Dans le cadre du combat collaboratif naval, nous souhaitons concevoir une architecture qui associe des senseurs hétérogènes de différentes plateformes pour effectuer des actions collaboratives, et ce de manière distribuée. La recherche sur les SMA met en avant les agents intelligents comme métaphore naturelle pour traiter de la résolution distribuée de problèmes complexes [13]. Ainsi, l'approche multi-agent rendrait possible l'allocation dynamique de tâches à des senseurs en réseau dans un contexte de collaboration multi-plateforme et multi-senseur. Cet article présente des concepts d'agents permettant de remplir cet objectif.
Les articles [16, 22, 23, 24, 25] ont étudié la problématique de l'allocation de tâches à des agents, cependant dans notre cas ce sont les agents qui proposent des tâches à nos senseurs qui sont alors considérés comme des ressources. L'allocation est un procédé qui fait correspondre des tâches à un ensemble de senseurs et qui tente de maximiser une utilité globale. L'objectif de nos travaux est d'optimiser l'utilisation des senseurs dans un contexte dense en nombre de menaces, il est donc nécessaire de sélectionner les tâches qu'ils effectueront en considérant des critères et des priorités.
Dans cet article, nous définissons le contexte dans lequel s'applique le besoin d'allocation de tâches à des senseurs, puis nous présentons des concepts d'agents permettant de répondre à ce besoin.
La section 2 de l'article présente des travaux relatifs à l'architecture des systèmes de senseurs et à l'allocation de tâches dans les SMA. Ensuite, nous formalisons le problème de l'allocation de tâches dans le contexte de défense naval en section 3. La section 4 présente notre conception des agents. Le développement de ces concepts et les méthodes choisies pour les évaluer sont présentés en section 5. Enfin, nous concluons en section 6 sur les travaux effectués et détaillons quelques pespectives à nos travaux.

2 Travaux connexes

Des travaux sur l'architecture des systèmes de senseurs et sur l'allocation de tâches dans les SMA sont présentés dans cette section.

2.1 Architecture

Actuellement, dans le domaine du combat aéronaval, chaque senseur est associé à une ou plusieurs fonctions qu'il peut accomplir. Dans l'architecture existante, les fonctions sont utilisées indépendamment les unes des autres et leur coordination est laissée à une initiative humaine.
Un travail de thèse sur l'architecture pour l'ordonnancement de systèmes multi-senseurs [7] utilise les concepts des SMA. Un agent est créé lors de la détection d'un objet sur le théâtre des opérations. Celui-ci assure le suivi et la collecte des données relatives à l'objet détecté (sa position, sa vitesse, etc.). Ces agents représentent la situation tactique globale et proposent des planifications de tâches senseurs dans le temps. Cette architecture est fonctionnelle dans un cadre mono-plateforme, notre approche étend cette solution à un ensemble de plateformes multi-milieux (aérien, terrestre et maritime).

2.2 L'utilisation des SMA pour l'allocation de tâches

Beal et al. [1] décrivent un système à base d'agents pour allouer des tâches aux senseurs dans le but de réduire le nombre de plateformes (ici des drones) pour accomplir un objectif. Des concepts d'agents sont présentés ainsi qu'un algorithme d'allocation de tâches.
Les auteurs proposent deux types d'agents : les agents *plateformes* et les agents *tâches*. Un agent *plateforme* est créé pour chaque ressource d'une plateforme (hypothèse : une seule ressource par plateforme). La particularité de cet agent est qu'il se décline en plusieurs agents *projetés*, en prenant en compte la position future anticipée de la plateforme. Un agent est créé pour chaque tâche et il communique avec les agents *plateformes* à portée de communication pour déterminer quelle plateforme accomplira la tâche. L'algorithme d'allocation de tâches prend en compte deux métriques afin de classer les tâches proposées : le nombre de plateformes qui couvrent déjà la tâche et la priorité de la tâche. Un budget correspondant au temps de disponibilité du senseur est fixé, les tâches assignées sont prises dans l'ordre d'importance et dans la limite de ce budget. Les travaux restent tout de même limités au partage de la ressource de senseurs hautement directifs (caméras embarquées par exemple). Les auteurs démontrent qu'il est possible d'utiliser moins de plateformes pour effectuer le même nombre de tâches en se servant uniquement du partage de la ressource disponible. Les senseurs ne collaborent donc pas dans le but de proposer de nouvelles fonctionnalités.
Ponda et al. utilise l'allocation de tâches afin que des agents hétérogènes dans un environnement dynamique puissent effectuer des missions [16]. Ces missions impliquent l'exécution de différentes tâches comme la reconnaissance, la surveillance, la classification de cibles, et les opérations de secours. Certains agents, des UAV (*Unmanned Aerial Vehicules*), peuvent effectuer des tâches différentes. Pour des cas avec un nombre important d'agents, les approches centralisées deviennent rapidement impossibles à mettre en œuvre en raisons des conflits d'allocations, de la complexité et du nombre d'interactions, il est donc nécessaire d'adopter des architectures décentralisées.
Dans l'article [25], le problème d'allocation de tâches s'intéresse plus particulièrement à une tâche unique, avec un seul robot, et des affectations à durée prolongée. Un agent effectue une tâche à la fois, et chaque agent peut se voir affecter plusieurs tâches dans un planificateur. Trouver la solution optimale à cette allocation de tâches dans un environnement temps-réel devient infaisable au niveau des calculs

à mesure que le nombre de tâches et d'agents augmente.
Des approches pour l'allocation de ressources sont étudiées dans [4], le modèle d'agent considéré suit trois sous-comportements (*acting*, *communicating* et *planning*). L'article soulève le problème du goulot d'étranglement dû aux communications, point névralgique de notre approche. De plus dans un système dynamique le problème d'allocation de ressources varie au cours du temps rendant impossible la résolution du problème de manière optimale. Dans l'approche proposée par les auteurs, la communication des agents n'est possible que si les agents font partie du même ensemble connecté, par message direct ou par diffusion « broadcast ».
Dans le cadre de la chasse aux mines sous-marines, Milot et al. [15] proposent une approche décentralisée d'allocation de tâches multirobots par enchère. La mission est composée de trois objectifs : détection, identification et neutralisation. Les robots se retrouvent en concurrence pour « acheter » des tâches lors d'enchères.

2.3 Méthodes d'allocation de tâches dans les SMA

Les articles [22, 23, 27] présentent un état de l'art sur les méthodes d'allocation de tâches dans les SMA. La performance optimale d'un système global est un concept qui dépend de la perception de chaque agent et des contraintes du système. De ce fait le terme « utilité », souvent utilisé dans d'autres travaux, représente une valeur ou un coût affilié à cette allocation de tâches.
Il existe de nombreuses méthodes d'allocation [23] : les enchères ou marchés ; la théorie des jeux ; les techniques basées sur l'optimisation qui englobent les heuristiques ou DCOP (*Distributed Constraint Optimization Problems* [4]), les optimisations déterministes (algorithme hongrois) ou les métaheuristiques (*PSO, Bees algorithm, ant colony*) ; les approches orientées apprentissages (*MARL : Multi-Agent Reinforcement Learning*) ; et enfin les approches hybrides qui combinent plusieurs stratégies.
Nous nous intéressons plus particulièrement aux méthodes qui semblent le mieux répondre à nos objectifs, à savoir les algorithmes d'enchères (*auction-based*) qui permettent l'allocation de tâches de manière décentralisée et flexible. Les méthodes d'enchères sont efficaces bien que non optimales, et la mise à l'échelle est possible, car les coûts en calcul et en communication sont modérés [23]. Ces caractéristiques sont importantes pour nos besoins et contraintes détaillés dans la section 3.2. Dans les méthodes par enchère, les agents misent sur des tâches, la mise la plus élevée remporte l'allocation de la tâche. La méthode habituelle pour déterminer le vainqueur est d'avoir un « commissaire-priseur » qui reçoit et évalue les mises centralement. Deux algorithmes par enchère sont proposés par Choi et al. [3], le CBAA (*Consensus-Based Auction Algorithm*) et le CBBA (*CB Bundle Algorithm*). Le premier répond au problème d'allocation unique, tandis que le second correspond à l'allocation multiple. Le CBBA est un algorithme décentralisé qui utilise d'abord une heuristique gloutonne pour sélectionner les tâches, puis il applique un consensus pour supprimer le problème du chevauchement de tâches. Il existe d'autres algorithmes, notamment des variantes du CBBA, mais aussi le CNP (*Contract Net Protocol*) qui est un protocole standardisé permettant d'allouer ainsi que de réassigner des tâches à des agents.
L'approche du CBBA permet une décision décentralisée, nécessaire lorsqu'il y a de nombreux échanges dans de grandes équipes d'agents. De plus, cet algorithme est polynomial en temps de calcul ce qui lui permet une mise à l'échelle facilitée avec le nombre de tâches ou la taille du réseau qui augmentent. Enfin, des objectifs de conception différents, des modèles d'agents et des contraintes peuvent être incorporés en définissant des fonctions de scores appropriées. Le besoin en synchronisation du CBBA rend son utilisation dans des applications temps réel moins efficace, les mêmes auteurs tentent de compenser cet inconvénient avec le ACBBA (Asynchronous CBBA), une extension de l'algorithme existant [12]. Le comportement des agents que nous proposons se rapproche des algorithmes par enchères, particulièrement le CBBA, qui devra être adapté à nos besoins et contraintes.

3 Formulation du problème

Cette section a pour but de présenter le cadre des études afin de comprendre l'approche menée et les choix effectués.

3.1 Définitions

Définition 1. *Une plateforme correspond à différentes entités militaires comme des bâtiments de surface ou des aéronefs. Elle est caractérisée par sa position géographique (longitude, latitude et altitude), sa vitesse, ses communications et les senseurs qu'elle possède définissant ainsi ses capacités.*
Définition 2. *Un senseur (communément appelé capteur) est un instrument qui détecte et répond à certaines entrées provenant d'un environnement [24]. Il est caractérisé par sa portée de détection, son type de senseur et les services qu'il propose.*
Dans nos études, les senseurs récupèrent des données électromagnétiques dans un théâtre d'opération. Nous considérons trois types de senseurs différents, proposant des capacités propres. Les senseurs de GE comme le RESM (*Radar Electronic Support Measure*) ont la capacité de détecter des signaux radar en restant discrets, ils peuvent aussi localiser et identifier des cibles. Les radars comme les FCR (*Fire Control Radar*) sont des senseurs actifs capables de détecter, de localiser et de classifier des cibles aériennes ou de surfaces en fournissant des données de distance, de direction et de vélocité [14]. Enfin, les senseurs optroniques sont utilisés pour faire de la veille infrarouge (ou IRST : *Infra-Red Search and Track*) qui consiste à repérer et localiser des menaces par leur chaleur. Chaque senseur ne rempli qu'une partie des informations concernant une cible. Par exemple, le radar fournit la position et la distance de la cible, la GE permet d'obtenir des droites de visée et ne collecte des informations de distance que par défilement (déplacement de la plateforme réceptrice). L'optronique ne mesure pas la

distance mais permet de recueillir les informations de position et d'identification.

Définition 3. *Une tâche représente une réservation de ressource pendant un intervalle de temps. Une tâche peut avoir des contraintes de précédence si une autre tâche doit être effectuée avant celle-ci (voir définition 5 plan senseur).*

Il existe des tâches mono-senseurs et multi-senseurs tout comme il y a des allocations instantanées ou prévues dans le temps. De plus, un senseur peut lui aussi effectuer une seule ou plusieurs tâches simultanément. Nous nous limiterons aux tâches instantanées dans un premier temps.

Définition 4. *Une ressource est nécessaire à l'exécution d'une tâche. Dans un cadre général, les ressources sont variées : des données, des capacités de calculs, de la mémoire, de l'énergie, ou d'autres entités [2].*

Les senseurs représentent les ressources principales de nos travaux, cependant, pour qu'ils puissent fonctionner, ils ont aussi besoin de ressources de calcul, de fréquences de fonctionnement, de mémoire, etc. Dans un premier temps, nous nous limiterons aux senseurs sans nous préoccuper des autres attributs des ressources.

Définition 5. *Un plan senseur P_k [7] est composé d'un ensemble de tâches T_i, avec des contraintes les liants entre elles, affectées à une ou plusieurs ressources. (voir Figure 1)*

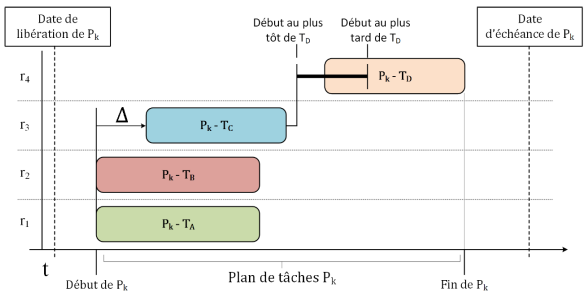


FIGURE 1 – *Schéma d'un plan senseur*, de [7] p.110

La plupart des plans senseurs ne comprennent qu'une seule tâche, mais d'autres peuvent en nécessiter plusieurs, par exemple : une tâche d'acquisition, puis une tâche de traitement, les deux ne pouvant être effectuées simultanément.

Définition 6. *Un service senseur est une capacité (ou une fonction) que proposent un ou plusieurs senseurs [18]. Les senseurs déclarent les services qu'ils peuvent accomplir dans un registre qui sera ensuite partagé à des consommateurs de services. Le service est caractérisé par une qualité de service (QoS), des règles, et des ressources qui permettent son utilisation. Le service répond à un besoin de l'utilisateur sans que celui-ci n'ai besoin de connaître son fonctionnement.*

Le modèle d'architecture à base de services permet l'intégration de nouveaux services avec des mécanismes de découverte de services. Les services peuvent évoluer ou être modifiés. Dans notre architecture, un service est décrit par un identifiant, une performance, un temps d'utilisation et des ressources dont il a besoin pour fonctionner. Ainsi, différentes ressources peuvent compléter un même service. Le choix du service s'effectuera donc en considérant sa faisabilité en fonction de critères tels que la position ou les performances des ressources, mais aussi la précision des mesures.

Définition 7. *Une requête opérationnelle est issue d'un ordre de mission interne ou externe à la plateforme. Cette requête décrit génériquement les besoins durant une opération, elle comprend notamment le service opérationnel, la zone géographique d'application, la durée effective de la requête, les contraintes, les priorités, la qualité de service désirée, etc.*

La localisation ou l'identification, deux fonctionnalités des senseurs vus dans la *Définition 2*, sont désignées comme des services opérationnels (ou hauts niveaux). Une requête de service opérationnelle de localisation discrète pourra par exemple être accomplie par différents services senseurs comme la TDOA (*Time Difference of Arrival*) ou la FDOA (*Frequency Difference of Arrival*) tout deux présentés dans [21]. Ces services permettent la géolocalisation passive, le radar ne sera alors pas utilisé, car contraint par un choix opérationnel de discrétion.

3.2 Besoins du système et métriques d'évaluation

Notre système multi-agent, en plus de répondre à la problématique d'allocation de tâches, devra montrer qu'il propose une solution sous certaines contraintes et avec des besoins définis :

- **Mise à l'échelle :** Le système doit pouvoir supporter une certaine charge, l'augmentation du nombre de plateformes et donc de senseurs impactera le nombre d'agents créés, il faut donc déterminer les limites de l'approche proposée ;
- **Contraintes temporelles :** En raison du contexte d'utilisation (le combat collaboratif), l'allocation de tâches devra être rapide, le temps mis pour obtenir une allocation de tâches effective devra être mesuré ;
- **Robustesse et résilience :** Le système doit s'adapter si des senseurs ou des plateformes sont ajoutés ou retirés, ce qui implique des changements sur l'allocation de tâches. Cela peut se mesurer en testant d'ajouter ou de supprimer des plateformes, ou en simulant une défaillance d'un senseur ;
- **Communication :** La communication entre les agents implique une augmentation des échanges inter-plateformes, il est donc nécessaire de quantifier le besoin en bande passante avec l'approche proposée ;
- **Modularité et adaptabilité :** Le contexte multi-milieu et multi-plateforme nécessite un travail sur le modèle des agents, nous devons pouvoir proposer de nouveaux services qui s'intégreront dans la liste des services, et ce avec des plateformes variées ;
- **Efficacité :** Le système doit maximiser le nombre de tâches accomplies tout en s'assurant de la prise en compte des tâches les plus importantes.

4 Conception des agents

Dans le cadre de nos travaux, nous étendons les concepts d'agents expliqués par Grivault et al. [9]. L'agent a pour objectif principal de collecter le plus de données possible sur un objet unique du théâtre des opérations grâce à des senseurs. Dans cette optique, l'agent en question va proposer des plans qui seront envoyés à un ordonnanceur. L'ordonnanceur sera chargé de positionner temporellement les tâches (durée, début et fin de tâche, etc.). L'architecture proposée est centralisée, les agents sont situés sur la même plateforme et il n'y a pas de collaboration avec d'autres plateformes. La vision de l'architecture interne d'un agent reste similaire dans notre approche multi-agent. La distribution des agents sur les différentes plateformes constitue une proposition d'architecture permettant l'usage de services multi-plateformes, ce qui étend donc les travaux des auteurs précédemment cités.
La conception architecturale de l'agent est présentée plus en détail dans l'article [8]. Sa structure interne comporte une mémoire, une messagerie et un ensemble de fonctions. L'agent récupère des informations de son environnement. Il alimente ses fonctionnalités grâce à la base externe de connaissances qui contient les capacités et informations collectés par les senseurs, ainsi que les données opérationnelles (par exemple des renseignements connus sur une cible potentielle). La mémoire, initialement vide pour chaque agent générique, se complète par les informations qui proviennent de la plateforme, des requêtes opérationnelles issues de l'exécutif et enfin des senseurs actifs qui engrangent des données sur les cibles. Un mécanisme de communication est présent dans le modèle agent mais n'est pas explicité, nous proposons d'utiliser le protocole AMQP (*Advanced Message Queuing Protocol*).
Dans cet article, nous proposons trois types d'agents qui dérivent d'un agent générique. Ils comprennent une mémoire interne, des fonctions (initialisation de l'agent, démarrage de l'agent, mise à jour de la mémoire, etc.) et un système de communication utilisant RabbitMQ. Dans cette section, nous présentons trois concepts d'agents :

- L'agent *plateforme* (voir Définition 1) a pour rôle de maintenir la base de connaissance de la plateforme qu'il symbolise et des plateformes alliées. Il crée un agent *service* et il actualise sa situation tactique locale par la création et la mise à jour des agents *tactiques* ;
- L'agent *service* représente la liste des services que fournit le système (voir Définition 6), chaque plateforme en possède un. Il a pour objectifs de proposer des services disponibles et de planifier les services en échangeant des messages avec les agents *tactiques* ;
- L'agent *tactique* [9] exprime un objet du théâtre des opérations. Il est donc créé en réponse à la détection d'une piste « senseur ». Dans nos travaux, cet agent propose des plans senseurs à l'agent *service*.

Ensemble, ils permettent d'allouer des tâches à des senseurs dans un contexte multi-plateforme.

4.1 Agent *plateforme*

L'agent *plateforme* représente logiciellement une plateforme. Sa mémoire interne comprend des informations sur sa position, sur ses senseurs et leur disponibilité, sur les connexions avec les autres plateformes, etc. Nous supposons que la faculté de communication entre deux plateformes est vérifiée en amont. Nous supposons que les liaisons sont directes dans un réseau maillé (en réalité, la communication peut s'effectuer par saut, ce qui engendre une surconsommation de la bande passante et une augmentation de la latence).
Les interactions avec les autres agents s'effectuent avec le modèle *publish/subscribe* que fournit le protocole AMQP. Les mécanismes d'échanges que nous proposons se rapprochent de ceux utilisés pour la plateforme Triskell3S [19]. Triskell3S propose un protocole de communication commun entre tous les agents. Les agents d'une plateforme A peuvent alors communiquer avec des agents d'une plateforme B par le mécanisme de *Publish/Subscribe* de MQTT (Message Queuing Telemetry Transport). Ainsi, dans nos travaux, pour chaque échange nous créons un producteur sur l'agent émetteur et un consommateur sur l'agent récepteur. Par exemple dans le cas de partage des ressources disponibles, l'agent *plateforme* publie les ressources sur une clé de routage (*routing key*) « platform.resource » et il s'abonne à une clé de liaison (*binding key*) « *.resource ».
Les échanges entre les agents sont présentés sur la figure 2, les messages d'une plateforme vers une autre transitent par le réseau tandis que les échanges internes ($Plateforme \leftrightarrow Service$ ou $Service \leftrightarrow Tactique$) se basent sur des données partagées.
L'agent *plateforme*, par des échanges avec ses pairs, met à jour la base de connaissance des ressources alliées. Nous nous intéressons principalement aux échanges entre des agents appartenant à des plateformes différentes, ainsi l'agent *service* pourra récupérer la liste des ressources directement depuis la plateforme à laquelle il appartient. De plus, les agents *plateformes* décident ensemble du choix des agents *tactiques* qu'ils prennent en charge. En effet, les plateformes possèdent une copie des données de tous les agents *tactiques*, mais un seul agent tactique concernant une même piste propose des plans. Cela permet d'avoir de la redondance et de ne pas perdre de données tactiques si une plateforme quitte le réseau.

4.2 Agent *service*

Pour chaque agent *plateforme*, un agent *service* est associé. Celui-ci possède la connaissance sur les services senseurs et leur disponibilité. Un service senseur a besoin de ressources pour fonctionner. L'agent *service* va donc récupérer la liste des ressources de l'ensemble des plateformes et créer une table interne de services uniques comme nous pouvons le voir dans la table 4.2. Cette table peut être élaborée en préparation de mission en référençant les services possibles, ils sont alors inutilisables par défaut et activés lorsque les plateformes et leurs ressources requises (RP_A dans la table pour « Ressource Plateforme A ») existent bien dans le réseau. Elle peut aussi être complétée en cours de

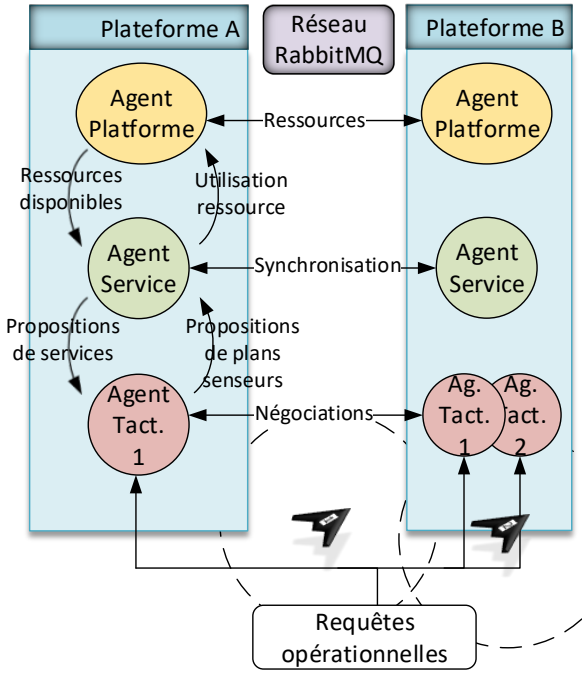


FIGURE 2 – Echanges entre agents

Service	Performance	Ressources	Durée
S1	580	RP_A, RP_B	10s
S2	450	RP_A, RP_C	10s

TABLE 1 – Connaissance de l'agent sur les services disponibles

mission, dans ce cas les plateformes doivent partager, en plus des ressources, les fonctionnalités de leurs senseurs qui permettront alors de les faire correspondre à des services senseurs. Par exemple, il faut au minimum deux senseurs de GE pour effectuer un service de TDOA, les plateformes capables d'effectuer ce service partagent sur le réseau leur capacité. L'agent *service* déclinera alors les choix possibles de ce service avec la durée nécessaire d'utilisation des ressources pour le mener à terme. Ainsi dans le cas de trois plateformes, il sera possible d'exploiter un service de TDOA utilisant les senseurs des plateformes A et B, ou bien les senseurs des plateformes B et C. Une performance est calculée selon plusieurs métriques comme la distance entre les plateformes ou les performances des senseurs.

L'agent *service* actualise à intervalle régulier sa table de services et partage ses mises à jour de services avec une performance acceptable à l'agent *tactique*. Celui-ci a également besoin de la durée d'utilisation de la ressource nécessaire à l'accomplissement du service.

Ensuite, il reçoit les propositions de plans senseurs des agents *tactiques* avec des scores associés et il planifie l'utilisation des services en réservant les meilleurs plans en fonction des ressources disponibles. L'agent *service* peut être vu comme un « commissaire-priseur » d'un algorithme de coordination par enchère, en effet il obtient des demandes d'allocation de manière centralisée tout en étant distribué sur différentes plateformes. Le score qu'il reçoit correspond donc à la mise que l'agent *tactique* lui propose. La phase de l'allocation locale (voir figure 4) se termine lorsque chaque agent tactique possède un plan d'accepté, ou lorsque la ressource est épuisée. La phase suivante, celle de l'allocation globale, consiste à demander l'activation de la ressource à l'agent plateforme. Afin de synchroniser les demandes de planification des agents services, une base de données distribuée existe au niveau de chaque plateforme. Chaque planification validée par l'agent service est envoyée sur un topic et mise en file d'attente FIFO pour validation auprès de la plateforme. Si le plan est finalement refusé, l'agent service partage le statut du plan à l'agent tactique qui retourne dans un état de planification. Un agent satisfait attend que son plan soit terminé pour en proposer de nouveaux. Un plan accepté peut aussi être refusé, si un meilleur plan qui utilise les mêmes ressources est proposé. Au cours du temps, l'agent service vérifie que les plans acceptés ne sont pas annulés et il incrémente le score des plans en cours pour éviter qu'ils soient annulés en fin d'utilisation du service.

4.3 Agent *tactique*

L'agent *tactique* est la représentation logicielle d'une piste (un objet ou une cible sur le théâtre d'opération). Sa mémoire interne comprend des informations sur la position, l'attitude et la vitesse de la cible, ses intentions pour compléter les connaissances de la piste et des informations d'identification (si disponibles). Les agents *tactiques* fonctionnent par requête opérationnelle, les besoins pour la mission vont influencer les choix de leurs plans. Par exemple, un opérateur a défini une zone de veille, si la menace détectée se situe hors zone alors l'agent *tactique* sera créé, mais inactif, et ce jusqu'à ce qu'une nouvelle requête lui ordonne le contraire.

L'agent *tactique* propose d'accomplir des objectifs opérationnels selon une suite logique nommée DRIL (Détection, reconnaissance, idenfication et localisation), pouvant être pris dans un ordre différent selon la situation. Ainsi, le choix des plans senseurs différera en fonction de l'objectif opérationnel en cours.

Il est possible que deux plateformes aient reçu la piste d'une même cible, donc deux agents *tactiques* sont créés pour un même objet (voir figure 2, où un objet situé entre les plateformes A et B amène à la création d'un agent *tactique* sur chaque plateforme). Dans la réalité, une corrélation entre les pistes doit être réalisée afin de déterminer qu'il s'agit bien de la même menace. Dans notre cas, nous supposons la corrélation comme parfaite en considérant que les plateformes connaissent a priori cette information.

L'agent *tactique* va étudier la faisabilité de plans senseurs en fonction des services proposés par l'agent *service*. Pour cela, il va calculer un score pour chaque plan qui sera valable pendant un court intervalle de temps. Ce score dépendra de la performance du service, de la position des plateformes possédant la ressource par rapport à la cible, de la priorité de la requête opérationnelle et des contraintes im-

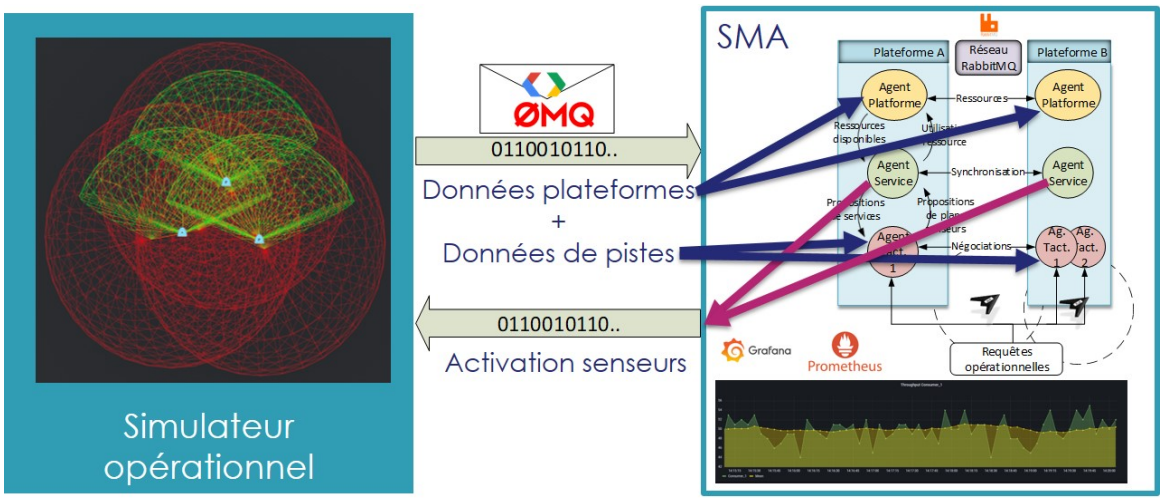


FIGURE 3 – Schéma de l'architecture globale

posées par cette requête (une requête avec contrainte de discrétion aura un score faible si les ressources proposées par le service ne sont pas discrètes également). Les plans senseurs qu'il soumettra comprendront : le service en question, le score calculé, le moment où le plan a été calculé, et les ressources nécessaires à sa réalisation. L'objectif de l'agent est de compléter ses données sur la cible afin d'améliorer sa connaissance tactique sur celui-ci. Dans ce but, il va fournir égoïstement (algorithme *glouton*) son ou ses meilleurs plans senseurs à l'agent *service*. Il ne se coordonne pas avec les autres agents avant d'envoyer ses plans et il ne sait pas si ses plans seront acceptés mais il maximise son propre score. Nous supposons que l'agent ne planifie qu'une utilisation de service à la fois, il ne pourra pas présenter un nouveau plan senseur tant que son plan n'est pas terminé ou annulé.

La figure 4 présente un exemple simplifié des échanges entre un agent *tactique* et un agent *service*.

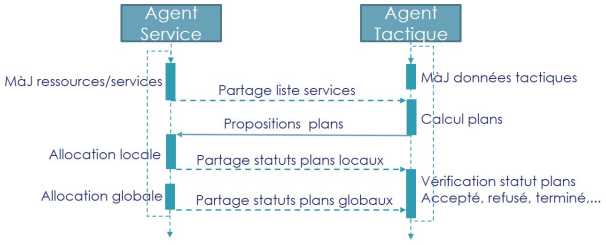


FIGURE 4 – Exemple simplifié d'un échange entre un agent tactique et un agent service

5 Expérimentations et méthodes d'évaluation

Dans la section précédente, nous avons présenté des concepts d'agents afin de répondre à la problématique de l'allocation de tâches à des senseurs. Dans cette partie, nous présentons les expérimenations et les méthodes d'évaluation dans le but de valider l'architecture multi-agent proposée.

Pour introduire cette partie, nous allons présenter un simulateur que nous utilisons. Ce simulateur, développé en C++, exploite le moteur de jeux vidéos *Unreal Engine 5*. Il permet la création de scénarios mettant en jeu plusieurs plateformes, il est possible de les ajouter et de les faire se déplacer, tout en activant/désactivant divers senseurs. L'exécution du scénario nous permet de récupérer des données des plateformes et des pistes qu'ils détectent. Avec un interfaçage combinant ZeroMQ [10] et le langage de description d'interface *Protocol Buffers*, nous pouvons récupérer ces précieuses données simulées dans le but de faire fonctionner notre logiciel multi-agent, qui est un logiciel externe au simulateur. Cependant, nous nous limiterons dans un premier temps à des données factices en envoyant des données chiffrées avec ZMQ au cours de la simulation, créant ainsi des agents. Le schéma de l'architecture globale est présenté en figure 3.

5.1 Développement de l'outil de validation

Les agents sont en cours de développement en langage Java. L'agent est représenté par un *thread* et ses actions suivent une machine à états finis. Par exemple, l'agent *plateforme* suit plusieurs états comme la mise à jour des données de la plateforme ou la mise à jour des agents tactiques.

Le récepteur ZeroMQ est exécuté sur un *thread* dédié, il réceptionne en permanence les données du simulateur. Chaque plateforme présente dans le scénario apportera la création d'un agent *plateforme*, tandis que la réception des données de pistes créera ou mettra à jour un agent *tactique*. Les agents *services* seront créés à l'initialisation des agents *plateformes*.

Le *Message-Oriented Middleware* (MOM) RabbitMQ implémente le protocole AMQP, nous l'utilisons pour modéliser les échanges entre les agents avec les concepts de *producteurs* et de *consommateurs* présentés dans l'article [17],

où les choix technologiques sont détaillés.
Dans le but de vérifier le bon fonctionnement des échanges entre les agents, nous affichons la mémoire de l'agent sous forme de tableau, comme nous pouvons le voir dans un exemple en figure 5. De la même façon, nous pouvons observer l'évolution de la liste des services et de l'utilisation des ressources en cours de simulation.

AGENT_NAME	AGENT_TYPE	AGENT_ID	POSITION	PLATFORM_RESOURCES
AgentPlatform_0	Platform Agent	3	[0.884; 0.313]	[Resource [resouceType=R1,...
AgentPlatform_1	Platform Agent	9	[0.131; 0.245]	[Resource [resouceType=R1,...
AgentPlatform_2	Platform Agent	15	[0.006; 0.412]	[Resource [resouceType=R1,...
AgentPlatform_3	Platform Agent	21	[0.257; 0.775]	[Resource [resouceType=R1,...

FIGURE 5 – Mémoire de l'agent plateforme

5.2 Méthodes d'évaluation et métriques considérées

Pour évaluer notre approche, nous devrons justifier d'une réponse à nos besoins mentionnées dans la sous-section 3.2. D'une part, nous voulons évaluer la faisabilité de nos concepts au regard des contraintes en bande passante entre les plateformes. L'utilisation de RabbitMQ [6] pour les échanges entre nos agents nous permet d'utiliser les outils Grafana et Prometheus pour la gestion du réseau, l'article [17] présente l'utilisation de ces outils dans le cadre de nos travaux. Les métriques de débits et de latences seront observées au cours de la simulation sur des graphes temporels. Le but étant de constater le besoin en bande passante de notre système en fonction de plusieurs paramètres : le nombre de plateformes, le nombre de senseurs actifs et le nombre de menaces dans l'environnement. De plus, l'évolution des paramètres devra être faite en dynamique, afin de constater la résilience de notre système. C'est une étape importante pour la validation de l'architecture multi-agent.
D'autre part, nous voulons améliorer la qualité de l'allocation de tâches en optimisant nos algorithmes. Les performances de l'allocation seront impactées par l'augmentation du nombre de services disponibles dans la base de données. Un délai trop important dans la proposition des plans senseurs pourrait impacter significativement les plans à caractères prioritaires et donc urgents. Il faut donc observer l'impact du nombre de services sur le système.
En dernier lieu, nous pourrons également observer la charge de calcul du système multi-agent sur chaque plateforme, en fonction du nombre de services et du nombres d'agents *tactiques* présents.

6 Conclusion et perspectives

Dans cette article, nous avons présenté des concepts d'agents pour la collaboration multi-plateforme dans un contexte de défense navale. Dans un premier temps, nous avons expliqué le cadre complexe du combat collaboratif naval. Puis nous avons détaillé le fonctionnement de trois agents : l'agent *plateforme*, l'agent *service* et l'agent *tactique*. Ensemble, ils forment un système multi-agent qui sera intégré dans les architectures des systèmes de senseurs futurs. Enfin, nous avons proposé des méthodes pour évaluer nos concepts vis-à-vis de nos besoins.
Pour la suite des travaux, nous évaluerons les performances de l'approche multi-agent proposée avec l'aide de l'outil Grafana et des métriques évoquées dans cet article. Ensuite, nous devrons proposer plusieurs scénarios opérationnels simulés afin de mettre en pratique des tests de performances du système pour s'assurer qu'il réponde bien aux besoins opérationnels. Enfin, une étude architecturale globale intégrant le système multi-agent sera menée avec notamment des réflexions sur le choix des interfaces, sur les échanges entre les composants, etc.
Remerciements. Ces travaux ont été effectués dans le cadre d'une thèse CIFRE sur l'architecture des systèmes navals chez Thales DMS en collaboration avec IMT Atlantique et le GIS Cormorant.

Références

[1] Jacob Beal, Kyle Usbeck, Joseph Loyall, Mason Rowe, and James Metzler. Adaptive opportunistic airborne sensor sharing. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 13(1) :1–29, 2018.

[2] Ellie Beauprez, Luc Bigand, Anne-Cécile Caron, Maxime Morge, and Jean-Christophe Routier. Réaffectation de tâches de la théorie à la pratique : état de l'art et retour d'expérience. In Jean-Paul Jamont, editor, *29ème francophones sur les systèmes multi-agents (JFSMA)*, JFSMA 2021. Collectifs cyber-physiques, pages 51–60, Bordeaux, France, June 2021. Cépaduès.

[3] Han-Lim Choi, Luc Brunet, and Jonathan P How. Consensus-based decentralized auctions for robust task allocation. *IEEE transactions on robotics*, 25(4) :912–926, 2009.

[4] Alaa Daoud, Flavien Balbo, Paolo Gianessi, and Gauthier Picard. Un modèle agent générique pour la comparaison d'approches d'allocation de ressources dans le domaine du transport à la demande. In *JFSMA 2021 : 29ème Journées Francophones sur les Systèmes Multi-Agents*, pages 127–136, Bordeaux, France, June 2021. Cépaduès.

[5] Ali Dorri, Salil S Kanhere, and Raja Jurdak. Multi-agent systems : A survey. *IEEE Access*, 6 :28573–28593, 2018.

[6] David Dossot. *RabbitMQ essentials*. Packt Publishing Ltd, 2014.

[7] Ludovic Grivault. *Architecture multi-agent pour la conception et l'ordonnancement de systèmes multi-senseur embarqués sur plateformes aéroportées*. PhD thesis, Sorbonne université, December 2018.

[8] Ludovic Grivault, Amal El Fallah-Seghrouchni, and Raphaël Girard-Claudon. Agent-based architecture for multi-sensors system deployed on airborne platform. In *2016 IEEE International Conference on Agents (ICA)*, pages 86–89. IEEE, 2016.

[9] Ludovic Grivault, Amal El Fallah-Seghrouchni, and Raphaël Girard-Claudon. Next generation of airborne

platforms from architecture design to sensors scheduling. In *2017 IEEE International Conference on Agents (ICA)*, pages 60–65. IEEE, 2017.

[10] Pieter Hintjens. *ZeroMQ : messaging for many applications*. " O'Reilly Media, Inc.", 2013.

[11] Michael N Huhns. *Distributed Artificial Intelligence : Volume I*, volume 1. Elsevier, 2012.

[12] Luke Johnson, Sameera Ponda, Han-Lim Choi, and Jonathan How. Asynchronous decentralized task allocation for dynamic environments. In *Infotech@ Aerospace 2011*, page 1441. 2011.

[13] Vicente Julian and Vicente Botti. Multi-agent systems. volume 9, page 1402. MDPI, 2019.

[14] Stéphane Kemkemian and Myriam Nouvel-Fiani. Toward common radar & ew multifunction active arrays. In *2010 IEEE International Symposium on Phased Array Systems and Technology*, pages 777–784. IEEE, 2010.

[15] Antoine Milot, Estelle Chauveau, Simon Lacroix, and Charles Lesire Cabaniols. Allocation par enchères et planification hiérarchique pour un système multi-robot, application au cas de la chasse aux mines. In *JFSMA 2022 : 30èmes Journées Francophones sur les Systèmes Multi-Agents*, 2022.

[16] Sameera Ponda, Josh Redding, Han-Lim Choi, Jonathan P How, Matt Vavrina, and John Vian. Decentralized planning for complex missions with dynamic communication constraints. In *Proceedings of the 2010 American Control Conference*, pages 3998–4003. IEEE, 2010.

[17] Paul Quentel, Yvon Kermarrec, Ludovic Grivault, Pierre Le Berre, and Laurent Savy. A rabbitmq-based framework to deal with naval sensor systems design complexity. Publication acceptée à *Computing Conference 2023*, Juin, Londres, Royaume-Uni, Inpress.

[18] Duncan Russell and Jie Xu. Service oriented architectures in the provision of military capability. In *UK e-Science All Hands Meeting*. Citeseer, 2007.

[19] Alexandre Schmitt, Valérie Renault, Florent Carlier, and Pascal Leroux. De l'iot à l'iot-a : une approche pour des communications dynamiques. In *27emes Journées Francophones sur les Systèmes Multi-Agents (JFSMA)*, 2019.

[20] A El Fallah Seghrouchni and L Grivault. Multi-agent paradigm to design the next generation of airborne platforms. *Aerospace Lab*, pages 1–8, 2020.

[21] Hugo Seuté, Laurent Ratton, and Antoine Fagette. Passive sensor planning for tdoa/fdoa geolocation under communication constraints. In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2020.

[22] Vaishnavi Singhal and Deepak Dahiya. Distributed task allocation in dynamic multi-agent system. In *International Conference on Computing, Communication & Automation*, pages 643–648, 2015.

[23] George Marios Skaltsis, Hyo-Sang Shin, and Antonios Tsourdos. A survey of task allocation techniques in mas. In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 488–497, 2021.

[24] Itshak Tkach and Yael Edan. *Distributed heterogeneous multi sensor task allocation systems*. Springer, 2020.

[25] Joanna Turner, Qinggang Meng, Gerald Schaefer, and Andrea Soltoggio. Distributed strategy adaptation with a prediction function in multi-agent task allocation. 2018.

[26] Michael Wooldridge. *An introduction to multiagent systems*. John Wiley & sons, 2009.

[27] Bing Xie, Jing Chen, and Lincheng Shen. Cooperation algorithms in multi-agent systems for dynamic task allocation : A brief overview. In *2018 37th Chinese Control Conference (CCC)*, pages 6776–6781, 2018.

MAKI: Une infrastructure à clefs publiques pour systèmes muti-agents

Arthur Baudet[1,2], Oum-El-Kheir Aktouf[1], Annabelle Mercier[1], Philippe Elbaz-Vincent[2]

[1] Univ. Grenoble Alpes, Grenoble INP, LCIS, 26000 Valence, France
[2] Univ. Grenoble Alpes, CNRS, IF, 38000 Grenoble, France

{arthur.baudet,oum-el-kheir.aktouf,annabelle.mercier}@lcis.grenoble-inp.fr
philippe.elbaz-vincent@math.cnrs.fr

Résumé

Ce papier présente une infrastructure à clefs publiques pour les systèmes multi-agents-embarqués ouverts. Ces derniers sont particulièrement vulnérables car ils permettent à des agents inconnus, et possiblement malicieux, d'entrer et d'interagir avec les agents existants. Nous nous intéressons notamment aux attaques portant sur les communications afin de permettre aux agents de communiquer sans risques de voir leurs communications manipulées. Pour ce faire, nous proposons une infrastructure mettant à profit l'autonomie des agents et permettant de s'assurer de la sécurité de leurs interactions.

Mots-clés

Déploiement de systèmes multi-agents, système multi-agent ouvert, agent embarqué, infrastructure à clefs publiques

Abstract

This paper presents a public key infrastructure for open multi-agent systems of embedded agents. These systems are very prone to attacks as they are confronted with unknown systems with unknown goals. We aim at securing the communications between agents to provide foundations for more advanced security solutions as well as allowing agents to communicate without the risk of their messages being tampered with. To do so, we deploy a key infrastructure taking advantage of the agents autonomy to allow authenticity and integrity checks and accountability of all interactions.

Keywords

Multi-agent system deployment, open multi-agent system, embedded agent, public key infrastructure

1 Introduction

Le paradigme multi-agent permet de construire des logiciels et systèmes plus résilients, pouvant plus facilement passer à l'échelle, et pouvant se confronter aux challenges actuels dont la complexités ne cesse de croitre. Notamment, avec l'augmentation constante du nombre d'objets connectés, le développement des systèmes autonomes, ce paradigme semble être de plus en plus pertinent pour réaliser le contrôle et la coordination de ces systèmes embarqués hétérogènes. Dans ce cadre, nous nous intéressons plus particulièrement aux systèmes ouverts, lesquels permettent à de nouvelles entités de les rejoindre ou de le quitter. Nous les nommons Systèmes Multi-Agent-Embarqués (SMAE) ouverts. De tels systèmes peuvent exister sous la forme de réseaux de drones coopérants, pour surveiller des feux de forêts par exemple, entrant et sortant de systèmes en fonction de leurs ressources. Un système pareil pourrait profiter d'une collaboration de constructeurs de drones, chacun déployant des drones au fur et à mesure que nécessaire.
Des études récentes [4, 7] montrent que les SMAE, et les systèmes similaires sont particulièrement vulnérables aux menaces venant de l'intérieur ainsi qu'aux attaques sur les communications. Ils montrent également que les Systèmes de Gestion de la Confiance (SGC) sont un moyen courant de mitiger ces menaces. Cependant, ces SGC prennent souvent des hypothèses fortes concernant les couches inférieures, notamment la couche cryptographique [11, 13]. Par exemple, ces hypothèses peuvent nécessiter la présence d'une tierce partie pour fournir une racine de confiance ou pour précharger des certificats dans les agents. Cela rend ces solutions inapplicables dans un contexte ouvert et décentralisé. Ainsi, une solution spécifique pour fournir aux agents des clefs cryptographiques leur permettant de communiquer de manière sécurisée est requise ici.
Le reste du papier est structuré de la sorte : la prochaine section présente les hypothèses et le modèle d'attaquant que nous considérons dans ce travail. Puis, on fonction de ce modèle, nous discutons des travaux connexes en Section 3. Ensuite, nous introduisons notre contribution dans la Section 4 et en proposons une validation dans la Section 5 sous deux aspects différents : d'une part à l'aide d'un outil de vérification de modèles et d'autre parts à travers le développement en simulation d'une preuve de concept. Enfin, nous concluons en Section 6.

2 Hypothèses et modèle de menaces

Nous cherchons à sécuriser les communications dans les systèmes multi-agents d'agents embarqués pour permettre l'utilisation des SGC sans risque de falsification des échanges. Notre proposition repose sur les hypothèses sui-

vantes :

— Les primitives cryptographiques utilisées, le matériel sur lequel elles sont exécutées et leurs implémentations sont sécurisées.
— Un SGC approprié est exécuté et prend en compte les attaques dont il peut faire l'objet.
— Un protocole de routage est utilisé pour permettre aux agents de communiquer entre eux.

Suivant ces hypothèses, nous définissons le modèle d'attaquant comme étant un attaquant disposant de ressources similaires à celles des agents du système et qui aurait le contrôle du canal de communication. Il serait capable d'écouter, de bloquer et d'altérer tout message. De plus, nous ne faisons aucune supposition sur les intentions des autres agents.

3 Travaux connexes

Les travaux présentés dans [14] et [9] reposent sur l'utilisation de cryptographie à seuil ou sur une infrastructure sur mesure pour fournir une infrastructure à clefs publiques (ICP) décentralisée. Cependant, ces deux solutions nécessitent toujours une vérification initiale ou le préchargement de certificats pour fournir une authentification.
Dans [5], les auteurs proposent une ICP distribuée pour le contrôle de systèmes industriels via l'utilisation d'un framework d'agents qui nécessite un opérateur pour ajouter ou supprimer les agents. Ceci n'est pas compatible avec la caractéristique d'ouverture des systèmes que nous étudions.
Dans les travaux [1, 6], l'ICP décentralisée repose sur une table de hachage distribuée pour permettre la signature, le stockage et la certification des certificats. Bien que ces dernières approches résolvent le problème du consensus dans la gestion des certificats, elles ne fournissent pas de moyens de filtrer les nœuds indignes de confiance.
Dans l'ensemble, toutes les approches ci-dessus fournissent un moyen de décentraliser ou de distribuer une ICP, mais elles ne peuvent pas s'appliquer à notre problématique, car elles reposent trop sur des tiers ou sur un contrôle externe.
Plus récemment, la plupart des efforts visant à concevoir une ICP décentralisée impliquent l'utilisation d'une blockchain [17, 18, 19]. La blockchain a été conçue pour fournir un consensus sur les informations dans des systèmes décentralisés sans confiance préexistante, ce qui en fait une solution idéale pour délivrer, stocker, et révoquer les certificats, comme pour deux travaux précédemment cités. Pourtant, les blockchains actuellement déployées ne sont pas adaptées à notre problématique. La plupart utilisent la preuve de travail pour leur algorithme de consensus et ce dernier n'est pas du tout adapté aux systèmes embarqués. Quant aux solutions utilisant des algorithmes alternatifs, comme la preuve d'enjeu, elles se basent sur des blockchains publiques accessibles sur un réseau externe (Internet dans la majorité des cas) et non pas en local entre les nœuds du système.
La cryptographie basée identité ou attributs semble aussi être adaptable aux systèmes décentralisés [16, 8]. Cependant, cela implique une connaissance préalable sur les agents, ce qui est une hypothèse difficile à satisfaire dans des systèmes hétérogènes ouverts.
Par conséquent, aucune approche de notre état de l'art ne nous fournit une infrastructure existante répondant aux exigences de décentralisation, d'ouverture et d'autonomie du type de systèmes étudiés. C'est pourquoi nous fournissons dans notre travail les fondations d'une telle ICP à travers Multi-Agent Key Infrastructure (MAKI). MAKI a pour but l'établissement et le partage de clefs asymétriques pour permettre aux agents de communiquer de façon sécurisée. Elle s'appuie également sur le SGC pour déployer une auto-organisation résiliente contre les attaques internes et le renforce en sécurisant les communications ainsi qu'en rendant possible l'exclusion d'intrus.

4 Multi-Agent Key Infrastructure

L'utilisation de signatures cryptographiques permet d'assurer les trois caractéristiques de sécurité nécessaires à des communications sécurisées :

L'intégrité d'un message doit être maintenue pour éviter qu'un attaquant modifie les communications ;

L'authenticité d'un message est nécessaire pour éviter qu'un attaquant prenne l'identité d'un autre agent ;

La non-répudiation d'un message est nécessaire pour qu'un attaquant ne puisse pas mentir et revenir sur une de ses interactions.

Cela nécessite simplement que les agents génèrent une paire de clefs asymétriques et qu'ils les utilisent pour signer chaque communication. Ce mécanisme répond à lui seul à l'exigence de cryptographie décentralisée que nous avons fixée. Cependant, cela permet des comportements abusifs tels que des agents utilisant plusieurs paires de clefs en même temps ou changeant leurs clefs au fil du temps. Nous empêchons ces comportements en liant l'identité d'un agent à la clef publique qu'il utilise et en mettant à profit le SGC pour rendre inefficace le changement d'identité (voir Section 5.2). Nous complémentons l'action du SGC en ajoutant la possibilité de révoquer un agent, à travers son certificat, au lieu de simplement l'ignorer.

4.1 Architecture

Puisque nous nous concentrons sur les systèmes hétérogènes ouverts, nous ne nous attendons pas à ce que les agents possèdent des certificats avant même de rejoindre MAKI. Et nous ne voulons pas non plus appliquer des protocoles d'authentification spécifiques. Au lieu de cela, nous avons conçu MAKI pour ne pas nécessiter d'authentification. Les agents sont anonymes et ne sont classés bienveillants ou malveillants que sur la base de leurs actions. Ceci est possible grâce à la non-répudiation apportée par l'utilisation de signatures cryptographiques. Cela signifie que l'identité d'un agent est liée aux clefs qu'il utilise pour interagir.
Nous avons conçu MAKI comme une ICP légère et décentralisée. Des principes de l'ICP, nous n'avons conservé que la fonction d'autorité de certification (AC), l'utilisation obligatoire des certificats et la révocation des certificats.

Pour imposer l'utilisation de la cryptographie, les messages non signés ou dont la signature n'est pas valide doivent être ignorés. De plus, les agents doivent s'assurer que la source des messages qu'ils reçoivent détient bien un certificat valide pour la clef utilisée pour signer le message. Les requêtes d'existence d'ACs ou les demandes de certification peuvent être signées avec des clefs non certifiées.

Ces certificats sont délivrés et révoqués par des ACs définies par l'utilisation d'un algorithme d'auto-organisation, capitalisant ainsi sur l'autonomie des agents. Un agent peut librement demander un certificat et l'AC est autonome pour y répondre ou non. Elles sont également autonomes dans le choix de révoquer un agent, mais il est attendu qu'elles le fassent lorsque la majorité des agents en qui elles ont confiance le demande. La révocation est effectuée à l'aide de deux mécanismes. D'une part, chaque AC maintient et diffuse une Liste de Révocation de Certificats (LRC). Cette méthode est directe et instantanée, mais, en fonction des capacités du réseau, les mises à jour des LRC peuvent prendre du temps. Ainsi, pour atténuer ce risque, nous utilisons d'autre part des certificats à courte durée de vie, qui ne seront pas renouvelés par les ACs ayant connaissance de la révocation.

4.2 Organisation

Le bon fonctionnement de l'auto-organisation dépend des règles que MAKI ajoute au SGC. Par exemple, la façon dont nous évitons le problème des AC auto-signées, la probabilité qu'un agent malveillant devienne AC ou les récompenses d'une certification croisée sont toutes expliquées dans la Section 4.3.

Les agents peuvent ont un rôle parmi deux : None ou AC. None est le rôle par défaut et n'a aucune responsabilité envers l'ICP. Les ACs sont chargées de délivrer des certificats aux agents None et aux autres AC, de révoquer les certificats, de stocker et distribuer une liste des certificats délivrés et une LRC. Comme MAKI ne s'appuie pas sur des tiers pour établir une racine de confiance, les ACs sont toutes initialement auto-signées et peuvent ensuite utiliser la certification croisée pour créer un réseau d'ACs dignes de confiance.

Les ACs sont auto-élues. Les agents capables d'être AC (en fonction de leur capacité de calcul, de stockage, d'énergie...) décident eux-mêmes s'ils vont devenir AC. L'algorithme 1 décrit comment le choix peut être fait. Cet algorithme a été conçu avec deux objectifs : (i) chaque agent est proche d'une AC et (ii) les ACs ne doivent pas devenir des points de défaillance uniques. Il conduit à une distribution uniforme des ACs avec une ou plusieurs (en fonction de `T`, la probabilité qu'un agent décide de devenir une AC redondante) ACs par groupe d'agents. Il est possible d'adapter la définition de « proche » pour réduire le nombre d'ACs. Il y aura plus d'ACs si « proche » signifie être à portée de communication que si cela signifie être à trois fois la portée de communication. Il est également possible d'adapter la valeur de `T` pour augmenter ou diminuer le nombre de ACs redondants. Si `T` est très élevé, presque tous les agents qui peuvent être AC le seront, mais si `T` est très faible, seuls les agents éloignés d'une AC le deviendront. La définition de « proche » et la valeur de `T` doivent être adaptées à l'application, à la densité et aux capacités de l'application sur laquelle fonctionne MAKI.

Algorithme 1 Algorithme décrivant comment la décision de devenir un CA est prise.

```
T ∈ [0, 1) ▷ La probabilité qu'un agent décide de devenir une AC même si d'autres ACs dignes de confiance sont proches.
Role ← None
CAs ← BROADCAST(CAListingRequest)
TrustedCAs ← FILTER(CAs, TrustLevel.Moderate)
if can become CA and (TrustedCAs is empty or RANDOM(0, 1) < T) then
    Role ← CA
```

Le choix d'une AC pour un agent None est similaire à la décision de devenir une AC. Des situations de point de défaillance unique peuvent survenir si les tous agents choisissent l'AC la plus fiable et que cette AC est considérée comme la plus fiable pour la plupart d'entre eux. Pour éviter de tels cas, un agent choisira l'une de celles auxquelles il fait le plus confiance, mais pas spécialement la plus fiable. Cela se traduit par un choix aléatoire pondéré par les valeurs de confiance. Cette façon de choisir garantit que la plupart du temps, une AC hautement fiable sera choisie, sans que ce soit toujours la même. Et cela mène aussi à choisir quelque fois des ACs considérées comme moins fiables, leur laissant l'occasion de faire leurs preuves.

L'ajout d'un nouvel agent dans MAKI est simple. L'agent déterminera d'abord s'il doit devenir une AC et, si non, demandera un certificat à une AC. Une fois fait, il peut décider de chercher une AC plus fiable en demandant à ses voisins ou garder l'AC qu'il a choisie. Dans tous les cas, il diffusera son certificat pour s'assurer que ses voisins en prennent connaissance.

Les ACs auto-signées ne sont pas sensibles aux mécanismes de révocation. La seule façon de les exclure est de les ignorer et de suggérer aux nouveaux agents de les éviter. Une façon de supprimer cet avantage est d'encourager la certification croisée. Cela signifie que les AC pourraient demander à d'autres AC de signer leurs certificats, ce qui les rendrait susceptibles à la révocation. Cela ne présente aucun avantage intrinsèque pour l'AC recourant à ce mode de certification, car cela ne fait que rendre plus difficile l'obtention et le maintien d'un certificat valide, mais peut être considéré comme une preuve de bonne foi et être récompensé au niveau du SGC.

Globalement, MAKI ne réduit pas la sécurité apportée par le SGC puisque, même si elles ne peuvent pas être révoquées, les AC auto-signées peuvent toujours être ignorées et leur mauvaise réputation partagée avec les nouveaux agents. Cela signifie que le nombre d'agents malveillants que MAKI peut gérer ne dépend que du SGC et de la complexité des attaques exécutées contre lui. Nous montrons dans la Section 5.2 comment, avec un SGC simple, MAKI gère les agents malveillants une fois que le SGC détecte leurs comportements malveillants.

TABLE 1 – Compromis entre les risques et les avantages pour chaque interaction entre les agents, en fonction de leurs rôles

Interaction	Risque	Bénéfice	Confiance requise
	Autorité de Certification		
Délivrer un certificat	Modéré. Permettre à un agent malveillant de participer.	Haut. Augmenter sa légitimité.	Modérée ou aucune†
Révoquer un certificat	Haut. Perdre la confiance des agents en désaccord avec la révocation. Exclure un agent bienveillant.	Haut. Exclure un agent malveillant.	Modéré.
Demander une certification croisée	Haut. Voir sa réputation* diminuer si le certificateur n'est pas de confiance. Donner plus de légitimité à une AC malveillante.	Modéré. Voir sa réputation augmenter.	Haute
Accepter une demande de certification croisée	Haut. Donner plus de légitimité à une AC malveillante.	Haut. Voir sa réputation augmenter.	Haute
	None		
Demander un certificat	Modéré. Donner plus de légitimité à une AC malveillante. Avoir besoin de changer d'AC.	Haut. Posséder un certificat est obligatoire pour participer au système.	Modérée ou aucune‡

* Le terme « réputation » est utilisé ici pour décrire la confiance moyenne globale envers un agent.
† Les ACs n'ont pas de moyen de s'assurer qu'un agent nouvellement entrant est digne de confiance, elles laissent une chance à ces agents dans un premier temps.
‡ Un agent nouvellement entrant n'a pas de moyen de déterminer l'AC la plus digne de confiance, il fait donc un choix non éclairé dans un premier temps.

4.3 Gestion de confiance

MAKI n'est pas conçu pour un SGC spécifique. De plus, définir un modèle de confiance pour chaque cas d'utilisation spécifique de MAKI n'est pas possible. Au lieu de cela, nous spécifions ici comment MAKI s'appuie sur le SGC pour déployer son auto-organisation.

Nous présentons dans le Tableau 1 une évaluation des risques des interactions dans MAKI et les seuils de confiance recommandés à atteindre pour les réaliser. Ces seuils de confiance représentent le niveau de confiance qu'un agent doit avoir dans les autres agents pour interagir avec eux.

En complément des interactions présentées, tout agent peut demander et partager un certificat sans risque ni confiance requise, cela permet de vérifier ou de prouver que l'exigence de détention d'un certificat valide est satisfaite. Il en va de même pour la requête et le partage de la LCR qui, elle aussi, ne contient que des informations publiques.

MAKI exploite également le SGC pour atténuer la prolifération des ACs malveillantes en ajoutant un coût au rôle d'AC. Une AC ne peut être légitime que si elle répond continuellement aux demandes de certification, de révocation, de partage de la liste des certificats valides en cours qu'elle a distribués et de sa LRC De cette façon, même une AC malveillante doit contribuer au système pour garder son rôle. Cela peut traduit notamment par une légère augmentation de la confiance une AC chaque fois qu'elle répond à une demande de certification. De plus, la confiance accordée à un agent certifié est pondérée par la confiance accordée à l'AC qui a signé son certificat. Cela a pour but d'encourager les agents à choisir des ACs en lesquelles ils ont confiance, mais en lesquelles les autres agents font également confiance. Cela dans le but de faire en sorte que les ACs se comportent correctement avec chaque agent et pas seulement avec certains d'entre eux.

Si nous avons expliqué comment atténuer le risque que des agents malveillants deviennent ACs, et donc des ACs auto-signées, nous pouvons également proposer un moyen de réduire le nombre d'ACs auto-signées en ajoutant une récompense pour les ACs à certification croisée. Cette récompense de confiance encouragera les ACs se certifier entre elles, prenant donc le risque de d'avoir leurs certificats révoqués et d'être exclues, pour garder, voire augmenter, leur légitimité.

4.4 Gestion des certificats

Les représentations ASN.1 [10] du certificat et de la LRC données en Figure 1. Le format du certificat est une version simplifiée du format X.509 dont les principales différences sont l'inclusion de la clef publique de l'émetteur, puisqu'elle fait partie de son identité, l'inclusion le champ supplémentaire, `subjectInfo`, qui est laissé à la discrétion des concepteurs du système, et la suppression de plusieurs champs non utiles pour notre ICP. En utilisant ce for-

```
Identity ::= SEQUENCE { name INTEGER, publicKey BIT STRING }
```

(a) Représentation ASN.1 du champ Identity utilisé dans le certificat et la LRC de MAKI.

```
Certificate ::= SEQUENCE {
  version [0]  INTEGER,
  serialNumber INTEGER,
  signature    BIT STRING,
  issuer       Identity,
  validity     SEQUENCE {
                 notBefore UTCTime,
                 notAfter  UTCTime
               },
  subject      Identity,
  subjectRole  Role,
  subjectInfo  SubjectInfo
}
Role ::= INTEGER { NONE(0), CA(1) }
```

(b) Représentation ASN.1 du certificat de MAKI.

```
CertificateRevocationList ::= SEQUENCE {
  version [0]         INTEGER,
  signature           BIT STRING,
  holder              Identity,
  thisUpdate          UTCTime,
  revokedCertificates SEQUENCE OF SEQUENCE {
    serialNumber        INTEGER,
    issuer              Identity,
    subject             Identity,
    revocationDate      UTCTime,
    reasonCode          ReasonCode
  }
}
ReasonCode ::= ENUMERATED {
   idCompromise(0),
   cessationOfOperation(1)
}
```

(c) Représentation ASN.1 de la LRC de MAKI.

FIGURE 1 – Représentation ASN.1 du certificat et de la LRC de MAKI.

mat, avec un champ `subjectInfo` vide, un `time_t` de 4 octets pour `UTCTime`, 193 octets pour la clef publique au format OpenSSH, 105 octets pour la signature brute, 1 octet pour `Version`, `Role` et `ReasonCode`, et 2 octets pour `SerialNumber` et `Name`, la taille d'un certificat est de 507 octets.

Comme MAKI ne s'appuie pas sur les autorités d'enregistrement pour délivrer et distribuer les certificats, la distribution des certificats incombe aux agents eux-mêmes. Les agents peuvent diffuser leurs certificats périodiquement et doivent les joindre aux premiers messages de chaque échange. Un agent peut également demander le certificat d'un autre agent. Ces méthodes de distribution sont moins efficaces que la collecte et le partage des certificats par un tiers, mais elles permettent une meilleure évolutivité et décentralisation tout en éliminant tout risque provenant de l'utilisation de cette tierce partie ainsi que tout risque de création de points de défaillance unique.

En ce qui concerne le format de la LRC, nous avons intégré le champ `reasonCode`, jusque-là optionnel, dans les champs obligatoires afin que les AC puissent être tenues responsables de chaque révocation. Nous réservons l'utilisation de la LRC à l'exclusion des agents, ainsi, parmi les dix valeurs possibles, nous n'avons conservé que les champs `KeyCompromise` (renommé `IdCompromise`) et `CessationOfOperation`. D'autres codes pourraient être ajoutés pour indiquer des comportements malveillants spécifiques à l'application. Avec les mêmes choix de format que pour le format de certificat, une LRC contenant $n \in \mathbb{N}$ certificats est de $305 + n \times 397$ octets. La distribution des LRC sont font, avec ou sans demande auprès des ACs.

5 Validation

5.1 Vérification de modèles

Nous avons employé une technique de vérification de modèles pour nous assurer que l'auto-organisation décrite en Section 4.2, partie centrale de MAKI, est valide. Pour cela, nous avons utilisé l'outil Model Checking for Multi-Agent Systems (MCMAS) [15] pour vérifier trois propriétés :

(A) Si l'organisation comprend au moins une AC, chaque agent finira par posséder un certificat valide.

(B) À partir d'un ensemble de None, où au moins un agent possède la capacité de devenir AC, une organisation comportant au moins une AC émerge.

(C) À partir d'une organisation comportant au moins une AC, si tous les ACs disparaissent, le système se réorganise pour retrouver une organisation à un moins une AC.

En utilisant l'enchaînement de (B) puis (A) ou (C) puis (A) nous pouvons être assurés que l'auto-organisation permet bien à chaque agent d'avoir l'opportunité d'obtenir un certificat, un prérequis à la participation dans le système.

La Figure 2 donne un aperçu du modèle d'un agent (Agent1) appliquant l'algorithme d'auto-organisation. Notamment, dans le cas où il reste None, il s'appuie sur un autre agent (Agent2) pour lui délivrer un certificat. L'ensemble des modèles sont disponibles sur le dépôt [3].

Néanmoins, MCMAS ne permet pas de modéliser les évolutions de confiances entre les agents, les résultats que nous obtenons avec ne concernent que des situations où les agents sont bienveillants. De plus, des problématiques, par exemple liées à l'exécution asynchrone des agents, ne sont pas prises en compte.

Ainsi, bien que certaines propriétés peuvent être validées avec une forte confiance en la preuve, ici on fait confiance en MCMAS pour produire des résultats justes. Tester lors d'exécutions est aussi nécessaire pour valider les aspects que nous n'avons pas pu modéliser.

5.2 Preuve de concept

Pour valider le principe général et nous assurer de la faisabilité de MAKI, nous avons développé une preuve de concept,

```
Agent Agent1
 ...
 Protocol:
  -- CA behavior
  role = CA and cert = none : { self_sign };
  role = CA and cert = self_signed and advertised = false : { advertise };
  role = CA and cert = self_signed and advertised = true and cert_asked = true : { deliver_cert };
  -- None behavior
  role = None and cert = none and cert_asked = false and knows_ca = true : { ask_cert };
  Other : { wait };
 end Protocol
 Evolution:
  -- CA evolution
  cert = self_signed if Action = self_sign;
  advertised = true if Action = advertise;
  cert_asked = true if Agent2.Action = ask_cert;
  cert_asked = false if Action = deliver_cert;
  -- None evolution
  cert_asked = true if role = None and Action = ask_cert;
  knows_ca = true if role = None and  Agent2.Action = advertise;
  cert = signed if role = None and Agent2.Action = deliver_cert;
 end Evolution
end Agent
```

FIGURE 2 – Aperçu synthétique d'un modèle MCMAS d'un agent MAKI.

disponible à [3], à l'aide de l'environnement de développement Mesa [12].

Paramétrage Concernant les choix cryptographiques, nous avons suivi les recommandations du NIST [2]. Ainsi, nous avons choisi l'algorithme de signature Elliptic Curve Digital Signature Algorithm, employé avec des clefs de taille 256-bits et basé sur la courbe P-256.

Nous avons aussi eu à choisir un modèle de confiance. Pour cela, nous avons un modèle simple, suffisant pour satisfaire les prérequis de MAKI. Ce modèle comprend une valeur de confiance initiale faible, une contremesure standard pour limiter les changements d'identité fréquents et le maintient simultané de plusieurs identités, ainsi que trois seuils de confiance : *Low*, *Moderate* et *High*. La confiance croit en suivant la fonction donnée en Équation 1 et décroit instantanément à 0 à la moindre détection de malveillance.

$$f : x \mapsto \frac{x}{x + 10} \tag{1}$$

La valeur 10 a été déterminée expérimentalement en fonction des autres paramètres de la simulation, notamment sa durée.

Les seuils *Low*, *Moderate* and *High* ont respectivement été fixés à 0.3, 0.7 et 0.9 et la valeur initiale a été fixée à *Low*. Dans ce modèle, une valeur de confiance inférieure à *Low* implique que l'agent est ignoré.

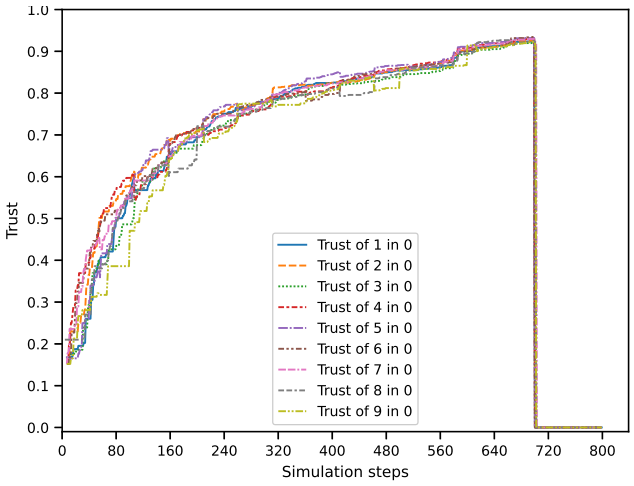
Le modèle inclut aussi les informations de confiance indirectes : un agent peut demander aux autres participants ce qu'ils pensent d'un certain agent. De plus, il est attendu que les agents écoutent toutes les communications pour s'assurer de la cohérence entre les dires et les actions de ses voisins, notamment concernant les distributions de certificats. Cela reste possible car nous ne cherchons pas à protéger la confidentialité des échanges propres à MAKI, ceux-ci restent publiques.

Puisque nous faisons l'hypothèse de la présence d'un protocole de routage correct, nous avons placé les agents de façon à ce qu'ils soient tous à portée de communication les uns des autres et avons développé un protocole très minimaliste dans lequel tous les messages sont diffusés, sans risque de perte et aucun accusé de réception n'est attendu. Néanmoins, les agents ne supposent pas que chaque demande recevra une réponse en temps voulu, voire reçue tout court. Enfin, nous faisons augmenter la confiance des agents de façon aléatoire au fil du temps pour émuler des interactions réussies et alimenter le SGC. Les interactions malveillantes sont, elles aussi, émulées : la détection d'un comportement malveillant est arbitraire et ne sert qu'à déclencher les processus de protection de MAKI, la révocation de certificats et la réorganisation.

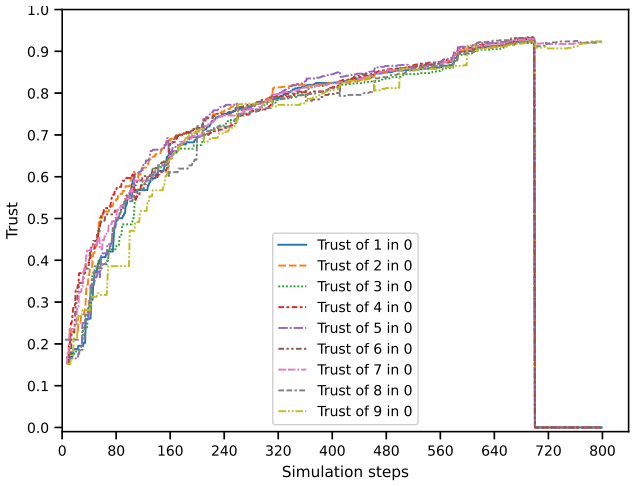
Exécutions Nous avons réalisé plusieurs simulations sous plusieurs scénarios d'attaque et en présentons deux ici. Chaque simulation comprend dix agents, $\mathcal{A}_0$ à $\mathcal{A}_9$, avec seulement $\mathcal{A}_6$, $\mathcal{A}_7$, $\mathcal{A}_8$ et $\mathcal{A}_9$ en capacité d'endosser le rôle d'AC.

Le premier est un scénario simple : l'agent $\mathcal{A}_0$ est détecté comme malveillant à l'étape 700 par $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$, $\mathcal{A}_4$, $\mathcal{A}_5$, et $\mathcal{A}_6$. Ces agents demandent et obtiennent ensuite la révocation du certificat d'$\mathcal{A}_0$, ce mène ce dernier à perdre la confiance du reste du système. La courbe de la confiance en $\mathcal{A}_0$ est donnée en Figure 3a. On y voit bien que la révocation d'un certificat mène à l'exclusion de l'agent certifié. En comparaison, la Figure 3b montre l'évolution de la confiance en désactivant la révocation des certificats. On y voit que, même si la confiance en $\mathcal{A}_0$ diminue un peu de par l'utilisation d'informations indirectes, seule la révocation permet une réelle exclusion de l'agent malveillant.

Le second scénario présenté correspond à une attaque coordonnée de l'ensemble des ACs à un instant donné : à l'étape 700 $\mathcal{A}_6$, $\mathcal{A}_8$ et $\mathcal{A}_9$, les ACs, sont tous détectés comme malveillants par le reste des agents. Ces trois agents sont donc



(a) Évolution de la confiance en $\mathcal{A}_0$ en fonction du temps avec révocation de son certificat.



(b) Évolution de la confiance en $\mathcal{A}_0$ en fonction du temps sans révocation de son certificat.

FIGURE 3 – Graphe montrant l'évolution de la confiance en $\mathcal{A}_0$ en fonction du temps avec et sans révocation de son certificat.

```
1098 main:step:701
1099 agent-7:<>:CertAdvert(Certificate(serial_number: 0, issuer: Identity(7, <public_key>), subject:
         Identity(7, <public_key>), subject_role: Role.CA, not_before: 702, not_after: 4702, version:
         1))
1100 DEBUG:agent:0:net:=>:Data(dest: Identity(7, <public_key>), payload: CertReq())
1101 DEBUG:agent:1:net:=>:Data(dest: Identity(7, <public_key>), payload: CertReq())
1102 DEBUG:agent:3:net:=>:Data(dest: Identity(7, <public_key>), payload: CertReq())
1103 DEBUG:agent:2:net:=>:Data(dest: Identity(7, <public_key>), payload: CertReq())
1104 DEBUG:agent:4:net:=>:Data(dest: Identity(7, <public_key>), payload: CertReq())
1105 DEBUG:agent:5:net:=>:Data(dest: Identity(7, <public_key>), payload: CertReq())
```

FIGURE 4 – Extrait simplifié de la trace d'exécution montrant la réorganisation du système après la perte de confiance dans les ACs.

ignorés, mais le système se retrouve sans AC de confiance. $\mathcal{A}_7$ change donc de rôle et devient AC. Voyant qu'un nouveau AC de confiance est disponible, le reste des agents font appelle à lui. Une partie de la trace d'exécution montrant cette réorganisation est donnée en Figure 4. On y voit $\mathcal{A}_7$ diffuser son nouveau certificat et $\mathcal{A}_0$, $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$, $\mathcal{A}_4$, $\mathcal{A}_5$ leur faire une requête de certificat. Les règles d'auto-organisation de MAKI permettent bien au système de réparer même si tous ses ACs disparaissent ou s'avèrent non dignes de confiance.

6 Conclusion

Dans cet article, nous avons introduit une infrastructure à clefs publiques décentralisée, MAKI, adaptée aux systèmes multi-agents ouverts d'agents embarqués. Cette infrastructure permet de l'utilisation de primitives cryptographiques pour sécuriser les communications entre agents ainsi que d'exclure de possibles intrus. L'exclusion se fait par l'utilisation de certificats délivrés et révoqués par des agents autorité de certification. Ces agents maintiennent un réseau d'autorités dignes de confiance sans requérir à de tierces parties grâce à un système de gestion de confiance. Une preuve de concept de MAKI montrant l'intérêt de l'utilisation de certificats ainsi que sa capacité d'adaptation lors d'attaques est présentée. Une description du processus de validation de l'auto-organisation par vérification de modèles est aussi présentée. Le code source de la preuve de concept et les modèles utilisés pour la validation sont disponibles en ligne [3].
Nous comptons finaliser la validation de MAKI par des simulations basées sur la preuve de concept. Nous explorons également une solution basée sur la blockchain pour fournir un moyen de, plus facilement, partager leurs certificats et d'auditer les autorités de certification.

Remerciements

Ce travail a bénéficié d'une aide de l'État gérée par l'Agence Nationale de la Recherche au titre du programme « Investissements d'avenir » portant la référence ANR-15-IDEX-02.

Références

[1] Agapios Avramidis, Panayiotis Kotzanikolaou, Christos Douligeris, and Mike Burmester. Chord-PKI : A distributed trust infrastructure based on P2P networks. *Computer Networks*, 56(1) :378–398, 2012.

[2] Elaine Barker and Quynh Dang. Recommendation for Key Management Part 3 : Application-Specific Key Management Guidance, 2015.

[3] Arthur Baudet. Code and data presented in jfsma 2023, 2023. https ://doi.org/10.5281/zenodo.7689499.

[4] Arthur Baudet, Oum-El-Kheir Aktouf, Annabelle Mercier, and Philippe Elbaz-Vincent. Systematic Mapping Study of Security in Multi-Embedded-Agent Systems. *IEEE Access*, 9 :154902–154913, 2021.

[5] Sergi Blanch-Torné, Fernando Cores, and Ramiro Moreno Chiral. Agent-based PKI for Distributed Control System. In *2015 World Congress on Industrial Control Systems Security (WCICSS)*, pages 28–35, 2015.

[6] Xavier Bonnaire, Rudyar Cortés, Fabrice Kordon, and Olivier Marin. A Scalable Architecture for Highly Reliable Certification. In *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, pages 328–335, 2013.

[7] Djallel Eddine Boubiche, Samir Athmani, Sabrina Boubiche, and Homero Toral-Cruz. Cybersecurity Issues in Wireless Sensor Networks : Current Challenges and Solutions. *Wireless Personal Communications*, 117(1) :177–213, 2021.

[8] Hui Cui and Robert H. Deng. Revocable and Decentralized Attribute-Based Encryption. *The Computer Journal*, 59(8) :1220–1235, 2016.

[9] Nicholas C. Goffee, Sung Hoon Kim, Sean Smith, Punch Taylor, Meiyuan Zhao, and John Marchesini. Greenpass : Decentralized, PKI-based Authorization for Wireless LANs. In *In 3rd Annual PKI Research and Development Workshop*, pages 26–41, 2004.

[10] ITU-T. X.680-X.693 : Information Technology - Abstract Syntax Notation One (ASN.1) & ASN.1 encoding rules, 2021.

[11] Rutvij H. Jhaveri and Narendra M. Patel. Attack-pattern discovery based enhanced trust model for secure routing in mobile ad-hoc networks. *International Journal of Communication Systems*, 30(7) :e3148, 2017.

[12] Jackie Kazil, David Masad, and Andrew Crooks. Utilizing Python for Agent-Based Modeling : The Mesa Framework. In *Social, Cultural, and Behavioral Modeling*, volume 12268, pages 308–317, 2020.

[13] Deepika Kukreja, S. K. Dhurandher, and B. V. R. Reddy. Power aware malicious nodes detection for securing MANETs against packet forwarding misbehavior attack. *Journal of Ambient Intelligence and Humanized Computing*, 9(4) :941–956, 2018.

[14] Francois Lesueur, Ludovic Me, and Valerie Viet Triem Tong. An efficient distributed PKI for structured P2P networks. In *2009 IEEE Ninth International Conference on Peer-to-Peer Computing*, pages 1–10, 2009.

[15] Alessio Lomuscio, Hongyang Qu, and Franco Raimondi. MCMAS : an open-source model checker for the verification of multi-agent systems. *International Journal on Software Tools for Technology Transfer*, 19(1) :9–30, 2017.

[16] Tatsuaki Okamoto and Katsuyuki Takashima. Decentralized Attribute-Based Encryption and Signatures. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E103.A(1) :41–73, 2020.

[17] Bo Qin, Jikun Huang, Qin Wang, Xizhao Luo, Bin Liang, and Wenchang Shi. Cecoin : A decentralized PKI mitigating MitM attacks. *Future Generation Computer Systems*, 107 :805–815, 2020.

[18] Ankush Singla and Elisa Bertino. Blockchain-Based PKI Solutions for IoT. In *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*, pages 9–15, 2018.

[19] Alexander Yakubov, Wazen M. Shbair, Anders Wallbom, David Sanda, and Radu State. A Blockchain-Based PKI Management Framework. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–6, 2018.

Exploiter l'équité d'un modèle d'apprentissage pour reconstruire les attributs sensibles de son ensemble d'entraînement

J. Ferry[1], U. Aïvodji[2], S. Gambs[3], M-J. Huguet[1], M. Siala[1]

[1] LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France
[2] École de Technologie Supérieure, Montréal, Canada
[3] Université du Québec à Montréal, Montréal, Canada

jferry@laas.fr

Résumé

Pour palier les biais non désirés en apprentissage supervisé, de nombreux travaux utilisent des métriques d'équité statistique, définies vis-à-vis de certains attributs sensibles. *Bien que ceux-ci ne soient généralement pas utilisés par le modèle appris au moment de l'inférence, ils le sont souvent pendant son entraînement pour contrôler l'équité. Nous montrons ainsi qu'un attaquant disposant d'un accès en boîte noire à un tel modèle peut utiliser le fait qu'il soit équitable pour reconstruire les attributs sensibles de son ensemble d'entraînement. L'approche proposée consiste à* corriger *une première reconstruction effectuée par un attaquant de la littérature, pour se conformer avec l'information de l'équité. Notre large évaluation expérimentale confirme que ce processus de correction permet d'améliorer les performances de l'attaque de manière significative.*

Mots-clés

Attaque de reconstruction, vie privée, équité, apprentissage, programmation linéaire en nombres entiers, programmation par contraintes.

Abstract

To face the undesirable biases in machine learning, a growing body of work consider statistical fairness metrics, defined with respect to some sensitive *attributes. Even though the later are generally not used by the learnt model for inference, they are often required at training time to ensure fairness. We then show that an adversary with black-box access to such model can leverage the fact that it is fair to reconstruct the sensitive attributes of its training set. The proposed approach consists in* correcting *a baseline reconstruction made by some adversary from the literature to comply with the fairness information. Our thorough experimental study demonstrates that this correction process significantly improves the performances of the performed attack.*

Keywords

Reconstruction attack, privacy, fairness, machine learning, integer linear programming, constraint programming.

1 Introduction

L'utilisation croissante de modèles d'apprentissage pour la prise de décisions à forts enjeux (*e.g.,* admissions à l'université, prédiction de récidive…) soulève de nombreux risques éthiques, parmi lesquels celui de discrimination. Pour faire face à cette problématique, de nombreux travaux proposent d'apprendre des modèles respectant des contraintes d'équité, exprimées vis-à-vis de certains attributs *sensibles* [3, 7, 23]. Ces derniers correspondent aux caractéristiques telles que le genre, l'âge ou l'origine ethnique [10], qui ne devraient pas influer sur les processus de prise de décision impactant les individus, pour des raisons légales, éthiques, sociales ou philosophiques [3].

Un autre aspect fondamental pour un apprentissage responsable est la protection de la vie privée. En effet, les modèles d'apprentissage sont souvent entraînés sur de grandes quantités de données personnelles. Il est alors important de s'assurer que ces modèles apprennent des motifs génériques utiles, sans révéler d'informations privées sur un ou plusieurs individus en particulier. Dans ce contexte, les *attaques d'inférence* [9, 14] visent à utiliser le résultat d'un calcul (*e.g.,* un modèle entraîné) pour retrouver des informations sur ses entrées (*e.g.,* un ensemble d'entraînement). Notre travail appartient à la famille des *attaques de reconstruction* (de jeux de données), dans lesquelles un attaquant essaie de reconstruire tout ou partie de l'ensemble d'entraînement d'un modèle [9]. Nous considérons le cas dans lequel l'attaquant tente de reconstruire la colonne des attributs sensibles de l'ensemble d'entraînement.

En fonction des données en sa possession *(connaissances adversariales)*, un attaquant peut adopter différentes stratégies pour reconstruire les attributs sensibles de l'ensemble d'entraînement d'un modèle. Nous proposons une méthode de post-traitement intitulée *correction de reconstruction*, qui prend en entrée une reconstruction initiale effectuée par un attaquant de la littérature, éventuellement associée à des scores de confiance pour chaque élément. Notre méthode modifie ensuite cette reconstruction initiale de manière à se conformer avec certaines contraintes définies par l'utilisateur. Notre travail considère le scénario dans lequel ces contraintes sont des contraintes d'équité, et l'attaquant uti-

TABLE 1 – Résumé des métriques d'équité considérées

Ref.	Métrique	Mesure égalisée	Expression de la contrainte
[13]	Statistical Parity (SP)	Probabilité de prédiction positive	$\forall s,\ \lvert\mathbb{P}(\hat{y}=1)-\mathbb{P}(\hat{y}=1 \mid s)\rvert \leq \epsilon$
[8]	Predictive Equality (PE)	Taux de Faux Positifs	$\forall s,\ \lvert\mathbb{P}(\hat{y}=1 \mid y=0)-\mathbb{P}(\hat{y}=1 \mid s,\ y=0)\rvert \leq \epsilon$
[19]	Equal Opportunity (EO)	Taux de Vrais Positifs	$\forall s,\ \lvert\mathbb{P}(\hat{y}=1 \mid y=1)-\mathbb{P}(\hat{y}=1 \mid s,\ y=1)\rvert \leq \epsilon$
[19]	Equalized Odds (EOdds)	Taux de Faux Positifs et de Vrais Positifs	Conjonction de la Predictive Equality et de l'Equal Opportunity

lise le fait qu'un modèle soit équitable pour améliorer sa reconstruction initiale. Cette *information de l'équité* peut être liée à des obligations légales, comme par exemple la "règle des 80 pourcents" [16] de la Commission américaine pour l'égalité des chances dans l'emploi *(US Equal Employment Opportunity Commission)* [15].

Les tensions entre équité et protection de la vie privée en apprentissage ont été récemment étudiées à travers les conflits théoriques et pratiques entre les métriques d'équité statistique et la confidentialité différentielle, méthode standard en protection de la vie privée [18]. Notre travail prend une direction différente mais démontre également que le fait d'imposer des contraintes d'équité pendant l'apprentissage peut compromettre la confidentialité des attributs sensibles. Ce travail complet a été présenté à la conférence internationale SATML 2023 *(The First IEEE Conference on Secure and Trustworthy Machine Learning)* [17].

2 Contexte et travaux antérieurs

2.1 Classification équitable

Soit M le nombre d'*attributs non sensibles* caractérisant un exemple. Pour $j \in \{1..M\}$, $\mathcal{X}_j$ est le domaine des valeurs possibles pour l'attribut j, qui peut être catégorique ou numérique, et $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_M$. De même, soit $\mathcal{S}$ (*resp.,* $\mathcal{Y}$) le domaine d'un *attribut sensible* (catégorique) (*resp., label*). $D = (X, S, Y)$ est un jeu de données issu d'une distribution sous-jacente (inconnue) sur $\mathcal{X} \times \mathcal{S} \times \mathcal{Y}$. Soit N le nombre d'*exemples* dans D, où $e_{i \in \{1..N\}} = (x, s, y) \in \mathcal{X} \times \mathcal{S} \times \mathcal{Y}$.

L'objectif d'un algorithme d'apprentissage supervisé est de construire un *classifieur* $\mathcal{L}(D) = h$ mappant l'espace des attributs à celui des labels. L'utilisation explicite d'un attribut sensible est en général interdite par la loi pour éviter un *traitement disparate* [4]. On considère donc que l'attribut sensible (utilisé pendant l'entraînement pour s'assurer de l'équité du modèle construit) n'est pas utilisé pour l'inférence : $h : \mathcal{X} \mapsto \mathcal{Y}$, et $\hat{Y} = h(X)$ sont les prédictions du modèle. De manière consistante avec la littérature de l'équité en apprentissage, on considère la tâche de classification binaire dans ce travail : $\mathcal{Y} = \{0, 1\}$. Néanmoins, notre approche peut facilement être étendue au cas de la classification non-binaire, étant donné des contraintes d'équité formulées dans ce cadre plus général.

Pour s'assurer que les modèles d'apprentissage ne reproduisent ni ne créent de *biais indésirables* (*e.g.*, menant à des discriminations), différentes notions d'équité ont été proposées dans la littérature [23]. Dans ce travail, nous considérons le cas dans lequel l'équité est exprimée avec des *métriques d'équité statistique* [13]. Celles-ci visent à s'assurer qu'une grandeur (*e.g.*, taux de vrais positifs) diffère d'au plus une valeur de tolérance donnée ϵ entre différents *groupes protégés* (définis par les attributs sensibles). De nombreuses méthodes ont été proposées [3, 7, 23] afin d'apprendre des modèles *équitables*. Elles peuvent être divisées en trois principales catégories, en fonction de l'étape du pipeline d'apprentissage à laquelle elles interviennent [5]. Les méthodes de *pre-processing* visent à atténuer les corrélations indésirables directement dans le jeu de données d'entraînement, avant d'utiliser des techniques classiques d'apprentissage sur le jeu de données ainsi modifié [21]. Les techniques d'*in-processing* consistent à modifier l'algorithme d'apprentissage lui-même, afin d'apprendre des modèles équitables. Enfin, les approches de *post-processing* [19] modifient les sorties d'un modèle préentraîné pour satisfaire certains critères d'équité.

Notre approche est agnostique au type de méthode utilisée pour apprendre le modèle équitable, puisque qu'elle dépend uniquement de ses prédictions et de l'information sur son équité. Nous utilisons quatre métriques d'équité statistique populaires dans la littérature : la Statistical Parity [13], la Predictive Equality [8], l'Equal Opportunity [19] et l'Equalized Odds [19]. Ces métriques sont présentées dans la Table 1, ainsi que les mesures qu'elles visent à égaliser entre les différents groupes protégés, et l'expression mathématique des contraintes associées.

2.2 Attaques de reconstruction

Notre approche est une *attaque d'inférence* [14], visant à retrouver des informations sur un jeu de données en observant les sorties d'un calcul sur celui-ci. Ici, le calcul en question est un algorithme d'apprentissage, et sa sortie est un modèle entraîné. Différents types d'attaques d'inférence ont été proposés contre les modèles d'apprentissage [9]. Notre attaque d'inférence est une *attaque de reconstruction* [9]. Elle nécessite seulement un accès en *boîte noire* au modèle équitable entraîné (*i.e.,* via une API de prédiction) et est agnostique au type de modèle, à l'algorithme d'apprentissage et à la méthode utilisée pour assurer l'équité.

Les attaques de reconstruction ont d'abord été étudiées dans le contexte des mécanismes d'accès aux bases de données. Dans le scénario considéré, une base de données contient des informations sur des individus, chaque exemple étant composé d'informations non privées ainsi que d'un *bit privé* (un par exemple/individu) [14]. L'objectif d'un attaquant est alors de reconstruire la colonne des *bits privés* de la base de données. Pour cela, il effectue des requêtes au mécanisme d'accès, dont les sorties sont des agrégations bruitées des bits privés des individus. De telles attaques de reconstruction ont été introduites et formalisées dans [11],

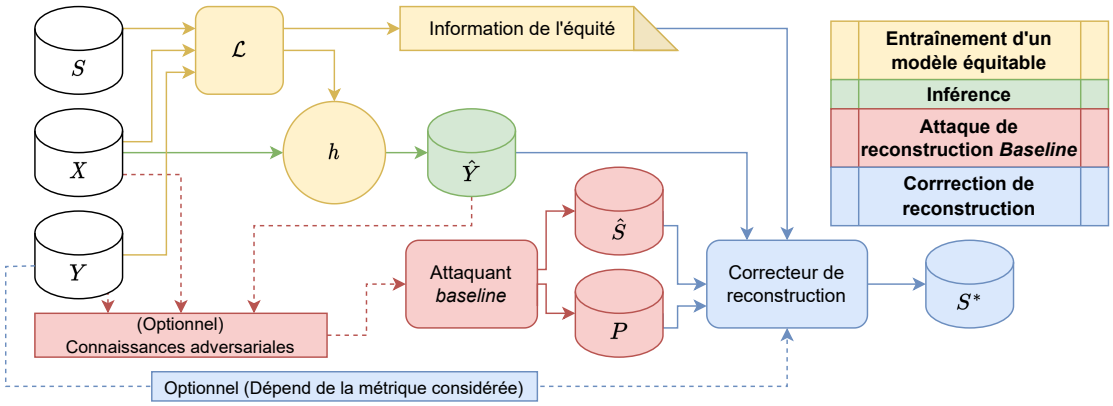


FIGURE 1 – Schéma de l'attaque proposée. Un modèle h est construit par la procédure d'apprentissage équitable $\mathcal{L}$ et utilisé pour l'inférence. Ensuite, un attaquant *baseline* tente de reconstruire les attributs sensibles S de l'ensemble d'entraînement de h. Cet attaquant produit une reconstruction $\hat{S}$ accompagnée de scores de confiance (optionnels) P. Notre contribution se situe au niveau du *Correcteur de reconstruction*, qui prend en entrée la reconstruction de l'attaquant *baseline* $\hat{S}$ et la modifie pour se conformer à l'information de l'équité, produisant ainsi S^*, la version corrigée de la reconstruction des attributs sensibles.

où elles sont formulées en utilisant un programme linéaire. Notre objectif est similaire à ces travaux : nous voulons reconstruire une *colonne* du jeu de données en utilisant la sortie d'un calcul utilisant cette colonne. Cependant, une différence fondamentale se trouve dans la nature du mécanisme accédant les données privées. D'un côté, les mécanismes d'accès aux bases de données considérés utilisent l'information privée pour calculer le résultat de chaque requête. De l'autre, dans notre cas, les attributs sensibles ne sont plus jamais utilisés au moment de l'inférence, et toute l'information les concernant est dévoilée en une seule fois, à travers le modèle entraîné lui-même ou ses prédictions.

Deux travaux sont proches du scénario considéré [1, 20]. D'un côté, [1] propose une attaque pour inférer les attributs sensibles d'un exemple étant donné les sorties d'un modèle pour cet exemple. En résumé, l'attaquant entraîne un modèle d'apprentissage en utilisant un *ensemble d'attaque* séparé, pour lequel il connaît les attributs sensibles. Cette attaque correspond à l'attaquant *baseline* considéré dans nos expériences (*cf.* Section 4.1). De l'autre côté, [20] propose un mécanisme dont le principe est assez proche de notre travail, mais considère une configuration très particulière où l'apprentissage est effectué de manière distribuée entre deux entités : un *learner* souhaitant apprendre un modèle équitable sur un certain jeu de données pour lequel il ne connaît pas les attributs sensibles, et un tiers qui les connaît. Le *learner* envoie itérativement les paramètres du modèle qu'il est en train de construire à ce tiers, qui lui indique si le modèle actuel respecte les contraintes d'équité. Il encode ensuite cette séquence d'informations d'équité dans un modèle de programmation en nombres entiers pour effectuer la reconstruction des attributs sensibles de l'ensemble d'entraînement. Tandis que l'intuition est similaire, notre travail couvre un cas d'usage plus général (en ne faisant aucune hypothèse sur l'algorithme d'apprentissage équitable utilisé) dans un cadre moins favorable (l'attaquant possédant uniquement l'information d'équité sur le modèle final).

3 Améliorer une reconstruction des attributs sensibles grâce à l'équité

Nous décrivons à présent l'approche proposée et montrons comment l'information sur l'équité d'un modèle peut être utilisée pour améliorer une reconstruction des attributs sensibles de son ensemble d'entraînement. Nous présentons le cadre considéré et positionnons le composant de *correction de reconstruction* que nous proposons, avant de décrire deux implémentations possibles pour celui-ci.

3.1 Principe de l'attaque

La Figure 1 illustre les différents composants de l'approche considérée. A partir d'un ensemble $D = (X, S, Y)$, un modèle h est entraîné par un algorithme d'apprentissage équitable $\mathcal{L}$, qui s'assure que h est équitable sur D selon une métrique d'équité définie par rapport à un certain attribut sensible S. Le classifieur h n'utilise pas l'attribut sensible S pour l'inférence afin d'éviter le *traitement disparate* [4]. Ainsi, h effectue ses prédictions en utilisant uniquement les attributs non sensibles X. Notre approche ne fait aucune hypothèse sur l'algorithme d'apprentissage équitable utilisé $\mathcal{L}$. En effet, le seul pré-requis de notre attaque est la connaissance de l'information de l'équité.

L'objectif de l'attaque est de reconstruire les attributs sensibles S de l'ensemble d'entraînement. Dans le schéma considéré, S est seulement utilisé par $\mathcal{L}$ pour s'assurer de l'équité de h (et n'est plus utilisé par la suite). Dans une première étape de l'attaque, un *attaquant baseline* effectue une première reconstruction $\hat{S}$ de S, en utilisant éventuellement certaines connaissances auxiliaires. L'attaquant *baseline* produit également un vecteur de probabilités P, qui reflète sa confiance dans chaque composant de sa reconstruction $\hat{S}$. Si l'attaquant ne calcule pas de scores de confiance, alors P correspond simplement au vecteur identité. Dans une seconde étape de l'attaque, un *correcteur de reconstruction* prend en entrée la reconstruction de l'at-

taquant *baseline* $\hat{S}$ et ses scores de confiance P. Il produit une nouvelle reconstruction S^* minimisant les changements (pondérés par les scores de confiance) par rapport à la reconstruction de l'attaquant *baseline*, tout en s'assurant du respect de certaines propriétés, telles que des contraintes d'équité statistique. Pour s'assurer du respect de telles contraintes, le *correcteur de reconstruction* nécessite également l'information de l'équité, les prédictions du modèle cible $\hat{Y}$ sur son ensemble d'entraînement, ainsi que (selon la métrique d'équité considérée, *cf.* Table 1) les vrais labels Y. Aucune hypothèse n'est faite sur le modèle cible h, qui peut être vu comme une boîte noire puisque l'attaque nécessite seulement l'accès à ses prédictions.

Le succès de l'attaque peut être évalué comme la *précision de la reconstruction* de S^* (*i.e.*, proportion d'éléments de S correctement prédits dans S^*). La contribution de notre travail se situe au niveau du *correcteur de reconstruction* qui, en incorporant uniquement l'information de l'équité, est capable d'améliorer significativement la qualité de la reconstruction des attributs sensibles. Cette amélioration peut être quantifiée en comparant la précision de la reconstruction de l'attaquant *baseline* $\hat{S}$ et celle de la version corrigée S^*.

3.2 Correcteur de reconstruction général

Nous présentons maintenant $\mathcal{RC}(\hat{S}, P, \hat{Y}, \epsilon)$, un modèle de Programmation Linéaire en Nombres Entiers implémentant le correcteur de reconstruction de la Figure 1, dans le cas d'un attribut sensible binaire. Son objectif est de modifier la reconstruction des attributs sensibles de l'ensemble d'entraînement de l'attaquant *baseline* pour se conformer à certaines contraintes (ici, l'information de l'équité) tout en minimisant les changements effectués (pondérés par la confiance de l'attaquant).

Entrées

- $\hat{s}_i \in \{0,1\},\ i = 1, \ldots, N$ (reconstruction initiale de l'attaquant *baseline*)
- $p_i \in \{0,1\},\ i = 1, \ldots, N$ (confiance de l'attaquant *baseline* pour $\hat{s}_i$)
- $\hat{y}_i \in \{0,1\},\ i = 1, \ldots, N$ (prédictions de h)
- Information de l'équité : h respecte une contrainte d'équité selon une métrique (*e.g.*, SP) et une valeur de tolérance ϵ

Variables de décision

- $s_i^* \in \{0,1\},\ i = 1, \ldots, N$ (reconstruction corrigée)

Modèle $\mathcal{RC}(\hat{S}, P, \hat{Y}, \epsilon)$

$$\min \sum_{i=1}^{N} (p_i \cdot (1 - \hat{s}_i) \cdot s_i^*) + \sum_{i=1}^{N} (p_i \cdot \hat{s}_i \cdot (1 - s_i^*)) \quad (1)$$

$$s.t. : 0 < \sum_{i=1}^{N} s_i^* < N \quad (2)$$

$$-\epsilon \leq \frac{\sum_{i=1}^{N} \hat{y}_i}{N} - \frac{\sum_{i=1}^{N} \hat{y}_i \cdot s_i^*}{\sum_{i=1}^{N} s_i^*} \leq \epsilon \quad (3)$$

$$-\epsilon \leq \frac{\sum_{i=1}^{N} \hat{y}_i}{N} - \frac{\sum_{i=1}^{N} \hat{y}_i \cdot (1 - s_i^*)}{\sum_{i=1}^{N} (1 - s_i^*)} \leq \epsilon \quad (4)$$

L'objectif (1) vise à minimiser les changements à $\hat{S}$ (pondérés par la confiance de l'attaquant). Chaque modification d'un élément $\hat{s}_i$ de la reconstruction initiale de l'attaquant *baseline* est pénalisée avec un coût p_i et le modèle minimise le coût total. La contrainte (2) s'assure que la reconstruction contienne au moins un exemple dans chaque groupe protégé. Enfin, les contraintes (3) et (4) encodent la métrique d'équité Statistical Parity. La contrainte (3) (*resp.,* la contrainte (4)) s'assure que le Taux de Prédictions Positives (TPP) sur le groupe 1 (*resp.,*, sur le groupe 0) diffère d'au plus ϵ du TPP global. Les prédictions $\hat{y}_i$ du modèle étant fixées, la contrainte d'équité est satisfaite en modifiant la reconstruction des attributs sensibles s_i^*.

Finalement, une solution optimale à notre modèle général de correction de reconstruction $\mathcal{RC}(\hat{S}, P, \hat{Y}, \epsilon)$ est une affectation des variables binaires s_i^* minimisant (1) tout en satisfaisant les contraintes (2) à (4). L'affectation S^* correspond aux changements de coût minimum à la reconstruction initiale de l'attaquant *baseline* $\hat{S}$ de sorte à satisfaire les contraintes d'équité. Si les changements effectués sont corrects la majorité du temps (ce qui est attendu si les scores de confiance fournis par l'attaquant *baseline* sont de qualité), alors la précision globale de la reconstruction sera améliorée. Dans tous les cas, l'algorithme est garanti de trouver une solution satisfaisant les contraintes d'équité - ce qui n'était pas forcément le cas de la reconstruction *baseline*. Par ailleurs, puisqu'il est capable de modifier les attributs sensibles de tous les exemples d'entraînement, le modèle peut atteindre n'importe quelle valeur des métriques d'équité dans la version corrigée de la reconstruction. Ainsi, la connaissance précise de la violation de l'équité (plutôt qu'une simple borne supérieure via ϵ) pourrait être utilisée pour réduire encore l'espace des reconstructions admissibles et améliorer les performances de la correction de reconstruction. Enfin, puisqu'il encode explicitement l'attribut sensible de chaque exemple de l'ensemble d'entraînement, $\mathcal{RC}(\hat{S}, P, \hat{Y}, \epsilon)$ peut être utilisé pour formuler n'importe quelle contrainte sur ces derniers.

3.3 Correcteur de reconstruction efficace

L'espace de recherche du modèle général de correction de reconstruction $\mathcal{RC}(\hat{S}, P, \hat{Y}, \epsilon)$ grandit exponentiellement avec le nombre d'exemples d'entraînement N. En effet, chaque élément du vecteur des attributs sensibles S étant considéré indépendamment des autres (et représenté comme une variable de décision binaire), la taille de l'espace de recherche est en $O(2^N)$, ce qui limite le passage à l'échelle. Cependant, une telle granularité n'est pas nécessaire lorsqu'on considère des métriques d'équité statistique. Plus précisément, pour satisfaire la contrainte d'équité, le correcteur de reconstruction peut réaliser exactement quatre actions différentes : (i) changer un élément de la reconstruction initiale $\hat{s}_i$ de 1 en 0 pour un exemple tel que $\hat{y}_i = 1$ (ii) changer $\hat{s}_i$ de 0 en 1, pour un exemple tel que $\hat{y}_i = 1$, (iii) changer $\hat{s}_i$ de 1 en 0, pour un exemple tel que $\hat{y}_i = 0$, ou (iv) changer $\hat{s}_i$ de 0 en 1, pour un exemple tel que $\hat{y}_i = 0$. Par ailleurs, pour l'opération choisie, le modèle commencera toujours par les exemples avec les scores de confiance les

plus faibles (puisqu'on minimise le coût des changements). Soit n_1^+ le nombre d'exemples d'entraînement prédits positivement par le modèle cible h et assignés au groupe 1 dans la reconstruction initiale de l'attaquant : $n_1^+ = \sum_{i=1}^{N} \hat{s}_i \cdot \hat{y}_i$. De même, soit $n_0^+ = \sum_{i=1}^{N} (1-\hat{s}_i) \cdot \hat{y}_i$, $n_1^- = \sum_{i=1}^{N} \hat{s}_i \cdot (1-\hat{y}_i)$, et $n_0^- = \sum_{i=1}^{N} (1-\hat{s}_i) \cdot (1-\hat{y}_i)$. Ces quatre nombres n_1^+, n_0^+, n_1^- et n_0^- sont les cardinalités des quatre groupes d'exemples définissant les quatre opérations possibles (*resp.,* (i), (ii), (iii) et (iv)) du point de vue de l'équité. Pour chaque groupe, nous trions et cumulons les scores de confiance associés à ces exemples pour obtenir les tableaux suivants : T_{1^+}, T_{0^+}, T_{1^-} et T_{0^-}. Par exemple, T_{1^+} contient les scores de confiance associés aux n_1^+ exemples prédits positivement par h et assignés au groupe 1 dans la reconstruction initiale de l'attaquant. $T_{1^+}[i]$ est la somme des i scores de confiance les plus bas au sein de ce groupe d'exemples. Ainsi, $T_{1^+}[i]$ est exactement le coût minimal pour changer la valeur de l'attribut sensible reconstruit de 1 à 0 pour i exemples prédits positivement par h. Nous utilisons quatre variables de décision entières, modélisant le nombre de fois où chacune des quatre opérations est effectuée pour corriger la reconstruction. Nous définissons alors notre modèle efficace de correction de reconstruction : $\mathcal{RC}_{\mathcal{E}}(\hat{S}, P, \hat{Y}, \epsilon)$.

Données

- Cardinalités de la reconstruction initiale n_1^+, n_0^+, n_1^- et n_0^-.
- Tableaux des scores de confiance de l'attaquant, triés et cumulés T_{1^+}, T_{0^+}, T_{1^-} et T_{0^-}.
- Information de l'équité : h respecte une contrainte d'équité selon une métrique (*e.g.*, SP) et une valeur de tolérance ϵ

Variables de décision

- $s_{01}^+ \in [0, n_0^+]$: nombre de changements de $\hat{s}_i$ de 0 en 1, pour des exemples tels que $\hat{y}_i = 1$.
- $s_{10}^+ \in [0, n_1^+]$: nombre de changements de $\hat{s}_i$ de 1 en 0, pour des exemples tels que $\hat{y}_i = 1$.
- $s_{01}^- \in [0, n_0^-]$: nombre de changements de $\hat{s}_i$ de 0 en 1, pour des exemples tels que $\hat{y}_i = 0$.
- $s_{10}^- \in [0, n_1^-]$: nombre de changements de $\hat{s}_i$ de 1 en 0, pour des exemples tels que $\hat{y}_i = 0$.

Modèle $\mathcal{RC}_{\mathcal{E}}(\hat{S}, P, \hat{Y}, \epsilon)$

$$\min T_{0^+}[s_{01}^+] + T_{1^+}[s_{10}^+] + T_{0^-}[s_{01}^-] + T_{1^-}[s_{10}^-] \quad (5)$$

$$s.t. : n_0^+ + n_0^- - s_{01}^+ - s_{01}^- + s_{10}^+ + s_{10}^- > 0 \quad (6)$$

$$n_1^+ + n_1^- - s_{10}^+ - s_{10}^- + s_{01}^+ + s_{01}^- > 0 \quad (7)$$

$$-\epsilon \leq \frac{\sum_{i=1}^{N} \hat{y}_i}{N} - \frac{n_1^+ - s_{10}^+ + s_{01}^+}{n_1^+ + n_1^- - s_{10}^+ - s_{10}^- + s_{01}^+ + s_{01}^-} \leq \epsilon \quad (8)$$

$$-\epsilon \leq \frac{\sum_{i=1}^{N} \hat{y}_i}{N} - \frac{n_0^+ - s_{01}^+ + s_{10}^+}{n_0^+ + n_0^- - s_{01}^+ - s_{01}^- + s_{10}^+ + s_{10}^-} \leq \epsilon \quad (9)$$

Tout comme pour le modèle général, l'objectif (5) minimise les changements effectués (pondérés par la confiance de l'attaquant). Il peut être implémenté efficacement en utilisant des contraintes `element` au sein d'un solveur de Programmation par Contraintes (PPC). De telles contraintes sont utilisées pour accéder à un tableau de constantes à la position donnée par la valeur d'une variable : $T_{0^+}[s_{01}^+] = \texttt{element}(T_{0^+}, s_{01}^+)$. Par ailleurs, pour minimiser uniquement le nombre de changements, il est également possible de simplement sommer les quatre variables de décision. L'objectif devient alors linéaire et le modèle peut dans ce cas être résolu par un solveur PLNE générique. Les contraintes (6) et (7) s'assurent simplement que la reconstruction contienne au moins un exemple de chaque groupe protégé. Enfin, les contraintes (8) et (9) encodent la contrainte d'équité pour la métrique Statistical Parity. Plus généralement, $\mathcal{RC}_{\mathcal{E}}(\hat{S}, P, \hat{Y}, \epsilon)$ peut être utilisé pour encoder n'importe quelle contrainte de taux (utilisant les attributs sensibles) sur les prédictions du modèle cible h, ce qui inclue toutes les métriques d'équité statistique.

Une fois le modèle résolu, les affectations optimales des quatre variables de décision définissent les changements de coût minimal (pondérés par la confiance de l'attaquant) qui doivent être effectués pour rétablir l'équité. Dans une étape de post-traitement, les mouvements associés sont appliqués aux exemples correspondants, par ordre croissant des scores de confiance (de telle sorte que le coût global soit exactement la valeur de la fonction objectif (5) du modèle résolu). On obtient alors la version corrigée de la reconstruction S^*.

3.4 Généralisation des modèles

Les modèles décrits encodent directement la métrique d'équité Statistical Parity, mais peuvent également être utilisés pour corriger des reconstructions d'attributs sensibles à partir des autres métriques présentées dans la Table 1.

En effet, la Predictive Equality (PE) vise à égaliser les taux de Faux Positifs (entre les différents groupes protégés), ce qui est équivalent à la satisfaction de la Statistical Parity *sur le sous-ensemble des exemples négatifs* de l'ensemble d'entraînement. Ainsi, il est possible d'utiliser directement le modèle de correction de reconstruction proposé, en l'applicant uniquement sur ce sous-ensemble. En effet, la PE ne donne aucune information sur les exemples positifs. De la même manière, l'Equal Opportunity égalise les taux de Vrais Positifs, et la correction de reconstruction peut être faite en utilisant les modèles proposés *sur le sous-ensemble des exemples positifs* de l'ensemble d'entraînement. Enfin, pour la métrique Equalized Odds, il est possible d'appliquer successivement la méthode de correction de reconstruction de la Predictive Equality et celle de l'Equal Opportunity.

4 Evaluation expérimentale

Dans cette section, nous présentons une évaluation expérimentale de notre approche de correction de reconstruction. Nous considérons trois jeux de données de la littérature, présentant des caractéristiques (taille, attribut sensible, prédiction) variées, quatre métriques d'équité et de nombreuses valeurs de tolérance ϵ. Nous décrivons tout d'abord l'attaquant *baseline* considéré et la configuration de nos expériences avant de présenter les résultats obtenus.

4.1 Reconstruction *baseline*

Nous instancions le cadre décrit dans la Figure 1 avec un attaquant de la littérature, noté $\mathcal{A}'$ (implémentant le composant "attaquant *baseline*"). De manière consistante avec la littérature des attaques de reconstruction [11], nous considérons que le jeu de données contient une *"grande quantité d'information identifiante non-privée et un bit secret, un par individu."* [14]. Ici, le bit secret (privé) de chaque individu i est la valeur de son attribut sensible s_i. L'attaquant *baseline* $\mathcal{A}'$ connaît ainsi les attributs non sensibles de l'ensemble d'entraînement X et les labels Y (*i.e.*, toutes les colonnes de l'ensemble d'entraînement à l'exception de la colonne *secrète*, qui est celle de l'attribut sensible dans notre cas). Par ailleurs, $\mathcal{A}'$ a également accès à un *ensemble d'attaque*, $D_A = (X_A, S_A, Y_A)$ issu de la même distribution que l'ensemble d'entraînement (mais disjoint avec celui-ci). Cet ensemble d'attaque modélise la connaissance d'une approximation de la distribution des attributs sensibles par rapport aux attributs non sensibles et aux labels. En effet, l'utilisation d'un tel *ensemble d'attaque* pour entraîner un *modèle d'attaque* est cohérent avec la littérature [1].

L'attaquant $\mathcal{A}'$, proposé par [1], a donc accès à toute l'information dont notre correcteur de reconstruction aura besoin (en réalité, même davantage), ce qui constitue l'attaquant *baseline* le plus fort, que nous allons comparer avec notre correction de reconstruction. En résumé, $\mathcal{A}'$ a accès à un ensemble d'attaque $D_A = (X_A, S_A, Y_A)$, aux attributs non sensibles de l'ensemble d'entraînement X et à ses labels Y. Il a également un accès en boîte noire au modèle entraîné h, qui lui permet de connaître ses prédictions sur l'ensemble d'entraînement $\hat{Y} = h(X)$ et sur l'ensemble d'attaque $\hat{Y}_A = h(X_A)$. Cet ensemble d'attaque lui permet d'entraîner un modèle d'apprentissage à prédire S_A à partir de $(X_A, Y_A, \hat{Y}_A)$. Enfin, il utilise ce *modèle d'attaque* pour prédire $\hat{S}$ étant donné $(X, Y, \hat{Y})$.

4.2 Scores de confiance

Le *modèle d'attaque* de l'attaquant *baseline* effectuant une tâche de classification binaire (puisqu'on considère le cas des attributs sensibles binaires), ses scores de confiance se situent naturellement entre 0.5 et 1.0. Utiliser directement ces scores pour pondérer l'objectif de notre problème de correction de reconstruction signifierait donc que modifier une prédiction dont le score de confiance serait 1.0 (l'attaquant était certain de sa reconstruction) serait moins coûteux que de modifier deux prédictions avec un score de confiance de 0.51 (pour lesquelles l'attaquant n'était pas du tout sûr). Pour encourager la correction de reconstruction à se concentrer sur les prédictions associées aux scores de confiance les plus faibles, nous normalisons tous les scores de confiance avant de leur appliquer un même exposant $k \geq 1$ pour accroître leurs différences. En pratique, l'exposant k est choisi pour maximiser la qualité de la reconstruction obtenue sur un ensemble de validation $D'_A \subset D_A$. En résumé, l'attaquant $\mathcal{A}'$ produit une reconstruction $\hat{S} = \{\hat{s}_{i \in \{1...N\}}\}$ des attributs sensibles de l'ensemble d'entraînement, accompagnée de scores de confiance $P = \{p_{i \in \{1...N\}}\}$ normalisés et mis à la puissance k.

4.3 Configuration

Jeux de données Nous considérons trois jeux de données de tailles différentes, et sélectionnons un attribut sensible différent pour chacun d'eux afin d'obtenir des scénarios suffisamment variés. Le premier jeu de données utilisé est UCI Adult Income [12], un jeu de données très populaire dans la littérature de l'équité. Il rassemble des données sur le recensement de 1994 aux États-Unis, et la tâche associée est de prédire si une personne gagne plus de $50\,000$\$ par an. L'attribut sensible considéré est le genre (femme/homme). Nous utilisons également deux jeux de données construits à partir des résultats d'une enquête du bureau américain de recensement intitulée "American Community Survey (ACS) Public Use Microdata Sample (PUMS)". Plus précisément, ces jeux de données sont issus des données collectées dans l'état du Texas en 2018. Le second, nommé ACSPublicCoverage [10], contient des données sur des individus âgés de moins de 65 ans et ayant un revenu inférieur à $30\,000$\$, la tâche associée étant de prédire s'ils sont couverts par une assurance-maladie publique. L'âge sert ici d'attribut sensible (quartile le plus jeune/autres). Enfin, le troisième jeu de données, ACSIncome [10], rassemble des informations sur des individus âgés de plus de 16 ans, qui ont indiqué travailler au moins une heure par semaine l'année passée, pour un revenu d'au moins 100\$. Comme pour UCI Adult Income, la tâche de classification associée est de prédire si les individus gagnent plus ou moins de $50\,000$\$ par an. Nous utilisons une version binarisée de l'origine ethnique ("blancs"/autres) comme attribut sensible.

Ces informations sont synthétisées dans la Table 2. Pour toutes les expériences, chaque jeu de données est partagé entre un ensemble d'entraînement $\left(\frac{1}{3}\right)$, un ensemble de test $\left(\frac{1}{3}\right)$ et un ensemble d'attaque $\left(\frac{1}{3}\right)$. Ici, l'ensemble de test sert uniquement à s'assurer que le modèle équitable a été correctement entraîné (en particulier, à montrer qu'il n'a pas *sur-appris*). L'ensemble d'attaque est connu par l'attaquant *baseline* (comme décrit dans la section 4.1).

Modèles équitables (cibles) Nous utilisons une méthode populaire d'apprentissage équitable, implémentée dans la librairie `Fairlearn` [6]. Cette méthode in-processing, nommée ExponentiatedGradient [2], formule le problème de classification équitable comme une séquence de problèmes de classification pondérée. Etant donné un modèle de base sensible aux coûts, l'approche consiste en un jeu à deux joueurs dans lequel un joueur entraîne le modèle de base tandis que l'autre adapte les poids des exemples d'entraînement. Nous utilisons des arbres de décision de la librarie `scikit-learn` [24] comme modèles de base, avec une profondeur maximale fixée à 8 et tous les autres paramètres laissés à leur valeur par défaut. Notons cependant que notre approche est agnostique au type d'algorithme d'apprentissage équitable mis en oeuvre, puisqu'elle utilise seulement l'information sur l'équité du modèle final.

Métriques d'équité Nous effectuons nos expériences pour les quatre métriques présentées dans la Table 1. Nous utilisons 49 valeurs différentes de tolérance d'inéquité ϵ, variant de manière non linéaire entre 0.0 (équité parfaite) et 0.20.

TABLE 2 – Synthèse des jeux de données utilisés dans nos expériences

Ref.	Nom	Tâche de prédiction (binaire)	#Exemples	#Attributs non sensibles	Attribut sensible
[12]	UCI Adult Income	Revenu > 50K$	45 222	7 catégoriques, 6 numériques	Genre (Femme/Homme)
[10]	ACSPublicCoverage*	Couvert par une assurance maladie publique	98 928	17 catégoriques, 1 numérique	Age (Premier Quartile/Autres)
[10]	ACSIncome*	Revenu > 50K$	135 924	7 catégoriques, 2 numériques	"Code" ethnique (Blancs/Autres)

*(État du Texas, 2018)

Modèles d'attaque Les modèles d'attaque utilisés par l'attaquant *baseline* $\mathcal{A}'$ sont des forêts aléatoires de la librarie `scikit-learn` [24]. Les attributs sensibles étant souvent déséquilibrés [1], nous utilisons une fonction objectif classe-pondérée. Les hyperparamètres des forêts aléatoires sont optimisés par la librarie `HyperOpt-Sklearn` [22], avec un maximum de 100 itérations pour son algorithme de recherche "Tree of Parzen Estimators". Cette configuration permet de s'assurer que l'attaquant *baseline* représente une base solide face à notre étape de correction de reconstruction, et correspond aux pratiques de la littérature.

Correction de reconstruction Notre modèle efficace de correction de reconstruction $\mathcal{RC}_{\mathcal{E}}(\hat{S}, P, \hat{Y}, \epsilon)$ (décrit dans la Section 3.3) est implémenté et résolu par le solveur commercial `IBM ILOG CP Optimizer Version 12.10` via l'API Python de modélisation `DOcplex` (version 2.21.207) dans sa configuration par défaut. Le nombre de threads utilisés par CP Optimizer est fixé à 1 et la tolérance d'optimalité (absolue et relative) est mise à 0.0. En effet, en raison du processus d'exponentiation présenté en Section 4.1, les valeurs des scores de confiance peuvent être très petites et seraient inférieures à la tolérance d'optimalité par défaut du solveur. Notre méthode de correction de reconstruction est implémentée comme une classe Python et est disponible sur notre dépôt [1].

Paramètres expérimentaux Nous fixons un temps d'exécution maximum d'une minute pour l'étape de correction de reconstruction (création et résolution du modèle). En pratique, cette limite n'a jamais été atteinte, et tous les modèles ont été résolus à l'optimum en quelques secondes (moins d'une seconde en moyenne). Chaque expérience est répétée 100 fois, avec différents paramètres pour l'initialisation des générateurs pseudo-aléatoires (pour la séparation du jeu de données et l'initialisation des algorithmes). Les résultats sont moyennés sur les 100 exécutions, et les déviations standard sont reportées. Toutes les expériences sont exécutées sur une plateforme de calcul équipée de processeurs Intel Xeon E5-2683 v4 Broadwell @ 2.1GHz CPU.

4.4 Résultats

Les résultats de nos expériences sont présentés pour les trois jeux de données et les quatre métriques d'équité considérés dans les Figures 2a, 2b et 2c. Ils démontrent l'efficacité de l'approche proposée. Comme mentionné en Section 4.1, l'attaquant $\mathcal{A}'$ utilise déjà toute l'information dont dispose notre composant de correction de reconstruction. Ainsi, toute amélioration de la reconstruction par notre processus de correction peut uniquement être expliquée par la sémantique de la contrainte d'équité intégrée explicitement dans notre modèle $\mathcal{RC}_{\mathcal{E}}(\hat{S}, P, \hat{Y}, \epsilon)$. Pour rappel, la précision de la reconstruction est la proportion d'exemples d'entraînement e_i pour lesquels l'attribut sensible $s_i \in S$ a été correctement reconstruit (par l'attaquant *baseline* dans $\hat{s}_i \in \hat{S}$ ou dans la reconstruction corrigée $s_i^* \in S^*$).

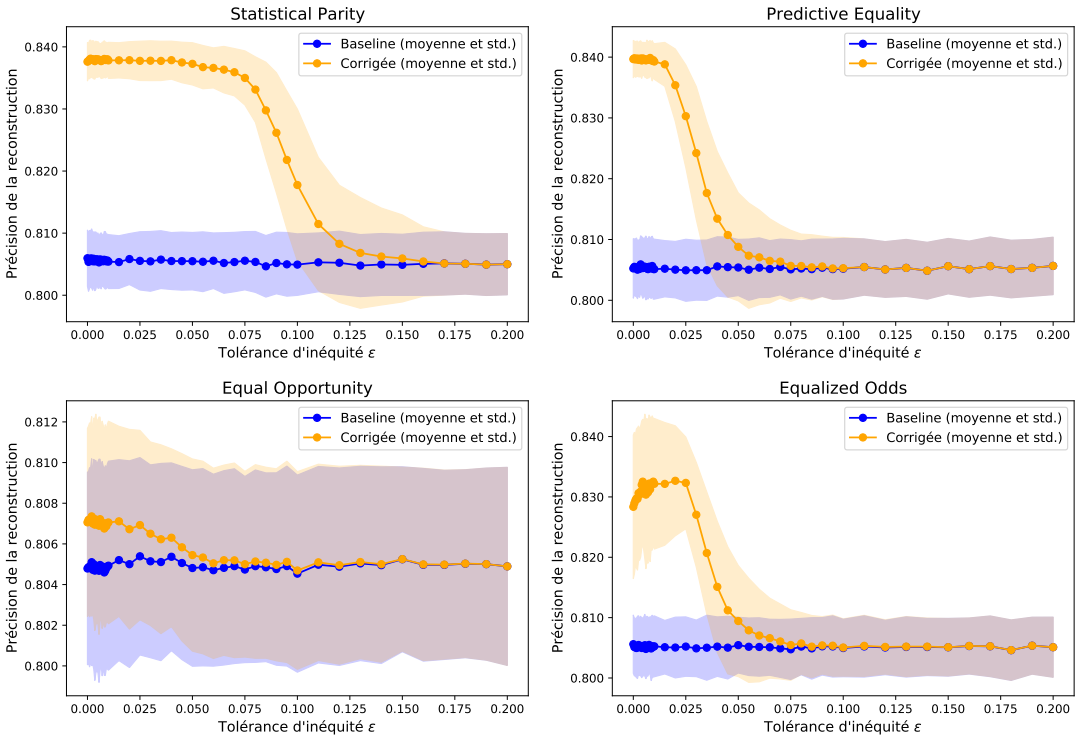
On peut observer que la reconstruction corrigée est toujours plus précise que la reconstruction originale de l'attaquant *baseline*, ce qui indique que les changements effectués sont corrects la plupart du temps. Par ailleurs, plus la contrainte d'équité utilisée est forte (*i.e.*, petites valeurs de tolérance d'inéquité ϵ), plus l'amélioration permise par la correction de reconstruction est importante. En effet, cette dernière est liée à la quantité de biais atténuée par la méthode d'apprentissage équitable, qui à son tour dépend de la métrique d'équité considérée, de la tolérance d'inéquité et du biais des données d'origine. Pour des contraintes d'équité fortes, on observe des améliorations dans la précision de la reconstruction allant jusqu'à 0.06 (ou 9%), comme dans le cas de la métrique Statistical Parity avec le jeu de données ACSIncome. De telles améliorations sont uniquement dues à l'information de l'équité, qui est la seule contrainte dans notre modèle de correction de reconstruction.

Pour rappel, la métrique Predictive Equality (*resp.,* Equal Opportunity) s'applique uniquement aux exemples négatifs (*resp.,* positifs). Ces métriques ne peuvent donc corriger qu'une partie de la reconstruction *baseline* (*cf.* Section 3.4). Les jeux de données utilisés étant déséquilibrés, avec une majorité d'exemples négatifs, la métrique Equal Opportunity ne s'applique donc qu'à une minorité d'exemples. Par conséquent, les améliorations de reconstruction observées avec cette métrique sont plus modestes que pour les autres. En effet, même avec un taux de modifications correctes proche, le nombre de corrections effectuées est plus faible.

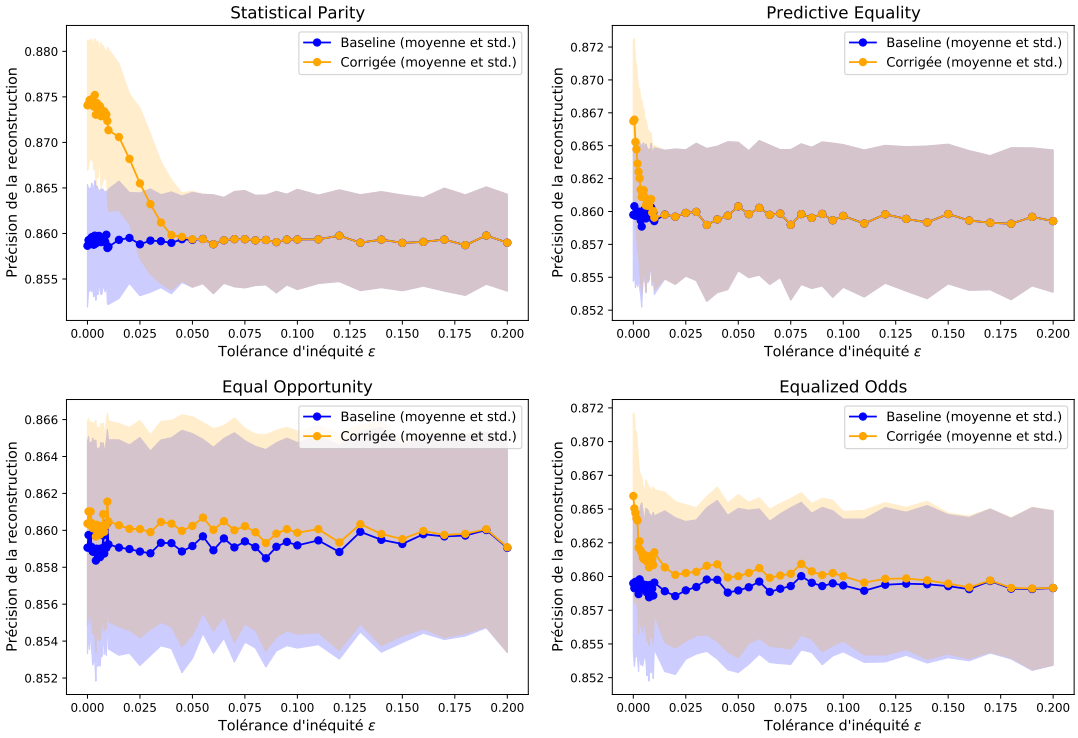
En variant la tolérance d'inéquité ϵ, la seule entrée de la méthode de reconstruction *baseline* qui est modifiée est le vecteur des prédictions du modèle équitable $\hat{Y}$ (et l'information de l'équité). Le fait que la précision de la reconstruction *baseline* de $\mathcal{A}'$ soit quasiment constante lorsque ϵ varie montre que les prédictions du modèle équitable sont peu utilisées par les modèles d'attaque construits. A l'inverse, notre méthode de correction sait exactement comment interpréter l'information de l'équité vis-à-vis de $\hat{Y}$, et est capable de l'utiliser pleinement pour améliorer la qualité de la reconstruction.

Finalement, les résultats expérimentaux montrent que notre méthode de correction de reconstruction est capable d'améliorer significativement la reconstruction des attributs sensibles de l'ensemble d'entraînement d'un modèle, même par rapport à un attaquant *baseline* aussi informé.

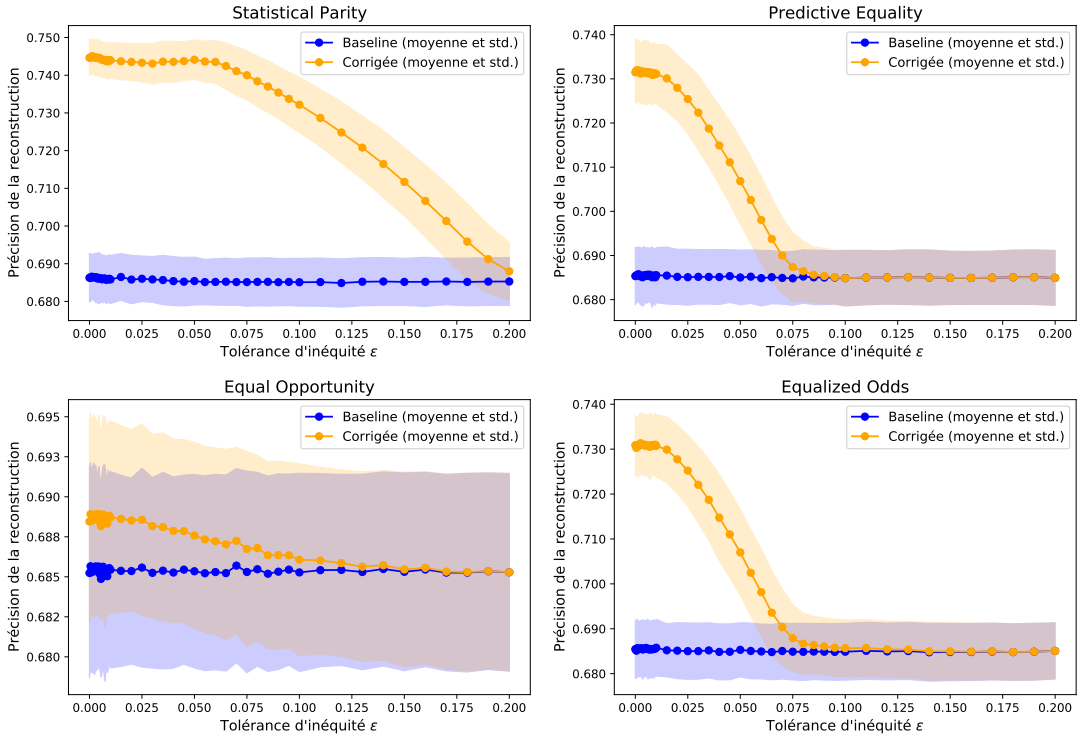
1. `https://github.com/ferryjul/SensitiveAttributesReconstructionCorrector/`



(a) Jeu de données UCI Adult Income.



(b) Jeu de données ACSPublicCoverage.



(c) Jeu de données ACSIncome.

FIGURE 2 – Qualité des reconstructions *baselines* (attaquant $\mathcal{A}'$) et corrigées, pour nos différentes expériences.

5 Discussion

Dans ce travail, nous proposons une approche novatrice utilisant la programmation déclarative pour améliorer les performances de reconstruction de n'importe quel attaquant *baseline*, en incorporant des contraintes définies par l'utilisateur. Bien que le problème général soit difficile d'un point de vue calculatoire, nous montrons que dans le cas des métriques d'équité statistique (ou d'autres contraintes formulées au niveau des groupes), il peut être reformulé et résolu de manière très efficace. Par ailleurs, notre large étude expérimentale montre que, parce qu'ils utilisent les attributs sensibles pour s'assurer de l'équité des modèles construits, les algorithmes d'apprentissage équitable divulguent intrinsèquement des informations sur ceux-ci. En effet, les contraintes d'équité fournissent des informations sur la distribution des prédictions (sur l'ensemble d'entraînement) d'un modèle équitable par rapport aux attributs sensibles (de l'ensemble d'entraînement). Même si cette information se situe au niveau des groupes, elle peut être utilisée par un attaquant pour améliorer sa reconstruction *baseline* des attributs sensibles. Par ailleurs, plus la contrainte d'équité appliquée est forte, plus les améliorations des reconstructions observées en pratique sont importantes.

Les travaux futurs incluent la combinaison de notre approche avec différents attaquants *baselines*, l'optimisation du traitement des scores de confiance P, ainsi que l'utilisation de notre méthode dans le contexte plus général des attributs sensibles non binaires. Enfin, tirer profit de la nature déclarative de l'étape de correction de reconstruction pour intégrer d'autres types de contraintes est également une direction de recherche intéressante.

Notre travail complet [17] contient un certain nombre de contributions que nous n'avons pas pu présenter ici pour des raisons d'espace. Nous présentons notamment un état de l'art plus détaillé, ainsi que des résultats expérimentaux supplémentaires, incluant (i) les performances des modèles équitables (cibles) appris en termes d'équité et de précision, (ii) les résultats de nos expériences utilisant un autre attaquant *baseline*, moins informé que $\mathcal{A}'$ et (iii) les résultats de nos expériences attaquant d'autres modèles équitables, entraînés avec des approches de pre-processing ou de post-processing. Nous discutons des contre-mesures possibles et montrons notamment que même si l'information sur l'équité du modèle n'est pas révélée, un attaquant peut adopter des stratégies relativement simples pour l'estimer. La correction de reconstruction avec la contrainte estimée présente alors de bonnes performances. Cela démontre l'applicabilité de l'approche proposée, et suggère que les métriques d'équité statistique, puisqu'elles utilisent explicitement les attributs sensibles, entrent intrinsèquement en conflit avec la protection de ces derniers. Nous démontrons par ailleurs que les deux modèles proposés (modèle général et modèle efficace) partagent le même ensemble de solutions optimales lorsque les contraintes considérées sont des contraintes d'équité statistique. Enfin, nous discutons de l'extension de ces modèles au cas général des attributs sensibles multi-valués.

Références

[1] Jan Aalmoes, Vasisht Duddu, and Antoine Boutet. Dikaios : Privacy auditing of algorithmic fairness via attribute inference attacks. *arXiv preprint arXiv :2202.02242*, 2022.

[2] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna M. Wallach. A reductions approach to fair classification. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 60–69. PMLR, 2018.

[3] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 2019. `http://www.fairmlbook.org`.

[4] Solon Barocas and Andrew D. Selbst. Big data's disparate impact. *California Law Review*, 104(3) :671–732, 2016.

[5] Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, et al. AI fairness 360 : An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *CoRR*, abs/1810.01943, 2018.

[6] Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, et al. Fairlearn : A toolkit for assessing and improving fairness in AI. Technical Report MSR-TR-2020-32, Microsoft, May 2020.

[7] Simon Caton and Christian Haas. Fairness in machine learning : A survey. *arXiv preprint arXiv :2010.04053*, 2020.

[8] Alexandra Chouldechova. Fair prediction with disparate impact : A study of bias in recidivism prediction instruments. *Big data*, 5(2) :153–163, 2017.

[9] Emiliano De Cristofaro. An overview of privacy in machine learning. *CoRR*, abs/2005.08679, 2020.

[10] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult : New datasets for fair machine learning. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34 : Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 6478–6490, 2021.

[11] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In Frank Neven, Catriel Beeri, and Tova Milo, editors, *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 9-12, 2003, San Diego, CA, USA*, pages 202–210. ACM, 2003.

[12] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[13] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226. ACM, 2012.

[14] Cynthia Dwork, Adam Smith, Thomas Steinke, and Jonathan Ullman. Exposed ! a survey of attacks on private data. *Annual Review of Statistics and Its Application*, 4(1) :61–84, 2017.

[15] The U.S. EEOC. Uniform guidelines on employee selection procedures. March 2, 1979.

[16] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In Longbing Cao, Chengqi Zhang, Thorsten Joachims, Geoffrey I. Webb, et al., editors, *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pages 259–268. ACM, 2015.

[17] Julien Ferry, Ulrich Aïvodji, Sébastien Gambs, Marie-José Huguet, and Mohamed Siala. Exploiting fairness to enhance sensitive attributes reconstruction. In *First IEEE Conference on Secure and Trustworthy Machine Learning*, 2023.

[18] Ferdinando Fioretto, Cuong Tran, Pascal Van Hentenryck, and Keyu Zhu. Differential privacy and fairness in decisions and learning tasks : A survey. *CoRR*, abs/2202.08187, 2022.

[19] Moritz Hardt, Eric Price, Nati Srebro, et al. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323, 2016.

[20] Hui Hu and Chao Lan. Inference attack and defense on the distributed private fair learning framework. In *The AAAI Workshop on Privacy-Preserving Artificial Intelligence*, 2020.

[21] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1) :1–33, 2012.

[22] Brent Komer, James Bergstra, and Chris Eliasmith. Hyperopt-sklearn : automatic hyperparameter configuration for scikit-learn. In *ICML workshop on AutoML*, volume 9, page 50. Citeseer, 2014.

[23] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6) :115 :1–115 :35, 2021.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, et al. Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, 12 :2825–2830, 2011.

SMOTE: Apprenons-nous à classifier ou à prédire la nature synthétique des données ?

N. Boudegzdame[1], K. Sedki[1], R. Tsopra[2,3,4], and JB. Lamy[1]

[1]LIMICS, INSERM, Université Sorbonne Paris Nord, Sorbonne Université, France
[2]INSERM, Université de Paris Cité, Sorbonne Université, Cordeliers Research Center, France
[3]HeKA, INRIA, France
[4]Department of Medical Informatics, Hôpital Européen Georges-Pompidou, AP-HP, France

nadaboudegzdame@gmail.com, {karima.sedki, jean-baptiste.lamy}@univ-paris13.fr

Résumé

Les algorithmes de suréchantillonnage sont des prétraitements utilisés dans l'apprentissage automatique dans le cas de données fortement déséquilibrées dans le but de rééquilibrer le nombre d'instances par classe et donc d'améliorer la qualité des modèles appris. Bien que le suréchantillonnage puisse être efficace pour améliorer les performances des modèles de classification sur les classes minoritaires, il peut également introduire plusieurs problèmes. Durant notre travail, nous avons remarqué que les modèles apprennent à detecter le bruit ajouté par les algorithmes de suréchantilloange au lieu d'apprendre les informations pertinentes. Dans cet article, nous définirons le suréchantillonnage et présenterons les techniques les plus courantes, avant de proposer une méthode pour évaluer les algorithmes de suréchantillonnage.

Mots-clés

Données déséquilibrées, suréchantillonnage, SMOTE, augmentation de données, apprentissage automatique, apprentissage profond, réseaux de neurones, données synthétiques.

Abstract

Oversampling algorithms are used as preprocess in machine learning, in the case of highly imbalanced data in an attempt to balance the number of samples per class, and therefore improve the quality of models learned. While oversampling can be effective in improving the performance of classification models on minority classes, it can also introduce several problems. From our work, it came to light that the models learn to detect the noise added by the oversampling algorithms instead of the underlying patterns. In this article, we will define oversampling, and present the most common techniques, before proposing a method for evaluating oversampling algorithms.

Keywords

Imbalanced data, oversampling, SMOTE, data augmentation, machine learning, deep learning, neural networks, synthetic data.

1 Introduction

Le suréchantillonnage est une technique utilisée pour résoudre le problème du déséquilibre des classes dans l'apprentissage automatique. Le déséquilibre des classes se produit lorsque le nombre d'instances d'une classe est beaucoup plus faible que le nombre d'instances de(s) autre(s) classe(s). Ceci génère d'un problème, car le classificateur aura du mal à apprendre à partir de la classe minoritaire. Les techniques de suréchantillonnage génèrent des instances supplémentaires appartenant à la classe minoritaire afin que le classificateur ait une meilleure chance d'apprendre à partir d'eux [12, 1].

Le suréchantillonnage crée de nouvelles instances des classes minoritaires en 1) reproduisant des instances existantes ou 2) en synthétisant des instances. Parmi les techniques les plus répandues, citons le suréchantillonnage aléatoire [5], *SMOTE* [5], *ADASYN* [11], *Borderline SMOTE* [10], *SMOTEN* [5], *Safe-Level SMOTE* [2] , et Minority Oversampling Technique (*MOTE*) [13]. Ces techniques consistent à dupliquer ou à générer des instances synthétiques de la classe minoritaire afin d'augmenter sa représentation dans le jeu de données. Beaucoup de ces techniques sont des variantes ou des extensions de *SMOTE*, une technique de suréchantillonnage largement utilisée qui interpole entre les instances minoritaires existants pour générer de nouvelles instances synthétiques.

Bien que le suréchantillonnage puisse s'avérer efficace pour améliorer les performances des modèles de classification sur les classes minoritaires, il peut également introduire plusieurs problèmes. Dans cet article, nous définirons les problèmes et les défis potentiels liés à l'utilisation du suréchantillonnage. L'un des problèmes les plus importants avec les techniques de suréchantillonnage, en particulier lorsque les données sont très déséquilibrées, est que les données originales représentent une petite fraction du jeu

de données résultant pour la classe minoritaire, la grande majorité des données étant représentée par les données synthétiques. En conséquence, cela peut entraîner une redéfinition du problème d'apprentissage. En fait, après le suréchantillonnage, la classe minoritaire contiendra principalement des données synthétiques, ce qui peut amener le modèle d'apprentissage automatique à apprendre à détecter la nature synthétique des données, c'est-à-dire le bruit ajouté par le suréchantillonnage, plutôt qu'à prédire la classe minoritaire à partir des motifs correspondants. Cela peut entraîner une mauvaise généralisation du modèle, conduisant ainsi à des faibles performances sur les données du monde réel [22, 8, 6, 20].

Par conséquent, nous proposons une méthode d'évaluation des techniques de suréchantillonnage sur un jeu de données précis et nous l'appliquerons à la prédiction des hémorragie causée par les médicaments. La méthode consiste à essayer d'apprendre un nouveau modèle capable de prédire le statut synthétique de l'instance ; les performances de la technique de suréchantillonnage sont inversement proportionnelles aux performances de ce modèle. Enfin, nous mettrons à l'épreuve les techniques de suréchantillonnage les plus courantes et évaluerons leur efficacité dans un exemple de cas d'utilisation.

2 Travaux antérieurs

Plusieurs techniques de suréchantillonnage ont été mises au point pour remédier au déséquilibre des classes dans les jeux de données. Voici quelques-unes des plus populaires :

- ***Suréchantillonnage aléatoire [5] :*** Cette technique simple consiste à dupliquer aléatoirement des instances de la classe minoritaire pour augmenter sa représentation dans le jeu de données.
- ***Synthetic Minority Oversampling Technique (SMOTE) [5] :*** crée de nouvelles instances synthétiques en interpolant entre des instances minoritaires existants. Elle sélectionne deux instances similaires et crée une nouvelle instance le long de la ligne qui les relie. De nombreuses modifications et extensions ont été apportées à la méthode *SMOTE* depuis sa proposition.
- ***Adaptive Synthetic (ADASYN) [11] :*** génère des instances synthétiques basés sur la densité de la distribution des données. Elle crée plus des instances synthétiques pour la classe minoritaire qui sont plus difficiles à apprendre et s'appuie sur la méthodologie de *SMOTE*.
- ***Borderline SMOTE [10] :*** cette technique est une variante de *SMOTE* et se concentre sur des instances qui se situent à la frontière entre la classe minoritaire et la classe majoritaire. Elle génère des instances synthétiques uniquement pour les instances situées à la frontière.
- ***SMOTEN [5] :*** est une extension de la technique de suréchantillonnage *SMOTE* qui peut traiter des jeux de données avec des attributs nominaux et continus. Elle utilise une métrique de distance adaptée aux types d'attributs mixtes afin de générer des instances synthétiques pour les attributs nominaux.
- ***Safe-Level SMOTE[2] :*** cette technique combine à la fois le *SMOTE* et le suréchantillonnage aléatoire. Elle génère des instances synthétiques pour la classe minoritaire, mais ne génère pas d'instances pour ceux qui entraîneraient un chevauchement avec la classe majoritaire.
- ***Minority Oversampling Technique (MOTE) [13] :*** est une variante de *SMOTE* qui sélectionne uniquement les instances mal classées par le modèle actuel et génère des instances synthétiques uniquement pour ces instances.

Depuis l'introduction du premier *SMOTE*, il y a plus de 20 ans, en 2002, de nombreuses nouvelles techniques ont été mises en œuvre pour l'améliorer. La première amélioration, *Borderline SMOTE*, a résolu le problème de surapprentissage qui peut survenir avec *SMOTE* lorsque les instances synthétiques générées sont uniquement pour les instances de classe minoritaire situées près de la frontière de décision. *ADASYN* a été la suivante, elle a ajusté la distribution de densité de l'espace des attributs pour générer plus d'instances synthétiques pour les instances de la classe minoritaire difficiles à apprendre. *Safe-Level SMOTE* a été développée pour réduire le risque de mauvaise classification en ne générant des instances synthétiques que pour les instances de la classe minoritaire situées à proximité d'une classe majoritaire sûre. La plus récente amélioration est *SMOTEN*, qui peut gérer des jeux de données avec des attributs nominaux et continus, en utilisant une approche différente pour générer des instances synthétiques pour les attributs nominaux.

Ces techniques de suréchantillonnage ont des forces et des faiblesses différentes et peuvent donner des résultats différents en fonction du jeu de données et du problème de classification. Pour résoudre les problèmes restants, plusieurs approches ont été proposées, telles que la combinaison du suréchantillonnage et du sous-échantillonnage, l'utilisation de techniques d'échantillonnage synthétique plus avancées ou l'ajustement du seuil de classification. Cependant, chaque approche a ses propres avantages et limites, et une attention particulière est nécessaire pour sélectionner la méthode de suréchantillonnage appropriée pour un jeu de données et un problème de classification particuliers.

3 Problème du Oversampling

La technique de suréchantillonnage la plus courante [12] et la plus efficace [1] est connue sous le nom de *SMOTE*

: Synthetic Minority Over-sampling Technique. Elle fonctionne en générant des instances synthétiques de la classe minoritaire par interpolation entre les instances existantes de la classe minoritaire et leurs k plus proches voisins dans l'espace des attributs, augmentant ainsi la représentation de la classe minoritaire sans dupliquer les instances existantes. Cette technique est souvent utilisée pour remédier au déséquilibre des classes, un problème courant dans de nombreux jeux de données du monde réel où une classe est considérablement sous-représentée. Bien que le suréchantillonnage puisse être efficace pour améliorer les performances des modèles de classification sur les classes minoritaires, il peut également introduire plusieurs problèmes.

On peut classer les problèmes de *SMOTE* dans les six catégories suivantes :
1. Tout d'abord, l'un des problèmes les plus courants associés au suréchantillonnage est qu'il peut introduire un biais en faveur de la classe minoritaire [22, 8]. Lorsque le suréchantillonnage est appliqué, la classe minoritaire est artificiellement gonflée en créant de nouvelles instances synthétiques, ce qui peut amener le modèle à prédire trop fréquemment cette classe au détriment de la classe majoritaire. En conséquence, le modèle peut avoir une grande précision sur les données d'apprentissage, mais mal fonctionner sur les données du monde réel, car la classe minoritaire est beaucoup moins fréquente.

2. Le suréchantillonnage peut par ailleurs entraîner des incohérences dans les types de données, puisque les points de données synthétiques peuvent générer des valeurs qui se situent en dehors de la plage typique de la variable ou dans un format différent. Par exemple, si les données originales ne contiennent que des nombres entiers pour l'âge, le suréchantillonnage peut générer des nombres décimaux qui ne sont pas présents dans les données du monde réel.

3. Les instances synthétiques créées par suréchantillonnage sont supposés appartenir à la classe minoritaire, mais cela peut ne pas être vrai. Il peut également produire des instances mal étiquetées appartenant à la classe majoritaire, ainsi que des instances "bruits" absurdes et ne correspondant à aucune classe ou réalité, telle qu'un patient de 3 ans et pesant 100 kg.

4. La distribution des données peut également être modifiée par des instances synthétiques. Par exemple, si la classe minoritaire comprend 50% d'enfants, mais que les données synthétisées n'en comprennent que 20% alors la distribution n'est pas la même.

5. Le suréchantillonnage peut réduire la diversité du jeu de données en créant des instances synthétiques très similaires aux instances existantes. Cela peut entraîner un surapprentisssage et avoir un impact négatif sur la capacité du modèle à généraliser à de nouvelles données. Le jeu de données suréchantillonné peut ne pas refléter avec précision la véritable diversité du problème.

Il est important d'examiner attentivement l'impact du suréchantillonnage sur la distribution et la diversité du jeu de données pour s'assurer que le modèle résultant reflète avec précision la véritable nature du problème.

6. Le suréchantillonnage peut augmenter le coût d'apprentissage d'un modèle, car il nécessite de générer des points de données supplémentaires pour la classe minoritaire [6, 20]. Travailler sur de grands jeux de données, implique que la génération de données synthétiques peut prendre beaucoup de temps et de ressources.

En outre, plus un jeu de données est déséquilibré, moins le jeu de données suréchantillonné reflète avec précision la véritable nature du problème [12]. Comme expliqué ci-dessus, l'algorithme de suréchantillonnage ajustera la distribution de classe du jeu de données. Ainsi, plus un jeu de données est déséquilibré, plus de données sont nécessaires pour ajuster la distribution des classes, ce qui entraînera plus de données synthétiques dans le jeu de données suréchantillonné.

Cela peut être particulièrement difficile lors de la manipulation des jeux de données de détection d'anomalies, car elles ont tendance à avoir des distributions de classe très déséquilibrées, car la survenue d'événements ou de conditions rares est peu fréquente par rapport à la population globale. Les jeux de données médicales et de détection de fraude sont des exemples courants d'ensembles de données très déséquilibrés où la détection d'anomalies est essentielle, mais ces anomalies sont rares dans l'occurrence [4].

Les jeux de données médicales sont extrêmement difficiles à suréchantillonner, ce sont souvent les jeux de données de distribution de classe les plus déséquilibrés en raison de la nature des données médicales. Dans les données médicales, la survenue de certaines maladies ou conditions médicales peut être rare par rapport à l'ensemble de la population. Par exemple, une maladie peut n'affecter qu'un petit pourcentage de la population, tandis que la majorité de la population peut être en bonne santé. En conséquence, le jeu de données aura une distribution de classe très déséquilibrée, la classe minoritaire étant la condition médicale d'intérêt [17].

De plus, le coût et la difficulté de la collecte de données dans le domaine médical peuvent également contribuer au déséquilibre de la répartition des classes. La collecte de données médicales nécessite souvent des procédures coûteuses et chronophages, telles que des tests médicaux ou des examens d'imagerie, qui peuvent être difficiles à réaliser sur une population nombreuse et diversifiée. Par conséquent, les données collectées peuvent être biaisées en faveur de certains groupes ou données démographiques, entraînant des distributions de classes déséquilibrées.

Dans la section suivante, nous illustrerons certains problèmes rencontrés avec le suréchantillonnage à partir d'un exemple médical.

4 Méthodologie

4.1 Description de la tâche initiale d'apprentissage automatique

Notre objectif initial était de prédire le risque d'hémorragie à partir des prescriptions médicales des patients de MIMIC, une base de données publique de dossiers médicaux électroniques dépersonnalisés pour les patients admis dans des unités de soins intensifs (USI) aux États-Unis. Elle contient des données cliniques complètes sur plus de 40 000 patients en soins intensifs, y compris des données démographiques, des diagnostics, des résultats de tests de laboratoire, des informations sur les médicaments et des signes vitaux [15].

L'objectif était d'identifier les patients à risque d'hémorragie en raison de certains médicaments, doses et antécédents médicaux. Les patients labellisés comme étant à risque d'hémorragie sont ceux qui ont subi une hémorragie. Il est crucial de les identifier car certains médicaments, doses et antécédents médicaux d'un individu peuvent augmenter le risque d'hémorragie, ce qui peut mettre en danger la vie du patient. Les médicaments couramment connus pour augmenter le risque d'hémorragie comprennent les anticoagulants tels que la warfarine, le dabigatran et l'apixaban, ainsi que les agents antiplaquettaires comme l'aspirine et le clopidogrel. D'autres médicaments, tels que les anti-inflammatoires non stéroïdiens (AINS) et les inhibiteurs sélectifs de la recapture de la sérotonine (ISRS), peuvent également augmenter le risque d'hémorragie, en particulier lorsqu'ils sont pris à fortes doses ou en association avec d'autres médicaments [14].

Nous définissons le problème de classification de l'apprentissage automatique comme suit :

Prédiction du risque d'hémorragie
Entrée: *Historique des prescriptions médicales des patients, historique des admissions des patients à l'hôpital.*
Sortie: *Le patient est-il à risque d'hémorragie ou non ?*

Pour étiqueter les données, nous avons d'abord dû définir comment extraire les informations sur les hémorragies induites par les médicaments. Pour ce faire, nous avons examiné le dossier d'admission à l'hôpital du patient, qui comprend la raison de l'admission codée à l'aide du système de classification internationale des maladies (CIM). Ce système est un système de classification médicale standardisé utilisé pour coder et classer les procédures médicales, les symptômes et les diagnostics [25]. En analysant le système de classification internationale des maladies, nous avons pu définir une liste de codes CIM qui représentent les hémorragies induites par les médicaments.

En ce qui concerne les données d'entrée, elles comprennent les informations relatives à l'admission à l'hôpital d'un patient et les détails de sa prescription actuelle. Les médicaments sont codés à l'aide du National Drug Code (NDC), un code unique à 10 chiffres utilisé pour identifier les médicaments aux États-Unis. Cependant, NDC est spécifique aux États-Unis et est trop spécifique, car des codes distincts existent pour les différents dosages, formes et présentations d'un médicament [23]. Nous avons donc utilisé le système de classification *ATC* (Anatomical Therapeutic Chemical), qui organise les médicaments en fonction de leurs propriétés thérapeutiques et de leur site d'action anatomique [24]. Pour remédier à cette différence, nous avons mis en correspondance les codes NDC à leurs codes *ATC* correspondants. Certains médicaments ont plusieurs codes *ATC* ; dans ce cas, nous les avons tous considérés.

Enfin, nous avons codé les médicaments du patient en utilisant one hot encoding. Il s'agit d'un processus utilisé dans l'apprentissage automatique pour convertir des données catégorielles en une représentation numérique pouvant être utilisée par des algorithmes d'apprentissage automatique. Cela implique de créer un vecteur binaire qui a une valeur pour chaque médicament possible, la valeur étant 1 si le médicament est présent et 0 sinon. Par exemple, s'il y a trois médicaments - M_1, M_2 et M_3 - chaque médicament ou ordonnance médicale serait représenté par un vecteur binaire de longueur trois. Le médicament M_1 serait représenté par le vecteur [1,0,0], le médicament M_2 serait représenté par [0,1,0] et le médicament M_3 serait représenté par [0,0,1]. Une ordonnance associant un médicament M_1 et M_2 serait représentée par [1,1,0]. Cela permet aux algorithmes de travailler avec des données catégorielles, ce qui peut être utile dans de nombreuses applications telles que la classification de texte.

Le jeu de données résultant était très déséquilibré avec une classe minoritaire représentant seulement 3,47 % des patients présentant un risque hémorragique. Cette nature déséquilibrée du jeu de données peut poser un défi important au modèle pour prédire avec précision la classe minoritaire. En effet, le modèle peut devenir biaisé en faveur de la classe majoritaire, ce qui a entraîné de mauvaises performances lors de la prédiction de la classe minoritaire. Pour résoudre ce problème, nous avons utilisé le suréchantillonnage comme technique courante pour équilibrer le jeu de données.

4.2 Problème rencontré avec l'oversampling

Après suréchantillonnage, nous avons remarqué que les performances du modèle s'amélioraient considérablement à la fois sur l'apprentissage et la validation qui étaient suréchantillonnées, mais a donné de mauvais résultats sur les données originales en termes de métriques de performance. De plus, prédire le risque d'hémorragie est une

tâche difficile, car il se produit rarement et il est difficile de prédire si une prescription entraînera une hémorragie. Cependant, nous avons obtenu un score f1 de 90% pour prédire l'hémorragie sur les données d'apprentissage ce qui semblait trop optimiste.

Pour étudier ce problème, nous avons procédé à une analyse des prédictions du modèle afin de déterminer s'il répondait toujours le problème initial. Nous formulons l'hypothèse que le modèle apprenait à prédire si une instance était synthétique, au lieu de prédire s'il appartient à la classe minoritaire, ce qui, en effet, revient presque au même, puisqu'une grande majorité des instances appartenant à la classe minoritaire sont synthétiques.

4.3 Une méthode d'exploration de la détectabilité des données synthétiques

Pour tester notre hypothèse, nous avons défini un nouveau problème d'apprentissage automatique pour détecter des données synthétiques. Nous avons généré un nombre d'instances synthétiques égal au nombre d'instances de la classe minoritaire en utilisant le suréchantillonnage, nous avons ensuite retiré des instances de la classe majoritaire et labellisé les instances comme étant synthétiques (1) ou originaux (0). Cette approche a été appliquée à différentes méthodes de suréchantillonnage afin de déterminer la facilité avec laquelle les données synthétiques générées par ces méthodes pourraient être détectées. Les données synthétiques de moindre qualité étant plus facilement détectées, cette méthode permet d'évaluer la qualité des différentes techniques de suréchantillonnage. Notre jeu de données ainsi affiné a été utilisé pour résoudre le problème suivant :

Détection de données synthétiques
Entrée : *Classe minoritaire VS données synthétiques produites par suréchantillonnage.*
Sortie : *Instance synthétique ou originale ?*

Nous avons évalué notre méthode en utilisant les métriques suivantes :

1. **Précision, rappel et score F1 :** La précision mesure la proportion d'instances positives correctement prédites parmi toutes les instances positives prédites, tandis que le rappel mesure la proportion d'instances positives correctement prédites parmi toutes les instances positives réelles. Le score F1 est la moyenne harmonique de la précision et du rappel. Ces mesures sont particulièrement utiles lorsqu'il s'agit de données très déséquilibrées, car elles fournissent une mesure de la capacité du modèle à identifier la classe minoritaire [12, 18].

2. **Area under the precision-recall curve (AUPRC) :** AUPRC fournit un score unique qui capture le compromis entre précision et rappel pour différents seuils de décision. Il s'agit d'une mesure utile pour les données très déséquilibrées, car elle se concentre sur la classe positive et peut fournir une évaluation plus informative que la précision ou ROC AUC [7].

3. **Receiver operating characteristic (ROC) et area under the curve (AUC):** ROC trace le taux de vrais positifs par rapport au taux de faux positifs pour différents seuils de décision. L'AUC mesure l'aire sous la courbe ROC et fournit un score unique qui indique la performance globale du modèle. ROC et AUC sont utiles pour comparer des modèles qui ont des seuils de décision différents [12, 18, 9].

4. **Matrice de confusion :** Une matrice de confusion fournit une répartition détaillée des prédictions du modèle, y compris les vrais positifs, les vrais négatifs, les faux positifs et les faux négatifs. Cela peut aider à identifier le nombre d'exemples correctement et incorrectement classés par le modèle pour chaque classe.

5. **Kappa de Cohen :** Le kappa de Cohen mesure l'accord inter-juges entre le jeu de données original et le jeu de données suréchantillonné. Cela peut être utile pour évaluer dans quelle mesure les données synthétiques capturent la vraie nature du problème [19].

En utilisant plusieurs métriques et techniques pour évaluer les performances du modèle avec précision, nous pouvons acquérir une compréhension plus complète des forces et des limites du modèle. Cela peut nous aider à prendre des décisions sur la manière d'améliorer le modèle ou de l'utiliser dans des applications pratiques, car aucune mesure unique ne peut fournir une image complète de l'efficacité du modèle.

Nous avons choisi le réseau de neurones comme approche d'apprentissage en nous basant sur des études précédentes qui ont montré l'efficacité de l'apprentissage profond pour les tâches de classification déséquilibrées [26]. Pour l'implémentation actuelle, nous avons utilisé un réseau de neurones avec deux couches cachées contenant respectivement 30 et 20 neurones. Pour éviter le problème de **"neurones morts"**, nous avons opté pour la fonction d'activation ***LeakyReLU***, qui s'est avérée performante dans des applications similaires [16]. La couche de sortie a été conçue avec une fonction d'activation ***sigmoid***, couramment utilisée dans les problèmes de classification binaire.

Pour garantir l'efficacité du modèle d'apprentissage, nous avons utilisé une technique de réduction du taux d'apprentissage appelée ***ReduceLROnPlateau***. Cette technique nous a permis d'ajuster dynamiquement le taux d'apprentissage de l'optimiseur pendant la phase apprentissage, en fonction d'une métrique surveillée telle que validation loss. Ce faisant, nous avons pu aider le modèle à sortir des plateaux et à continuer de s'améliorer, même à l'approche de la convergence. Notre modèle a été

entrainé sur 100 époques, ce qui était suffisant pour assurer un apprentissage complet et la convergence du modèle.

4.4 Résultats et analyse

Les résultats du tableau 1 indiquent que, pour les quatre techniques de suréchantillonnage, le réseau de neurones a obtenu de bons résultats en termes de métriques d'évaluation et a donc été en mesure de prédire des données synthétiques avec un degré élevé de précision. Parmi toutes les techniques de suréchantillonnage, *SMOTEN* a obtenu le score le plus élevé en termes de score f1, de rappel, de précision, d'accuracy, de cohen kappa et d'AUC. L'algorithme Borderline *SMOTE* conduit également à des scores élevés dans toutes les mesures d'évaluation, à l'exception de l'AUC.

Par conséquent, nous pouvons facilement prédire si une instance est synthétique ou non. Cette prédiction est beaucoup plus facile que celle du risque d'hémorragie. Cela confirme donc notre hypothèse : le modèle initial prédisait en fait la nature synthétique des données au lieu du risque d'hémorragie.

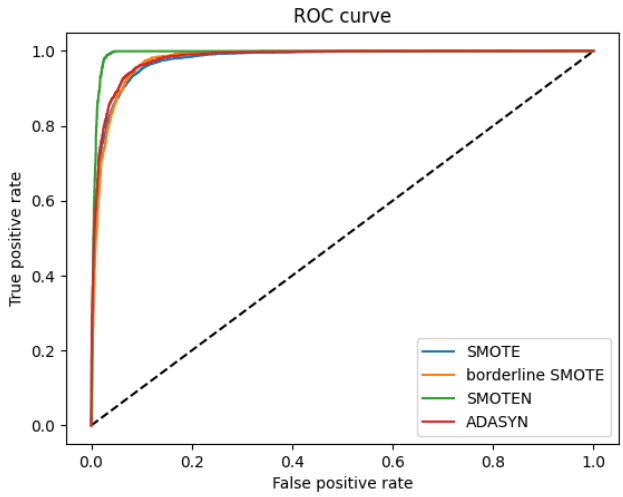


Figure 1: Courbe ROC pour la détection de la nature synthétique des données.

Comme expliqué ci-dessus, la courbe ROC et la courbe Precision-Recall fournissent des informations importantes sur les performances d'un modèle de classification binaire. Par conséquent, nous avons tracé les deux courbes pour obtenir une évaluation plus complète des performances du modèle. La figure 1 résume la courbe ROC pour les quatre algorithmes de suréchantillonnage. Elle indique que le modèle est très précis dans la distinction entre les instances positives et négatives. En fait, une AUC de 0,5 suggère une classification aléatoire, tandis qu'une AUC de 1 suggère une classification parfaite. Les valeurs AUC pour *SMOTE*, borderlineSMOTE et *ADASYN* sont

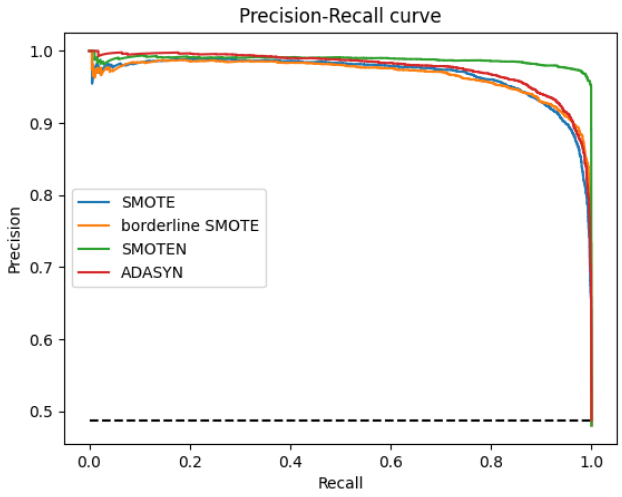


Figure 2: Courbe de précision et de rappel pour la détection de la nature synthétique des données.

de 0,97, ce qui indique que les performances du modèle sont très proches de la perfection, avec seulement un petit nombre de faux positifs et de faux négatifs. De plus, nous avons observé que les données suréchantillonnées générées par *SMOTEN* sur nos données étaient les plus faciles à détecter, comme le confirme la figure 2, qui résume la courbe Rappel-Précision.

Par conséquent, le tableau 1 et les figures 1 et 2 suggèrent que les techniques de suréchantillonnage peuvent être facilement détectées dans une large mesure. Cependant, le choix de l'algorithme de suréchantillonnage doit dépendre des caractéristiques spécifiques du jeu de données et des mesures d'évaluation qui nous intéressent.

Bien qu'il soit connu que l'algorithme de suréchantillonnage ne se comporte pas de la même manière sur différents jeux de données, les résultats des tests sur les données médicales, comprenant des prescriptions de médicaments, et constituant un jeu de données fortement déséquilibré, indique fortement que le suréchantillonnage ne sera pas une technique considérable pour équilibrer nos données. Une analyse et une expérimentation plus approfondies peuvent être nécessaires pour déterminer l'approche la plus efficace pour équilibrer le jeu de données des prescriptions médicales en question.

4.5 Comprendre pourquoi les données synthétiques sont facilement détectées

Les prescriptions de médicaments suréchantillonnées peuvent ne pas refléter fidèlement les données du monde réel, car elles sont facilement détectables par les algorithmes d'apprentissage automatique. Pour mieux comprendre cette problématique, nous avons formulé les hypothèses suivantes :

	F1 Score	Recall	Precision	Accuracy	Cohen Kappa	AUC
SMOTE	0.92	0.94	0.90	0.92	0.84	0.97
Borderline *SMOTE*	0.93	0.96	0.91	0.93	0.86	0.97
SMOTEN	0.97	0.99	0.96	0.97	0.95	0.99
ADASYN	0.92	0.93	0.91	0.92	0.84	0.97

Table 1: Évaluation comparative des métriques de performance de divers algorithmes d'oversampling pour la classification de données synthétiques.

Hypothèse 1 : Sur ou sous-représentation des drogues. (Problème #4 dans la section 3)
Les prescriptions médicales contiennent généralement un nombre limité de médicaments. Cependant, les données synthétiques générées peuvent contenir un nombre plus petit ou plus grand de médicaments, ce qui entraîne une sous-représentation ou une surreprésentation des médicaments, ce qui pourrait entraîner des écarts entre les données synthétiques et réelles.

Le tableau suivant 2 montre que les quatre méthodes de suréchantillonnage (*SMOTE*, *Borderline SMOTE* , *SMOTEN* et *ADASYN*) ont entraîné une diminution du nombre moyen de codes *ATC* pour les médicaments dans les données suréchantillonnées par rapport aux données originales. Cela indique une sous-représentation des médicaments dans les instances suréchantillonnés.

	Nombre Moyen de Codes *ATC*
Données Originales	34.78
SMOTE	20.11
Borderline SMOTE	21.13
SMOTEN	20.76
ADASYN	19.49

Table 2: Distribution des médicaments dans les données originales et synthétiques.

Hypothèse 2 : Changement de la nature des données. (Problème #2)
SMOTE peut introduire de petites perturbations dans les valeurs des attributs afin de créer des instances synthétiques, ce qui peut entraîner des valeurs non entières ou à virgule flottante pour les attributs discrètes [21]. Par exemple, les médicaments sont représentés par des valeurs discrètes de 0 ou 1, indiquant la présence ou l'absence du médicament dans une ordonnance. Cependant, les données synthétiques générées à des fins d'analyse peuvent contenir des médicaments avec des valeurs continues, ce qui peut entraîner des inexactitudes dans les résultats.

Après une enquête plus approfondie, nous avons constaté que l'application de *SMOTE*, Borderline *SMOTE*, *SMOTEN* et *ADASYN* n'a entraîné aucun changement significatif dans la nature des données suréchantillonnées. Les quatre méthodes de suréchantillonnage appliquées à nos données n'ont pas modifié la nature des données.

Hypothèse 3 : Incohérences dans les codes* ATC *. (Problème #3)
Certains médicaments comme l'aspirine ont plusieurs codes *ATC* , et nous les avons associés à tous leurs codes correspondants dans les données originales. Toutefois, dans les instances synthétiques, un tel médicament peut être associé à un seul de ses codes. Par exemple, une prescription d'aspirine pourrait être codée comme un inhibiteur de l'agrégation plaquettaire mais pas comme un analgésique dans les instances synthétiques.

Hypothèse 4 : Incohérences dans les associations médicamenteuses. (Problème #3)
Les ordonnances synthétiques générées peuvent inclure des associations médicamenteuses incohérentes. Par exemple, des médicaments comme le ramipril et l'énalapril, qui sont tous deux des inhibiteurs de l'enzyme de conversion de l'angiotensine et qui ont les mêmes effets, ne sont donc jamais associés. Cependant, de telles incohérences peuvent se produire dans les instances synthétiques.

Nous sommes encore en train d'expérimenter et d'essayer de valider ces hypothèses.

5 Discussion

Dans cet article, nous avons décrit un problème que nous avons rencontré lors de l'utilisation du suréchantillonnage : le modèle d'apprentissage automatique apprenait à détecter la nature synthétique des données suréchantillonnées plutôt que les informations pertinentes initialement dans les données originales. Nous avons proposé une méthode pour identifier ce problème et évaluer les méthodes de suréchantillonnage, consistant à essayer d'apprendre dans quelle mesure des données synthétiques peuvent être détectées.

Dans la littérature, de nombreuses études ont exploré les problèmes associés au suréchantillonnage et au *SMOTE*, cependant, à notre connaissance, aucune d'entre elles n'a mentionné l'apprentissage de la nature synthétique des données ni proposé une méthode pour la quantifier.

Les travaux de Tarawneh et al. [22] sont un article de synthèse complet sur le problème de la résolution du déséquilibre des classes dans l'apprentissage automatique et met en évidence la technique de suréchantillonnage

couramment utilisée pour y remédier. Les auteurs affirment que le suréchantillonnage peut entraîner un surapprentissage, une augmentation des coûts de calcul et une réduction des performances de généralisation.

L'article souligne également que le suréchantillonnage peut augmenter le risque de biais du modèle et peut entraîner une diminution des performances de généralisation, en particulier lorsque les données suréchantillonnées sont utilisées pour les tests. En outre, le suréchantillonnage peut augmenter le coût de calcul des modèles de formation, particulièrement dans le cas des grands jeux de données, car il nécessite de générer et de stocker un grand nombre d'instances synthétiques. Les auteurs discutent des limites de l'approche de suréchantillonnage et suggèrent des méthodes alternatives, telles que l'apprentissage sensible au coût et la détection d'anomalies, qui peuvent fournir des solutions plus efficaces au problème de déséquilibre des classes.

Les travaux de R. Buda et al. [3] étudient l'impact du déséquilibre des classes sur les performances des CNN pour les tâches de classification d'images et évalue l'efficacité de différentes stratégies, y compris le suréchantillonnage. Cependant, l'article suggère que le suréchantillonnage seul peut ne pas être suffisant pour résoudre le déséquilibre de classe dans les CNN. En effet, le suréchantillonnage peut entraîner un surapprentissage dans les CNN, où le modèle mémorise les données d'apprentissage et fonctionne mal sur les nouvelles données. De plus, le suréchantillonnage peut créer des instances irréalistes et redondants, entraînant une utilisation inefficace des ressources de calcul.

Plusieurs études ont proposé des modifications à la technique de suréchantillonnage pour résoudre ces problèmes. Rodríguez-Torres et al. [20] ont proposé le suréchantillonnage aléatoire à grande échelle (LRO) pour résoudre les problèmes de déséquilibre des classes sur les grands jeux de données.

Dans cette étude, les performances de LRO ont été comparées à celles de plusieurs autres méthodes de suréchantillonnage, telles que *SMOTE* et *Borderline-SMOTE*. Les résultats ont montré que LRO atteignait une précision et un score F1 plus élevés que les autres méthodes, et était également plus efficace en termes de calcul. Les auteurs ont identifié certaines limites et difficultés de la méthode *SMOTE*, telles que son incapacité à générer des instances diversifiées et sa sensibilité au bruit. Dans l'ensemble, les auteurs suggèrent que LRO pourrait fournir une solution plus efficace et plus évolutive pour les problèmes de déséquilibre des classes sur de grands jeux de données.

En résumé, la littérature met en évidence les limites et les défis potentiels du suréchantillonnage et du *SMOTE* pour traiter les données déséquilibrées dans l'apprentissage automatique, et suggère des approches alternatives et des modifications pour résoudre ces problèmes. Les articles présentés couvrent divers aspects du suréchantillonnage et des problèmes *SMOTE*, notamment le surapprentissage, l'évaluation des performances, la gestion de grands jeux de données, le déséquilibre multi-classes, la gestion du bruit et le suréchantillonnage synthétique.

6 Conclusion et perspectives

Le suréchantillonnage, en conclusion, peut être un outil précieux pour améliorer les performances des modèles d'apprentissage automatique sur des jeux de données déséquilibrés. Cependant, nos résultats suggèrent que les algorithmes de suréchantillonnage peuvent introduire un certain nombre de problèmes. La mauvaise qualité des données synthétiques dans la classe minoritaire peut amener le modèle d'apprentissage automatique à apprendre à prédire les données synthétiques plutôt que les informations pertinentes, ce qui entraîne de mauvaises performances sur les données du monde réel.

Par conséquent, il est essentiel d'analyser en profondeur les méthodes de suréchantillonnage pour s'assurer que de tels problèmes sont évités. Nous avons proposé une méthode d'évaluation des algorithmes de suréchantillonnage qui tient compte à la fois de leur efficacité et de leur potentiel à introduire du bruit détectable. En évaluant la capacité du modèle à distinguer les données synthétiques des données réelles, nous pouvons identifier les techniques de suréchantillonnage qui introduisent trop de bruit et ne sont pas efficaces. Cette approche permet aux chercheurs de sélectionner les meilleures techniques de suréchantillonnage pour leurs jeux de données spécifiques et d'améliorer la précision et la généralisation de leurs modèles. En plus, cela peut aider à déterminer si le suréchantillonnage est une option viable pour équilibrer le jeu de données.

Pour les recherches futures, notre objectif principal est de développer un algorithme de suréchantillonnage spécialement conçu pour répondre aux particularités uniques des données médicamenteuses. En outre, nous souhaitons étudier les avantages potentiels de différentes techniques existantes, telles que l'apprentissage par transfert, l'apprentissage par ensemble, d'augmentation des données et l'apprentissage sensible aux coûts, afin d'améliorer les performances des modèles d'apprentissage automatique sur des jeux de données déséquilibrés. Ainsi, une direction future intéressante de ce travail serait (1) d'essayer différentes approches individuellement et de comparer leurs performances ; et (2) d'essayer de combiner les différentes approches ensemble.

Remerciements

Ce travail a été financé par l'agence nationale de la recherche (ANR) via le projet ABiMed [grant number ANR-20-CE19-0017-02].

References

[1] Batista, G. E., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1), 20–30.

[2] Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2009). Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. *In Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 475-482, Springer.

[3] Buda, R., Maki, A., Mazurowski, M. A. (2018). A Systematic Study of the Class Imbalance Problem in Convolutional Neural Networks. *Neural Networks*, 106, 249-259.

[4] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3), 15.

[5] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16(1), 321–357.

[6] Chen, C., Liaw, A., & Breiman, L. (2004). Using random forest to learn imbalanced data. *University of California, Berkeley*, 110, 24–31.

[7] Davis, J. and Goadrich, M. (2006). The Relationship Between Precision-Recall and ROC Curves. In *Proceedings of the 23rd International Conference on Machine Learning*, 233-240.

[8] Drummond, C., & Holte, R. (2003). C4. 5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In *Proceedings of the ICML'03 Workshop on Learning from Imbalanced Datasets*

[9] Fawcett, T. (2006). An Introduction to ROC Analysis. In *Pattern Recognition Letters*, 27(8), 861-874.

[10] Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *International Conference on Intelligent Computing*, 878-887, Springer.

[11] He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *IEEE International Joint Conference on Neural Networks*, 1322–1328.

[12] He, H. and Garcia, E.A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), 1263-1284.

[13] Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1-3), 489–501.

[14] Johnathan W. Hamrick, Diane Nykamp (2015). Drug-Induced Bleeding. *US Pharmacist*, 40(12), HS17-HS21.

[15] Johnson, A., Bulgarelli, L., Pollard, T., Celi, L. A., Mark, R., & Horng, S. (2021). MIMIC-IV (version 1.0). *PhysioNet*.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[17] Longadge, Rushi and Dongre, Snehalata (2013). Class imbalance problem in data mining: Review. In *International Journal of Computer Science and Network*

[18] Powers, D.M.W. (2011). Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness and Correlation. In *Journal of Machine Learning Technologies*, 2(1), 37-63.

[19] McHugh, M.L. (2012). Interrater Reliability: The Kappa Statistic. In *Biochemia Medica*, 22(3), 276-282.

[20] Rodríguez-Torres F, Martínez-Trinidad JF, Carrasco-Ochoa JA (2022). An Oversampling Method for Class Imbalance Problems on Large Datasets. *Applied Sciences*, 12(7), 3424.

[21] Rok Blagus and Lara Lusa (2013). SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics*, 14,106 .

[22] Tarawneh, S., Al-Betar, M. A., & Mirjalili, S. (2022). Stop oversampling for class imbalance learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2), 340-354.

[23] U.S. Food and Drug Administration. (n.d.). National Drug Code (NDC) Directory.

[24] WHO Collaborating Centre for Drug Statistics Methodology. (2013). Guidelines for ATC classification and DDD assignment 2013. Oslo, Norway: WHO Collaborating Centre for Drug Statistics Methodology.

[25] World Health Organization. (2016). International classification of diseases, 11th revision (ICD-11). Geneva: World Health Organization.

[26] Xu, Youjun & al. Deep Learning for Drug-Induced Liver Injury. *Journal of chemical information and modeling* vol. 55,10 (2015): 2085-93.

A Review of Validation and Verification of Neural Network-based Policies for Sequential Decision Making

Q. Mazouni[1], H. Spieker[1], A. Gotlieb[1], M. Acher[2]

[1] Simula Research Laboratory, Oslo, Norway
[2] Univ Rennes, Inria, INSA Rennes, CNRS, IRISA, France

quentin@simula.no

Résumé

Les réseaux de neurones sont aujourd'hui couramment utilisés pour représenter et apprendre la stratégie des agents pour la prise de décision séquentielle. Ce domaine d'application implique de nouveaux défis d'évaluation de la qualité des logiciels que les pratiques de validation et de vérification traditionnelles ne sont pas en mesure de résoudre. En conséquence, des approches novatrices ont émergé pour adapter ces techniques aux stratégies basées sur les réseaux de neurones pour la prise de décision séquentielle. Ce document vise à résumer ces nouvelles contributions et à proposer de futures directions de recherche. Nous avons effectué une revue de la récente littérature (de 2018 à 2023), dont les sujets couvrent des aspects du test ou de la vérification des stratégies basées sur les réseaux de neurones. La sélection des travaux a de plus été enrichie par un processus boule de neige à partir de ceux précédemment sélectionnés, afin d'étendre la portée de cette étude et de fournir au lecteur des informations sur les défis de vérification similaires et leurs récentes solutions. Finalement, nous avons selectionné 18 articles. Nos résultats témoignent d'un intérêt croissant pour cette problématique. Ils mettent en évidence la diversité des problèmes exacts considérés et des techniques utilisées pour y faire face.

Mots-clés

Test, Réseau de neurones, Prise de décision séquentielle.

Abstract

In sequential decision making, neural networks (NNs) are nowadays commonly used to represent and learn the agent's policy. This area of application has implied new software quality assessment challenges that traditional validation and verification practises are not able to handle. Subsequently, novel approaches have emerged to adapt those techniques to NN-based policies for sequential decision making. This survey paper aims at summarising these novel contributions and proposing future research directions. We conducted a literature review of recent research papers (from 2018 to beginning of 2023), whose topics cover aspects of the test or verification of NN-based policies. The selection has been enriched by a snowballing process from the previously selected papers, in order to relax the scope of the study and provide the reader with insight into similar verification challenges and their recent solutions. 18 papers have been finally selected. Our results show evidence of increasing interest for this subject. They highlight the diversity of both the exact problems considered and the techniques used to tackle them.

Keywords

Software testing, Neural networks, Sequential decision making.

1 Introduction

Last years have seen tremendous advances in solving sequential decision making problems with neural networks (NNs). For example, in game playing [29, 30] or Artificial Intelligence (AI) Planning [35, 21, 20]. Testing software systems that employ these NNs as policies is difficult and encounters several novel challenges. On one hand, most of the work on NN verification focus on single calls (like in image classification), without thus accounting for the sequential decisions made by the policy tested. On the other hand, traditional validation and verification (V&V) techniques for sequential decision making must deal with the major paradigm shift involved by the use of NNs, where the logic of the program has been learned rather than "coded". In fact, they can either leverage the subsequent, specific information of NN white-box testing (like NNs' architecture and weights) or assume no information at all about the NN-based policy under test (i.e, black-box testing). Fortunately, a new research area dedicated to the V&V of NN-based policies has emerged. However, its contributions are very diverse: they don't share the same testing goals and assumptions, and their respective limitations are unclear.
This paper aims at covering this recent literature (from 2018 to beginning of 2023), by detailing all the sub-problems addressed, the methodologies used and their current limitations. To cope with the diverse nature of the contributions and their respective degree of maturity, we categorise them (methodology-wise) and include in their review insight into their applicability. Furthermore, we highlight the subsequent remaining challenges and suggest possible ideas to address them. Our purpose is to provide the reader with

a succinct, yet comprehensive view of this evolving area of research, in order to stimulate the latter and guide interested researchers towards the uncovered problems. We thus address the following questions:

- What approaches have been recently proposed to address the V&V of NN-based policies?
- What are remaining gaps with respect to the V&V of NN-based policies?

Other works have reviewed related topics with different methodologies than ours. Corso et al. (2022) [8] cover autonomous cyber-physical systems (CPSs) rather than NN-based policies. Closer to this work, Zhang and Li (2020) [40] review NN-based CPSs. However, they propose a systematic literature review (SLR), which drastically differs from our scope and methodology. Besides, it does not include the most recent contributions (the papers were gathered from 2011 to 2018). Eventually, Tambon et al. (2022) [32] propose another SLR, that aims at answering the question of the certification of learned-based safety-critical systems. Therefore, their study adopts a broader approach than ours and, as [40], lacks the very last contributions this review covers.
We recognise that our survey is not exhaustive and that it does not cover all the literature directory (as the previously aforementioned SLRs do). Instead, we aim at analysing a more restricted and precise research topic, which is the V&V of NN-based policies for sequential decision making. The rest of the paper is organised as follows. Section 2 introduces the relevant notions and concepts that are discussed throughout this paper. We then describe our paper search and selection methodology in Section 3, as well as statistical results over the 18 papers analysed. Section 4 details the review of the papers. In Section 5, we synthesise the observed limitations. Section 6 concludes this paper by elaborating on future research directions.

2 Background

In this section, we introduce the key concepts which are at the core of this study: sequential decision making and neural networks as policies.

2.1 Sequential Decision Making

Informally, sequential decision making refers to tasks that can be solved by any decision theory in a step by step manner and which accounts for the dynamics of the environment [1]. In our study, we consider goal-oriented sequential decision making problems, where an agent starting from an initial state of the world can interact with the environment (e.g, simulations) through step-wise observation-decision-action processes until a satisfying state is reached. A typical example is the case of path planning in Robotics, where the agent is expected to safely reach a given position from an initial situation. The papers studied in this survey formalise sequential decision making problems as Markov Decision Processes (MDPs). It is defined as 4-tuple $\langle S, A, R, P \rangle$ where:

- S is a set of states. Referred as observation space, it specifies what the agent can know about its environment.
- A is the set of actions. Referred as action space, it specifies how the agent can act on its environment.
- R is the reward function. It reflects the agent's performance by associating any pair of state-action with a numerical value. In goal-oriented problems, such functions are often sparse, meaning that the agent receives positive rewards only for goal states (0 otherwise).
- P is the transition function, which is a probability distribution over the observation and the action space. It depicts which state the environment will transit to after an action is executed.

Solutions to MDPs are called policies (noted π), which are functions mapping from the observation space S to the action space A.

2.2 Neural Networks as Policies

The papers studied in this review consider policies as NNs. In such a context, the inputs of the networks are usually the observation space of the MDP of a decision making problem (or a slightly adapted version), while their outputs describe a probability distribution over the possible actions. Consequently, an agent following a policy π means that at every time step t, it chooses the next action a_{t+1} whose probability is the highest, i.e, $a_{t+1} = \arg\max \pi(s_t))$. Furthermore, we introduce stateless and stateful agents, since some contributions specifically target one or the other. Stateless agents follow policies modeled by feed forward neural networks (FFNNs), which consist in multiple hidden layers and admit no cycles [2]. On the other hand, stateful agents rely on recurrent neural networks (RNNs), whose ability to employ sequential data let them recall information [23] (thus providing the agent with a "memory"). Note that such stateful problems have an extended definition compared to the one defined above. Precisely, the observation space is a set of states (i.e, the current and past observations).

3 Methodology

In this section, we first elaborate on the selection strategy of papers for our literature review. Then, we provide a succinct statistical analysis of the papers selected.

3.1 Paper Search Strategy

Since the topic of this survey is quite narrow, we did not conduct a keyword-based, automated search in digital libraries (as systematic literature reviews do). We adopted an iterative process instead, whose steps would increasingly enlarge the scope of the search. That way we were able to precisely control, for each paper, the relation between its contribution and the topic of the study, as well as monitor the total number of papers. Each iteration consisted in a traditional, two-step process where a first batch of papers

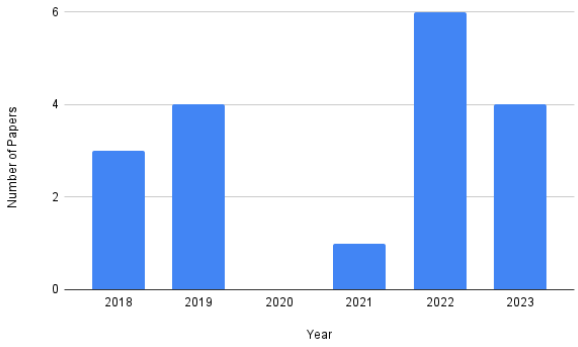


Figure 1: Number of analysed papers per year.

is reviewed and then we snowballed from their references. We sourced the papers in the first step from the 2022 edition of highly ranked conferences interested in either decision making and AI (ICAPS, IJCAI, AAAI) or software testing and engineering (ICSE, ISSTA, ESEC/FSE, IEEE TSE) and looked for terms related to V&V practices (e.g, testing, verification). In addition, we examined papers reviewed in other related surveys [8, 40, 32] whose topics match the scope of our study.

3.2 Selection Criteria

For each paper gathered during our iterative search, we looked at its title, keywords (if any) and read the abstract. We then completed the reading if (i) the abstract described interests in the validation and verification of programs for solving any decision making problem and (ii) whose assumptions correspond to the ones of NN-based policies (e.g, any learned models, black-box agents).
We adopted loose quality selection criteria. Indeed, the research topic studied is very active and, since we aim at providing the reader with as many diverse approaches as possible, we argue that only considering peer-reviewed papers would not have let us fulfill this goal. Therefore, we considered preprints too.

3.3 Statistical Results

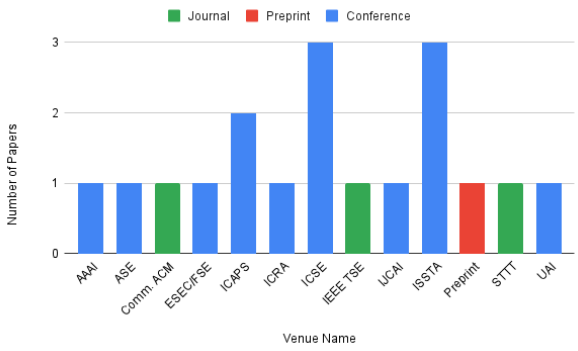


Figure 2: Venue distribution of the selected papers. The venue types are indicated with different colors.

Figure 1 shows the distribution of the papers' year of publication. We can see that most of the papers have been published in the last three years. Actually, more than half of them have been published either in 2022 or early 2023. This observation highlights how V&V of NN-based policies has recently become popular. Besides, Figure 2 depicts the venues represented by our paper selection along with their type. 14 papers were published in conferences, 3 in journals and 1 is a preprint.

4 Review

We derived from our literature review a set of categories that we use to classify the 18 papers selected. Figure 3 shows the result as a comprehensive taxonomy tree, where each leaf denotes a software V&V technique that best describes the general approach used by at least one of the papers selected. In the remainder of the section, we present each paper by describing its contribution, summarising its assumptions and highlighting its limitations.

4.1 Formal Verification Methods

In our context, verification methods ("Formal Verification" node on the left side of Figure 3) aim at proving that the agent following the policy under test is safe with respect to safety properties. These methods return SAT if the specifications are always satisfied (i.e the policy is verified) or UNSAT – with the associated counterexample – if they are not. A counterexample can for example be an entire execution trace of the agent (interacting with the environment) or a state of the world the agent led the simulation to.

4.1.1 Statistical Model Checking

Statistical Model Checking (SMC) [18] is an alternative to Model Checking [6] that aims at alleviating the well-known state explosion problem by combining simulation and statistical methods to provide statistical evidence for the satisfaction or violation of the specification. In essence, the model under test is simulated to generate samples, which are then evaluated with respect to a given property. As such, SMC provides an estimation of the value of the property, along with statistical results on the potential error. Gros et al. (2022) [15] propose Deep Statistical Model Checking (DSMC), where the NN-based policy under test is used as an action oracle to select the actions to perform during the execution of the MDP model of the problem. In other words, the verification is done through statistical model checking of the MDP whose transitions follow the policy tested. The authors evaluate their approach on a simplified version of the Racetrack benchmark[1], an autonomous driving challenge, where the objective is to reach the goal in a minimal number of steps without bumping into a boundary wall. Interestingly, DSMC has already been used to improve reinforcement learned policies under test [13], and has been integrated inside a toolbox, called MoGym [14], which enables the training along with the verification of machine-learned agents in an unified framework. In conclu-

[1] https://racetrack.perspicuous-computing.science/

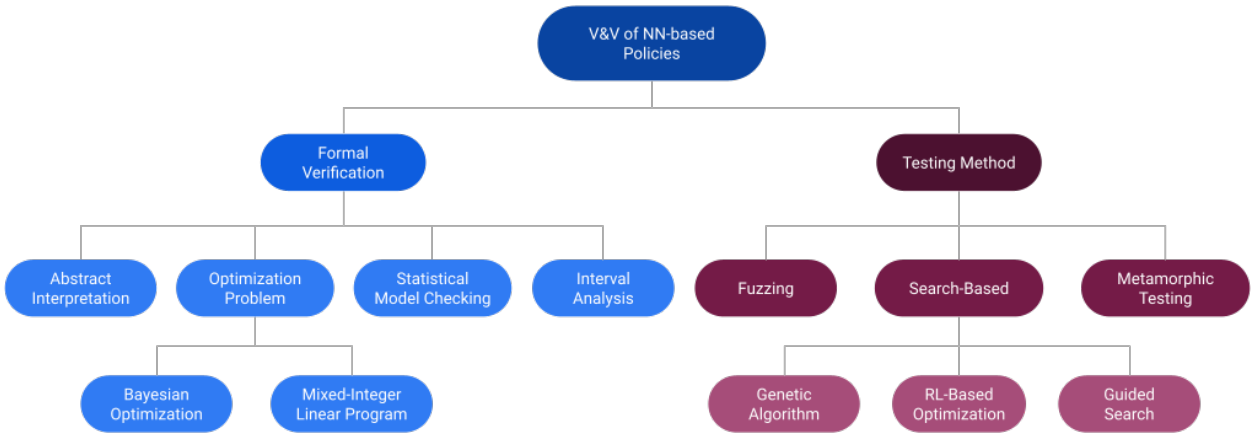


Figure 3: Representation of the taxonomy of the papers reviewed. Each leaf denotes a software testing approach leveraged by at least one of the works presented.

sion, this methodology verifies black-box NN-based policies, but requires a formal (and executable) model of the decision making problem (current implementation uses JANI [4] models).

4.1.2 Abstract Interpretation

One of the abstract interpretation techniques for formal verification consists in checking the specification against an abstract model which over-approximates the concrete model of the system under test. Thanks to such an over-approximation, the satisfaction of the property by the abstract model also proves the initial model's correctness. Vinzent et al. (2022) [37] use predicate abstraction [12] to compute a policy abstract state space and checks that none of its states violates a given safety property. They evaluate their approach with problems taken from the AI Planning literature, adapted to include unsafety conditions and non-deterministic actions. The results show that their framework outperforms standard predicate abstraction (thus ignoring the policy) and is more applicable than explicit enumeration and bounded model checking baselines.

This methodology has a limited applicability, as it requires a white-box model of the NN tested, as well as a formal model of the decision making problem. More importantly, the technique inputs the abstract predicates (i.e, they are not computed automatically), which significantly increases the amount of testing efforts. Besides, most of the optimizations studied only apply to NNs whose activation functions are rectified linear unit (ReLU).

4.1.3 Interval Analysis

Formal verification of NNs with reachability methods [38] leverages interval analysis [26] to compute and analyse the possible output sets of each layer of the network. However, the usual definition of safety properties does not allow their verification in the case of NN-based policies, where the networks' outputs typically encode a probability distribution over the actions. Corsi et al. (2021) [7] extend this approach to consider the multiple outputs of NN-based policies by introducing behavioral properties, and propose to deal with huge input state spaces of decision making problems (i.e, all the possible situations of the world) by computing an iterative bisection of the input intervals. In essence, their methodology consists in splitting the input space into areas for which the outputs' boundaries of the NN-based policy never overlap (i.e, the policy can be unambiguously evaluated). By doing so, the proposed framework is able to quantify the number of violations, from which a violation rate is derived (as the percentage of the input area that causes a violation). This metric brings better insight into how the policy performs with respect to the given properties (than the usual SAT/UNSAT output of verification procedures). Besides, their implementation, called ProVe, takes advantage of the computation independence of the intervals to parallelize the process. Consequently, the experimental evaluation conducted shows significant performance improvements over state-of-the-art NN verification tools. In conclusion, this technique does not need to know the dynamics of the decision making problem, but requires the NN-based policy tested to be a white-box and the safety properties have to be translated in behavioral ones.

4.1.4 Optimization Problem

The formal verification task can be seen as an optimization problem, where the verification procedure aims at finding a counterexample as fast as possible.

Bayesian Optimization. Ghosh et al. (2018) [11] encode safety properties as constraints and use Bayesian Optimization (BO) to solve the problem. To do so, the authors first compute a parse tree of the properties and estimate confidences of the lower bound values of the predicates with Gaussian Process. Then, they search for a counterexample in a active testing loop by iteratively minimising the predicates' variables through BO. The key idea is that the aforementioned variables are actually parameters of the environment, so each new environment selected minimizes

the worst case prediction of violating the properties. As a result, the exhaustive search is effectively guided towards adversarial environments. This methodology considers both the simulator and the NN-based policy as black-boxes, but requires a bound value on the agent's trajectories (i.e, bounded verification).

Mixed-Integer Linear Program. Akintunde et al. (2019) [3] tackle the case of ReLU-RNN-based policies (i.e, stateful policies) by unrolling the neural network to enable the use of existing white-box verification techniques for FFNNs. The subsequent problem is then solved with Mixed-Integer Linear Programming (MILP). They implement their approach in a tool called RNSVerify and compare two unrolling methods, namely: Input on Start (IOS) and Input on Demand (IOD). Experimental results show that IOD performs systematically better than IOS, since the number of variables and constraints are lower. In conclusion, this first research effort for the verification of stateful agents requires access to both the NN and the model of the problem. As for noticeable limitations, one can remark that such a model has to be linearly-definable (or linearly approximated), the exhaustive search is bounded (like [11]) and the experimental results revealed scalability issues.

4.2 Testing Methods

Testing methods (referred as "Testing Method" on Figure 3) aim at generating test cases to probe the quality of the policy under test with respect to evaluation criteria. The exact form and meaning of the test cases generated vary from paper to paper, as well as the definitions of the criteria. In any case, such criteria rely on the availability of an oracle (i.e, the expected correct output for a given input), which can be explicit translations of safety properties, computed automatically during testing (e.g, metamorphic testing [5]) or based on more advanced testing techniques like differential testing [24].

4.2.1 Metamorphic Testing

Zhang et al. (2018) [41] test image input-based NNs for autonomous driving systems to detect behavior inconsistencies with metamorphic testing (MT) [5]. MT is a testing technique that replaces test oracle checking with metamorphic relations (MRs) which assess the results of multiple program executions by specifying how given changes to an input should affect the output. As such, MR definitions are usually based on properties of the algorithms implemented. For example, a program that adds a and b should return the same result for the inputs (a, b) and (b, a) – whatever the actual result is – since the addition function is commutative.

In this work, the MRs induce weather-based scene changes which are assumed to keep the original semantic for the neural network (i.e, its outputs should not change). These transformations are generated by a model that learns to compute different versions (e.g, rainy, snowing) of a single input. To do so, the model combines a generative adversarial network and a variational autoencoder. In their experimental evaluation, their implementation – called DeepRoad – show better synthetic image transformations compared to DeepTest [34] (a competitive approach, described in 4.2.2), but this comes with the cost of training the aforementioned model first. In particular, such a training requires pair-set data collections, where images of same driving situations under different weather conditions have to be regrouped together.

4.2.2 Search-Based Testing

In the following, we analyse contributions whose overall frameworks are inspired by Search-Based Software Testing [25]. In general, search-based approaches consist in searching for input test cases and gathering the ones whose outputs reveal wrong behaviors of the policy under test. Similarly to other testing methods, erroneous behaviors are detected thanks to test oracles. Since the search space is in most cases very large, the challenge is to develop optimizing techniques to efficiently find fault-revealing test cases within the available test budget.

Genetic Algorithm. Zolfagharian et al. (2023) [42] and Haq et al. (2022) [16] optimize the search with genetic algorithms [19]. Such algorithms consider test cases as individuals of a population. The general idea is to iteratively let the population evolve (through crossover and mutation transformations) and only retain the most promising individuals (with a selection function) for the next iteration. Therefore, a typical genetic algorithm repeats the following until the test budget is consumed: (i) generating a new set of individuals with the crossover and mutation operators from the current population; (ii) calculating their fitness scores by executing the policy for every individual; (iii) keeping the test cases which revealed wrong behaviors (given a test oracle); (iv) selecting the individuals of the population for the next iteration with the selection function.

Zolfagharian et al. [42] test the policy of Reinforcement Learning (RL) based agents in a data-box testing setting (i.e, the training data is accessible) by finding faulty episodes. As such, the individuals of the population are episodes, and their genes are the state-action pairs of the execution traces of the latter. An abstract representation of the observation space (of the MDP model of the problem) is first computed with the aforementioned training data, which then lets their framework reason over abstract states to combine individuals of the population. Precisely, the combination of two episodes consists of the genes of the first individual but whose last genes, starting from a crossover point (randomly selected), are replaced by the ones of the second individual. The authors best preserve the consistency of the new episode (i.e, it can be executed by the RL agent) by checking the two concrete states designated by the crossover point belong to the same abstract class. They define three fitness functions, which favor low-reward episodes, episodes that maximise policy's uncertainty level and the ones that minimise the probability of functional faults, respectively. The probability of those faults is predicted with Machine Learning (Random Forest) whose model is trained in an initial step of the methodology with the training data of the agent under test.

In addition to the restricted scope of this work (it only applies to RL agents), we point out several weaknesses. First, the authors had a hard time ensuring the consistency or realism of the faulty episodes, since they are mutated. Furthermore, the validity of the test cases is only taken into account when they are finally compared with the execution of the agent, which may have a negative impact on performance. A more deeply-rooted weakness lies in the possible incomplete computation of the abstract state space. Indeed, since the latter is based on the episodes of the agent's training data, the abstraction of mutated episodes can be impossible (i.e, unseen abstract states are needed).

Haq et al. [16] also use a genetic algorithm, but aim at reducing the testing computation cost by assisting the search with surrogate models to avoid the expansive calls to the simulator. More precisely, they reduce the number of fitness function computations (and so the simulation calls) by predicating the best test cases through cooperation between global and local searches. The results of the predictions of each iteration are then used to improve the surrogate models' accuracy. Furthermore, to overcome the trade-off challenge of finding a balanced number of local surrogate models (accuracy versus performance), they introduce a clustering-based approach that generates one local surrogate model per cluster composed of test cases that belong to the same promising area.

RL-based Optimization. Lu et al. (2023) [22] and Haq et al. (2022) [17] turn the search problem into a RL task. The general idea is to train an agent to change in real-time the environment of the simulations towards situations where the policy under test exhibits faults. Lu et al. [22] consider complex, highly configurable environments for testing autonomous vehicles and learn a Deep Q-learning agent to find scenarios that maximise their collision. They compare the safety and current distances between the vehicle under test and its surrounding obstacles to estimate a collision probability (in a worst-case scenario) that is then used to define the reward function of the underlying MDP of the RL task. By linking the agent's rewards with the collision probability, their framework DeepCollision effectively trains the agent to guide the simulation towards collision-prone scenarios. On the other hand, Haq et al. [17] check several safety requirements (i.e, the problem statement is a many-objective search). Consequently, their framework MORLOT considers a Q-table per safety requirement and the choice of every next action is based on the Q-table whose related safety property has not been violated yet and whose reward for the previously chosen action was the maximum. Regarding the reward function definitions, they are not based on the maximisation of the collision probability of the vehicle under test (as proposed by Lu et al. [22]) but, rather, on the degree violation of the requirements. Those functions have thus to be defined for each safety property, as well as their respective maximum acceptable violation threshold.

Interestingly, by leveraging the same overall approach, these two works highlight its limitations and points of concern. Indeed, we can note that they both define the MDP model of the RL task with context-dependent knowledge (e.g, changing weather or traffic conditions). More importantly, they also have to constrain the agent with handcrafted, behavioral rules to ensure the consistency or realism of the simulations. Eventually, the two methodologies involve a significant number of parameters, whose definitions and performance impacts might be difficult to define and measure, respectively (more details in Section 5).

Coverage-guided Search. Tian et al. (2018) [34] detect erroneous behaviors of image input-based neural networks for driving autonomous cars as a neuron-coverage-guided greedy search. At each step, the input state space is further explored with new synthetic images which are generated with MT. The metamorphic operations to create those synthetic images involve weather condition changes (like adding fog or rain), and are assumed to keep the semantic of the original ones. As such, the metamorphic oracle checks if the outputs of the NN for the original and the new images are identical (given an error threshold). The search keeps track of the images which significantly increase the current neuron coverage to expand the input space. Even though the subsequent test cases do not eventually depict action scenarios like most of the works reviewed do, we decided to mention this work because we find the use of MT inside a guided search worth being mentioned. Regarding the scope and limitations of this approach (called DeepTest), one can note the use of neuron coverage. Indeed, in addition to recent concerns regarding the effectiveness of such a metric to guide search-based testing of NNs, it also restricts the approach to white-box testing setting. Besides, the metamorphic relations used in this work can misclassify correct behavior (since it is not guaranteed that the input transformations preserve the semantic of the images), meaning that the test suite generated is likely to include false positives.

Pei et al. (2019) [28] also consider behavior inconsistencies testing of image input-based NNs. This framework, DeepXplore, relies on a similar neuron-coverage-guided search (as DeepTest [34]). However, the oracle problem is not alleviated with MRs but, rather, with differential testing [24], thus detecting erroneous behaviors when the NNs' outputs are not at all the same. Consequently, the search is formulated as a joint optimization problem to maximise both the neuron coverage and these output differences. Of course, DeepXplore is therefore only geared towards testing of multiple (white-box) neural networks.

Du et al. (2019) [9] study quantitative analysis of stateful RNN-based systems. They define an abstract model computation algorithm of the NN under test along with several quantitative indicators to enable adversarial attack detection and coverage-guided testing. More precisely, they construct a Discrete-Time Markov chain by first applying state and transition abstractions on a set of concrete traces (of the RNN tested), and then computing transition probability distributions for each abstract state. The coverage metrics over the abstract model quantify its relative part exercised by a given concrete trace of the RNN, either state or transi-

tion wise. Those metrics are used to guide a test case search procedure similar to DeepTest [34]: new inputs are derived from the current test input with domain specific metamorphic transformations which are assumed to be semantically preservative. Consequently, faults are detected if the outputs of the RNN for the new inputs are not close enough to the initial one. The benefit of guiding the search with coverage metrics over an abstract model of the RNN under test lets the framework consider the latter as a black-box. However, the subsequent weakness is that the guidance efficiency depends on the accuracy of the abstraction. Indeed, in their experimental evaluation, the authors report that the level of abstraction granularity greatly impacts the resulting sensitivities of the coverage criteria.

Eventually, Tappler et al. (2022) [33] propose an unusual, yet demanding search-based framework for safety testing of RL agents. The authors aim at generating what they call boundary test cases, that correspond to safety-critical situations. Precisely, a safety-critical situation is defined as a state of the environment in which an action can lead to the violation of a given safety property. The crucial difference with all the other search-based methodologies reviewed is that the framework does not look for those boundary states but, rather, retrieved the latter from the state space explored by an initial backtracking-based, depth-first search for a solution of the decision making problem. The authors then define several test suites from those states. For instance, the "simple test suite" consists of all the states belonging to the paths that end in a boundary state. In conclusion, the key limitation of this methodology lies in the fact that it requires to solve the problem, and supposes that some boundary test cases will be found through the resolution computation. We further analyse that such a framework is difficult to apply to stochastic environments.

4.2.3 Fuzzing

This subsection covers works whose testing procedure relies on fuzzing. In general, fuzzing frameworks employ a pool of test case candidates (or seeds). At each step, a seed is taken from the pool and used to create new test cases (with random transformations or mutation operations). The ones that successfully make the policy violate the testing objective(s) are added to the test suite. Those whose selection criterion value is high enough are inserted in the pool (or their associated seed). Additionally, some methodologies define specific strategies – often called seed selection strategies – to pick the most promising candidate from the pool, rather than using common random sampling for example.

Pang et al. (2022) [27] consider initial simulation environments as seeds to test NN-based policies. Test cases are generated by randomly mutating those initial environments with user-defined operations. For each test case executed, the framework computes the state sequence density of the execution trace of the agent. The test cases with the highest density values are then used to feed the pool. The authors also propose to guide the choice of the next test case (to pick from the pool) with a metric called sensibility, which prioritises test cases that minimise the accumulated reward obtained by the agent (similarly to one of the fitness functions used in [42]). Their intuition is that execution traces with low returns would guide the search towards situations where the agent is less robust and therefore, less safe. Their approach, which considers both the simulator and the policy as black-boxes, is implemented in a generic testing tool called MDPFuzz. However, the mutation operations as well as the test oracles (to detect errors among the execution traces) are input parameters and context-dependent.

Xie et al. (2019) [39] study to what extent fuzzing frameworks are actually relevant for testing NNs in the first place. Like DeepTest, they consider image input-based NNs for decision making which are therefore not tested with simulation scenarios but, rather, with non-related test cases. They implement their approach in a tool called DeepHunter and extensively benchmark combinations of several seed selection and semantically preservative metamorphic mutation strategies, as well as well-known testing criteria to maintain the pool of candidates.

Eventually, Steinmetz et al. (2022) [31] and Eniser et al. (2022) [10] investigate the bug confirmation problem for NN-based action policies. More precisely, Steinmetz et al. [31] consider that a policy π contains a bug if another policy π' does better. The authors explore the use of heuristic functions used in classical AI Planning to automatically and efficiently compute test oracles (instead of computing such π' policies). They also introduce a policy quality bias in the action selection of the random walks involved in the fuzzing framework.

As for Eniser et al. [10], they leverage a similar fuzzing, pool-based testing framework but rely on MT to automatically derive both the new environment to test the policy with and the associated test oracle. To do so, the authors design the metamorphic operations around state relaxation, a well-studied concept also taken from the AI Planning community. Their idea is that a relaxed version of a given environment should represent an easier problem than the original one. Therefore, the agent's policy under test contains a bug if it solves the original problem but fails to solve its "relaxed" counterpart.

Some limitations regarding the works above are worth being mentioned. In MDPFuzz [27], the consistency of mutation operations are checked arbitrarily, which is a similar weakness we found in DeepCollision [22] and MORLOT [17]. Similarly, Xie et al. [39] constrain the metamorphic mutation operations of DeepHunter with conservative parameters in order to best ensure the semantic equivalence with the original images. Still, they eventually assume that they are sufficient to keep the semantics of the mutated images. The approach proposed by Eniser et al. [10] is currently limited to invariant checking (i.e, non-temporal failure condition that must hold in every simulation state) and the state relaxation functions are input parameters (i.e, context-dependent). As for the work of Steinmetz et al. [31], it currently comprises strong restrictions. In particular, they consider decision making problems of classical AI Planning, where the dynamics of the environ-

ment are specified (i.e, white-box environment) and deterministic.

5 Limitation Summary

We identify redundant limitations as well as similar challenges among the contributions reviewed. We think that synthesising these findings can serve as guidelines for future researchers. Note that these limitations come on top of already recognized difficulties related to either V&V practices or NNs, such as the high computational cost of some verification methods or the black-box nature of NN-based systems.

Context-dependent transformations. Methodologies that involve mutation operations or environment transformations usually define the latter with respect to specific knowledge. As a result, the concerned frameworks, while being generically applicable, actually demand domain expertise. For instance, [42, 27, 17] use mutation operations whose definitions are bound to the model of each decision making problem.

Assessing the consistency of the test cases is challenging. Most of the works that mutate the test cases and/or change the environment of the simulations end up arbitrarily checking the consistency of the results. It is for example the case of MORLOT [17] and DeepCollision [22], which enforce the consistency of the online simulation modifications with hand-coded rules (e.g, the RL agents cannot modify the weather conditions "too quickly"). Interestingly, MDPFuzz [27] mitigates this issue by generating the test cases before testing (instead of applying online modifications).

Hyperparameters. One can distinguish two issues: methodologies with a significant number of parameters (often RL-based) and/or the ones for which the values of the parameters greatly impact the performance. For instance, the work of Zolfagharian et al. [42] has an important number of parameters which, as a consequence, involves a great amount of effort to fine-tune. On the other hand, Lu et al. [22] reported that too short time intervals between each action of their RL agent sometimes led simulations to chaotic driving situations.

Spurious results. Some works rely on test oracles which might misclassify actually correct behavior. For example, [39, 41, 34] involve metamorphic operations that suppose the new inputs share the same semantic of their original counterparts, which is unfortunately not guaranteed (and difficult to assess). Similarly, the test oracles of MORLOT [17] depend on maximum acceptable degrees of violation, whose thresholds are left to hyperparameters.

Scope of applicability. Some frameworks have significant applicability restrictions, that we see as noteworthy limitations. They can be on-purpose limitations (e.g, RNSVerify [3] specifically addresses the verification of RNN policies), strong testing assumptions (e.g, DeepXplore [28] test multiple, white-box NNs) or are due to the early nature of the methodologies (e.g, Eniser et al. [10] assume single non-temporal failure conditions and leave temporal properties as future work).

6 Future Research Direction

Based on our literature review, we conclude this paper by elaborating on future research directions to guide and stimulate efforts for the V&V of NN-based policies.

Extending existing works. An important part of the papers analysed are pioneering works (e.g, [10, 31, 37]). As such, they have opened a new research area and raised questions which are now left to be addressed. For instance, as mentioned previously, Eniser et al. [10] plan to extend their framework to temporal safety properties. The generalisation of the V&V techniques could also be increased by relaxing their current testing setting requirements (e.g, white-box to black-box, deterministic to stochastic environments).

Refining existing works. Research efforts are needed towards improving the methodologies themselves: whether it is performance enhancement (e.g, Haq et al. [16] investigated surrogate models for many-objective search) or ease of applicability (e.g, enabling automated design of state relaxations in [10]). To that regard, Vinzent and Hoffmann (2022) [36] have recently enabled automated predicate abstraction computation (introduced in [37]) with counter-example guided abstraction refinement (CEGAR).

Combining different approaches/techniques. Several research opportunities could consist in combining parts of existing works with each other to begin with. For instance, fuzzing frameworks can benefit from better guidance for their seed selection strategy and testing criterion. Furthermore, we find research interests in the investigation of different well-known software testing techniques than the ones the works reviewed opted for. For example, Akintunde et al. [3] noted that they could have solved the unrolled FFNN verification task with SMT (instead of MILP).

Benchmarking and comparison studies. Empirical studies are needed to better assess the applicability of some approaches. Moreover, this emerging research area lacks comparison evaluations. To that regard, we especially think that a scaling comparison between formal verification and testing methods could reveal possible limitations regarding the former.

Explainability of the policies tested. Typical verification frameworks output SAT/UNSAT (possibly with a counterexample), which is not enough in the context of V&V of intangible policies such as NN-based models. More informative results would help software engineers to understand and fix the non-compliant behaviors reported. Some of the works reviewed have already contributed to this end, like ProVe [7] or DSMC [15], where a violation rate that quantifies the number of specification violations is introduced and a complete quality analysis of the neural network is provided, respectively. Similarly, finding fault-revealing test cases is far from being the only desirable feedback software engineers need. We argue that it is the very first step instead. Sharing this observation, we report early works among the ones reviewed. For example, MDPFuzz [27] investigates the visualisation of the distributions of the activated neurons by the fault-revealing states. Additionally,

we recall that [31, 10] investigate bug confirmation, that captures avoidable failures among the defects of the policy under test.

Acknowledgements

This work is funded by the Norwegian Ministry of Education and Research and part of the RESIST_EA Inria-Simula associate team.

References

[1] *The Cambridge Handbook of Artificial Intelligence*. Cambridge University Press, 2014. doi: 10.1017/CBO9781139046855.

[2] Oludare Isaac Abiodun, Aman Bin Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 2018.

[3] Michael E. Akintunde, Andreea Kevorchian, Alessio Lomuscio, and Edoardo Pirovano. Verification of rnn-based neural agent-environment systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[4] Carlos E. Budde, Christian Dehnert, Ernst Moritz Hahn, Arnd Hartmanns, Sebastian Junges, and Andrea Turrini. Jani: Quantitative model and tool interaction. In Axel Legay and Tiziana Margaria, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, 2017.

[5] T.Y. Chen, S.C. Cheung, and S.M. Yiu. Metamorphic Testing: A New Approach for Generating Next Test Cases. Technical report, Department of Computer Science, Hong Kong University of Science and Technology, 1998.

[6] Edmund M. Clarke and Bernd-Holger Schlingloff. Chapter 24 - model checking. In *Handbook of Automated Reasoning*. North-Holland, 2001.

[7] Davide Corsi, Enrico Marchesini, and Alessandro Farinelli. Formal verification of neural networks for safety-critical tasks in deep reinforcement learning. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, 2021.

[8] Anthony Corso, Robert Moss, Mark Koren, Ritchie Lee, and Mykel Kochenderfer. A survey of algorithms for black-box safety validation of cyber-physical systems. *J. Artif. Int. Res.*, 2022.

[9] Xiaoning Du, Xiaofei Xie, Yi Li, Lei Ma, Yang Liu, and Jianjun Zhao. Deepstellar: Model-based quantitative analysis of stateful deep learning systems. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2019.

[10] Hasan Ferit Eniser, Timo P. Gros, Valentin Wüstholz, Jörg Hoffmann, and Maria Christakis. Metamorphic relations via relaxations: An approach to obtain oracles for action-policy testing. In *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2022.

[11] Shromona Ghosh, Felix Berkenkamp, Gireeja Ranade, Shaz Qadeer, and Ashish Kapoor. Verifying controllers against adversarial examples with bayesian optimization. *CoRR*, 2018.

[12] Susanne Graf and Hassen Saidi. Construction of abstract state graphs with pvs. In *Computer Aided Verification*, 1997.

[13] Timo P. Gros, Daniel Höller, Jörg Hoffmann, Michaela Klauck, Hendrik Meerkamp, and Verena Wolf. Dsmc evaluation stages: Fostering robust and safe behavior in deep reinforcement learning. In Alessandro Abate and Andrea Marin, editors, *Quantitative Evaluation of Systems*, 2021.

[14] Timo P. Gros, Holger Hermanns, Jörg Hoffmann, Michaela Klauck, Maximilian A. Köhl, and Verena Wolf. Mogym: Using formal models for training and verifying decision-making agents. In Sharon Shoham and Yakir Vizel, editors, *Computer Aided Verification*, 2022.

[15] Timo P. Gros, Holger Hermanns, Jörg Hoffmann, Michaela Klauck, and Marcel Steinmetz. Analyzing neural network behavior through deep statistical model checking. *International Journal on Software Tools for Technology Transfer*, 2022.

[16] Fitash Ul Haq, Donghwan Shin, and Lionel Briand. Efficient online testing for dnn-enabled systems using surrogate-assisted and many-objective optimization. In *2022 IEEE/ACM 44th International Conference on Software Engineering (ICSE)*, 2022.

[17] Fitash Ul Haq, Donghwan Shin, and Lionel Briand. Many-objective reinforcement learning for online testing of dnn-enabled systems, 2022.

[18] Thomas Hérault, Richard Lassaigne, Frédéric Magniette, and Sylvain Peyronnet. Approximate probabilistic model checking. In Bernhard Steffen and Giorgio Levi, editors, *Verification, Model Checking, and Abstract Interpretation*, 2004.

[19] John H. Holland. Genetic algorithms. *Scientific American*, 1992.

[20] Murugeswari Issakkimuthu, Alan Fern, and Prasad Tadepalli. Training deep reactive policies for probabilistic planning problems. *Proceedings of the International Conference on Automated Planning and Scheduling*, 2018.

[21] Rushang Karia and Siddharth Srivastava. Learning generalized relational heuristic networks for model-agnostic planning. *CoRR*, 2020.

[22] Chengjie Lu, Yize Shi, Huihui Zhang, Man Zhang, Tiexin Wang, Tao Yue, and Shaukat Ali. Learning configurations of operating environment of autonomous vehicles to maximize their collisions. *IEEE Transactions on Software Engineering*, 2023.

[23] Danilo P. Mandic and Jonathon A. Chambers. Recurrent neural networks for prediction: Learning algorithms, architectures and stability. 2001.

[24] William M. McKeeman. Differential testing for software. *Digit. Tech. J.*, 1998.

[25] Phil McMinn. Search-based software testing: Past, present and future. In *2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops*, 2011.

[26] Ramon Edgar Moore. *Interval Arithmetic and Automatic Error Analysis in Digital Computing*. PhD thesis, 1963.

[27] Qi Pang, Yuanyuan Yuan, and Shuai Wang. Mdpfuzz: Testing models solving markov decision processes. In *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2022.

[28] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. Deepxplore: Automated whitebox testing of deep learning systems. *Commun. ACM*, 2019.

[29] David Silver, Aja Huang, Christopher Maddison, Arthur Guez, Laurent Sifre, George Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 2016.

[30] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 2018.

[31] Marcel Steinmetz, Daniel Fišer, Hasan Ferit Eniser, Patrick Ferber, Timo P. Gros, Philippe Heim, Daniel Höller, Xandra Schuler, Valentin Wüstholz, Maria Christakis, and Jörg Hoffmann. Debugging a policy: Automatic action-policy testing in ai planning. *Proceedings of the International Conference on Automated Planning and Scheduling*, 2022.

[32] Florian Tambon, Gabriel Laberge, Le An, Amin Nikanjam, Paulina Stevia Nouwou Mindom, Yann Pequignot, Foutse Khomh, Giulio Antoniol, Ettore Merlo, and François Laviolette. How to certify machine learning based safety-critical systems? a systematic literature review. *Automated Software Engg.*, 2022.

[33] Martin Tappler, Filip Cano Córdoba, Bernhard K. Aichernig, and Bettina Könighofer. Search-based testing of reinforcement learning, 2022.

[34] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th International Conference on Software Engineering*, 2018.

[35] Sam Toyer, Felipe W. Trevizan, Sylvie Thiébaux, and Lexing Xie. Asnets: Deep learning for generalised planning. *CoRR*, 2019.

[36] Marcel Vinzent and Joerg Hoffmann. Neural Policy Verification via Predicate Abstraction: CEGAR. page 9, 2022.

[37] Marcel Vinzent, Marcel Steinmetz, and Jörg Hoffmann. Neural network action policy verification via predicate abstraction. *Proceedings of the International Conference on Automated Planning and Scheduling*, 2022.

[38] Weiming Xiang, Hoang-Dung Tran, Joel A. Rosenfeld, and Taylor T. Johnson. Reachable set estimation and safety verification for piecewise linear systems with neural network controllers. In *2018 Annual American Control Conference (ACC)*, 2018.

[39] Xiaofei Xie, Lei Ma, Felix Juefei-Xu, Minhui Xue, Hongxu Chen, Yang Liu, Jianjun Zhao, Bo Li, Jianxiong Yin, and Simon See. Deephunter: A coverage-guided fuzz testing framework for deep neural networks. In *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2019.

[40] Jin Zhang and Jingyue Li. Testing and verification of neural-network-based safety-critical control software: A systematic literature review. *Information and Software Technology*, 2020.

[41] Mengshi Zhang, Yuqun Zhang, Lingming Zhang, Cong Liu, and Sarfraz Khurshid. Deeproad: Gan-based metamorphic testing and input validation framework for autonomous driving systems. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, 2018.

[42] Amirhossein Zolfagharian, Manel Abdellatif, Lionel Briand, Mojtaba Bagherzadeh, and Ramesh S. A search-based testing approach for deep reinforcement learning agents, 2023.

Amélioration de la recherche d'architecture neuronale en combinant un algorithme FireFly avec une évaluation sans apprentissage

N. Mokhtari, A. Nédélec, M. Gilles, P. De Loor

Lab-STICC (CNRS UMR 6285) - ENIB
Centre Européen de Réalité Virtuelle
Brest, France

{nassim.mokhtari, alexis.nedelec, marlene.gilles, pierre.deloor}@enib.fr

Résumé

Les algorithmes de recherche d'architecture neuronale (Neural Architecture Search - NAS) sont utilisés pour automatiser la conception de réseaux de neurones profonds. Trouver la meilleure architecture pour un jeu de données peut prendre beaucoup de temps car ces algorithmes doivent explorer un grand nombre de réseaux et les évaluer pour choisir le plus approprié. Dans ce travail, nous proposons une nouvelle métrique qui utilise le score Intra-Cluster Distance (ICD) pour évaluer la capacité d'un modèle non entraîné à distinguer les données sans avoir à les entraîner.Pour la recherche de l'architecture, nous utilisons une méta-heuristique de type FireFly améliorée, plus robuste face au problème des optimums locaux que l'algorithme FireFly de base, comme technique de recherche pour trouver le meilleur modèle de réseau de neurones pour un ensemble spécifique de données. Les résultats expérimentaux sur les différents NAS Benchmarks montrent que notre métrique est valable pour l'évaluation des réseaux de neurones convolutifs ainsi que des réseaux de neurones récurrents, et que l'algorithme FireFly que nous proposons peut améliorer les résultats obtenus par les méthodes sans entraînement les plus récentes.

Mots-clés

Apprentissage profond, recherche d'architecture neuronale, score sans entraînement, distance intra-groupe, apprentissage automatique, méta-heuristiques, algorithme FireFly.

Abstract

Neural Architecture Search (NAS) algorithms are used to automate the design of deep neural networks. Finding the best architecture for a given dataset can be time consuming since these algorithms have to explore a large number of networks, and score them according to their performances to choose the most appropriate one. In this work, we propose a novel metric that uses the Intra-Cluster Distance (ICD) score to evaluate the ability of an untrained model to distinguish between data in order to approximate its quality. We also use an improved version of the FireFly algorithm, more robust to the local optimums problem than the baseline FireFly algorithm, as a search technique to find the best neural network model adapted to a specific dataset. Experimental results on the different NAS Benchmarks show that our metric is valid for either scoring CNNs and RNNs, and that our proposed FireFly algorithm can improve the result obtained by the state-of-art training-free methods.

Keywords

Deep Learning, Neural Architecture Search, Training-free Score, Intra Cluster Distance, Machine Learning, Metaheuristics, FireFly algorithm

1 Introduction

Ce document est la traduction de notre travail publié dans *International Joint Conference on Neural Networks (IJCNN) 2022* [1].

Les avancées récentes dans le domaine de la classification d'images et de la reconnaissance vocale liées à la recherche sur l'apprentissage profond, en particulier sur les réseaux de neurones convolutifs, ont montré l'utilité de ces derniers pour l'extraction de caractéristiques et la classification, et semblent les mieux adaptées à de nombreux problèmes de classification [2–5].

En raison de la croissance du nombre d'hyperparamètres (couches cachées, unités cachées, etc.) utilisés pour définir les architectures d'apprentissage profond et, par conséquent, la hausse du nombre d'organisations de réseau possibles (augmentation exponentielle) impliquées par ces hyperparamètres, un nouveau défi consiste à trouver des solutions pour concevoir l'architecture elle-même à l'aide d'algorithmes plutôt que manuellement. Pour ce faire, la communauté de l'apprentissage profond a introduit la recherche d'architecture neuronale *(Neural Architecture Search - NAS)*, des algorithmes capables d'automatiser la découverte d'architectures efficaces [8–13].

Afin de comparer l'efficacité des NAS, la communauté du NAS a conçu des *benchmarks* spécifiques destinés à cet effet : [14–17]. En effet, le choix de l'architecture d'un réseau de neurones peut être vu comme un problème combi-

natoire, où l'objectif est de trouver la combinaison d'hyperparamètres (nombre de couches, taille des couches, etc.) qui offre les meilleures performances. Les *benchmarks* fournissent un espace de recherche fini d'architectures, qui peut être utilisé pour comparer les algorithmes de NAS. Les méta-heuristiques sont souvent utilisées pour résoudre ce type de problèmes, et il existe plusieurs travaux qui exploitent ces méthodes [18–22]. L'utilisation d'une méta-heuristique implique l'évaluation des solutions (architectures de réseaux de neurones) afin d'évaluer leurs qualités.
L'un des principaux problèmes liés à l'évaluation des performances d'un modèle est la phase d'apprentissage qui prend beaucoup de temps, entraînant un temps de recherche énorme qui peut prendre des jours, même en utilisant des centaines de GPU [23]. Il existe donc des méthodes permettant de contourner cette phase d'apprentissage et d'évaluer une architecture à partir de métriques basées sur la distribution de l'activation des neurones par rapport à différentes valeurs d'entrée rassemblées dans un *mini-batch* de données, comme la métrique de Mellor [23] et la proposition de Lopes et al. [24].
Notre contribution dans ce travail peut être résumée comme suit :

1. Une nouvelle métrique ***training-free*** qui peut approcher la qualité d'un réseau de neurones en évaluant sa capacité à distinguer les données. Cette métrique peut être utilisée pour évaluer plus de types de réseaux de neurones que les métriques précédentes.
2. Une version améliorée de l'algorithme *FireFly (Improved Firefly Algorithm - IFA)* qui utilise des opérateurs d'algorithmes génétiques, permettant à notre IFA d'être plus robuste aux optimums locaux que la version standard.
3. Une combinaison de la métrique proposée et de l'algorithme FireFly amélioré afin de trouver le modèle le plus intéressant dans un espace de recherche d'architecture de réseaux de neurones.
4. Une évaluation de notre proposition qui montre qu'elle surpasse l'état de l'art en terme de performance pour trouver l'architecture la plus adaptée à un problème de *machine learning*.

Le reste du document s'organise comme suit : la Section 2 introduit une synthèse des différents travaux réalisés dans le cadre de l'évaluation des réseaux de neurones pour les NAS. La Section 3 présente l'algorithme *FireFly*. Dans la Section 4, nous présentons notre méthode pour évaluer un réseau de neurones non entraîné et une proposition d'amélioration de l'algorithme *FireFly*. En Section 5 nous présentons les résultats expérimentaux obtenus sur 4 *benchmarks NAS*, montrant l'efficacité de notre proposition (valide et améliore les scores de l'état de l'art). Enfin, nous résumerons dans la Section 6 les résultats de ce travail ainsi que les améliorations possibles.

2 Travaux connexes

L'espace de recherche (ensemble de tous les réseaux possibles) étant très grand, l'évaluation de l'efficacité d'un algorithme NAS ne peut être faite de manière exhaustive. Ceci a conduit à la création de plusieurs benchmarks [14–17] qui consistent en des espaces de recherche NAS et des méta-données relatives à l'entraînement de ces réseaux dans cet espace de recherche [23].
Le NAS-Bench-101 est composé de 423 624 réseaux de neurones (CNN) qui ont été entraînés de manière exhaustive, avec trois initialisations différentes, sur le jeu de données CIFAR-10 pour 4, 12, 36 et 108 itérations (423K * 3 * 4 = 5M modèles entraînés au total) [14]. Le NAS-Bench-201 comprend 15 625 réseaux entraînés plusieurs fois sur CIFAR-10, CIFAR-100 et ImageNet-16-120 [15]. Le NAS-BENCH-NLP est composé de 14K réseaux de neurones (RNN) entraînés sur le jeu de données *Penn Tree Bank* et le jeu de données *WikiText-2* [17].
Les algorithmes NAS explorent l'espace de recherche afin de trouver le meilleur réseau. Cependant, l'entraînement de chaque architecture de réseau de neurones pour sélectionner la plus appropriée est un processus qui prend du temps. Par conséquent, la possibilité d'évaluer la qualité d'une architecture sans l'entraîner est une alternative pour trouver le réseau de neurones le plus approprié sans passer des jours à faire des calculs.
Mellor et al. [23] ont proposé une façon d'évaluer un réseau de neurones sans entraînement préalable (les poids du réseaux sont définis aléatoirement) , en identifiant un indicateur binaire qui se concentre uniquement sur les unités linéaires rectifiées (ReLU) du réseau (0 pour une unité inactive, et 1 pour une unité active). L'intuition derrière leur approche est que plus les codes binaires associés à deux entrées sont similaires, plus il est difficile pour le réseau d'apprendre à discriminer ces entrées. Un mini-batch de données X = [x_1, x_2, ..., x_n] est envoyé à travers un réseau de neurones (composé de ReLU et de plusieurs autres types d'unités) afin d'obtenir des codes binaires C = [c_1, c_2, ...,c_n], où chaque c_i (obtenu uniquement à partir des sorties ReLU) fait référence au code binaire de x_i. Pour évaluer le réseau de neurones, Mellor et al. [23] calculent le logarithme du déterminant d'une matrice K_h, où chaque composante est calculée en utilisant la distance de Hamming (Eq. (1)). Plus le score est élevé, meilleur est le réseau

$$K_h[i,j] = N_A - Hamming\_distance(c_i, c_j) \quad (1)$$

où N_A est le nombre d'unités ReLU.
Lopes et al. [24] ont proposé une autre façon d'évaluer un réseau de neurones sans entraînement. En se basant sur les travaux de Mellor et al. [23], ils ont proposé d'utiliser les codes binaires pour calculer une matrice jacobienne (J). L'objectif est de déterminer si un réseau non entraîné peut distinguer les opérateurs linéaires locaux pour chaque point de données, mais aussi obtenir des résultats similaires pour des points de données similaires (appartenant à la même classe dans une approche supervisée). Pour estimer ce comportement, ils évaluent la corrélation des valeurs de J par rapport à leur classe en calculant une matrice de covariance pour chaque classe présente dans J. Les corrélations sont d'abord évaluées individuellement, car elles peuvent avoir

des tailles différentes en raison du nombre de données par classe. Le score final est la somme des scores des matrices de corrélation individuelles.
L'exploration de toutes les architecture du benchmark NAS afin de trouver le meilleur modèle peut être une tâche difficile et longue, même si nous utilisons ce type de métrique pour éviter l'entraînement du réseau, en raison des millions de modèles inclus dans cet espace de recherche. Pour éviter cela, il existe plusieurs travaux qui exploitent des méta-heuristiques pour trouver le réseau le plus approprié à une tâche donnée afin d'éviter une recherche exhaustive. Par exemple, Sun et al. [20] ont utilisé un algorithme génétique pour concevoir automatiquement un réseau de neurones convolutif, en utilisant la précision du réseau entraîné comme fonction objectif de leur algorithme génétique.
Rere et al. [22] ont proposé plusieurs méta-heuristiques (algorithmes SA, DE et HS). Leur fonction objectif était l'erreur standard sur l'ensemble d'entraînement. Carvalho et al. [21] ont proposé une fonction objectif basée sur l'erreur d'entraînement et l'erreur de test, qui a été utilisée avec les algorithmes VNS, SA, GEO et GA. Ayumi et al. [19] ont utilisé un algorithme de recuit microcanonique *(Microcanonical Annealing Algorithm - MAA)* pour concevoir un réseau de neurones, en utilisant l'erreur durant la phase d'apprentissage comme fonction objectif. Strumberger et al. [18] ont préféré utiliser un algorithme FireFly pour concevoir leur CNN, où la fonction objectif utilisée était basée sur l'erreur obtenue sur l'ensemble de test. Mellor et al. [23] ont proposé d'utiliser un algorithme d'évolution régularisée assistée *(Assisted Regularised Evolution Algorithm - AREA)*, une version améliorée de REA proposée par Real et al. [13], combiné à la métrique qu'ils ont proposée pour trouver la meilleure architecture de réseau de neurones.
Dans ce travail, nous nous intéressons à l'utilisation d'une version améliorée de l'algorithme FireFly, combinée avec une métrique d'évaluation de modèle sans entraînement utilisée comme fonction objectif. Notons que cette métrique sera exploitable avec n'importe quel type de cellules.
Notre choix d'utiliser l'algorithme FireFly vient du fait que cette méthode inclut plusieurs méta-heuristiques telles que le recuit simulé *(Simulated Annealing - SA)*, ou l'optimisation par essaims de particules *(Particle Swarm Optimization - PSO)*, en plus de sa convergence rapide vers un optimum [25].

3 Présentation de l'algorithme FireFly (FA)

Inspiré par le comportement des lucioles, cet algorithme a été initialement développé par Xin-She Yang en 2008 [25]. Il est basé sur trois règles :

1. Les lucioles sont unisexes, donc une luciole est attirée par une autre sans tenir compte de son genre.
2. L'attirance est proportionnelle à la luminosité et toutes deux sont inversement proportionnelles à la distance (l'attirance et la luminosité diminuent lorsque la distance augmente). Pour toute paire de lucioles, la moins brillante sera attirée par la plus brillante, et se dirigera donc vers elle. S'il n'y a pas de luciole plus brillante, cette dernière se déplacera au hasard.
3. La luminosité d'une luciole (solution) est donnée par la fonction objectif.

Le mouvement d'une luciole i vers une luciole j plus lumineuse est défini par Eq. (2).

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}(x_j^t - x_i^t) + \alpha\tau_i^t \qquad (2)$$

— r : distance entre deux lucioles.
— α : poids du déplacement aléatoire.
— β_0 : attirance à r = 0.
— γ : coefficient d'absorption (paramètre à renseigner).
— x_i^t : position de la luciole i à l'instant t.
— τ_i^t : vecteur de nombres aléatoires donnés par une distribution gaussienne à l'instant t pour la luciole i.

Le fonctionnement de l'algorithme FireFly est présenté dans l'algorithme 1.

Algorithm 1 FireFly Algorithm

Générer aléatoirement une population de n lucioles x_i=(i=1,2,...,n).
L'intensité lumineuse I_i de chaque luciole x_i est donnée par f(x_i) où f est la fonction objectif
Définir les coefficients α, β et γ.
while critères d'arrêt non atteint **do**
  for i=1 to n **do**
    for j=1 to n **do**
      if I_i< I_j **then**
        Déplacer la luciole i vers la luciole j
      end if
      Faire varier l'attractivité en fonction de la distance r.
      Évaluer la nouvelle solution et mettre à jour l'intensité lumineuse.
    end for
  end for
  Renvoyer la meilleure solution
end while

4 Méthode proposée

Dans cette partie, nous présentons notre approche basée sur :

1. L'évaluation de la qualité d'un réseau de neurones avec une métrique sans entraînement.
2. L'utilisation d'un algorithme FireFly amélioré pour guider le choix parmi l'énorme quantité d'architectures possibles.

4.1 Évaluation de la qualité d'un modèle

Nous proposons une nouvelle façon d'évaluer un réseau de neurones sans l'entraîner, en faisant passer un mini batch de données par le réseau à évaluer afin d'obtenir le code binaire de chaque donnée d'entrée (0 pour une valeur négative et 1 pour une valeur positive). Nous suivons la méthodologie de Mellor et al. [23] sauf que nous utilisons toutes les unités du réseau, pas seulement celles ayant une fonction ReLU.

Notre intuition est que les unités qui utilisent une fonction d'activation autre que ReLU (ou n'en utilisant pas) peuvent aussi être utilisées pour formaliser la manière dont le modèle interprète les données (codes binaires). Puisque nous cherchons à évaluer la capacité d'un modèle à distinguer des données, nous pouvons éviter de vérifier les similarités (en calculant la co-variance comme le fait Lopes el al. [24] par exemple) et nous concentrer davantage sur le degré de différence entre les représentations. Pour ce faire, nous proposons d'utiliser la distance intra-cluster (ICD), calculée sur les codes binaires, comme métrique pour évaluer le réseau non entraîné. La figure 1 illustre notre processus d'évaluation du réseau, pour un mini-batch de données contenant 4 échantillons.

L'ICD est généralement utilisée pour évaluer la qualité d'une méthode de *clustering* où l'objectif est de trouver un *clustering* fournissant des groupes qui ont une petite distance intra-cluster. L'idée est de calculer la distance moyenne entre chaque point de données et le centre du cluster (moyenne des données).

Dans notre travail, nous évaluons le réseau selon l'ICD calculé en utilisant Eq. (3).

$$ICD = \frac{\sum_{n=1}^{N} d(\bar{c}, c_i)}{N} \tag{3}$$

Où d est la distance euclidienne, c_i est le code binaire de l'entrée x_i, $\bar{c}$ est le centre des codes binaires (moyenne des codes binaires) et N est le nombre total de codes binaires (nombre d'échantillons dans le mini-lot de données).

Une petite valeur ICD signifie que le cluster est compact (composé d'échantillons similaires), alors qu'une grande valeur signifie que le cluster est étiré (les échantillons sont différents). Puisque nous recherchons un réseau capable de fournir différents codes binaires pour différents échantillons, le cluster composé de ces codes binaires doit être le plus étiré possible, ce qui donne une valeur ICD élevée. Plus le score est élevé, meilleur est le réseau.

4.2 Improved FireFly Algorithm

Grâce au mécanisme d'attraction, l'algorithme FireFly atteint rapidement un optimum, ce qui augmente les chances de rester bloqué dans un optimum local, même si l'algorithme FireFly inclut une diversification (la marche aléatoire). Afin d'améliorer la diversification, nous proposons d'utiliser les opérateurs de l'algorithme génétique (sélection, croisement et mutation) qui pourraient permettre une meilleure exploration de l'espace de recherche en combinant les composants des solutions déjà explorées. Plus de détails sur l'implémentation de ces opérateurs de métaheuristiques peuvent être trouvés dans la Section 4.C.

Nous proposons d'exécuter une itération de l'algorithme génétique chaque fois que l'algorithme FireFly est bloqué dans un optimum local, afin de créer une nouvelle population, complètement différente de l'ancienne. Nous considérons que l'algorithme FireFly est bloqué dans un optimum local, s'il n'arrive plus à améliorer la meilleure solution après un certain nombre d'itérations (chances). Avant chaque exécution de l'algorithme génétique, l'optimum actuel est stocké dans une liste (*candidats*), le résultat final de l'exploration (lorsque le critère d'arrêt est atteint) est la meilleure solution de la liste des candidats. L'algorithme 2 illustre l'approche proposée.

L'utilisation de ce mécanisme donne la chance à notre algorithme proposé d'effectuer une diversification "mineure" (marche aléatoire de FireFly) qui lui permet de mieux explorer une région de l'espace de recherche, avant d'en explorer une autre obtenue par une diversification "majeure" grâce à l'algorithme génétique.

La sélection de la meilleure solution dans la liste des *candidats* est effectuée en fonction des performances des modèles après l'entraînement. Comme notre métrique ne peut qu'approximer la qualité du modèle et ne peut la mesurer exactement, le fait de conserver une liste de candidats augmente les chances de trouver une bonne architecture.

Le critère d'arrêt a été fixé en fonction du nombre de populations générées.

Algorithm 2 Improved FireFly Algorithm

```
Générer aléatoirement la population de solution
Définir MaxChances
chances = MaxChances
candidats = [ ]
LocalBest = NULL
while critères d'arrêt non atteint do
    Exécution d'une itération de FireFly
    Best_t = la meilleure solution pour la population actuelle
    if LocalBest = NULL then
        LocalBest = Best_t
    else
        if fitness(Best_t) ≥ fitness(Best_{t-1}) then
            LocalBest = Best_t
        else
            chances - -
        end if
    end if
    if chances = 0 then
        ajouter LocalBest à candidats
        LocalBest = NULL
        Effectuer une itération de l'Algorithme Génétique
        chances = MaxChances
    end if
end while
Déterminer la meilleure solution à partir de la liste des candidats
```

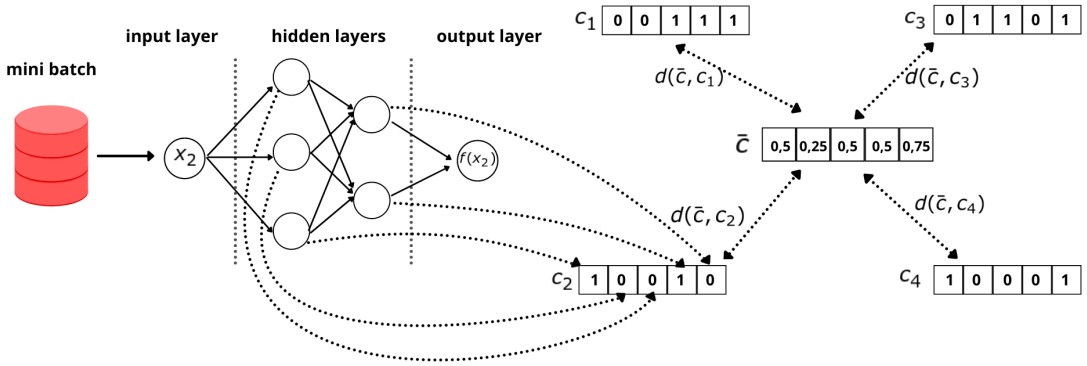


FIGURE 1 – Le processus proposé pour l'évaluation de la qualité d'un réseau de neurones : chaque échantillon x_i du mini-batch de données est utilisé pour générer un code binaire c_i, en utilisant les sorties des unités cachées données par x_i comme entrée du réseau de neurones. La moyenne de tous les codes binaires est $\bar{c}$ et est utilisée pour calculer la valeur du ICD.

4.3 Implémentation

Chaque solution (luciole) représente une architecture réseau contenue dans le NAS benchmark, puisque chaque benchmark a sa propre représentation d'une architecture, nous devons créer une implémentation dédiée pour chacun d'entre eux. Dans ce qui suit, nous décrirons notre choix pour représenter une solution, et l'implémentation des opérateurs de la métaheuristique.

4.3.1 NAS-BENCH-101

Dans le NAS-BENCH-101, tous les réseaux partagent le même squelette, qui est composé de 3 piles, chacune comprenant 3 cellules, comme le montre la partie gauche de la figure 2. Les réseaux sont différents dans le "module" (cellule), qui est représenté par des graphes acycliques dirigés (jusqu'à 7 sommets et 9 arêtes). Les opérations valides à chaque sommet sont "convolution 3x3", "convolution 1x1", et "max-pooling 3x3" [14]. La partie droite de la figure 2 montre un exemple de module dans le NAS-BENCH-101.

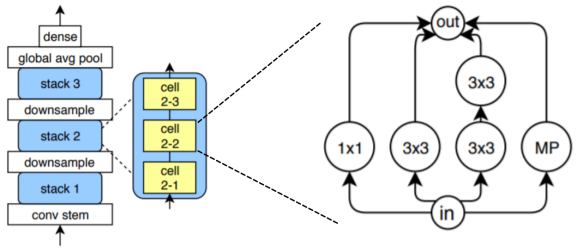


FIGURE 2 – Architecture d'un réseau dans le modèle NAS-BENCH-101 [14] : La partie gauche est le squelette partagé par tous les modèles, la partie centrale représente une pile de cellules alors que la partie droite représente un exemple de cellule (module).

Dans le NAS-BENCH-101, une architecture possible est représentée par la matrice d'adjacence du module, et une liste contenant les opérations à chaque sommet. L'exemple illustré dans la figure 2 peut être représenté à l'aide de la matrice d'adjacence présentée à la figure 3 et de la liste d'opérations suivante : [INPUT, CONV1X1, CONV3X3, CONV3X3, CONV3X3, MAXPOOL3X3, OUTPUT].

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

FIGURE 3 – matrice d'adjacence pour NAS-BENCH-101

Afin de représenter notre solution, nous choisissons de conserver la matrice d'adjacence, et d'encoder l'opération de la chaîne de caractères en nombres réels (entre 0 et 1), afin de rendre possible le déplacement de la luciole. L'encodage se fait comme suit :

- MAXPOOL3X3 = 0
- CONV1X1 = 0.33
- CONV3X3 = 0.66

Les opérateurs de la métaheuristique sont implémentés comme suit :

- Croisement : après la sélection de deux solutions (parents), une nouvelle solution est générée par :
  - Séparation horizontale des matrices des parents en deux parties, puis jointure de la partie supérieure du premier parent avec la partie inférieure du second, afin de créer la nouvelle matrice d'adjacence.
  - Séparation verticale des listes d'opérations codées des parents (aux sommets) en deux parties, puis jointure de la partie gauche du premier parent avec la partie droite du second parent, pour créer la nouvelle liste d'opérations codées.

— Mutation : sélectionner aléatoirement un sommet, puis générer un nouveau réel pour son opération (nombre aléatoire entre 0 et 1).
— Déplacement d'une luciole : le déplacement est effectué en calculant Eq.(2) par élément (pour chaque élément de la matrice et chaque élément de la liste des opérations codées). Après chaque déplacement, les valeurs inférieures à 0 sont mises à 0 et celles qui sont supérieures à 1 sont mises à 1.

Afin de rechercher une nouvelle architecture (donnée par les opérateurs de la métaheuristique) dans le NAS-BENCH-101, les opérateurs de la valeur des sommets (x) sont décodés comme suit :

$$operateur(x) = \begin{cases} MAXPOOL3X3, & \text{si } x < 0.33 \\ CONV1X1, & \text{si } 0.33 \leq x < 0.66 \\ CONV3X3, & \text{otherwise} \end{cases}$$

4.3.2 NAS-BENCH-201

Les architectures du NAS-BENCH-201 partagent le même squelette, qui commence par une convolution 3 par 3 et une couche de normalisation par lots (*batch normalization*). Trois piles de cellules sont ensuite reliées par un bloc résiduel. Le squelette se termine par une couche de *pooling* globale utilisant la moyenne (*global average pooling*), suivie d'une couche de classification utilisant *softmax* [15]. Le NAS-BENCH-201 supporte 5 opérations qui sont codées comme suit :

— none = 0
— skip connection = 1
— 1-by-1 convolution = 2
— 3-by-3 convolution = 3
— 3-by-3 average pooling = 4

De la même manière que le NAS-BENCH-101, les cellules du NAS-BENCH-201 peuvent être représentées sous la forme d'un graphe acyclique dirigé [15], elles peuvent donc être représentées à l'aide d'une matrice d'adjacence (M), où $M[i, j]$ définit l'opération existant entre les noeuds i et j. La figure 5 illustre la matrice d'adjacence utilisée pour représenter la cellule illustrée dans la figure 4.

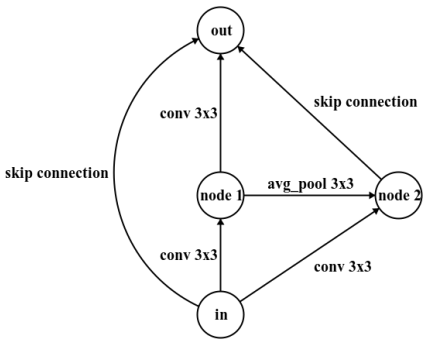


FIGURE 4 – Exemple d'un module (cellule) dans NAS-BENCH-201

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 3 & 4 & 0 & 0 \\ 1 & 3 & 1 & 0 \end{bmatrix}$$

FIGURE 5 – matrice d'adjacence pour NAS-BENCH-201

Les opérateurs de la métaheuristique sont implémentés comme suit :

— Croisement : effectué de la même manière que le croisement des matrices d'adjacence NAS-BENCH-101.
— Mutation : sélectionner aléatoirement un élément de la matrice, puis générer un nouvel entier pour sa valeur (nombre aléatoire entre 0 et 4).
— Déplacement d'une luciole : se fait de la même manière que le déplacement des matrices d'adjacence NAS-BENCH-101. Après chaque déplacement, les valeurs sont d'abord arrondies à l'entier le plus proche, puis, celles inférieures à 0 sont mises à 0 et celles supérieures à 4 sont mises à 4.

5 Expérimentation

Dans cette section, nous allons présenter les résultats expérimentaux obtenus sur différents benchmarks/datasets. Tous les tests ont été réalisés sur un ordinateur portable équipé d'un CPU Intel i7-11850H, de 32 Go de RAM et d'un GPU NVIDIA RTX A3000. Notez que dans l'évaluation suivante du modèle **non-entraîné**, seuls 100 échantillons ont été utilisés pour noter les réseaux neuronaux.

5.1 Validation de la métrique

Afin de valider notre proposition (basée sur l'ICD) pour l'évaluation d'un réseau de neurones sans entraînement, nous suivons le même protocole de test que Mellor et al. dans [23].

Nous avons utilisé 2000 architectures de chacun des benchmarks : NAS-BENCH-101, NAS-BENCH-201, NAS-BENCH-NLP et NDS. Tous ces benchmarks fournissent la précision et l'erreur pour chaque architecture après son entraînement, ce qui nous permet d'évaluer et de valider notre métrique sur un grand nombre de données en peu de temps. Nous avons calculé le τ de Spearman pour vérifier s'il existe une corrélation entre le score du réseau (sans apprentissage) et sa performance finale (après entraînement), qui est la précision dans le cas de NAS-BENCH-101, NAS-BENCH-201 et NDS, et l'erreur dans le cas de NAS-BENCH-NLP. Les résultats obtenus sur les différents benchmarks/datasets sont illustrés dans le tableau 1.

D'après les τ de Spearman présentés dans le tableau 1, nous pouvons dire qu'il existe une corrélation dans tous les benchmarks/datasets testés, atteignant 0,798 et 0,796 dans NAS-BENCH-201 pour CIFAR-10 et CIFAR-100, ce qui signifie qu'il existe une forte corrélation entre notre score avant l'apprentissage et la précision finale après l'apprentissage du réseau. La valeur p est égale à 0, signifiant que la probabilité que la valeur de τ soit due au hasard est nulle. La

Benchmark	Dataset	Spearman's τ	p-value
NAS-BENCH-101	CIFAR-10	0.499497	2.212611e-129
NAS-BENCH-201	CIFAR-10	0.798250	0.0
NAS-BENCH-201	CIFAR-100	0.796226	0.0
NDS (DARTS)	CIFAR-10	0.624632	7.627274e-217
NDS (Amoeba)	CIFAR-10	0.256211	2.408139e-31
NDS (ENAS)	CIFAR-10	0.501247	1.031932e-127
NDS (NASNET)	CIFAR-10	0.387336	1.369921e-72
NDS (PNAS)	CIFAR-10	0.490744	1.044203e-121
NAS-BENCH-NLP	Penn TreeBank	-0.420435	4.588134e-73

TABLE 1 – Valeur du τ de Spearman et valeur p du score ICD sur différents benchmarks/dataset.

plus petite valeur de τ était de 0,256 obtenue sur le benchmark NDS (Amoeba) en utilisant le jeu de données CIFAR-10, signifiant que la corrélation est faible.
Sur le NAS-BENCH-NLP, nous pouvons remarquer que la valeur du τ de Spearman est négative (-0,42) : il existe une corrélation entre le score avant l'entraînement et le résultat final du réseau. Dans ce cas, la corrélation est négative car le réseau est évalué en fonction de l'erreur, donc plus l'erreur est faible, meilleur est le réseau.
La figure 6 illustre les corrélations entre les scores de l'ICD (avant apprentissage) et la précision finale du test (obtenue après la apprentissage) à l'aide d'un nuage de points. Nous pouvons remarquer que dans la plupart des cas, le graphique obtenu a la forme d'une ligne avec une pente positive, montrant que plus le score avant l'apprentissage est élevé, plus la précision du test final est élevée. Cela nous permet de dire que la métrique que nous proposons est valide et peut être utilisée pour évaluer un réseau de neurones sans apprentissage, que le réseau soit un CNN ou un RNN.

5.2 Comparaison avec Mellor et al. [23]

Dans cette partie, nous comparons les résultats obtenus à l'aide de notre métrique avec la proposition de Mellor et al [23]. La comparaison se fait sur les scores τ de Kendall calculés sur les différents benchmarks NAS. Le tableau 2 montre les résultats obtenus.

Benchmark	Dataset	ICD	Mellor et al.
NAS-BENCH-101	CIFAR-10	0.353	0.285
NAS-BENCH-201	CIFAR-10	0.595	0.574
NAS-BENCH-201	CIFAR-100	0.594	0.611
NDS (DARTS)	CIFAR-10	0.457	0.467
NDS (Amoeba)	CIFAR-10	0.185	0.223
NDS (ENAS)	CIFAR-10	0.362	0.365
NDS (NASNET)	CIFAR-10	0.271	0.304
NDS (PNAS)	CIFAR-10	0.352	0.382

TABLE 2 – τ de Kendall pour l'ICD (notre proposition) ainsi que la proposition de Mellor et al. sur différents benchmarks/datasets

Nous pouvons remarquer que les τ de Kendall obtenus par la proposition de Mellor et al. et ceux obtenus par notre score ICD sont similaires de manière générale, avec quelques meilleurs résultats pour la proposition de Mellor dans certains cas à l'exemple du NDS (Amoeba) avec CIFAR-10, et de meilleurs résultats pour notre proposition dans d'autres cas à l'exemple du NAS-BENCH-101 avec CIFAR-10, mais aucune différence significative n'a été observée.
D'après les différents résultats, nous pouvons dire que la métrique proposée par Mellor et al. et notre proposition ICD sont équivalentes pour évaluer les réseaux sur une tâche de classification. Cependant, l'utilisation de toutes les unités du réseau rend notre méthode plus générique, puisqu'elle supporte les modèles RNNs (valable sur le benchmark NAS-BENCH-NLP).

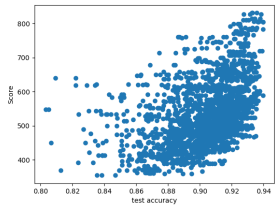
5.3 Comparaison avec d'autres méthodes sans entraînement

Dans cette partie, nous comparons notre proposition : la métrique proposée combinée à l'algorithme FireFly amélioré (IFA), aux méthodes sans entraînement les plus récentes, ainsi qu'à l'algorithme FireFly de base.
Toutes les expériences ont été réalisées en utilisant les paramètres suivants :

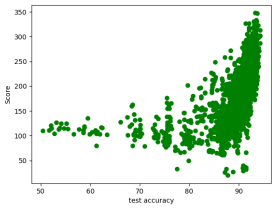
- Critères d'arrêt = 100 générations
- Taille de la population = 20
- Max chances = 5 (pour IFA uniquement)
- β_0 = 0.95
- γ = 0.15
- α = 0.5

5.3.1 NAS-BENCH-101

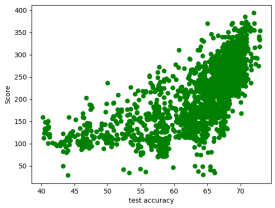
Nous avons comparé nos résultats obtenus (pour 10 exécutions) des FA, GA et IFA sur le benchmark NAS-BENCH-101, utilisant le jeu de données CIFAR-10, avec les méthodes NASWOT et AREA proposées par Mellor et al. [23]. NASWOT consiste à choisir le meilleur réseau parmi les N possibilités générées aléatoirement tandis que REA est la méthode proposée par Rere et al. [22]. Le temps de recherche moyen, la précision moyenne du test et l'écart-type sont résumés dans le tableau 3.
Tout d'abord, nous pouvons remarquer que l'IFA est plus performant que la version basique de l'algorithme FireFly, en donnant une meilleure précision de test avec un plus petit écart-type (plus stable). Ensuite, nous pouvons également remarquer que notre proposition surpasse les autres



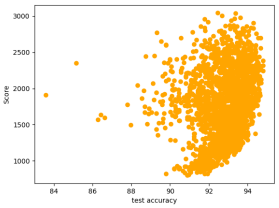
(a) NAS-BENCH-101 (CIFAR-10)



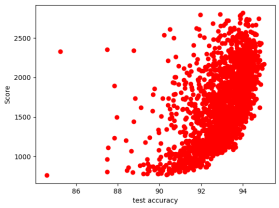
(b) NAS-BENCH-201 (CIFAR-10)



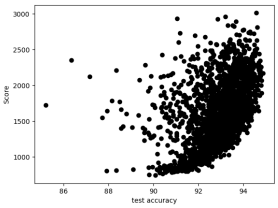
(c) NAS-BENCH-201 (CIFAR-100)



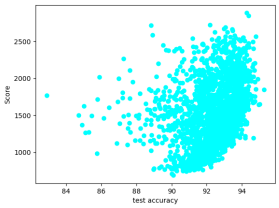
(d) NDS-Amoeba (CIFAR-10)



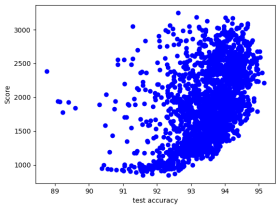
(e) NDS-DARTS (CIFAR-10)



(f) NDS-ENAS (CIFAR-10)



(g) NDS-NASNET (CIFAR-10)



(h) NDS-PNAS (CIFAR-10)

FIGURE 6 – Graphiques de notre score ICD pour les architectures **non-entraînées** par rapport à la précision des tests lorsqu'elles sont entraînées pour le NAS-Bench-201, NAS-Bench-101 et le NDS . Les entrées lors du calcul du score pour chaque graphique proviennent de CIFAR-10, sauf pour (c) qui utilise CIFAR-100.

Méthode	Recherche (s)	Précision de test
NASWOT (N=100)	23	91.77 ± 0.05
REA	12000	93.87 ± 0.22
AREA	12000	93.91 ± 0.29
FA (our)	3260	93.00 ± 1.44
IFA (our)	3596	$\mathbf{94.03 \pm 0.12}$

TABLE 3 – Comparaison sur le NAS-BENCH-101 (CIFAR-10) de notre proposition et des méthodes de l'état de l'art. Performances présentées en termes de précision avec une moyenne±std

méthodes, en donnant 0,12 % de plus de précision que la méthode AREA et plus de stabilité (0,12 pour IFA contre 0,22 pour AREA). Tout cela en étant plus de 3 fois plus rapide.

5.3.2 NAS-BENCH-201

Nous avons comparé les résultats obtenus (pour 10 exécutions) de FA, GA et IFA sur le benchmark NAS-BENCH-201, en utilisant les jeux de données CIFAR-10, CIFAR-100, ImageNet16-120, avec les méthodes NASWOT et EPE-NAS. EPE-NAS a été introduit par Lopes et al. [24] pour être combiné à leur métrique, leur algorithme de recherche est similaire à NASWOT (basé sur des sélections aléatoires). Comme Lopes et al. [24], nous exécutons notre proposition sur le jeu de données CIFAR-10, les architectures obtenues sont ensuite entraînées sur le CIFAR-100 et ImageNet16-120. La précision moyenne du test et l'écart-type sont résumés dans le tableau 4.

D'après les résultats obtenus, nous constatons que l'IFA que nous proposons surpasse les méthodes sans apprentissage de l'état de l'art sur les trois jeux de données du NAS-BENCH-201, avec une meilleure moyenne de précision pour chaque jeu de données.

Nous pouvons également remarquer que la version de base de l'algorithme FireFly (FA), qui utilise notre métrique ICD, surpasse les méthodes sans apprentissage de l'état de l'art sur les jeux de données CIFAR-10 et ImageNet16-120, et obtient la même précision moyenne que EPE-NAS proposée par Lopes et al. [24] sur le jeu de données CIFAR-100, mais le FA réussi à produire une valeur std plus petite, signifiant qu'il est plus stable que EPE-NAS.

En comparant FA et IFA, nous pouvons remarquer que l'IFA obtient de meilleurs résultats avec une précision supérieure et un std plus petit. Comme prévu, l'utilisation des opérations génétiques rend notre méthode plus robuste au problème des optimums locaux.

Method	CIFAR-10	CIFAR-100	ImageNet16-120
NAS-WOT (N=10)	92.47 ± 0.04	69.20 ± 1.05	42.20 ± 1.37
EPE-NAS (N=10)	92.63 ± 0.32	$\mathbf{70.10 \pm 1.71}$	41.92 ± 4.25
NAS-WOT (N=100)	91.41 ± 2.24	67.18 ± 4.14	41.42 ± 1.53
EPE-NAS (N=100)	91.59 ± 0.87	67.19 ± 3.82	38.80 ± 5.41
NAS-WOT (N=500)	91.71 ± 1.37	67.54 ± 2.23	39.84 ± 3.68
EPE-NAS (N=500)	92.27 ± 1.75	69.33 ± 0.66	42.05 ± 3.09
NAS-WOT (N=1000)	91.20 ± 2.04	68.95 ± 0.72	38.08 ± 1.58
EPE-NAS (N=1000)	91.31 ± 1.69	69.58 ± 0.83	41.84 ± 2.06
FA (our)	$\mathbf{92.90 \pm 1.07}$	$\mathbf{70.10 \pm 1.56}$	$\mathbf{43.38 \pm 1.80}$
IFA (our)	$\mathbf{93.58 \pm 0.15}$	$\mathbf{70.27 \pm 0.75}$	$\mathbf{44.53 \pm 1.54}$

TABLE 4 – Comparaison sur le NAS-BENCH-201 entre notre proposition et les méthodes de l'état de l'art. Performances indiquées en termes de précision avec moyenne±std

Benchmark(Dataset)	Méthode	score	notre score
NAS-BENCH-101(CIFAR-10)	GA-NAS [27]	94.23	94.03
NAS-BENCH-201(CIFAR-10)	β-DARTS [26]	94.36	93.58
NAS-BENCH-201(CIFAR-100)	β-DARTS [26]	73.51	70.27
NAS-BENCH-201(ImageNet)	β-DARTS-RS [26]	46.71	44.53

TABLE 5 – Comparaison sur différents benchmarks/datasets entre notre proposition et les méthodes NAS les plus performantes de l'état de l'art en fonction de la précision des tests.

5.4 Comparaison avec les méthodes NAS les plus performantes de l'état de l'art

Dans ce qui suit, nous allons comparer notre méthode proposée avec les méthodes NAS les plus performantes de l'état de l'art (basées sur l'apprentissage), en fonction de la précision des tests.

Le tableau 5 comprend la comparaison avec GA-NAS proposé par Rezaei el al. [27] sur le NAS-BENCH-101, et également la comparaison avec β-DARTS et β-DARTS-RS proposés par Peng et al. [26]. Nous remarquons que, même si la méthode sans entraînement que nous proposons n'atteint pas les mêmes performances que celles utilisant l'entraînement, l'écart entre elles est faible.

6 Conclusion

Les algorithmes NAS sont capables d'automatiser la découverte d'architectures efficaces dans un espace de recherche. Trouver le modèle le plus intéressant peut prendre beaucoup de temps car les algorithmes NAS doivent évaluer les réseaux en fonction de leurs performances pour choisir le plus adapté.

Afin d'éviter l'entraînement des modèles lors de l'exécution d'un algorithme NAS, nous avons proposé une nouvelle métrique, qui permet d'approcher la qualité d'un modèle sans aucun entraînement. Cette dernière est basée sur l'utilisation du score Intra-Cluster Distance (ICD). Les résultats obtenus sur différents benchmarks NAS (NAS-BENCH-101, NAS-BENCH-201, NDS et NAS-BENCH-NLP) montrent que notre métrique est valable pour évaluer les CNN et les RNN.

Nous avons proposé d'utiliser un algorithme Firefly amélioré (IFA) comme technique de recherche pour trouver la meilleure architecture pour un ensemble spécifique de données. Cet algorithme utilise les opérateurs de l'algorithme génétique ce qui lui permet d'être plus robuste que l'algorithme FireFly de base face au problème des optimums locaux. Les résultats expérimentaux sur les benchmarks NAS-BENCH-101 et NAS-BENCH-201 montrent que notre IFA, combiné à la métrique proposée, surpasse les méthodes sans entraînement de l'état de l'art ainsi que la version basique de l'algorithme FireFly.

Dans le cadre de travaux futurs, nous envisageons de trouver un moyen générique pour représenter les hyperparamètres d'une architecture de réseau de neurones (nombre de couches, nombre d'unités, type de cellules, etc.) en y incluant d'autres types de couches en plus des CNNs, ce qui nous permettrait d'exploiter notre proposition dans un problème d'apprentissage autre que la classification d'images.

Remerciements

Ce travail a été réalisé dans le cadre du projet franco-canadien DOMAID financé par l'Agence Nationale de la Recherche (ANR-20-CE26-0014-01) et le FRQSC.

Références

[1] N. Mokhtari, A. Nédélec, M. Gilles and P. De Loor, "Improving Neural Architecture Search by Mixing a FireFly algorithm with a Training Free Evaluation," 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 2022, pp. 1-8, doi : 10.1109/IJCNN55064.2022.9892861.

[2] Cao, X., Yao, J., Xu, Z., and Meng, D. : Hyperspec-tral image classification with convolutional neural net-work

and active learning.IEEE Transactions on Geo-science and Remote Sensing, 58(7) :4604–4616, 2020.

[3] Martins, V., Kaleita, A., Gelder, B., Silveira, H., and Abe,C. : Exploring multiscale object-based convolutional neural network (multi-ocnn) for remote sensing image classification at high spatial resolution.ISPRS Journal of Photogrammetry and Remote Sensing,168 :56–73,2020.

[4] Mustaqeem and Kwon, S. : Mlt-dnet : Speech emotion recognition using 1d dilated cnn based on multi-learning trick approach.Expert Systems with Applications, 167, 2020.

[5] Zhang, N., Wang, J., Wei, W., Qu, X., Cheng, N., and Xiao,J. : Cacnet : Cube attentional cnn for automaticspeech recognition. In2021 International Joint Conference on Neural Networks (IJCNN), pages 1–7, 2021.

[6] Elsken, T., Metzen, J. H., and Hutter, F. Neural architecture search : A survey. Journal of Machine Learning Research, 20(55) :1–21, 2019.

[7] Wistuba, M., Rawat, A., and Pedapati, T. A survey on neural architecture search. arXiv preprint arXiv :1905.01392, 2019.

[8] Zoph, B. and Le, Q. V. Neural architecture search with reinforcement learning. In International Conference on Learning Representations, 2017.

[9] Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[10] Pham, H., Guan, M. Y., Zoph, B., Le, Q. V., and Dean, J. Efficient neural architecture search via parameter sharing. In International Conference on Machine Learning, 2018.

[11] Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., and Le, Q. V. MnasNet : Platform-aware neural architecture search for mobile. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.

[12] Liu, H., Simonyan, K., and Yang, Y. DARTS : Differentiable architecture search. In International Conference on Learning Representations, 2019.

[13] Real, E., Aggarwal, A., Huang, Y., and Le, Q. V. Regularized evolution for image classifier architecture search. In Proceedings of the AAAI Conference on Artificial Intelligence, 2019.

[14] Ying, C., Klein, A., Christiansen, E., Real, E., Murphy, K. &amp ; Hutter, F. : NAS-Bench-101 : Towards Reproducible Neural Architecture Search. Proceedings of the 36th International Conference on Machine Learning, in Proceedings of Machine Learning Research, 2019.

[15] Dong, X. and Yang, Y., "NAS-Bench-201 : Extending the Scope of Reproducible Neural Architecture Search", arXiv e-prints, 2020.

[16] Radosavovic, Ilija & Johnson, Justin & Xie, Saining & Lo, Wan-Yen & Dollár, Piotr : On Network Design Spaces for Visual Recognition, 2019.

[17] Klyuchnikov, Nikita & Trofimov, Ilya & Artemova, Ekaterina & Salnikov, Mikhail & Fedorov, Maxim & Burnaev, Evgeny. NAS-Bench-NLP : Neural Architecture Search Benchmark for Natural Language Processing, 2020.

[18] I. Strumberger, E. Tuba, N. Bacanin,M. Zivkovic, M. Beko, and M. Tuba. Designing convolutional neural network architecture by the firefly algorithm. In2019 International Young Engineers Forum (YEF-ECE), pages 59–65, 2019.

[19] Vina Ayumi, L.M. Rere, Mohamad IvanFanany, and Aniati Arymurthy. Optimization of convolutional neural network using microcanonical annealing algorithm. 102016.

[20] Yanan Sun, Bing Xue, Mengjie Zhang, and Gary Yen. Automatically designing cnn architectures using Genetic Algorithm for image classification, 08 2018.

[21] Adenilson Carvalho, Fernando Ramos, and Antonio Chaves. Metaheuristics for the feed forward artificial neural network (ann)architecture optimization problem.Neural Computing and Applications, 20, 10 2010.

[22] L.M. Rere, Mohamad Ivan Fanany,and Aniati Arymurthy.Metaheuristic algorithms for convolution neuralnetwork.Computational Intelligence and Neuroscience, 2016, 05 2016.

[23] Mellor, J., Turner, J., Storkey, A. Crowley, E.J. : Neural Architecture Search without Training. Proceedings of the 38th International Conference on Machine Learning, in Proceedings of Machine Learning Research,2021

[24] Lopes, Vasco et al. "EPE-NAS : Efficient Performance Estimation Without Training for Neural Architecture Search." ICANN (2021).

[25] Xin-She Yang. 2008. Nature-Inspired Metaheuristic Algorithms. Luniver Press.

[26] Ye, Peng and Li, Baopu and Li, Yikang and Chen, Tao and Fan, Jiayuan and Ouyang, Wanli. β-DARTS : Beta-Decay Regularization for Differentiable Architecture Search,2022

[27] Rezaei, Seyed Saeed Changiz and Han, Fred X and Niu, Di and Salameh, Mohammad and Mills, Keith and Lian, Shuo and Lu, Wei and Jui, Shangling. Generative Adversarial Neural Architecture Search,arXiv,2021

Etude de l'apport de l'Intelligence Artificielle pour l'innovation de produit

F. HAMDANI[1], D. MONTICOLO[1,] V. BOLY[1]

[1] laboratoire ERPI, Université de Lorraine, F-54000 Nancy, France

Résumé

L'analyse de besoins des clients est considérée comme un défi majeur pour les entreprises de tous les secteurs. C'est une procédure critique et une condition essentielle pour le développement réussi de nouveaux produits (NPD). L'adoption de technologies et de méthodes d'intelligence artificielle (IA) pourrait améliorer le processus de l'analyse de besoins et assister les entreprises à renforcer leur position concurrentielle dans un environnement commercial en constante évolution et dynamique. Cependant, malgré sa popularité, de nombreuses organisations sont réticentes à adopter l'IA dans leur processus en raison de l'ambiguïté et l'incertitude quant aux avantages potentiels qu'elle peut apporter.

Dans ce contexte, cet article examine le rôle de l'intelligence artificielle et l'efficacité de l'IA sur le processus d'analyse de besoins en présentant un aperçu des travaux de recherche menés dans ce domaine. Cette étude contribue à la théorisation des capacités de l'IA pour l'analyse de besoins et fournit des orientations pour les futures recherches dans ce domaine.

Mots-clés

Intelligence Artificielle, Analyse des besoins Clients, Processus d'innovation, Apprentissage Automatique, Apprentissage Profond, Traitement du Langage Naturel, Analyse de Sentiments, Processus d'Innovation.

Abstract

Customer needs analysis is considered a major challenge for businesses across all sectors. It is a critical procedure and an essential condition for developing new products (NPD) successfully. The adoption of artificial intelligence (AI) technologies and methods could improve the needs analysis process and assist companies in strengthening their competitive position in a constantly evolving and dynamic business environment. However, despite its popularity, many organizations are reluctant to adopt AI in their processes due to ambiguity and uncertainty about the potential benefits it can bring.

In this context, this article examines the role of artificial intelligence and the effectiveness of AI in the needs analysis process by providing an overview of research conducted in this area. This study contributes to the theorization of AI capabilities for needs analysis and provides guidance for future research in this field.

Keywords

Artificial Intelligence, Customer Needs, Customer Management, Machine Learning, Deep Learning, Natural Language Processing, Sentiment Analysis, Innovation Process.

1 Introduction

L'importance de l'analyse de besoins des clients dans le développement de produits a été largement reconnue dans les milieux universitaires et industriels (Zhou et al., 2013). C'est l'un des principaux facteurs de succès du développement de produits sur le marché. L'analyse efficace des besoins des clients est le moteur du développement de nouveaux produits innovants En revanche, une mauvaise compréhension des besoins des clients et des hypothèses inexactes faites lors de du processus de l'analyse peut avoir des implications négatives significatives sur la conception et la fabrication du produit en termes de qualité, de délai et de coût.

L'analyse des besoins des clients est un processus particulièrement difficile qui implique un processus de gestion fastidieux entre les clients, les spécialistes du marketing et les concepteurs. Le développement de produits et de services avec une qualité perçue élevée et l'implication des clients dans la co-conception ne sont plus une option pour l'entreprise. Cela implique de travailler étroitement avec les clients pour les aider à développer une compréhension plus précise de leurs besoins, ce qui n'est pas toujours trivial. De plus, il s'agit d'un processus avec des difficultés inhérentes de sémantique et de terminologie. Cela conduit à l'adoption de méthodes et de technologies qui peuvent rendre l'échange fluide et qui peuvent gérer le manque de cohérence et de cohésion dans la communication et l'échange d'informations. Au cours de la dernière décennie, des approches basées sur l'IA ont été largement utilisées dans divers domaines pour résoudre des problèmes complexes. De même, l'adoption de l'IA a démontré son potentiel dans la conception personnalisée de produits et la gestion des clients. L'IA avancée offre un grand potentiel pour améliorer le processus de gestion des besoins. Dans ce contexte, ce travail présente un aperçu sur les travaux et des avancées sur l'adoption de l'IA pour analyser les besoins des clients, afin de comprendre les multiples dimensions des capacités clés de l'IA, ses défis, ses limites et ses orientations pour dans la gestion des besoins des clients.

2 Les challenges liés à l'analyse de besoins

L'analyse des besoins est une étape cruciale pour le développement des produits. Cependant, cette étape peut s'avérer difficile et complexe, en particulier lorsque les besoins des utilisateurs sont mal compris ou mal articulés. Dans cette

section, nous présentons les défis courants liés à l'analyse des besoins,

Les défis linguistiques : les besoins clients sont exprimés en langage naturel et en termes linguistiques, ce qui entraîne des différences linguistiques en termes de sémantique et de terminologie. Les clients, les équipes marketing et les designers expriment leurs besoins dans des contextes, des sémantiques et des terminologies différents, les clients manquent souvent de termes et de terminologie spécifiques pour décrire la spécification technique, ce qui crée un écart entre "ce qui est conçu" et "ce qui est désiré".

Les besoins latents : les besoins des clients sont composés de besoins explicites et latents. Les besoins latents sont des besoins que les clients peuvent ne pas être conscients ou ne pas être en mesure d'exprimer consciemment. Toutefois, si ces besoins sont satisfaits, les clients peuvent être ravis, mais s'ils ne le sont pas, ils peuvent être déçus. Ils sont difficiles à identifier, ne sont pas directement mesurables et ne sont pas toujours exprimés par le langage naturel. Ils peuvent être exprimés par les émotions, les comportements, et les données physiologiques. L'identification des besoins latents peut nécessiter l'utilisation de réseaux de capteurs spécifiques et sophistiqués, ce qui rend l'identification des besoins latents difficile.

Les changements rapides et dynamiques des besoins clients : les besoins clients ne sont pas un concept statique, mais plutôt un concept très rapide, dynamique et temporellement changeant. Les besoins clients changent rapidement avec l'avancement des produits/services concurrents. Les données obtenues sur une période spécifique deviennent rapidement obsolètes et peuvent ne pas suivre la même distribution à un moment ultérieur.

Les défis de l'optimisation des ressources : l'identification des besoins clients est un processus fastidieux qui demande souvent une multitude de ressources d'experts du domaine. Les retards dans l'identification des besoins clients peuvent être longs, avec un effort constant de l'équipe de développement de produits. En effet, une grande partie des retards dans l'identification des besoins clients est due au temps que l'équipe consacre à l'observation et à l'analyse des transcriptions et à la synthèse des besoins clients. L'effort requis entraîne souvent des retards sur le marché et limite l'échelle des données et des informations à prendre en compte. Cependant, dans l'environnement concurrentiel actuel, l'optimisation du temps et des coûts de développement de produits est décisive.

La complexité du processus d'innovation : le processus d'innovation implique des décisions critiques avec une dimension d'incertitude (Rejeb et al., 2011). Elle implique également des équipes avec des caractéristiques individuelles et des points de vue différents, ce qui en fait un processus difficile qui nécessite un soutien proactif pour le processus de prise de décision humain. De plus, la complexité liée à la compréhension des besoins dans la phase amont en raison de déclarations de clients imprécises, vagues et peu claires peut affecter négativement la créativité de la conception de l'ingénierie.

Les défis numériques : les opportunités d'expansion numérique sont depuis longtemps une priorité claire, même avant la pandémie de COVID-19, bien qu'elles soient devenues plus apparentes lorsque toutes les interactions quotidiennes ont été déplacées en ligne. Le nouvel état normal est numérique, agile, durable, écologique, coopératif et personnalisé. Le marché complexe et changeant force les entreprises à suivre le progrès technologique (Lindemann et al., 2020). La transformation numérique et sa capacité à générer des informations perspicaces et prévisibles à partir des données ont émergé comme un principe de légitimité numérique pour toutes les entreprises qui souhaitent rester compétitives et prospères. Cette évolution conduit à un fort besoin d'aspects technologiques et numériques ainsi que de nouvelles méthodes modernes, authentiques et dynamiques pour gérer les besoins des clients et comprendre leurs besoins en temps réel.

Les défis de la gestion des données : les défis de gestion des données liés à la croissance rapide du commerce électronique et des plateformes de médias sociaux, où de grandes quantités d'informations de groupes divers de clients sur une multitude de produits / services sont disponibles en ligne. Bien que ces données soient une source précieuse pour identifier les besoins et les préférences des clients, les défis incluent le traitement de grandes quantités de données structurées et non structurées provenant de diverses sources, la sélection d'informations à valeur ajoutée, la gestion de spam et de faux besoins, la difficulté à extraire des informations significatives de la quantité de données et l'analyse en temps réel.

Plusieurs méthodes conventionnelles ont été largement utilisées et ont fait leurs preuves pour relever les défis liés à l'analyse des besoins. Cependant, elles restent insuffisantes pour relever les défis actuels. Elles reposent fortement sur les capacités et la volonté des clients à exprimer explicitement leurs préférences et leurs besoins. La qualité des acquisitions dépend fortement de la formulation des questionnaires d'enquête présentés aux clients actuels/potentiels (Tucker & Kim, 2011). Aussi, ces méthodes dépendent fortement des experts, ce qui peut être très coûteux en temps et en argent. Elles sont souvent opérées de manière subjective, et la qualité des acquisitions dépend de la compétence. Les facteurs subjectifs impliqués peuvent compliquer le problème et même égarer la sollicitation. Par exemple, la méthode de conception empathique est chronophage nécessite une compétence et une expertise considérables, elle repose également sur une interprétation subjective et constitue une approche basée sur des connaissances d'experts. L'effort manuel requis entraîne souvent des retards sur le marché et limite l'échelle des données et des informations à traiter. La taille de l'ensemble de données d'enquête est souvent assez limitée en raison des coûts en temps et en argent. De plus, les approches les plus courantes sont qualitatives, ce qui peut conduire à des résultats trompeurs et insuffisants. Par exemple, les catégories de Kano sont généralement qualitatives, et elles ne peuvent pas mesurer précisément le degré de satisfaction des clients (Chen et al., 2013). Pour combler l'écart sémantique entre les clients et les concepteurs, les méthodes traditionnelles reposent fortement sur l'action humaine et sont toutes exécutées manuellement, ce

qui peut être chronophage et nécessiter une connaissance étendue du domaine (Y. Wang, Li, & Mo, 2021). Les méthodes traditionnelles ne sont ni efficaces ni efficientes pour traiter de grandes quantités de données générées par les clients.

Dans un autre contexte, la technologie de l'IA se propage rapidement dans tous les contextes pour résoudre des problèmes et des défis complexes, et de plus en plus de chercheurs et d'experts de l'industrie ont proposé des modèles et des méthodes d'IA pour résoudre les problèmes de la vie humaine et améliorer l'efficacité du travail. L'objectif de cet article est de présenter les capacités de l'IA, de son utilité, de ses défis pour supporter le processus de l'analyse de besoins.
.

3 L'Intelligence Artificielle peut-elle améliorer le processus d'analyse des besoins des clients ?

L'intelligence artificielle est une technologie émergente qui offre de nouvelles possibilités pour améliorer le processus d'analyse de besoins en automatisant certaines tâches et en fournissant des insights précis en temps réel, L'annexe 1 présente une liste non exhaustive d'études utilisant l'IA pour l'analyse des besoins. Dans cette section, nous allons examiner le potentiel de l'IA pour améliorer le processus d'analyse des besoins des clients, en explorant les avantages, les défis et les limites de cette approche.

Défis linguistiques et terminologiques : pour l'asymétrie cognitive et le gap sémantique entre les domaines des clients et des ingénieurs/concepteurs. Contrairement aux méthodes traditionnelles qui reposent fortement sur l'action humaine et qui sont effectuées manuellement pour combler le gap sémantique, ce qui est chronophage et nécessite des connaissances étendues dans le domaine, l'IA peut améliorer efficacement la cartographie des besoins en spécifications de conception de produit, faciliter l'interprétation et la traduction en spécifications de produit/service, et contribuer considérablement à raccourcir les délais de développement de produits. Des solutions basées sur les réseaux de neurones conventionnels pour cartographier automatiquement les spécifications de produit (Y. Wang et al., 2022) (Y. Wang & Li, 2021a) (Y. Wang & Li, 2021b) (Y. Wang, Zhao, et al., 2020) (Y. Wang, Li, et al., 2020), et cartographier les paramètres de conception (Y. Wang, Mo, & Tseng, 2018). Un apprentissage automatique supervisé (machine à vecteurs de support) pour combler automatiquement le gap sémantique entre les besoins perceptuels et les spécifications de produit (Y. Wang & Zhang, 2017).

Cependant, l'espace de solution pour les spécifications de produit n'est pas en expansion et comprend également des solutions fixes, ce qui peut entraver le processus d'innovation. De plus, le manque d'approches pratiques pour de nombreuses questions concernant la commodité dans des contextes réels et son accessibilité ressortent.

Les changements rapides et dynamiques du besoin : la dimension temporelle de la compréhension des besoins constitue un paramètre décisif pour les entreprises ayant un taux élevé d'innovation afin d'identifier et de développer des nouveaux produits (Arco et al., 2019). En effet, la rapidité pour atteindre ses objectifs dans le temps requis pour atteindre les changements du marché est crucial, de même qu'une compréhension à grande échelle des CN est une exigence clé pour la conception innovante. Les entreprises adoptent dans leur posture compétitive la manière de collecter, stocker et analyser les informations clients, pour acquérir rapidement et efficacement des connaissances exploitables pour avoir une longueur d'avance. L'analyse des données client dès le stade initial de la conception de produits/services s'est avérée efficace dans le succès global du développement de nouveaux produits/services (Zhou et al., 2020) (Zhou et al., 2015). Un nombre limité d'études (Kilroy et al., 2022; Luo & Xu, 2021 ; Jin et al., 2016) répondent aux défis des besoins des clients en évolution rapide et dynamique. Des approches plus robustes, incluant des informations clients dynamiques, sont nécessaires pour une meilleure élucidation des besoins des clients, en particulier avec le développement de la technologie de l'Internet des objets et de la conception de produits basée sur les données, qui peuvent soutenir une réponse rapide et aider à suivre le rythme des changements rapides des besoins des clients

Les besoins latents : la capacité offerte par l'apprentissage automatique à évaluer des détails massifs et à découvrir les besoins tacites et spontanés des clients a été exploitée dans certaines des études. Par exemple, l'étude (Zhou et al., 2015) formule l'élicitation des besoins latents en se basant sur un modèle à deux niveaux utilisant le raisonnement par cas et l'analyse des sentiments. Cependant, l'évaluation des besoins latents dans cette étude était limitée aux études de cas exceptionnelles. Une analyse de sentiment basée sur un paradigme utilisant l'analyse des sentiments des avis de produits en ligne a été proposée par (Zhou & Jiao, 2016), pour répondre au défi de l'identification explicite et latente des besoins des clients. Le paradigme combine l'apprentissage automatique non supervisé pour extraire les attributs des produits, l'apprentissage automatique supervisé avec des lexiques affectifs pour la prédiction des sentiments, et une mesure de similarité des termes, pour faciliter le processus d'dentification des besoins latents et explicites des clients. Dans le contexte de la santé, un modèle d'analyse de sentiment non supervisé et une méthode basée sur l'apprentissage non supervisé ont été utilisés pour identifier les maladies infectieuses latentes du monde réel en exploitant les données des réseaux sociaux (Lim et al., 2017). Un algorithme d'extraction de fonctionnalités d'extraction de données pour identifier les fonctionnalités potentielles des utilisateurs avancés et des produits latents à partir des données sociales (Tuarob & Tucker, 2015). Au niveaux des approches basées sur l'IA l'identificatio automatique des besoins explicites est une zone bien explorée avec des résultats intéressants et importants. Néanmoins, une quantité limitée de recherche permet l'identification des besoins latents.

Analyse émotionnelle/comportementale : l'utilisation de l'analyse de sentiment et de l'apprentissage automatique a été employée dans de nombreuses études pour analyser les émotions, les attentes et les sentiments des clients à partir de

données en ligne. Un système de gestion de l'indice de satisfaction des produits basé sur l'analyse de sentiment et la modélisation de sujets a été proposé pour améliorer la qualité des produits et répondre aux attentes des clients (Sun et al., 2021). Une approche de fouille de texte pour capturer les sujets d'intérêt des clients et leur satisfaction à partir des interactions avec les chatbots et les clients (Akhtar et al., 2019). Analyse de sentiment pour les commentaires en ligne basée sur l'apprentissage profond pour modéliser le sentiment et la prédiction des évaluations (Luo et Xu, 2021). Modélisation de sujets et analyse de sentiment pour évaluer les sentiments et les opinions qui surgissent avant et après l'introduction d'un nouveau produit à travers deux langues et quatre pays (Wedel et al., 2021). Pour comprendre et analyser le comportement des clients, de nombreuses études ont émergé ces dernières années (Zhao et al., 2021 ; (Chan et Fan, 2020 ; Wu et al., 2019 ; Roh et al., 2019 ; Ameen et al., 2021).

L'un des principaux inconvénients de l'analyse émotionnelle et comportementale est le fait que les approches utilisent des techniques sentimentales ou d'exploration d'opinions pour classer les avis des clients en catégories positives ou négatives sans contexte ni raisons liées au sentiment défini. Cela peut entraîner des résultats incohérents ou trompeurs, car les opinions des clients peuvent être nuancées et influencées par des facteurs externes tels que leur expérience passée avec le produit ou le service, leur état émotionnel actuel, ou même leur environnement physique et social. Il est donc important de prendre en compte ces facteurs contextuels et de comprendre les raisons sous-jacentes derrière les émotions et les comportements des clients, afin de fournir des insights précis et utiles pour améliorer l'expérience client et l'efficacité des stratégies commerciales.

Dépendance à l'expertise et aux compétences des experts : Au niveau de l'IA les ressources utilisées restent limitées par rapport aux méthodes traditionnelles, elles nécessitent moins de compétences d'experts. Notez que la majorité des études qui mobilisent des ressources sont pour le codage et l'étiquetage des données pour les approches supervisées, et parfois pour l'intégration et la validation des résultats.

Prise de décisions proactives : les approches basées sur l'IA aide à comprendre et à imiter les comportements des concepteurs et à développer de nouveaux systèmes intégrés à l'intelligence humaine pour augmenter la conception computationnelle. Une approche technique basée sur des algorithmes prédominants en régression et en classification pour modéliser et valider l'utilité du point de vue du concepteur pour sélectionner des critiques clients représentatives (Y. Liu et al., 2013). Une plateforme cognitive basée sur des agents pour simuler et analyser les processus et performances de conception humaine pour modéliser les caractéristiques de plusieurs individus dans de petites équipes de conception a été proposée par (McComb et al., 2015). Un apprentissage inversé basé sur un algorithme d'optimisation bayésien pour imiter les démonstrations humaines de recherche de solutions, pour poursuivre l'exécution d'une recherche de solutions, même après que les participants humains ont abandonné le problème (Sexton & Ren, 2017). Une étude des capacités et comportements d'apprentissage humain pour résoudre des problèmes de conception de configuration a été menée, en utilisant une chaîne de Markov et une modélisation basée sur des agents (McComb et al., 2017a). Un processus d'extraction d'heuristiques humaines basé sur une chaîne de Markov cachée (McComb et al., 2017b). Un modèle descriptif pour comprendre les stratégies individuelles de prise de décision pour l'acquisition séquentielle d'informations, sous contraintes de coûts et de complexité de la tâche (Chaudhari & Panchal, 2019). Un cadre pour imiter la conception humaine, la génération de conception et la visualisation sans aucune information spécifique sur les opérations de conception en utilisant l'apprentissage profond (Raina et al., 2019). Une comparaison de modèles bayésiens de divers Un cadre pour imiter la conception humaine et la génération et la visualisation de la conception sans aucune information spécifique sur les opérations de conception en utilisant l'apprentissage en profondeur (Raina et al., 2019). Une comparaison de modèles bayésiens de divers heuristiques pour fournir la meilleure présentation des décisions d'acquisition d'informations d'un concepteur sous plusieurs sources d'informations et des contraintes budgétaires limitées (Chaudhari et al., 2020). Un logiciel de générateur de motifs émotionnels a été développé pour créer des motifs automatiquement en fonction des émotions et des sentiments de l'utilisateur en utilisant l'intelligence computationnelle (logique floue) (Trautmann et Piros, 2020). Une approche d'apprentissage en profondeur pour modéliser le comportement de conception et prédire les décisions de conception séquentielles en se basant sur le modèle fonction-comportement-structure et le modèle de mémoire à court terme (Rahman et al., 2021). Un cadre prédictif basé sur un réseau de neurones récurrents et la théorie des jeux non coopératifs pour modéliser les décisions futures des individus pour les processus de conception séquentielle en concurrence (Bayrak et Sha, 2021).

Cependant, bien que les approches basées sur l'IA offrent de nombreux avantages pour l'analyse des besoins, il existe également des limites qui doivent être prises en compte. En ce qui concerne la faisabilité économique, l'estimation du coût de mise en œuvre des approches basées sur l'IA par rapport aux approches traditionnelles reste vague. De plus, l'utilisation de telles approches nécessite des compétences difficiles et coûteuses à acquérir, ce qui peut être un obstacle pour les petites et moyennes entreprises.

En termes de représentativité, l'utilisation de grandes quantités de données peut ne pas refléter une grande variété d'informations pertinentes et de qualité, et l'utilisation de données générées par les clients peut ne pas refléter le véritable besoin de l'ensemble ou de différentes catégories de la société. De plus, le manque d'adaptabilité à l'analyse de données de langue mixte peut limiter l'efficacité de l'identification des besoins des clients.

Enfin, en ce qui concerne l'applicabilité, un nombre limité d'études ont fourni des outils et des méthodologies qui peuvent être facilement mis en œuvre et adoptés dans le processus de l'innovation. Il n'est pas clair non plus comment les résultats peuvent être intégrés dans le processus d'innovation et la

manière dont ces approches peuvent soutenir et aider la conception de produits est rarement discutée. Les recherches futures devront mener des études supplémentaires pour vérifier l'applicabilité et la généralisabilité des méthodes proposées dans différents domaines et dans des environnements pratiques réels.

4 Conclusion

Le processus d'analyse de besoins est d'une importance cruciale pour le succès des produits/services d'une entreprise. En effet, il est évident que si l'identification des besoins n'est pas bien faite, il y a peu ou pas de chance que des résultats satisfaisants soient obtenus et que les besoins soient satisfaits. La clé du succès des produits/services repose sur une meilleure compréhension de la voix du client et des liens plus étroits entre les préférences et les besoins des clients. La mauvaise compréhension des besoins des clients et les hypothèses inexactes faites pendant l'identification et l'analyse ont des implications négatives significatives sur la conception et la fabrication du produit en termes de qualité, de délai et de coût.

L'IA peut avoir le plus grand impact sur la gestion du processus de besoins. En particulier, les avancées de l'IA ont le potentiel d'améliorer les processus de gestion des clients et d'augmenter la connaissance des entreprises sur les préférences, les émotions et les comportements des clients. Le déploiement stratégique des technologies d'IA à différents points de contact clés avec les clients peut donc apporter des avantages significatifs aux entreprises et une possible augmentation de la satisfaction des clients. L'IA peut rationaliser et améliorer la flexibilité et la réactivité de la gestion des processus clients. Par rapport aux approches qualitatives, les techniques quantitatives basées sur l'IA et la découverte de connaissances semblent plus précises et objectives en termes d'interprétation des Besoins.

L'IA peut soutenir la gestion évolutive des besoins des clients, en termes de collecte de données des consommateurs à partir de nombreuses sources et de fourniture d'informations et de prévisions importantes. De plus, l'adoption du paradigme de l'IA pourrait améliorer les performances de gestion des processus clients dans l'environnement commercial et de marché rapide et dynamique d'aujourd'hui. Elle peut aider les entreprises à automatiser le processus d'identification des besoins, à se protéger contre les coûts de main-d'œuvre élevés et à faciliter les opérations de l'entreprise. L'IA peut être considérée comme un facteur de conduite pour que les entreprises soient plus compétitives et innovantes et les soutenir dans leur émergence de la digitalisation.

5 Annexes

Annexe 1 : approches basées sur l'IA pour l'analyse des besoins (Deep Learning « apprentissage profond », Machine Learning «apprentissage automatique» (ML), Texte Mining «fouille de textes » (TM), Sentiment Analysis (SA), Opinion Mining (OM), Data Mining (DM), Natural Language Processing «Traitement automatique des langues» (NLP), Multi-agent system «Système multi-agents » (MSA), Ontologies (Ot), Case-based reasoning «Raisonnement à partir de cas» (CBR), L'intelligence computationnelle (IC))

Référence	Téchniques	Automatisation	Scalabilité	Gestion de données	Changements rapides et dynamiques	Opinion/sentiment	Capacités d'analyse comportementale	Elicitation des besoins latents	Défis sémantiques
(Kilroy et al., 2022)	NLP, ML	v	v	v	v				
(Y. Wang et al., 2022)	NL, DL	v	v	v					v
(T. Wang, 2022)	NLP	v	v	v					v
(Y. Wang, Li, Zhang, et al., 2021)	NLP, DL	v	v	v					
(Y. Wang & Li, 2021a)	NLP, DL	v	v	v					v
(Jayashree et al., 2021)	SA, NLP	v	v	v		v			
(Y. Wang, Li, & Mo, 2021)	NLP, DL	v	v	v					v
(Malik et al., 2021)	NLP, ML	v	v	v					
(Y. Wang & Li, 2021b)	NLP, DL	v	v	v					v
(Y. Wang, Zhao, et al., 2020)	NLP, DL	v	v	v					v
(Luo & Xu, 2021)	NLP, DL	v	v	v	v	v			
(Peddireddy et al., 2021)	DL	v	v	v					

(Han & Moghaddam, 2021)	NLP, DL	v	v	v		v			
(Wedel et al., 2021)	NLP, TM, · SA	v	v	v		v			
(Bansal, 2019)	NLP, SA	v	v	v		v			
(E. J. Xu et al., 2020)	NLP, SA, DL	v	v	v		v			
(Zhou et al., 2020)	ML, TM, SA	v	v	v		v			
(Kühl et al., 2020)	NLP, ML	v	v	v					
(Y. Wang, Li, et al., 2020)	NLP, DL	v	v	v					v
(Q. Liu et al., 2020)	NLP, ML	v	v	v					
(Chiu & Tsai, 2020)	MSA	v	v	v		v			
(Sharif et al., 2019)	NLP, ML	v	v	v					
(Cantwell & Hayashi, 2019)	TM	v	v	v					
(Timoshenko & Hauser, 2019)	NLP, DL	v	v	v		v			v
(Roh et al., 2019)	NLP, ML	v	v	v					
(S. G. Kim et al., 2019)	NLP, DL	v	v	v					v
(Y. Wang et al., 2018)	NLP, DL	v	v	v		v			
(Haque et al., 2018)	NLP, SA, ML	v	v	v					
(H. Xu et al., 2017)	TM	v	v	v					
(Rangu et al., 2017)	TM	v	v	v					
(Alan et al., 2016)	DM, IC	v	v	v					
(Eckstein et al., 2016)	ML	v	v	v					
(Algur & Biradar, 2016)	NLP, SA	v	v	v		v			
(Y. Xu et al., 2016)	DL,IC	v	v	v		v			
(Zhou et al., 2015)	SA, NLP, CBR,	v	v	v				v	
(Jin et al., 2015)	NLP, ML	v	v	v					v
(Chen et al., 2013)	Ot,ML	v	v	v					
(Kutschenreiter-Praszkiewicz, 2013)	ML	v	v	v					
(K. Jae Kim, 2011)	DM	v	v	v			v		
(Wu et al., 2019)	DL	v	v	v		v		v	
(Tuarob & Tucker, 2015b)	NLP, DM	v	v	v			v		
(Rahman et al., 2021)	DL, Ot	v	v	v		v			
(Giatsoglou et al., 2017)	SA, ML	v	v	v		v			
(H. Liu et al., 2021)	NLP, DM, SA	v	v	v		v			
(L. Wang et al., 2011)	TM	v	v	v		v			
(Tuarob & Tucker, 2015a)	NLP, DM, TM	v	v	v		v			
(Jin et al., 2019)	OM	v	v	v		v			
(Jin et al., 2016)	OM	v	v	v	v	v			

6 Références

Akhtar, M., Neidhardt, J., & Werthner, H. (2019). The potential of chatbots: Analysis of chatbot conversations. Proceedings - 21st IEEE Conference on Business Informatics, CBI 2019, 1, 397–404. https://doi.org/10.1109/CBI.2019.00052

Algur, S. P., & Biradar, J. G. (2016). Rating consistency and review content based on multiple stores review spam detection. Proceedings - IEEE International Conference on Information Processing, ICIP 2015, 685–690. https://doi.org/10.1109/INFOP.2015.7489470

Ameen, N., Tarhini, A., Reppel, A., & Anand, A. (2021). Customer experiences in the age of artificial intelligence. Computers in Human Behavior, 114(August 2020), 106548. https://doi.org/10.1016/j.chb.2020.106548

Arco, M. D., Presti, L. lo, Marino, V., & Resciniti, R. (2019). Embracing AI and Big Data in customer journey mapping: From a literature review to a theoretical framework. Innovative Marketing, 15(4), 102–115. https://doi.org/10.21511/im.15(4).2019.09

Bayrak, A. E., & Sha, Z. (2021). Integrating Sequence Learning and Game Theory to Predict Design Decisions under Competition. Journal of Mechanical Design, Transactions of the ASME, 143(5). https://doi.org/10.1115/1.4048222

Cantwell, J., & Hayashi, T. (2019). A paradigm shift in technologies and innovation systems. In Paradigm Shift in Technologies and Innovation Systems. https://doi.org/10.1007/978-981-32-9350-2

Chaudhari, A. M., Bilionis, I., & Panchal, J. H. (2020). Descriptive Models of Sequential Decisions in Engineering Design: An Experimental Study. Journal of Mechanical Design, Transactions of the ASME, 142(8). https://doi.org/10.1115/1.4045605

Chaudhari, A. M., & Panchal, J. H. (2019). An experimental study of human decisions in sequential information acquisition in design: Impact of cost and task complexity. In Smart Innovation, Systems and Technologies (Vol. 134). Springer Singapore. https://doi.org/10.1007/978-981-13-5974-3_28

Chen, X., Chen, C. H., Leong, K. F., & Jiang, X. (2013). An ontology learning system for customer needs representation in product development. International Journal of Advanced Manufacturing Technology, 67(1–4), 441–453. https://doi.org/10.1007/s00170-012-4496-2

Eckstein, L., Kuehl, N., & Satzger, G. (2016). Towards Extracting Customer Needs from Incident Tickets in IT Services. Proceedings - CBI 2016: 18th IEEE Conference on Business Informatics, 1, 200–207. https://doi.org/10.1109/CBI.2016.30

Han, Y., & Moghaddam, M. (2021). Eliciting Attribute-Level User Needs from Online Reviews with Deep Language Models and Information Extraction. Journal of Mechanical Design, Transactions of the ASME, 143(6). https://doi.org/10.1115/1.4048819

Haque, T. U., Saber, N. N., & Shah, F. M. (2018). Sentiment analysis on large-scale Amazon product reviews. 2018 IEEE International Conference on Innovative Research and Development, ICIRD 2018, June, 1–6. https://doi.org/10.1109/ICIRD.2018.8376299

Jin, J., Ji, P., Liu, Y., & Johnson Lim, S. C. (2015). Translating online customer opinions into engineering characteristics in QFD: A probabilistic language analysis approach. Engineering Applications of Artificial Intelligence, 41, 115–127. https://doi.org/10.1016/j.engappai.2015.02.006

Jin, J., Liu, Y., Ji, P., & Liu, H. (2016). Understanding big consumer opinion data for market-driven product design. International Journal of Production Research, 54(10), 3019–3041. https://doi.org/10.1080/00207543.2016.1154208

Kilroy, D., Healy, G., & Caton, S. (2022). Using Machine Learning to Improve Lead Times in the Identification of Emerging Customer Needs. IEEE Access, 10, 37774–37795. https://doi.org/10.1109/ACCESS.2022.3165043

Kim, S., Kim, S., Min, S., Yang, M., Choi, J., & Akay, H. (2019). AI for design : Virtual design assistant CIRP Annals - Manufacturing Technology AI for design : Virtual design assistant. CIRP Annals - Manufacturing Technology, 68(1), 141–144. https://doi.org/10.1016/j.cirp.2019.03.024

Kühl, N., Mühlthaler, M., & Goutier, M. (2020). Supporting customer-oriented marketing with artificial intelligence: automatically quantifying customer needs from social media. Electronic Markets, 30(2), 351–367. https://doi.org/10.1007/s12525-019-00351-0

Kühl, N., & Satzger, G. (2021). Needmining: Designing Digital Support to Elicit Needs from Social Media. 1–36. http://arxiv.org/abs/2101.06146

Lim, S., & Tucker, C. S. (2016). A Bayesian Sampling Method for Product Feature Extraction from Large-Scale Textual Data. Journal of Mechanical Design, Transactions of the ASME, 138(6), 1–26. https://doi.org/10.1115/1.4033238

Lim, S., Tucker, C. S., & Kumara, S. (2017). An unsupervised machine learning model for discovering latent infectious diseases using social media data. Journal of Biomedical Informatics, 66, 82–94. https://doi.org/10.1016/j.jbi.2016.12.007

Lindemann, M., Briele, K., & Schmitt, R. H. (2020). Methodical data-driven integration of customer needs from social media into the product development process. Procedia CIRP, 88, 127–132. https://doi.org/10.1016/j.procir.2020.05.023

Liu, Q., Wang, K., Li, Y., & Liu, Y. (2020). Data-driven concept network for inspiring designers' idea generation. Journal of Computing and Information Science in Engineering, 20(3), 1–12. https://doi.org/10.1115/1.4046207

Liu, Y., Jin, J., Ji, P., Harding, J. A., & Fung, R. Y. K. (2013). Identifying helpful online reviews: A product designer's perspective. CAD Computer Aided Design, 45(2), 180–194. https://doi.org/10.1016/j.cad.2012.07.008

Luo, Y., & Xu, X. (2021). Comparative study of deep learning models for analyzing online restaurant reviews in the era of the COVID-19 pandemic. International Journal of Hospitality Management, 94(January), 102849. https://doi.org/10.1016/j.ijhm.2020.102849

McComb, C., Cagan, J., & Kotovsky, K. (2015). Lifting the Veil: Drawing insights about design teams from a cognitively-inspired computational model. Design Studies, 40, 119–142. https://doi.org/10.1016/j.destud.2015.06.005

McComb, C., Cagan, J., & Kotovsky, K. (2017a). Capturing Human Sequence-Learning Abilities in Configuration Design Tasks Through Markov Chains. Journal of Mechanical Design, Transactions of the ASME, 139(9), 1–12. https://doi.org/10.1115/1.4037185

McComb, C., Cagan, J., & Kotovsky, K. (2017b). Mining Process Heuristics from Designer Action Data Via Hidden Markov Models. Journal of Mechanical Design, Transactions of the ASME, 139(11), 1–12. https://doi.org/10.1115/1.4037308

Nordin, V. J. (2002). The voice of the customer. Forestry Chronicle, 78(3), 343–345. https://doi.org/10.4324/9780080496313-9

Peddireddy, D., Fu, X., Shankar, A., Wang, H., Joung, B. G., Aggarwal, V., Sutherland, J. W., & Jun, M. B. G. (2021). Identifying manufacturability and machining processes using deep 3D convolutional networks. Journal of Manufacturing Processes, 64(February), 1336–1348. https://doi.org/10.1016/j.jmapro.2021.02.034

Rahman, M. H., Xie, C., & Sha, Z. (2021). Predicting sequential design decisions using the function-behavior-structure design process model and recurrent neural networks. Journal of Mechanical Design, Transactions of the ASME, 143(8). https://doi.org/10.1115/1.4049971

Raina, A., McComb, C., & Cagan, J. (2019). Learning to design from humans: Imitating human designers through deep learning. Proceedings of the ASME Design Engineering Technical Conference, 2A-2019(November), 1–11. https://doi.org/10.1115/1.4044256

Rejeb, H. ben, Boly, V., & Morel-Guimaraes, L. (2011). Attractive quality for requirement assessment during the front end of innovation. TQM Journal, 23(2), 216–234. https://doi.org/10.1108/17542731111110258

Roh, T., Jeong, Y., Jang, H., & Yoon, B. (2019). Technology opportunity discovery by structuring user needs based on natural language processing and machine learning. PLoS ONE, 14(10). https://doi.org/10.1371/journal.pone.0223404

Sexton, T., & Ren, M. Y. (2017). Learning an optimization algorithm through human design iterations. Journal of Mechanical Design, Transactions of the ASME, 139(10). https://doi.org/10.1115/1.4037344

Sun, Y. F., Lu, A. P., Zhuo, L., Li, G., Jia, J., Liu, W., & Hu, C. J. (2021). Quality Big Data Analysis and Management Based on Product Satisfaction Index. IOP Conference Series: Materials Science and Engineering, 1043(3). https://doi.org/10.1088/1757-899X/1043/3/032004

Timoshenko, A., & Hauser, J. R. (2019). Identifying customer needs from user-generated content. Marketing Science, 38(1), 1–20. https://doi.org/10.1287/mksc.2018.1123

Trautmann, L., & Piros, A. (2020). The concept of EmPatGen (Emotional Pattern Generator). SN Applied Sciences, 2(5). https://doi.org/10.1007/s42452-020-2765-5

Tuarob, S., & Tucker, C. S. (2015). Automated discovery of lead users and latent product features by mining large-scale social media networks. Journal of Mechanical Design, Transactions of the ASME, 137(7). https://doi.org/10.1115/1.4030049

Tucker, C., & Kim, H. M. (2011). Predicting emerging product design trend by mining publicly available customer review data. ICED 11 - 18th International Conference on Engineering Design - Impacting Society Through Engineering Design, 6(August), 43–52.

Wang, L. (2011). Detc2011-48338 Using Web Based User-Generated Content. Asme 2011, 1–15.

Wang, M. (2015). A Data-Driven Network Analysis Approach to Predicting Customer Choice Sets for Choice Modeling in Engineering Design. 137(July), 1–11. https://doi.org/10.1115/1.4030160

Wang, T. (2022). A Novel Approach of Integrating Natural Language Processing Techniques with Fuzzy TOPSIS for Product Evaluation. Symmetry, 14(1). https://doi.org/10.3390/sym14010120

Wang, X., Wang, Y., & Liu, A. (2020). Determining Customer-Focused Product Features through Social Network Analysis. Procedia CIRP, 91, 704–709. https://doi.org/10.1016/j.procir.2020.02.227

Wang, Y., & Li, X. (2021a). Addressing the Semantic Gap in the Consumer-to-Manufacturer Strategy Using Dual Convolutional Neural Network. 2021 IEEE International Conference on Industrial Engineering and Engineering Management, IEEM 2021, 624–628. https://doi.org/10.1109/IEEM50564.2021.9673094

Wang, Y., & Li, X. (2021b). Mining Product Reviews for Needs-Based Product Configurator Design: A Transfer Learning-Based Approach. IEEE Transactions on Industrial Informatics, 17(9), 6192–6199. https://doi.org/10.1109/TII.2020.3043315

Wang, Y., Li, X., & Mo, D. (2021). Knowledge-Empowered Multi-Task Learning to Address the Semantic Gap Between Customer Needs and Design Specifications. IEEE Transactions on Industrial Informatics, 3203(c). https://doi.org/10.1109/TII.2021.3067141

Wang, Y., Li, X., & Tsung, F. (2020). Configuration-based smart customization service: A multitask learning approach. IEEE Transactions on Automation Science and Engineering, 17(4), 2038–2047. https://doi.org/10.1109/TASE.2020.2986774

Wang, Y., Luo, L., & Liu, H. (2022). Bridging the Semantic Gap Between Customer Needs and Design Specifications Using User-Generated Content. In IEEE Transactions on Engineering

Management (Vol. 69, Issue 4, pp. 1622–1634). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/TEM.2020.3021698

Wang, Y., Mo, D., Wang, Y., Mo, D. Y., & Tseng, M. M. (2018). Mapping customer needs to design parameters in the front end of product design by applying deep learning CIRP Annals - Manufacturing Technology Mapping customer needs to design parameters in the front end of product design by applying deep learning. CIRP Annals - Manufacturing Technology, 67(1), 145–148. https://doi.org/10.1016/j.cirp.2018.04.018

Wang, Y., Mo, D. Y., & Tseng, M. M. (2018). Mapping customer needs to design parameters in the front end of product design by applying deep learning. CIRP Annals, 67(1), 145–148. https://doi.org/10.1016/j.cirp.2018.04.018

Wang, Y., & Zhang, J. (2017). Bridging the semantic gap in customer needs elicitation: A machine learning perspective. Proceedings of the International Conference on Engineering Design, ICED, 4(DS87-4), 643–651.

Wang, Y., Zhao, W., & Wan, W. X. (2020). Needs-Based Product Configurator Design for Mass Customization Using Hierarchical Attention Network. IEEE Transactions on Automation Science and Engineering, January, 1–10. https://doi.org/10.1109/tase.2019.2957136

Wedel, I., Palk, M., & Voß, S. (2021). A Bilingual Comparison of Sentiment and Topics for a Product Event on Twitter. Information Systems Frontiers. https://doi.org/10.1007/s10796-021-10169-x

Wu, Q., Hsu, W. L., Xc, T., Liu, Z., Ma, G., Jacobson, G., & Zhao, S. (2019). Speaking with Actions - Learning Customer Journey Behavior. Proceedings - 13th IEEE International Conference on Semantic Computing, ICSC 2019, 279–286. https://doi.org/10.1109/ICOSC.2019.8665577

Xu, E. J., Tang, B., Liu, X., & Xiong, F. (2020). Automatic aspect-based sentiment analysis (AABSA) from customer reviews. CEUR Workshop Proceedings, 2614, 47–66.

Xu, K., Liao, S. S., Li, J., & Song, Y. (2011). Mining comparative opinions from customer reviews for Competitive Intelligence. Decision Support Systems, 50(4), 743–754. https://doi.org/10.1016/j.dss.2010.08.021

Zhao, Y., Zhao, J., Jiang, L., Tan, R., Niyato, D., Li, Z., Lyu, L., & Liu, Y. (2021). Privacy-Preserving Blockchain-Based Federated Learning for IoT Devices. IEEE Internet of Things Journal, 8(3), 1817–1829. https://doi.org/10.1109/JIOT.2020.3017377

Zhou, F., Ayoub, J., Xu, Q., & Yang, X. J. (2020). A machine learning approach to customer needs analysis for product ecosystems. Journal of Mechanical Design, Transactions of the ASME, 142(1). https://doi.org/10.1115/1.4044435

Zhou, F., & Jiao, R. J. (2016). Latent customer needs elicitation for big-data analysis of online product reviews. IEEE International Conference on Industrial Engineering and Engineering Management, 2016-Janua, 1850–1854. https://doi.org/10.1109/IEEM.2015.7385968

Zhou, F., Jiao, R. J., & Linsey, J. S. (2015). Latent customer needs elicitation by use case analogical reasoning from sentiment analysis of online product reviews. Journal of Mechanical Design, Transactions of the ASME, 137(7). https://doi.org/10.1115/1.4030159

Zhou, F., Qu, X., Jiao, J., and Helander, M. G., (2014). Emotion Prediction From Physiological Signals: A Comparison Study Between Visual and Auditory Elicitors," Interact. Comput., 26(3), pp. 285–302.

Encodage Logique et Explications Visuelles pour l'Argumentation

Théo Duchatelle[1]

[1] Université Toulouse III - Paul Sabatier, IRIT

theo.duchatelle@irit.fr

Résumé

En argumentation abstraite, le but est d'extraire des groupes d'arguments qui satisfont certaines contraintes, en se basant sur les conflits qui opposent les arguments. Ces groupes d'arguments peuvent être calculés en utilisant des outils logiques, dont certains prennent en compte des cas de généralisation du cadre de base. Une fois ces groupes d'arguments calculés, on peut vouloir chercher à les expliquer. On tâche d'apporter des éléments de réponse à cette question en se basant sur des critères visuels.

Mots-clés

Argumentation, Explicabilité, Logique, Graphes

Abstract

In abstract argumentation, the aim is to extract groups of arguments that satisfy some constraints, based on the conflicts that exist between the arguments. These groups of arguments can be computed using logical tools, some of them capturing cases of generalization of the basic framework. Once these groups of arguments are computed, we may seek to explain them. We try to bring some elements of answer, that are based on visual criteria, to this question.

Keywords

Argumentation, Explainability, Logic, Graphs

1 Introduction

L'argumentation abstraite est un mécanisme de prise de décision, introduit par Dung 1995, [4]. L'idée est de modéliser des entités abstraites appelées « arguments », ainsi que les conflits qui émergent entre eux (i.e. le fait qu'un argument en contredise un autre). Tout cela est appelé un cadre d'argumentation. On peut imaginer un cadre d'argumentation comme étant une sorte de débat, par exemple politique, pour décider de propositions à appliquer, ou juridique pour décider d'un jugement à rendre. Une fois un cadre établi, l'objectif est d'en extraire les arguments qui « gagnent » le débat, appelés collectivement extension, en utilisant des sémantiques qui définissent la manière de « gagner ».

Certains travaux se sont intéressés à généraliser cette notion. Ces travaux proposent ce qu'on appellera des « enrichissements » du cadre de base. On peut citer l'introduction d'une relation positive, de support, entre les arguments, le fait que plusieurs arguments contredisent en coalition un autre argument, ou le fait qu'un argument contredise le fait qu'un autre argument en contredise un troisième. On parle, dans le dernier cas, de relation d'ordre supérieur.

Il existe plusieurs méthodes pour calculer les extensions d'un cadre d'argumentation. L'une d'elles consiste à définir un encodage logique, pour obtenir les extensions via un mécanisme d'inférence. Dans cette lignée, Besnard et al. 2022a, [2] propose un encodage générique, qui permet de capturer plusieurs enrichissements et leurs combinaisons.

Une personne à qui une extension est présentée pourrait remettre en cause sa validité. Cette personne chercherait alors une explication au fait que les arguments présentés forment une extension. Čyras et al. 2021, [3] donne un état de l'art des méthodes d'explicabilité pour l'argumentation abstraite, ou l'utilisant pour expliquer. Les travaux de Besnard et al. 2022b, [1] exhibent des explications pour les processus d'argumentation qui reposent sur des critères visuels. Ces explications vont dans le sens de Vesic et al. 2022, [6], qui montre que des supports visuels aident à mieux comprendre le fonctionnement d'un processus argumentatif, et plus généralement de Miller 2019, [5] qui plaide pour des explications plus personnalisées et se reposant moins sur des mécanismes accessibles uniquement aux experts.

Ce papier a pour but de présenter brièvement Besnard et al. 2022a, [2] et Besnard et al. 2022b, [1].

2 Argumentation abstraite

Un *cadre d'argumentation* est un graphe orienté, où les noeuds représentent les *arguments* et la relation binaire représente les conflits. Un argument en *attaque* un autre s'il existe un arc du premier vers le deuxième (notion généralisable à l'attaque par un ensemble d'arguments).

Dans un contexte politique, les arguments pourraient être les propositions avancées et les conflits une incompatibilité entre les propositions. Dans un contexte juridique, les arguments pourraient être les jugements possibles et les éléments du procès, et les conflits une réfutation de véracité.

Un argument est *acceptable* selon un ensemble d'arguments si cet ensemble attaque tous les attaquants de l'argument. L'acceptabilité est au coeur des sémantiques classiques qui permettent de sélectionner les arguments qui « gagnent » le débat. Un ensemble d'arguments est dit *sans conflit* s'il n'existe pas d'arc qui relie deux de ses arguments (i.e. l'ensemble est cohérent). Il est dit *admissible* s'il est sans conflit et que tous ses arguments sont acceptables selon lui-même (i.e. il se défend contre tous ses attaquants).

Exemple. Dans le cadre d'argumentation de la figure 1,

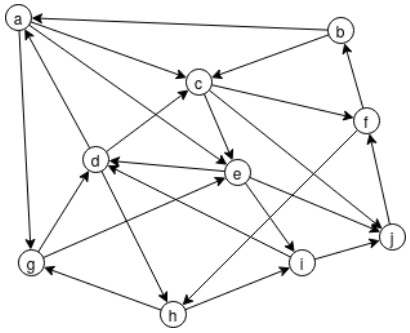


FIGURE 1 – Un exemple de cadre d'argumentation

l'ensemble $\{a, i, f\}$ est sans conflit et admissible.

3 Encodage logique

Pour calculer les extensions des sémantiques classiques d'un cadre d'argumentation, Besnard et al. 2022a, [2] a mis au point un encodage logique. Il utilise la logique du premier ordre, avec égalité, sans symbole de fonction. Il comprend deux parties : (i) une première qui encode uniquement le graphe ; (ii) une deuxième qui correspond à la sémantique dont il faut calculer les extensions.

On peut montrer que chaque modèle de la théorie encode une extension, et que pour toute extension, il existe un modèle de la théorie qui l'encode.

L'encodage logique est générique pour pouvoir capturer plusieurs enrichissements. Cela signifie que les formules de la partie qui encode la sémantique utilisent des prédicats laissés sans définition, dont il faut instancier la signification selon les enrichissements apportés. On illustre ce principe avec les formules correspondant à la sémantique sans conflit. Il s'agit des formules génériques (1).

$$\forall \alpha \in Att[Act(\alpha) \rightarrow \exists x \in E(T(\alpha, x) \wedge NA(x))] \quad (1a)$$

$$\forall x \in E\big(NA(x) \rightarrow \neg Acc(x)\big) \quad (1b)$$

$Att(x)$ signifie que x est une attaque, $Acc(x)$ signifie que x est dans l'extension, $NA(x)$ signifie que x ne peut pas être dans l'extension, $T(\alpha, x)$ signifie que x est la cible de α. $E(x)$ et $Act(x)$ sont ici des prédicats qui n'auront de signification que dans une instanciation particulière d'enrichissements. Ils représentent respectivement le fait que x peut faire partie de l'extension et que x est une interaction activable. Si le cadre possède des relations d'ordre supérieur, il faut instancier les prédicats E et Act avec les formules (2).

$$\forall x\big(E(x) \leftrightarrow Arg(x) \vee Att(x)\big) \quad (2a)$$

$$\forall \alpha[Act(\alpha) \leftrightarrow (\forall a \in Arg(S(\alpha, a) \rightarrow Acc(a)) \wedge Acc(\alpha))] \quad (2b)$$

Ici, on ajoute $Arg(x)$ qui signifie que x est un argument. Si le cadre d'argumentation a une relation de support, il faut instancier les prédicats E et Act avec les formules (3).

$$\forall x\big(E(x) \leftrightarrow Arg(x)\big) \quad (3a)$$

$$\forall \alpha[Act(\alpha) \leftrightarrow (\forall a \in Arg(S(\alpha, a) \rightarrow (Acc(a) \wedge Supp(a))))] \quad (3b)$$

Ici, on ajoute $Supp(x)$ qui signifie que x doit posséder un support. L'aspect modulaire de l'approche se ressent quand on combine les enrichissements. Par exemple, des formules (2) et (3), on peut facilement déduire les formules (4), qui correspondent au cas d'un cadre avec une relation de support et des relations d'ordre supérieur.

$$\forall x\big(E(x) \leftrightarrow Arg(x) \vee Att(x)\big) \quad (4a)$$

$$\forall \alpha[Act(\alpha) \leftrightarrow (\forall a \in Arg(S(\alpha, a) \rightarrow (Acc(a) \wedge Supp(a))) \wedge (Acc(\alpha) \wedge Supp(\alpha)))] \quad (4b)$$

4 Explications visuelles

Supposons qu'une personne utilise un système d'argumentation abstraite pour prendre une décision. Le système lui calcule une extension (sans conflit, admissible, ou autre selon des contraintes) dont les arguments servent de base à sa décision. Cependant, la personne est surprise par le résultat. Peut-être s'attendait-elle à autre chose, ou n'imaginait même pas que ce résultat était possible. Elle cherche donc à savoir ce qui en fait un résultat valable. Autrement dit, elle cherche à savoir pourquoi l'ensemble qui lui est présenté est une extension de la sémantique choisie.

Exemple. Une personne veut prendre une décision se basant sur le cadre d'argumentation de la figure 1. Les contraintes spécifiées sur la décision correspondent à une extension admissible. Le système lui renvoie l'ensemble $\{a, i, f\}$. Mais la personne est surprise car, par exemple, elle ne s'attendait pas à la présence de a dans l'extension. Elle cherche donc à savoir pourquoi $\{a, i, f\}$ est une extension admissible.

En ce sens, la réponse qui sera fournie à la personne, et qu'on appellera explication, doit pouvoir lui permettre de vérifier si un certain ensemble (a priori le résultat sur lequel elle s'interroge) est bien une extension de la sémantique choisie ou non. Si elle a accès au cadre d'argumentation initial, elle pourrait le vérifier elle-même en utilisant les définitions. Mais le graphe pouvant potentiellement être très grand et rempli, il semble pertinent de chercher à lui faciliter le travail. Afin de tirer parti de la nature *visuelle* des graphes, on cherchera donc à calculer une partie pertinente du cadre d'argumentation initial, telle que le processus de vérification puisse se reposer sur des conditions pouvant être *vues* dans cette partie (i.e. des conditions *structurelles*). Nos explications, telles que définies dans Besnard et al. 2022b, [1], sont donc des sous-graphes du cadre initial. On n'abordera ici que les cas du sans conflit et de l'admissibilité. Pour montrer qu'un ensemble est sans conflit, on calcule le sous-graphe induit par cet ensemble. On prouve que l'ensemble est sans conflit (propriété sémantique) si et seulement le sous-graphe induit ne possède pas d'arcs (propriété structurelle / visuelle). Pour expliquer l'admissibilité

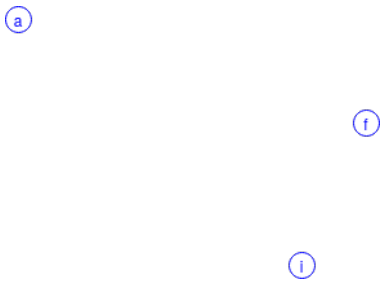


FIGURE 2 – Explication pour le fait que $\{a, i, f\}$ est sans conflit dans la figure 1 (aucun arc entre les arguments)

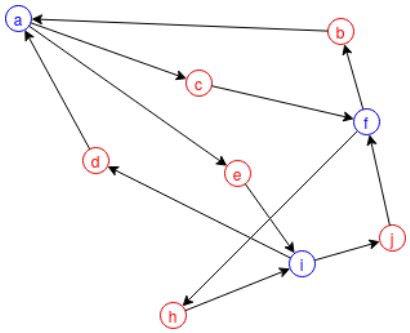


FIGURE 3 – Explication pour le fait que a, i, et f sont acceptables selon $\{a, i, f\}$ dans la figure 1 (ils ont des attaquants, en rouge, mais aucun n'est un noeud source)

d'un ensemble, on calcule le sous-graphe induit par l'ensemble et ses attaquants, puis on ne garde que les attaques qui vont de l'ensemble vers ses attaquants ou vice versa. On prouve que tous les arguments de l'ensemble sont acceptables (propriété sémantique) si et seulement s'il n'existe pas de noeud source parmi les attaquants dans l'explication (propriété structurelle / visuelle). Ainsi, ce sont les deux sous-graphes décrits ici qui, ensemble, forment une explication pour l'admissibilité d'un ensemble d'arguments.

Exemple. Les figures 2 et 3 *montrent* pourquoi (et donc forment une explication pour le fait que) $\{a, i, f\}$ est une extension admissible dans la figure 1. À l'inverse, la figure 4 *montre* pourquoi $\{i, f\}$ n'est pas une extension admissible dans la figure 1.

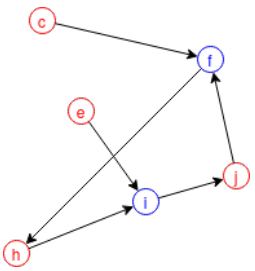


FIGURE 4 – Explication pour le fait que i et f ne sont pas acceptables selon $\{i, f\}$ dans la figure 1 (ils ont chacun un attaquant qui est un noeud source)

5 Conclusion et perspectives

Pour conclure, on a développé dans Besnard et al. 2022a, [2] une théorie logique de l'argumentation abstraite qui permet de calculer les extensions des sémantiques classiques. Cette théorie est générique car elle peut être paramétrée pour correspondre à des cadres plus ou moins enrichis. Cependant, certains enrichissements ne peuvent pas être retrouvés par cette théorie, par exemple les sémantiques graduelles, car ayant un fonctionnement fondamentalement trop différent. En plus de cela, on a défini dans Besnard et al. 2022b, [1] des explications qui permettent de certifier si un ensemble d'argument est une extension d'une sémantique ou non en se basant sur des propriétés visuelles. Cela permet de vérifier des résultats obtenus via des méthodes d'argumentation abstraite, même en étant non expert, car la vérification ne se base alors plus que sur des propriétés structurelles des graphes. En l'état ces explications ne permettent hélas pas encore de répondre à des questions plus raffinées, par exemple contrastives, comme « Pourquoi tel argument est présent dans l'extension et pas tel autre ? ».
Par la suite, on souhaiterait étendre notre outil logique pour lui permettre de calculer également les explications que l'on a définies. Cela aurait l'avantage supplémentaire de pouvoir étudier ces explications via le prisme des formalismes logiques et donc d'en extraire des qualités mieux exprimables mathématiquement que le fait de se baser sur des critères visuels, ou de pouvoir les comparer à des notions déjà existantes d'explications dans ces formalismes.

Remerciements

Je tiens à remercier Marie-Christine Lagasquie-Schiex et Sylvie Doutre pour leur bienveillance et leurs conseils avisés, ainsi que Philippe Besnard pour sa capacité à me guider durant mes débuts en tant que doctorant.

Références

[1] P. Besnard, S. Doutre, T. Duchatelle, and M.-C. Lagasquie-Schiex. Explaining semantics and extension membership in abstract argumentation. *Intelligent Systems with Applications*, 16 :200118, 2022.

[2] P. Besnard, S. Doutre, T. Duchatelle, and M.-C. Lagasquie-Schiex. Generic logical encoding for argumentation. *Journal of Logic and Computation*, 2022.

[3] K. Čyras, A. Rago, E. Albini, P. Baroni, and F. Toni. Argumentative XAI : A survey. In *Proc. of IJCAI*, pages 4392–4399, 2021.

[4] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intel.*, 77(2) :321–357, 1995.

[5] T. Miller. Explanation in artificial intelligence : Insights from the social sciences. *Artif. Intel.*, 267 :1–38, 2019.

[6] S. Vesic, B. Yun, and P. Teovanovic. Graphical representation enhances human compliance with principles for graded argumentation semantics. In *Proc. AAMAS*, 2022.

Génération de données synthétiques à partir d'une forêt aléatoire

J. Gonzalez[1], F. Dama[1]

[1] Centre de Recherche et d'Innovation de Talan, France

Résumé

Dans le cadre de l'apprentissage incrémental avec les forêts NCMF, nous proposons une stratégie de génération de données synthétiques locale à la volée et sur demande. En outre, cette stratégie permet, contrairement à l'algorithme original, de continuer à incrémenter le modèle sans nécessiter l'accès aux anciennes données ; également, elle permet de synthétiser un dataset complet. Nos expériences ont été réalisées sur des ensembles de données de référence de UCI et offrent des résultats prometteurs.

Mots-clés

Apprentissage incrémental, Données synthétiques, NCMF.

Abstract

In the context of incremental learning with NCMF forests, we propose a strategy for generating local synthetic data on the fly and on demand. Moreover, this strategy allows, unlike the original algorithm, to continue to increment the model without requiring access to old data ; also, it allows to synthesize a complete dataset. Our experiments were performed on UCI benchmark datasets and offer promising results.

Keywords

Incremental Learning, Synthetic data, NCMF.

1 Introduction

Dans de nombreuses applications, les données sont disponibles que par petits lots au fil du temps [2]. La question se pose donc de savoir ce que l'on fait du modèle entraîné sur un sous-jeu de données. La stratégie qui consiste à réentraîner le modèle depuis le début (*from scratch*) n'est clairement pas efficace [4]. D'une part, cette méthode est très coûteuse en temps de calcul, d'autre part, elle empêche l'intégration de nouvelles données en temps réel et n'est pas réalisable lorsque les données initiales ne sont plus disponibles. Il apparaît ainsi nécessaire de mettre à jour un modèle existant de manière incrémentale.

Il a été démontré que les forêts aléatoires (RF) [1], en plus de leur nature multi-classes et leur capacité à généraliser, ont également la capacité de s'incrémenter en données et en classes [5]. Les forêts de type NCM (*Nearest Class Mean*) proposent une fonction de séparation différente des forêts classiques, ainsi que diverses stratégies d'incrémentation [6]. Cependant, les stratégies proposées pour continuer de faire croître les arbres pendant la phase incrémentale nécessitent l'accès aux données précédentes, ce qui peut être problématique. Nous proposons ici une stratégie qui permet aux NCMF de poursuivre leur incrémentation sans accéder aux anciennes données, mais plutôt en les générant sur demande de manière synthétique.

La structure de l'article est décrite comme suit. Dans la section 2, nous décrivons la stratégie d'incémentation proposée. La section 3 est dédiée à la génération de données synthétiques avec la NCMF et le modèle de mélange gaussien (utilisé pour comparaison). Ensuite en section 4, la capacité générative de la NCMF et la pertinence de la stratégie d'incrémentation proposée sont évaluées, en s'appuyant sur des datasets benchmarks de UCI. La dernière section est réservée pour la conclusion.

2 La stratégie proposée : IGTLGSS

2.1 Nearest Class Mean Forest (NCMF)

Nearest Class Mean Forest (NCMF) [6] est une forêt aléatoire (RF) [1] dont les nœuds sont des classifieurs NCM (Nearest Class Mean) [3]. Dans ces nœuds, des centroïdes conditionnels aux classes sont calculés et affectés à un noeud fils gauche ou droit. Une observation parcourant le noeud est alors dirigée vers le noeud fils associé au centroïde le plus proche. Une fois entraînées sur un premier jeu d'entraînement, les NCMFs peuvent être mises à jour avec de nouvelles données. La stratégie d'incrémentation IGT (*Incremental Growing Tree*) peut être utilisée à cet effet.

2.2 Incrémentation avec IGT et limite

La stratégie d'incrémentation IGT, introduite dans [6], procède comme suit : la nouvelle observation est propagée dans chaque arbre de la forêt pré-entraînée ; ensuite, les distributions de classes dans chacune des feuilles qu'elle atteint sont mises à jour. Lorsque l'incrément engendre un changement de classe majoritaire dans une feuille, alors cette feuille est transformée en nœud et la construction récursive d'un sous-arbre local débute. Une limite inhérente à la stratégie IGT est qu'elle nécessite l'accès aux anciennes données d'entraînement. Pour pallier cette limite, nous proposons la méthode IGTLGSS (*Incremental Growing Tree with Local Generation of Synthetic Samples*).

2.3 La stratégie IGTLGSS

La méthode IGTLGSS consiste en la génération de données synthétiques locales au moyen de la loi normale multivariée. Pour rendre cela possible, nous avons stocké les matrices de covariance conditionnelles aux classes au ni-

veau des noeuds, en plus des centroïdes. Dans la suite, nous notons $\mathcal{K} = \{k_1, ..., k_l\}$ l'ensemble de l labels et $(X^{[INCR]}, y^{[INCR]})$ l'ensemble à partir duquel la forêt pré-entraînée doit s'incrémenter.
Pour chaque observation (x, y) dans $(X^{[INCR]}, y^{[INCR]})$:

1. Chaque arbre de la forêt propage l'observation jusqu'à une feuille pour prédire un label $k_i \in \mathcal{K}$.
2. Chaque arbre t, pour lequel la prédiction k_i ne correspond pas au label y, est incrémenté. L'idée ici est de mettre à jour seulement les arbres en difficulté en exécutant les étapes suivantes :
   (i) (x, y) est propagée jusque dans une feuille, puis la répartition $S^t(x) = [S^t(k_1|x), ..., S^t(k_l|x)]$ stockée dans celle-ci est extraite, où $S^t(k_i|x)$ correspond au nombre d'occurrences de la classe k_i dans la feuille.
   (ii) Pour chaque classe k_i, nous utilisons la loi normale multivariée locale correspondant à cette classe (dont les paramètres sont stockés dans le noeud parent, *i.e.*, l'ancienne feuille). De cette manière, nous générons un ensemble synthétique local $(X^{[SYNTH]}, y^{[SYNTH]})$.
   (iii) L'observation (x, y) est ajoutée à $(X^{[SYNTH]}, y^{[SYNTH]})$ et, depuis cette position, la construction récursive du sous-arbre se poursuit comme dans une phase d'apprentissage classique.

Le modèle $NCMF$ devient alors capable de s'incrémenter à partir de nouvelles données sans avoir besoin d'accéder aux anciennes données.

3 Génération de données synthétiques

3.1 *Nearest Class Mean Forest* (NCMF)

Pour générer des données synthétiques à partir d'une NCMF entraînée, nous proposons la procédure suivante. Pour chaque classe k_i, une observation est générée en trois étapes : (i) un arbre t est tiré aléatoirement dans la forêt ; (ii) ensuite, une feuille est tirée aléatoirement, de telle sorte que plus une feuille contient d'occurrences de la classe k_i, plus elle a de chance d'être choisie ; (iii) enfin, une observation est générée à partir de la loi normale multivariée locale correspond à la classe k_i (dont les paramètres sont stockés dans le noeud parent). Les étapes précédentes sont répétées pour obtenir le bon nombre de données.

3.2 Gaussian Mixture Model (GMM)

Le GMM est un modèle probabiliste exprimé sous forme de mélange de lois Gaussiennes (lois normales multivariées) [8]. Il permet d'estimer de façon paramétrique la distribution d'un ensemble de variables aléatoires interdépendantes en la modélisant comme la somme de K Gaussiennes. De plus, chaque loi Gaussienne est associée à une classe spécifique.
Les paramètres du modèle GMM sont généralement estimés de façon non-supervisée au moyen de l'algorithme Espérance-Maximisation (EM) [9]. Cependant, lorsque les données sont labélisées, comme c'est le cas dans notre étude, une loi Gaussienne peut être ajustée à chaque classe de façon supervisée.
Le GMM est un modèle probabiliste qui permet de générer des données synthétiques. La génération de données est faite en deux étapes : (i) dans un premier temps, la classe de l'observation est générée ; (ii) ensuite, l'observation est générée à partir de la loi Gaussienne correspondante. Les étapes (i) et (ii) sont répétées jusqu'à l'obtention d'un dataset synthétique.

4 Expérimentations

4.1 Datasets

Dans nos expérimentations, nous avons considéré six datasets benchmarks de UCI dont la description est fournie dans le tableau 1.

Dataset	$\lvert\mathcal{K}\rvert$	n_{obs}	n_{feat}
Raisin Dataset ($R.$)	2	900	4
Breast Cancer Coimbra ($B.C.$)	2	116	10
Banknote authentication ($B.K.$)	2	1372	5
Avila ($A.V.$)	12	20867	10
Speaker Accent Recognition ($A.R.$)	6	329	12
Optical Recognition of Handwritten Digits ($H.G.$)	10	5620	64

TABLE 1 – Description des datasets. De gauche à droite : nombre de classes, d'observations et de features (https ://archive.ics.uci.edu/ml/datasets/).

4.2 Protocole expérimental

4.2.1 Évaluation de la capacité générative de la NCMF

Les cinq premiers datasets (c.f. Table 1) ont été considérés dans cette expérimentation. Chaque dataset a été découpé en deux parties : un jeu d'entraînement (80%) et un jeu de test (20%), à l'exception de $A.V.$ pour lequel un découpage est déjà fourni. Ensuite, une NCMF et un GMM ont été entraînés sur chaque jeu d'entraînement. Enfin, les modèles obtenus ont été utilisés pour générer des datasets synthétiques $(X^{[SYNTH]}_{GMM}, y^{[SYNTH]}_{GMM})$ et $(X^{[SYNTH]}_{NCMF}, y^{[SYNTH]}_{NCMF})$, de taille identique aux datasets d'entraînement et qui respectent la répartition des classes de ces derniers.
Pour évaluer la qualité des données synthétiques précédemment générées, nous avons suivi la méthode proposée dans [7]. Plusieurs modèles de classification sont entraînés sur les données réelles et synthétiques. Ensuite, la performance des modèles obtenus est évaluée sur un jeu d'évaluation (composé uniquement de données réelles). Lorsque la perte de performance résultant de l'utilisation des données synthétiques est faible, les données synthétiques sont considérées suffisamment similaires aux données réelles.
Nous avons considéré les modèles de classification suivants : forêt aléatoire standard (RF), NCMF, *Naive Bayes* (NB) et SVM. La mesure de performance utilisée est la

moyenne des métriques d'*accuracy* obtenues pour chacun des modèles. À noter que l'optimisation des hyperparamètres des différents modèles n'entre pas dans le cadre de cet article.

4.2.2 Apprentissage incrémental

Pour comparer les stratégies d'incrémentation IGTLGSS et IGT, nous avons considéré le dataset H.G. (c.f. Table 1). Le protocole utilisé s'apparente à celui décrit dans [5]. Les données d'entraînement (80%) sont découpées aléatoirement sous forme de 50 sous-ensembles (*batchs*) de taille et de distributions de classes identiques. La forêt s'entraîne sur le premier batch, puis s'incrémente sur les autres. L'*accuracy* du modèle est mesurée à la fin de chaque incrémentation.

4.3 Résultats et Analyse

Le tableau 2 compare les performances des modèles entraînés sur les jeux de données réelles et synthétiques. Comme attendu, les modèles entraînés sur les données réelles obtiennent de meilleures performances en comparaison à ceux entraînés sur les données synthétiques (à l'exception du *Raisin Dataset*). Par ailleurs, les résultats montrent que les modèles entraînés sur $(X_{NCMF}^{[SYNTH]}, y_{NCMF}^{[SYNTH]})$ surpassent, dans la majorité des cas, ceux entraînés sur $(X_{GMM}^{[SYNTH]}, y_{GMM}^{[SYNTH]})$. De plus, la perte de performance obtenue suite à l'utilisation des données synthétiques générées par le modèle NCMF est seulement de 0.038 ± 0.029. Ce résultat est prometteur car, à notre connaissance, nous n'avons pas vu de forêts aléatoires être en capacité de générer des données synthétiques.

Dataset	Réel	GMM	NCMF
$R.$	$\underline{0.825} \pm 0.026$	0.829 ± 0.014	$\mathbf{0.846} \pm 0.021$
$B.C.$	$\mathbf{0.635} \pm 0.133$	0.583 ± 0.076	$\underline{0.615} \pm 0.205$
$B.K.$	$\mathbf{0.945} \pm 0.093$	0.924 ± 0.080	$\underline{0.928} \pm 0.102$
$A.V.$	$\mathbf{0.677} \pm 0.280$	0.348 ± 0.041	$\underline{0.642} \pm 0.201$
$A.R.$	$\mathbf{0.686} \pm 0.090$	$\underline{0.629} \pm 0.045$	0.591 ± 0.113

TABLE 2 – Moyenne des métriques d'*accuracy* obtenues pour différents modèles de classification en utilisant les datasets réels et synthétiques.

La figure 1 décrit l'évolution des performances des NCMF incrémentées suivant les stratégies IGT ou IGTLGSS. Nous pouvons observer que les performances sont très satisfaisantes, 0.01 point de moins pour les données synthétiques. La courbe montre que malgré le fait que les données soient synthétiques, la NCMF est capable d'améliorer ses performances jusqu'au *batch* numéro 35. Au delà, la stratégie IGTLGSS semble atteindre un plafond, là où la méthode IGT semble permettre un gain de performance. Il serait intéressant d'observer le comportement de la stratégie IGTLGSS sur des datasets plus larges permettant de considérer par exemple une centaine de *batchs*.

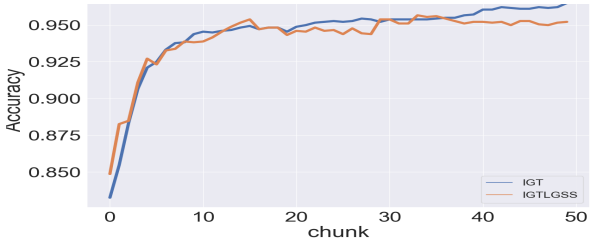


FIGURE 1 – Dataset H.G. : évolution de l'*accuracy* de la $NCMF$ en fonction du nombre d'incrémentations.

5 Conclusion

Notre proposition permet, contrairement à l'algorithme original, de continuer à incrémenter la NCMF sans nécessiter l'accès aux anciennes données. Nos premiers résultats sont encourageants et ouvrent des pistes pour de futures expérimentations.

Références

[1] L. Breiman. Random Forests. *Machine Learning*, 45(1) :5–32, October 2001.

[2] A. Gepperth and B. Hammer. Incremental learning algorithms and applications. In *European symposium on artificial neural networks (ESANN)*, 2016.

[3] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning : data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[4] J. He, R. Mao, Z. Shao, and F. Zhu. Incremental learning in online scenario. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13926–13935, 2020.

[5] R. Pecori, P. Ducange, and F. Marcelloni. Incremental learning of fuzzy decision trees for streaming data classification. In *11th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT 2019)*, pages 748–755, 2019.

[6] M. Ristin, M. Guillaumin, J. Gall, and L. Van Gool. Incremental learning of ncm forests for large-scale image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, page 3654–3661, 2014.

[7] A. Torfi and E. A. Fox. Corgan : Correlation-capturing convolutional generative adversarial networks for generating synthetic healthcare records. *arXiv preprint arXiv :2001.09346*, 2020.

[8] H. Wan, H. Wang, B. Scotney, and J. Liu. A novel gaussian mixture model for classification. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 3298–3303. IEEE, 2019.

[9] M.S. Yang, C.Y. Lai, and C.Y. Lin. A robust em clustering algorithm for gaussian mixture models. *Pattern Recognition*, 45(11) :3950–3961, 2012.

Change-Relaxed Active Fairness Auditing

A. Godinot[1,2,3], E. Le Merrer[2], C. Penzo[3], F. Taïani[1], G. Tredan[4]

[1] Université de Rennes
[2] Centre Inria de l'Université de Rennes
[3] Pôle d'Expertise de la Régulation Numérique
[4] LAAS/CNRS

Abstract

The pervasive deployment of user-facing automated decisions systems raises concerns over their impact on society. The sheer amount of such online platforms and their growing complexity highlights the need for automated and robust audits to assess their impact on users. This paper focuses on a recent theoretical advance named manipulation-proofness. It aims at guaranteeing successive audits of a platform cannot be gamed by the platform, provided the labels returned on the audit dataset do not change.
While this constitutes a decisive step for reliable audits, it is too restrictive, as models naturally evolve with time in practice. This paper thus explores how manipulation-proofness can be adapted to better fit actual scenarios, by studying the effects of relaxing the constraint on the amount of change the remote model can operate while being audited. Our results on the COMPAS dataset demonstrate a request gain in one of the two models considered, while also noticing the surprisingly good performance of the random strawman approach. We believe this constitutes an interesting step for further attempts to improve reliable and manipulation proof audits.

Keywords

Artificial Intelligence, Algorithmic Auditing, Black-box models, Active Learning

1 Introduction

The pervasive deployment of user-facing automated decisions systems raises concerns over their impact on society. The sheer amount of such online platforms and their growing complexity highlights the need for automated and robust audits to assess their impact on users. The advent of highly publicized audits, such as ProPublica's story on COMPAS [12] or Reuters study on Amazon's recruiting tool [8], has led to the algorithmic audit field gaining significant traction. For the public to trust Artificial Intelligence (AI) systems, and more broadly algorithmic decision systems, we need methods to explain the decision of such systems [19, 13], certify their implementation [22, 20] and automatically and robustly detect misconduct [14, 18].
Inspired by "traditional" financial audits, we focus in this work on *external certification audits*. In this type of audit, an external auditor (e.g. a regulator, or an auditing company) is commissioned by a platform to certify some desirable property (the absence of bias, for example) of its system. The system consists in a Machine Learning (ML) model h^* (see subsection 3.1) which is accessed by users through an interface (e.g. a web-page or an Application Programming Interface). To restrict the scope of this work, we consider that h^* is a binary classifier. Furthermore, we assume that the answers presented through the interface are faithful to that of the model h^*. We assume that the platform does not give access to the weights or implementation of the model h^*. The goal of the auditor is thus to certify the system as implemented and as seen by the users. The only information the auditor knows about the audited system is the hypothesis class $\mathcal{H}$ of the model $h^* \in \mathcal{H}$. We dub this setting *remote black-box certification*. Yan and Zhang [22] recently proposed a theoretical framework to model the problem of remote black-box auditing. They provide an algorithm to select a minimal set of points S to estimate a property $\mu(h^*)$ (demographic parity for example) of the remote model h^*. While the model h^* behind the API is allowed to change after the audit, the auditor is guaranteed that the value $\mu(h)$ of any model $h \in \mathcal{H}$ that agrees with h^* on S will be close to their estimation $\widehat{\mu}(h^*)$. This new estimation problem coined *manipulation-proof estimation* by Yan and Zhang is a step towards robust auditing as it provides a framework amenable to theoretical analysis. In practice, even if the type of model stays the same, because of retraining, arrival of new users or small tweaks, models served by platforms change over time. Thus, the requirement that the output of the API on the audit points does not change is too restrictive in practice. Moreover, Yan and Zhang only experimented with linear models on small datasets. In this work, we relax this recent formalization and empirically analyze its performance.

Contributions. This paper makes the following contributions. It first reviews the algorithmic audit setup, and recalls the concept and shortcomings of manipulation proofness (Sections 1 and 3). It then proposes a relaxation (coined r-AFA+) on the tolerated audit errors. We then evaluate this relaxation in Section 5, with two model classes and the COMPAS dataset, before we conclude.

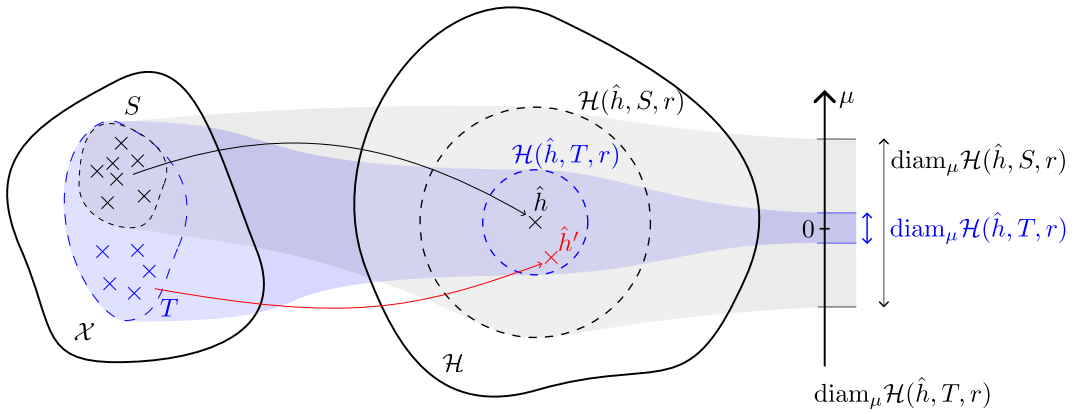


Figure 1: Schematic overview of the r-AFA+ algorithm. On the left, S and T are subsets of the input set $\mathcal{X}$. The goal of the approach is to identify a small set of inputs (left) that quickly reduce the version space (center) to models having close μ-values (right). Those inputs are iteratively improved to refine an estimate of the property of interest (μ-value).
At each outer iteration (line 2), the algorithm consists in three steps. **1. (black)** Train a surrogate $\hat{h}$ on the current labeled dataset $(S, h^*(S))$. **2. (blue)** Find dataset T that minimizes the μ-diameter of the version space. **3. (red)** Merge T in S, query the API on T to train a new surrogate $\hat{h}'$.

2 Related work

AI audits. The AI audit field seeks to understand Artificial Intelligence (AI) algorithms as part of a larger socio-technical system. Most of the published audits include two phases (see for example [2, 15]). First, the auditor analyses the context of the algorithm: the training data, the users or the team who built said system. Then, they typically perform a statistical study to discover potential biases in the algorithm's output. Recently, efforts have been made to formalize requirements dictated by regulatory bodies (such as the Data Minimization Principle [18]) and provide algorithms to help their enforcement. One challenge of remote black-box auditing is to limit the number of queries used to perform the audit. Issuing too few queries prevents any meaningful analysis but if the auditor requests large bursts of queries, they risk being blacklisted.

Robust auditing. Audits relying on statistical studies often make the simplifying assumption that the audited platform is cooperative and honest. While very practical, this is overly optimistic since there were examples of companies trying to evade high-stakes audits in the past [11]. Without this assumption, many simple audits become theoretically impossible [21]. To overcome this limit, the notion of manipulation-proof estimation has recently emerged [22]. Intuitively, this approach aims at constructing an auditing procedure that is resilient to arbitrary manipulation by the auditee while making as few assumptions as possible on the audited target.

Distribution testing. The field of tolerant distribution testing is interested in answering the question: given samples from an unknown distribution p, is this distribution ϵ_1-close ($min_{q\in\mathcal{P}}(d, q) \leq \epsilon_1$) to the set of distributions $\mathcal{P}$ or is it ϵ_2-far ($min_{q\in\mathcal{P}}(d, q) \geq \epsilon_2$) from it? Some fairness measures (such as demographic fairness) can be formulated as the independence between the output of positive labels (e.g. granting a loan, recruiting a new employee) and sensitive attributes (e.g. gender, ethnicity, religious beliefs, political views). Thus, as a specific tolerant distribution test, testing for independence could certify demographic parity. For an introduction to distribution testing and its extension to tolerant distribution testing, refer to [4] and [5]. It is however not straightforward how to certify other fairness properties.

Active learning. Active learning is a form of interactive learning where the learning algorithm can iteratively select the examples to train on. At each training step, the learning algorithm can use the performance of the trained model and past training examples to decide which training example to select next in order to optimize the learning process. The literature on active learning proved that interacting with the trained model could dramatically reduce learning sample complexity [10]. Even if the trained model is treated as a black box, it is possible to iteratively select training points based on the model's outputs to reduce the number of training points needed [7]. These methods cannot be directly applied to black-box remote auditing because they need to probe the model on the whole dataset at each iteration, whereas we try to minimize the number of queries to said model. Yet, the method we present in this work builds on this idea to select the audit dataset by interacting with a *surrogate* of the API instead of the API itself. (We discuss this notion of surrogate model in more detail in Section 3.)

3 Manipulation proofness with AFA

In this section, we present in more detail the notion of *manipulation-proof estimation* introduced by Yan and Zhang [22] in the context of *remote black-box auditing*. We first introduce some key notations and assumptions and formalize the auditing process as a game between the auditor and the platform. We then define the notion of manipulation-proof estimation and present the general intu-

ition behind the AFA algorithm proposed by Yan and Zhang to solve the auditing game in a manipulation-proof manner.

3.1 Notations and assumptions

We consider a platform that seeks to solve a classification task (e.g. whether or not to grant a loan) based on user features (some information regarding the prospective borrower) grouped in a vector $x \in \mathcal{X}$. We assume that the space of all possible inputs—the *sample space* $\mathcal{X}$—is finite. Should $\mathcal{X}$ not be finite, it is suffices to sample a fixed number of instances in $\mathcal{X}$ and treat them as a finite sample space. As explained in [7], it is then possible to adapt the bounds obtained for a finite $\mathcal{X}$.

When the platform trains its model, it effectively chooses a hypothesis $h^* \in \mathcal{H}$ in a set of possible (deterministic) models—the *hypothesis space* $\mathcal{H}$. The hypothesis space could for example be the set of linear binary classifiers on $\mathcal{X}$. Since the platform solves a classification task, the set of possible outputs—the *output space* $\mathcal{Y}$—is also finite. Therefore, because $\mathcal{X}$ and $\mathcal{Y}$ are finite, the space $\mathcal{Y}^{\mathcal{X}}$ of all functions from $\mathcal{X}$ to $\mathcal{Y}$ (and by extension $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$) is also finite. For any set S, we write $|S|$ its cardinal and $\mathcal{P}(S)$ the set of all its subsets.

The auditor seeks to test whether the model h^* used by the platform respects some desirable property $\mu : (\mathcal{P}(\mathcal{X}), \mathcal{H}) \to \mathbb{R}$. For simplicity, we use the notation abuse $\mu(h) = \mu(\mathcal{X}, h)$. Define the *demographic parity* μ_{DP} as

$$\mu_{\text{DP}}(S,h) = \frac{1}{|S \cap A|} \sum_{x \in S \cap A} \mathbb{1}\{h(x) = 1\} - \frac{1}{|S \cap A^c|} \sum_{x \in S \cap A^c} \mathbb{1}\{h(x) = 1\} \quad (1)$$

where $\mathbb{1}\{P\}$ is the indicator function for the predicate P, A is the set of samples in $\mathcal{X}$ with a positive sensitive attribute and A^c its complementary in $\mathcal{X}$.

We define the Hamming distance $d_H(h(S), h^*(S)) = |x \in S : h(x) \neq h^*(x)|$. Simply put, the Hamming distance is the number of points in S on which two hypotheses (or models) h and h^* disagree. Finally, for any subset of the hypothesis class $V \subset \mathcal{H}$, the *μ-diameter* $\text{diam}_\mu(V)$ is the largest difference in the value of μ between any two models in V.

$$\text{diam}_\mu(V) = \max_{h,h' \in V} |\mu(h) - \mu(h')| \quad (2)$$

3.2 The Auditing Game

The auditing process can be modeled as a game between the auditor and the audited platform. First, the auditor decides on the fairness measure $\mu : (\mathcal{P}(\mathcal{X}), \mathcal{H}) \to \mathbb{R}$ on which they want to evaluate the platform. We assume the auditor can directly query the platform's output on a given input $x \in \mathcal{X}$, either through an API or through scraping. By gathering outputs on well-chosen inputs, the auditor seeks to construct an *audit dataset* $S \subseteq \mathcal{X}$, which the auditor will then use to estimate how well the platform respects the fairness measure μ, by using $\widehat{\mu} = \mu(S, h^*)$ as an estimation of the true value $\mu(\mathcal{X}, h^*)$. In practice, platforms regularly retrain their model h^*, for instance to account for new users or to improve it. As a result, h^* is likely to evolve after it has been audited. Thus, constructing a robust estimator intuitively means constructing an estimator that does not change too much even if the model is slightly modified after the audit. More formally, this auditing game can be described as follows:

Phase 1. At time t_0, the auditor constructs an audit dataset $S \subset \mathcal{X}$ to build its estimator $\widehat{\mu}(S, h^*_{t_0})$ by interacting with the model $h^*_{t_0}$ served by the platform. The time needed to construct the audit dataset is supposed to be negligible and the remote model is assumed not to change during this phase.

Phase 2. At any time $t > t_0$ after the audit, and for any reason (retraining, new user or even adversarial change), we allow the model to change slightly. By re-querying the new model h^*_t on the same dataset S, we verify that answers to queries in S have not changed $d_H\left(h^*_t(S), h^*_{t_0}(S)\right) = 0$. The auditor's goal is to detect through their estimation $\widehat{\mu}(S, h^*_t)$ when $\mu(\mathcal{X}, h^*_t)$ deviates too much from some target boundary, in which case the certificate must be revoked.

3.3 Manipulation-proof estimation.

The auditor's algorithm solves the above auditing game if it can produce an auditing set S such that for all $h \in \mathcal{H}$, if $d(h(S), h^*_t(S)) = 0$, then the μ-value of h cannot be at a distance larger than ϵ to $h^*_{t_0}$. To formalize the auditor's goal, we first define the notion of *version space*. It is the set of models h whose output agree with that of h^* on S.

$$\mathcal{H}(S, h^*) = \{h \in \mathcal{H} \ : \ d(h(S), h^*(S)) = 0\} \quad (3)$$

Then, an estimator $\widehat{\mu}(S, h^*)$ of $\mu(\mathcal{X}, h^*)$ is said (r, ϵ)-*manipulation-proof* i.i.f.

$$\text{diam}_\mu \mathcal{H}(S, h^*) < \epsilon \quad (4)$$

The auditor only queries the labels $h^*(x)$ of points $x \in S$ therefore, they can only base their estimation $\widehat{\mu}(S, h^*)$ of μ on $(S, h^*(S))$. Multiple models in $\mathcal{H}$ can have the same answers on S and the auditor does not have any means to know which one of them is behind the API. Thus, there is an uncertainty on the true value $\mu(\mathcal{X}, h^*)$. The μ-diameter evaluates how well different audit datasets S might lead to a smaller/larger uncertainty on $\mu(\mathcal{X}, h^*)$. In their paper, Yan and Zhang frame the auditing game as a minimax game and prove a lower-bound on the number of queries required to reach ϵ-manipulation proofness. Inspired by the *Multiplicative Weight Update* method [1], they provide a randomized approximate algorithm AFA (Active Fairness Auditing, see our adapted version algorithm 1) to compute a solution with a query competitive ratio of $\mathcal{O}(\log(\mathcal{H}) \log(\mathcal{X}))$.

We present in algorithm 1 the core structure of the algorithm proposed in [22] with the modifications discussed in Section 4. The intuition behind this algorithm is to use the black-box teaching algorithm introduced in [7]. To avoid probing the API on the entire $\mathcal{X}$, we assume that we have access to an oracle $\mathcal{O} : \mathcal{P}(\mathcal{X}) \to \mathcal{H}$ providing surrogates of

h^* trained on S (line 4). The oracle $\mathcal{O}$ is assumed to be mistake bounded, that is there exits $M > 0$ such that for any sequence $(x_i)_i$ of points from $\mathcal{X}$ and their corresponding labels $(y_i)_i$, $\sum_{k=1}^{+\infty} \mathbb{1}\left\{\mathcal{O}((x_1, \ldots, x_k))(x_k) \neq y_k\right\} \leq M$. Example of such oracles include the perceptron algorithm [16] and the halving algorithm [3]. This surrogate is then used with the black-box teaching algorithm in [7] to find a subset T of $\mathcal{X}$ maximizing $\text{diam}_\mu \mathcal{H}(\mathcal{O}(S), T)$.

4 Robust auditing in practice: giving some slack on the radius

Because our goal is to systematically analyze the performance of [22] in practice, we had to modify it to account for more realistic settings. The original AFA algorithm requires that after the audit, the labels of the queried points must not change. We argue that this assumption needs to be relaxed for two reasons. First, as we said for practical reasons the model might change slightly over time, modifying a small fraction of labels. Second, the auditor might not have access to an exact description of $\mathcal{H}$. This implies that the hypothesis space used for the audit $\mathcal{H}_{\text{surrogate}}$ does not match the one used by the platform $\mathcal{H}_{\text{API}}$. Because of this mismatch, the return condition line 19 might never be met. For these reasons, we relaxed the condition to $d(h(S), h^*(S)) < r$ in the definition of the version space (Equation 3) and adapted the algorithm accordingly. We name this method r-AFA+ (r-radius Active Fairness Auditing) and provide the pseudo-code in algorithm 1.

$$\mathcal{H}\left(S, h^*, r\right) = \left\{h \in \mathcal{H} \ : \ d\left(h(S), h^*(S)\right) \leq r\right\} \qquad (5)$$

Theoretically, it is still unclear how these allowed errors might influence the query complexity of r-AFA+ compared to AFA. On one hand it definitely increases the cardinal of the version space, potentially increasing the μ-diameter for a given budget, requiring a larger T at each inner iteration. On the other hand, the exit condition of the algorithm (line 19) is less restrictive and decreases the number of outer iterations (and thus the total number of queries in S). As this is still preliminary work, we leave a more in-depth analysis of r-AFA+ for future work, and discuss the empirical results that follow from this relaxation in the next section.

5 Evaluation

We now quantify the impact of relaxing AFA with a tolerance radius of r changes, with r-AFA+.

Implementation Based on the notebooks provided in the supporting material of [22] we reimplemented algorithm 1 (with our modifications discussed in section 4). The implementation differs from the pseudocode in two ways. First, we modify the termination of the algorithm. The joint requirements for termination of estimated μ-diameter smaller that ϵ (line 10) and surrogate/API agreement (line 19) are replaced by the condition that $|S|$ does not exceed the budget. Second, instead of querying all the points in T (line 17) we only query one of them and re-enter the inner loop.

Algorithm 1 Remote black-box certification with r-AFA+

Require: Hypothesis class $\mathcal{H}$, mistake M-bounded oracle $\mathcal{O}$, target error ϵ, property μ, confidence δ, radius r

Ensure: audit dataset S

1: $S \leftarrow \emptyset$
2: **while** true **do**
3: $\quad T \leftarrow \emptyset$
4: $\quad \hat{h} \leftarrow \mathcal{O}(S)$
5: $\quad w(x) \leftarrow \frac{1}{|\mathcal{X}|}, \ \forall x \in \mathcal{X}$
6: $\quad \tau(x) \sim \text{Exp}\left(\ln\left(\frac{M}{\delta}\,|\mathcal{H}|^2\right)\right)$
7: $\quad$ **while** true **do**
8: $\quad\quad$ ▷ *Estimate the μ-diameter of the current version space* ◁
9: $\quad\quad (h_{\min}, h_{\max}) \leftarrow \arg\min / \max_h \mu(h)$ s.t. $d(h(T), \hat{h}(T)) \leq r$
10: $\quad\quad$ **if** $\mu(h_{\max}) - \mu(h_{\min}) < \epsilon$ **then**
11: $\quad\quad\quad$ **break**
12: $\quad\quad \Delta\left(h_{\max}, h_{\min}\right) = \left\{x \in \mathcal{X} \ : \ h_{\max}(x) \neq \hat{h}(x) \text{ or } h_{\min}(x) \neq \hat{h}(x)\right\}$
13: $\quad\quad$ ▷ *Multiplicative weight update* ◁
14: $\quad\quad$ **while** $\sum_{x \in \Delta(h_{\max}, h_{\min})} w(x) \leq 1$ **do**
15: $\quad\quad\quad w(x) \leftarrow 2\, w(x), \ \forall x \in \Delta\left(h_{\max}, h_{\min}\right)$
16: $\quad\quad T \leftarrow \{x \in \mathcal{X} \ : \ w(x) \geq \tau(x)\}$
17: $\quad$ query h^* on T
18: $\quad S \leftarrow S \cup T$
19: $\quad$ **if** $\hat{h} \in \mathcal{H}\left(h^\star, S, r\right)$ **then**
20: $\quad\quad$ **return** S

Dataset We run our experiments on three datasets : student performance [6], COMPAS [12] and the reconstructed adults dataset [9]. In this preliminary version of our work, we only showcase results on the COMPAS dataset. COMPAS is a tool used by the US Department of Justice to evaluate the risk of recidivism among defendants, based on individual features such as age, gender, localization, origins amongst others. The COMPAS dataset consists in a list of 6172 defendants with their individual features and recidivism status.

Classifier model We run our experiments with multiple API hypotheses classes adapted to the classification task on tabular data: linear regression, support vector machines, decision trees and gradient boosted decision trees. Again, because it is a preliminary version of our work, we only analyse here the case of decision trees and linear classifiers, as implemented in `scikit-learn` [17]. We perform classical hyperparameter optimization with 5-fold validation to train the model behind the API.

Auditing algorithms The simplest algorithm we test is a random sampling baseline. Given a budget b, b points are uniformly sampled in $\mathcal{X}$ without replacement to form the audit dataset S. The second baseline is the AFA algorithm. Then we test our method r-AFA+ with two values of r.

Evaluation results In Figure 2, we plot the value of the μ-diameter $\text{diam}_\mu \mathcal{H}(h^*, S, 5)$ against the audit budget $|S|$. On

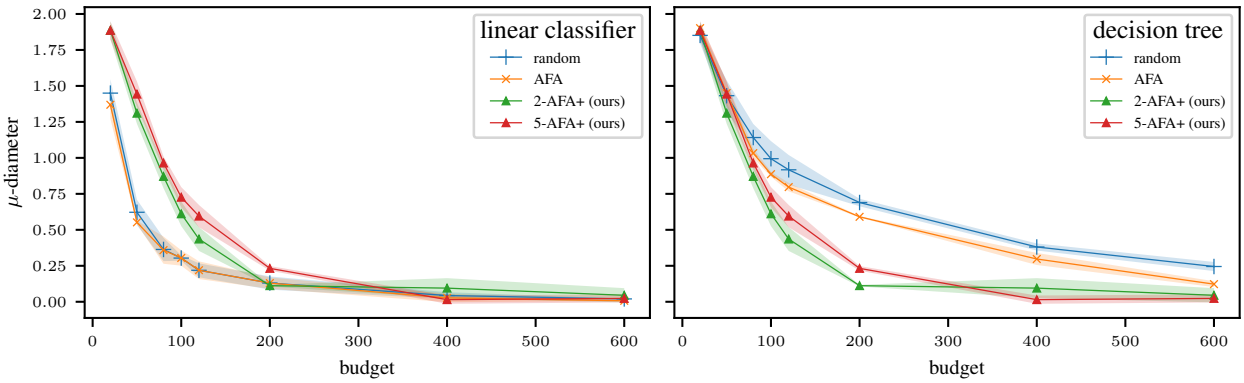


Figure 2: μ-diameter of the version space for budgets ranging from 200 to 600 requests. On the left, $\mathcal{H}$ is the set of linear classifiers. On the right, $\mathcal{H}$ is the set of decision trees. The figure compares two baselines (random sampling and original AFA) against r-AFA+ for two radius values ($r = 2$ and $r = 5$).

the right, $\mathcal{H}$ is the set of decision trees. The figure compares the two baselines (random sampling and original AFA) and our method. Note that we are evaluating the audit algorithms in the more realistic setting with $r = 5 > 0$.

In the case of linear classifiers, both our methods and the baselines tend to a null μ-radius. Our methods even reduce slightly the convergence speed of the μ-radius to 0 (2-AFA+ needs $\sim$ 50 more queries that random and AFA to reach a μ-diameter of 0.25). Yet, after 200 queries, all methods become equivalent in terms of μ-diameter. In addition, this plot highlights the performance of the simplest random baseline: it is the best performing method on this (dataset, api model) combination.

The second situation gives a totally different picture of the comparison between AFA and r-AFA+. While AFA performs as well as the random baseline, our methods allow to save up to 400 queries ($\sim$ 66%) to reach a μ-diameter of 0.25. The intuition behind the performance gap of r-AFA+ between the decision tree and linear APIs is linked to the regularity of the decision function. If the decision boundary is smoother (as is the case for linear models), two models that do not agree on a given set of points would not agree on the remaining points with a high probability. On the other hand, if the decision boundary is very irregular (as is the case for decision trees), two models that do not agree on a set of points might still be very close on the other points. Thus in this case, it seems that allowing for more disagreement (a.k.a. increasing r) between $\hat{h}$ and the μ-optimal model h in line 9 helps to include models similar to the API h^* even if $\hat{h}$ is far from it in the beginning.

The takeways from this evaluation are that i) the gains in terms of budget are highly hypothesis dependant, 2) AFA is never substantially better than random, which questions its utility (high complexity w.r.t. random selection), and 3) r-AFA+ is at least as competitive as random and AFA on the long run (i.e. for small diameters).

6 Conclusion

Being robust to slight model changes is a practical requirement to take into account the practices of deployed ML systems that often evolve. In this context, the promising auditing approach of producing one-shot certificates might frequently require auditors to re-audit the target model after each slight update. This paper explores how certificates can be designed to be robust to such modifications and presented preliminary results that support this direction.
We have empirically shown that the r-AFA+ relaxation can provide an interesting gain over AFA in one scenario, and that the random and computationnaly cheap strawman approach is also to be considered. We leave to futurework a full characterization of the model families on which these observations generalize. Futurework also includes the study of the impact of removing the assumption that the hypothesis class is known by the auditor. More precisely, allowing for a restricted hypothesis space while preserving the accuracy of audits seems like an important next step for reliable and practical audits.
As a final remark, throughout this work, we used the term "Active Auditing" coined by the authors of [22]. Yet since, this algorithm guarantees that *if* the platform does not change $\mathcal{H}$ *then* we can "easily" verify that our estimated value still holds. Thus, a more accurate term would be "active certification". This splits the goal of algorithmic auditing: trying to build certificates for platforms to defend themselves, or finding estimators that are able to uncover misconduct robust to concealment attempts from the platform.

Acknowledgements

We would like to thank Tom Yan (co-author of [22]) for the exchange we had upon implementing their algorithm.

References

[1] Sanjeev Arora, Elad Hazan, and Satyen Kale. “The Multiplicative Weights Update Method: A Meta-

Algorithm and Applications". In: *Theory of Computing* 8.6 (May 1, 2012), pp. 121–164.

[2] Solon Barocas, Moritz Hardt, and Arvind Narayanan. "Fairness in Machine Learning". In: *Nips tutorial* 1 (2017).

[3] Ja M. Barzdin. "On the Prediction of General Recursive Functions". In: *Soviet Mathematics Doklady*. Vol. 13. 1972, pp. 1224–1228.

[4] Clément L. Canonne. *A Survey on Distribution Testing: Your Data Is Big. But Is It Blue?* Graduate Surveys 9. Theory of Computing Library, Aug. 15, 2020. 100 pp.

[5] Clement L. Canonne et al. "The Price of Tolerance in Distribution Testing". In: *Proceedings of Thirty Fifth Conference on Learning Theory*. Conference on Learning Theory. PMLR, June 28, 2022, pp. 573–624.

[6] Paulo Cortez and Alice Silva. "Using Data Mining to Predict Secondary School Student Performance". In: *EUROSIS* (Jan. 1, 2008).

[7] Sanjoy Dasgupta et al. "Teaching a Black-Box Learner". In: *Proceedings of the 36th International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, May 24, 2019, pp. 1547–1555.

[8] Jeffrey Dastin. "Amazon Scraps Secret AI Recruiting Tool That Showed Bias against Women". In: *Reuters. Retail* (Oct. 10, 2018).

[9] Frances Ding et al. "Retiring Adult: New Datasets for Fair Machine Learning". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 6478–6490.

[10] Steve Hanneke. "Theory of Disagreement-Based Active Learning". In: *Foundations and Trends® in Machine Learning* 7.2-3 (June 11, 2014), pp. 131–309. ISSN: 1935-8237, 1935-8245.

[11] Russell Hotten. "Volkswagen: The Scandal Explained". In: *BBC News. Business* (Sept. 22, 2015).

[12] Jeff Larson et al. "How We Analyzed the COMPAS Recidivism Algorithm". In: *ProPublica* (May 23, 2016).

[13] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[14] J. Nathan Matias, Austin Hounsel, and Nick Feamster. "Software-Supported Audits of Decision-Making Systems: Testing Google and Facebook's Political Advertising Policies". In: *Proceedings of the ACM on Human-Computer Interaction* 6 (CSCW1 Apr. 7, 2022), 118:1–118:19.

[15] Danaë Metaxa et al. "Auditing Algorithms: Understanding Algorithmic Systems from the Outside In". In: *Foundations and Trends® in Human–Computer Interaction* 14.4 (2021), pp. 272–344. ISSN: 1551-3955, 1551-3963.

[16] Albert B. Novikoff. *On Convergence Proofs for Perceptrons*. STANFORD RESEARCH INST MENLO PARK CA, 1963.

[17] Fabian Pedregosa et al. "Scikit-Learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12.85 (2011), pp. 2825–2830. ISSN: 1533-7928.

[18] Bashir Rastegarpanah, Krishna Gummadi, and Mark Crovella. "Auditing Black-Box Prediction Models for Data Minimization Compliance". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 20621–20632.

[19] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. New York, NY, USA: Association for Computing Machinery, Aug. 13, 2016, pp. 1135–1144. ISBN: 978-1-4503-4232-2.

[20] Ali Shahin Shamsabadi et al. "Confidential-PROFITT: Confidential PROof of FaIr Training of Trees". In: The Eleventh International Conference on Learning Representations. Feb. 1, 2023.

[21] Ali Shahin Shamsabadi et al. "Washing The Unwashable : On The (Im)Possibility of Fairwashing Detection". In: Advances in Neural Information Processing Systems. Oct. 31, 2022.

[22] Tom Yan and Chicheng Zhang. "Active Fairness Auditing". In: *Proceedings of the 39th International Conference on Machine Learning*. International Conference on Machine Learning. PMLR, June 28, 2022, pp. 24929–24962.

Un modèle pour la généricité des agents conversationnels vocaux multi-domaines

Maya Medjad[1 2], Mathieu Buonomo[2], Raphaël Szymocha[2], Frédéric Armetta[1]
[1] Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, UMR5205, F-69622 Villeurbanne, France
nom.prenom@liris.cnrs.fr
[2] Reecall, reecall.com

5 mai 2023

Résumé

Les agents conversationnels vocaux sont de plus en plus utilisés pour des services automatisés d'assistance téléphonique (support, réservations, commandes, etc). Après avoir décrit la chaîne de traitement d'un agent conversationnel, nous soulignons des spécificités propres au déploiement de ces systèmes sur plusieurs domaines et proposons un modèle permettant d'étendre les possibilités de paramétrage et de généralisation à différents domaines d'activité.

Mots-clés

Traitement naturel du langage, Agent conversationnel, Apprentissage profond, Chatbot

Abstract

Conversational voice agents are increasingly used for automated phone assistance services (support, reservations, orders, etc.). After describing the processing chain of a conversational agent, we highlight specific features for the deployment of such systems over several domains and propose a model to extend the possibilities of parameterisation and generalisation to different domains of activity.

Keywords

Natural language processing, Conversationnal agent, Deep learning, Chatbot

1 Introduction

Les agents conversationnels sont de plus en plus utilisés pour leurs capacités d'interaction en langage naturel avec des utilisateurs, afin de fournir des réponses rapides, pertinentes et personnalisées dans différents domaines (suivi de commandes, etc.). Au-delà de l'automatisation, ceux-ci offrent aussi une meilleure disponibilité lors de fortes sollicitations des services, tout en permettant d'orienter l'utilisateur vers un conseiller lorsque cela est approprié.
Dans le cadre de ce travail, nous nous intéressons au déploiement d'agents conversationnels multi-domaines, qui permettent de traiter des conversations pour des domaines d'application variés. Nous soulignons dans ce travail des spécificités que nous avons identifiées propres au déploiement multi-domaine, et proposons un flux de traitement générique afin de fiabiliser le déploiement de tels systèmes.

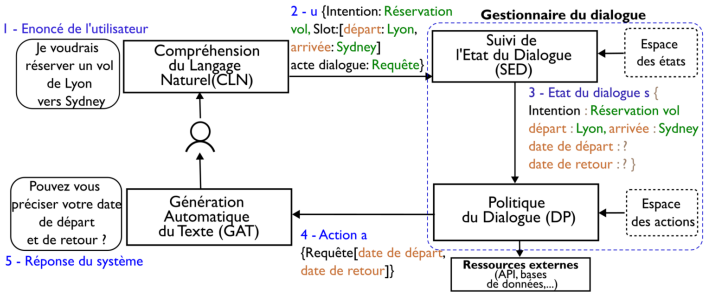


FIGURE 1 – Gestion d'un tour de conversation [1]

En section 2, nous décrivons les méthodes utilisées pour le déploiement de ces systèmes. Nous introduisons ensuite en section 3 notre proposition de modèle
pour la mise en application d'un système multi-domaines, et concluons en section 4.

2 Les agents conversationnels

La gestion de la conversation s'articule autour de différents composants qui permettent à partir d'un message de l'utilisateur formulé en langage naturel d'identifier l'intention portée par le message, pour le contextualiser au sein de la conversation et fournir une réponse adaptée à l'utilisateur.
Nous retrouvons ces composants sur la figure 1, tel que présenté par [1] :

- Le composant CLN [4] (Compréhension du Langage Naturel) permet d'identifier (souvent grâce a des réseaux de neurones) l'intention de l'utilisateur tout en extrayant des informations utiles à la conversation. Sur la figure, on identifie l'intention de réserver et certaines données utiles telles que la ville de destination.
- L'information ainsi recueillie permet au SED (Suivi de l'État du Dialogue) de contextualiser l'état de progression de la conversion, en identifiant des informations encore non renseignées telle que la date de départ.
- Le DP (Politique du Dialogue) détermine alors la prochaine action du système, tel que la demande d'informations complémentaires permettant

de compléter la transaction.
— Le composant de GAT (Génération automatique de texte) génère quant à lui une réponse appropriée en langage naturel.

Toutes ces étapes peuvent être implémentées de façon modulaire ou fusionnée selon l'approche choisie. Les systèmes de gestion de la conversation peuvent ainsi prendre plusieurs formes, avec des méthodes dites manuelles [6] (la conversation suit alors son cours en parcourant différents états et transitions pré-établies), de l'apprentissage automatique (supervisé ou par renforcement [2]) ou des modèles hybrides [7]. Les méthodes manuelles permettent d'expliciter les étapes de la conversation (données à recueillir, informations à communiquer, etc.). Les méthodes issues du deep learning requièrent beaucoup de données parfois difficiles à rassembler. Dans le domaine des systèmes de conversation orientés but on peut utiliser des approches hybrides pour contrôler le fil de la conversation, avec un paramétrage des étapes de la conversation explicite (diagramme d'états et de transitions) tout en utilisant des classifieurs pour l'identification des intentions de l'utilisateur [9].

Les données auxquelles font face ces systèmes pour identifier les intentions des utilisateurs ne sont jamais connues entièrement à l'avance. On définit les domaines d'application pour lesquels l'agent répondra, ce qui ne correspond pas est alors identifié "hors du champ" [3] (*out of scope*).

Parmi les différentes application propres à ce type de système, le besoin d'étendre le périmètre de compétence de ces agents à plusieurs domaines soulève de nombreux défis [5] tel que la disponibilité des datasets.

Dans cet article, on s'intéresse à étendre la détection des intentions de l'utilisateur à plusieurs domaines, en limitant l'introduction de classifieurs spécifiques difficiles à paramétrer.

3 Des agents conversationnels génériques

3.1 La spécificité a un coût

La société Reecall [1] propose différents services à ses clients dans des domaines variés tels que l'immobilier ou le e-commerce. Afin d'adapter les agents conversationnels suivant les besoins spécifiques des clients, la gestion du dialogue s'effectue par une modélisation à états finis des étapes de la conversation, dont les transitions sont déclenchées par les intentions de l'utilisateur identifiée par différents classifieurs (composant CLN).

Un scénario correspond alors à un échange composé de questions et d'actions entre l'agent et son interlocuteur afin de traiter sa demande. De nouveaux domaines sont régulièrement ajoutés au système, ce qui nécessite l'introduction de classifieurs spécifiques, dont l'apprentissage nécessite également différents jeux de données dédiés. La gestion de la conversation doit elle-même être adaptée à ces nouveaux domaines. L'introduction de ces classifieurs peut

1. La société Reecall est spécialisée dans la mise en place d'agents conversationnels vocaux `reecall.com`

J'ai besoin de parler avec un technicien.	Ce serait pour prendre un rendez-vous.	J'ai oublié mon mot de passe.
READ Escalation	CREATE Main Object	UPDATE Account

FIGURE 2 – Détection d'intentions primaire (en bleu) et secondaires (en vert)

être complexe et introduire des anomalies (jeux d'entraînement déséquilibrés, hors sujet (*out-of-scope*) difficiles à définir), la définition de nouvelles politiques de conversation peut également introduire des incertitudes et des délais supplémentaires pour le déploiement. Afin d'améliorer la satisfaction des utilisateurs et de fiabiliser un tel système, nous proposons ainsi un modèle plus générique qui a pour vocation de pouvoir s'appliquer et s'adapter aux différents domaines rencontrés, tout en maintenant des possibilités de spécialisation des services proposés.

3.2 Un modèle multi-domaine plus générique

Nous proposons un modèle de détection d'intentions qui s'appuie sur la détection conjointe d'intentions primaires et secondaires, tel que présenté sur la figure 2. Une interaction entre l'utilisateur et l'agent est ainsi décomposée de cette façon : 1) L'utilisateur émet une demande, 2) l'intention primaire de l'utilisateur est détectée (action à effectuer), 3) l'intention secondaire est détectée à partir de la même phrase formulée (objet concerné), 4) l'agent identifie ainsi le contexte et répond à l'utilisateur suivant la combinaison des intentions primaire et secondaire identifiées.

L'intention primaire et secondaire se complètent donc pour exprimer la demande de l'utilisateur. Les détections des deux classifieurs dédiés permettent de détecter l'entièreté de la demande exprimée par l'utilisateur (par exemple "UDPATE" et "ACCOUNT" sur la figure 2).

Si l'une des deux classifications échoue (ce qui se produit régulièrement en environnement bruité par exemple, l'échec de reconnaissance pouvant être détecté par différentes méthodes de seuils de confiance non détaillées dans cet article, une détection avec un seuil trop faible ne sera pas prise en compte), l'agent conversationnel peut tout de même déclencher un scénario adapté à l'action ou au contexte identifié. Il pourra par exemple demander des précisions sur l'objet de la mise à jour ("UPDATE" détecté), ou inversement demander des précisions sur l'opération à réaliser sur l'objet ("ACCOUNT" détecté).

En procédant ainsi, le classifieur d'intention primaire qui identifie l'action est commun et réutilisable pour tous les clients. Celui associé aux intentions secondaires est malgré tout à différencier suivant le domaine concerné ("*Main Object*" correspondra a un "rendez-vous" pour le domaine des rendez-vous, ou au "produit" pour le domaine du "e-commerce", avec des formulations spécifiques à définir pour l'apprentissage), mais pourront être utilisés pour différents agents conversationnels abordant des thèmes communs, garantissant ainsi une meilleure fiabilité. Le déroulement de la conversation pourra ainsi être défini au cas par cas suivant les combinaisons d'intentions primaires et se-

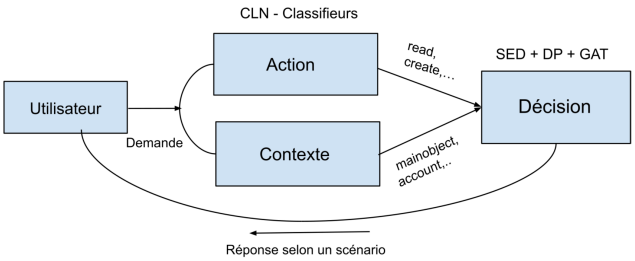


FIGURE 3 – Le système

condaires identifiées détectées au cours de la conversation.

3.3 Des actions génériques pour représenter toutes les configurations

Nous avons sélectionné les actions (ou intentions primaires) en s'inspirant du modèle CRUD [8] du domaine des bases de données, dédié à la manipulation d'objets (*create, delete, read, update*). Afin de traiter les actions inconnues, nous complétons cette liste par l'action oos (*out-of-scope*).
Concernant le Contexte (intentions secondaires), nous avons identifié sept classes (*mainobject, invoice, quote, account, payment, escalation, oos*) qui peuvent être explicitées de façon différente suivant le domaine concerné. Ce qui correspond à l'objet principal de l'échange pour un domaine sera qualifié de *mainobject*, *escalation* correspond à la mise en relation avec un opérateur humain, *invoice* concerne la facturation, etc.
Ainsi, si le système détecte "create(ip) account(is)", cela signifie que l'utilisateur souhaite créer un compte, et le scénario suivi permet alors de récupérer les informations complémentaires nécessaires. Le contexte est ainsi identifié par la combinaison de l'intention primaire et de l'intention secondaire (par domaine). Certaines applications nécessitent toutefois l'utilisation d'intentions secondaires issues de plusieurs domaines, les classifieurs dédiés aux intentions secondaires peuvent alors se compléter, et étendre ainsi les contextes qui peuvent être identifiés, et orienter le déroulement de la conversation.
Avec un tel système, nous pouvons donc définir une configuration de base stable et éprouvée, selon les besoins du client pour l'identification des intentions (module CLN), et créer ou adapter la gestion de la conversation (module SED et DP).

4 Conclusion

Nous avons présenté dans cet article une mise en application pratique pour le développement d'agents conversationnels vocaux plus génériques. En proposant un système plus générique, adaptable à différents domaines, nous souhaitons fiabiliser le paramétrage et l'agencement des différents classifieurs dédiés à l'identification des intentions de l'utilisateur. Améliorer la qualité de détection des intentions de l'utilisateur est en effet primordial pour garantir la pertinence des réponses apportées à l'utilisateur par le module conversationnel. Cette approche permet également un déploiement plus rapide, tout en gardant la possibilité de personnalisation des scénarios suivant les cas d'utilisation. Le modèle proposé s'intéresse principalement à améliorer le module de reconnaissance des intentions dans un contexte multi-domaine. Pour de futurs travaux, nous pourrons étudier l'extension de la généralisation au module conversationnel, afin d'automatiser le couplage entre la configuration utile à la reconnaissance des intentions, et le module conversationnel, pour les agents conversationnels multi-domaines.

Références

[1] Hayet Brabra, Marcos Báez, Boualem Benatallah, Walid Gaaloul, Sara Bouguelia, and Shayan Zamanirad. Dialogue management in conversational systems : a review of approaches, challenges, and opportunities. *IEEE Transactions on Cognitive and Developmental Systems*, 2021.

[2] Zhi Chen, Lu Chen, Xiang Zhou, and Kai Yu. Deep reinforcement learning for on-line dialogue state tracking. *arXiv preprint arXiv :2009.10321*, 2020.

[3] L Manik, Zaenal Akbar, Hani Febri Mustika, Ariani Indrawati, Dwi Setyo Rini, Agusdin Dharma Fefirenta, and Tutie Djarwaningsih. Out-of-scope intent detection on a knowledge-based chatbot. *International Journal of Intelligent Engineering and Systems*, 14(5) :446–457, 2021.

[4] Marjorie McShane. Natural language understanding (nlu, not nlp) in cognitive systems. *AI Magazine*, 38(4) :43–56, 2017.

[5] Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. Towards scalable multi-domain conversational agents : The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696, 2020.

[6] Cameron Smith, Nigel Crook, Daniel Charlton, Johan Boye, Raul Santos De La Camara, Markku Turunen, David Benyon, Björn Gambäck, Oli Mival, Nick Webb, et al. Interaction strategies for an affective conversational agent. *Presence*, 20(5) :395–411, 2011.

[7] Kai Sun, Su Zhu, Lu Chen, Siqiu Yao, Xueyang Wu, and Kai Yu. Hybrid dialogue state tracking for real world human-to-human dialogues. In *INTERSPEECH*, pages 2060–2064, 2016.

[8] Ciprian-Octavian Truica, Florin Radulescu, Alexandru Boicea, and Ion Bucur. Performance evaluation for crud operations in asynchronously replicated document oriented database. In *2015 20th International Conference on Control Systems and Computer Science*, pages 191–196. IEEE, 2015.

[9] Jason D. Williams, Kavosh Asadi, and Geoffrey Zweig. Hybrid code networks : practical and efficient end-to-end dialog control with supervised and reinforcement learning. *CoRR*, abs/1702.03274, 2017.

Metrics for community dynamics applied to unsupervised attacks detection

J. Michel[1,2], P. Parrend[1,2]

[1] Laboratoire des sciences de l'ingénieur, de l'informatique et de l'imagerie, Icube
[2] Laboratoire de Recherche de L'EPITA , LRE

May 4th 2023

Abstract

Attack detection in big networks has become a necessity. Yet, with the ever changing threat landscape and massive amount of data to handle, network intrusion detection systems (NIDS) end up being obsolete. Different machine-learning-based solutions have been developed to answer the detection problem for data with evolving statistical distributions. However, no approach has proved to be both scalable and robust to passing time. In this paper, we propose a scalable and unsupervised approach to detect behavioral patterns without prior knowledge on the nature of attacks. For this purpose, we define novel metrics for graph community dynamics and use them as feature with unsupervised detection algorithm on the UGR'16 dataset. The proposed approach improves existing detection algorithms by 285.56% in precision and 222.82% in recall when compared to usual feature extraction (FE) using isolation forest.

Keywords

Features Engineering, Graph community metrics, Scalability, Graph representation, Unsupervised detection approach, Dynamic graphs, Attacks detection

1 Introduction

Attack detection in big networks requires processing an increasing amount of data. Furthermore, the behaviour of the data changes over time in ways that cannot be predicted. This phenomenon called concept drift renders prior existing models invalid. Thus, it has come to light that there is a need for scalable and adaptable solutions. Due to the evolving nature of attacks to detect at any given time, methods which do not use knowledge of specific attacks have gained the attention of the community [1,5]. To solve this kind of problems, unsupervised approaches for anomaly detection have been studied [8]. But unsupervised approaches, especially outlier detection algorithms have shortcomings in their inability to choose the right criteria for anomalies if the configuration scope is unconstrained. This configuration scope is controlled through hyperparameter tuning, but especially through feature engineering [3]. Only features relevant for detecting attacks of interest should be fed to the anomaly detection algorithm.

Our contributions In this paper we present graph community metrics used as features for anomaly detection and specifically applied to the detection of behaviour patterns for attack detection. $Density$ and $Externality$ show remarkable results, but they do not take into account the evolution of data over time. We therefore additionally define $Local$ and $Global\ Stability$ values as metrics for graph community dynamics. They are included in the set of candidate features for anomaly detection, and fed to anomaly detection models such as Isolation Forest. The evaluation of these metrics shows that they have a high correlation rate with specific attacks such as port scan and dos. Therefore, they are highly relevant for enhancing detection capabilities. As a first evaluation of our approach, we apply our pipeline on the UGR'16 data set [4].

2 Preliminaries

2.1 Problem definition

In this paper, we address a specific attack detection problem, namely the detection using unsupervised detection models. The objective is to loosen the dependency on labelled historical data with attacks and to better perceive novel, abnormal behaviours. We more specifically focus on the relation between anomaly detection and the characterization of these anomalies as actual attacks [6]. Anomalies are defined as rare data points or outliers in the data set, and the features of the data points are the mean to highlight those outliers. Most of the time, the available features of the data are not sufficient to properly discriminate attacks from benign data [2]. This can be explained either by the fact that the features of the dataset do not discriminate attacks more than benign data or that the attacks are better characterized not by a single feature, but by a relationship between two features : identical values (as in IP loops where IP sources and destination are identical), differences between scalars (differences of throughput), etc. Another reason for the bad discrimination of attacks is that the amount of data is too high and then most outliers are in fact only statistical anomaly.

To address this problem, we rely on machine learning outlier detection algorithms, but we make use of graph representation and community detection as a mean to extract new

features. Those features are used to better discriminate attacks from other data, and thus to increase the detection rate for considered attacks. We then evaluate which graph community metrics are features relevant to attack detection.

2.2 Metrics for graph community analysis

The following metrics have shown different definitions in the literature [7] or none. We therefore provide here explicit definitions for those metrics :

Density in community i is the chance for any node inside i to be adjacent to another given node in i. It is defined as :

c_i **:** Number of connections in i

C_i **:** Maximum number of connections in i (if i were a clique)

$$Dens_i = \frac{c_i}{C_i} \tag{1}$$

Externality is not defined in the literature. It is similar to the "expansion" found in [7] and in community i is the proportion of communication of between i and others communities compared to any communication involving nodes in i as defined as :

M **:** The number of edge with at least one vertex in i

Me **:** The number of edge with exactly one vertex in i

$$Ext_i = \frac{Me}{M} \tag{2}$$

In addition to those metrics, we define local and global stability of community in dynamic graph as a contribution :

Local stability is a ratio of similarity between the state of a community C at time $t : C_t$, and $t + 1 : C_{t+1}$ defined as :

V_t **:** Set of nodes belonging to C_t

V_{t+1} **:** Set of nodes belonging to C_{t+1}

$$Ls_i = \frac{|V_t \cap V_{t+1}| - |(V_t \cap \bar{V}_{t+1}) \cup (V_{t+1} \cap \bar{V}_t)|}{|V_t \cup V_{t+1}|} \tag{3}$$

Global stability of a community C at time n is the mean of all local stabilities between time 0 and n defined as :

$$Gs_i = \frac{\sum_{i=1}^{n} Ls_i}{n-1} \tag{4}$$

2.3 Performance of metrics extraction

Graph community metrics for dynamic graphs are extracted using sliding windows for a given time interval. Our approach shall be applicable to a real time data stream, and therefore should be scalable. In order to determine the performance of our community metrics extraction algorithm, we measure the time spent on its application on a time windowed graph and reproduce the process for increasing amounts of data (Figure 2).

3 Evaluation

3.1 Relevance of graph community to detection

Performance metrics are compared for different models using the Isolation Forest algorithm which shows the best performances on the same sample of the UGR'16 dataset [4] in Table 1. Except for the baseline which does not make use of feature selection (**FS**) and Hyper-parameter tuning (**HPT**), the same process is applied for the different models. The process is repeated 10 times and the average of each performance metrics are reported.

In addition to the comparison between the different models including those taking dynamic graphs community metrics such as stability, the attack distribution in relation to those metrics is observed.

3.2 Scalability of metrics extraction

In order to evaluate the scalability of our algorithms, we test them on samples of different size of a same week of the dataset UGR'16. The biggest sample entails about 539 millions data points extracted for one complete week on the target system. Every other sample is then an evenly distributed proportion of the complete week. The algorithm is applied on those samples and the time spent for the extraction of graph community metrics by our algorithms on each time windows of the dataset is observed.

4 Results

4.1 Relevance of graph community to detection

TABLE 1 – Performance evaluation of the chosen approach

Isolation Forest on UGR'16	F-Score	Precision	Recall
Baseline	0.00049984	0.00103121	0.00032987
Feature extraction(FE)	0.05641072	0.03454118	0.15377554
FE + Feature Selection(FS)	0.30761931	0.32660442	0.29073454
FE + FS + Hyper Parameter Tuning(HPT)	0.32155489	0.30152777	0.34444449
FE+FS+HPT+ IP Graph(5 min)	0.48283249	0.46699468	0.49990232
FE+FS+HPT+ Ip&(Ip,port) Graph (5, 10 & 20 min)	0.80421724	0.85484417	0.75928304
Previous+local stability	0.81157849	0.86104321	0.76750087

The results in Table 1 show that graph community metrics as features can significantly improve detection performances. We observe a F-score up to **0.804** with graph community metrics against **0.322** for common feature extraction. Moreover, we show that using different features, or combinations of features for node as well as extracting the community metrics for different time intervals lead to a significant improvement of performance. A **0.804 F-score** is observed, using both **IP** and **couple of IP and port** as node with **5, 10 and 20 minutes time interval** against a **0.423 F-score** using IP as node and 5 minutes time interval.

4.2 Relevance of graph community dynamics

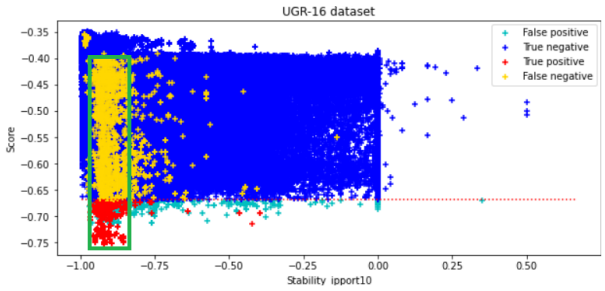


FIGURE 1 – Distribution of positives and negatives in detection with regard to global stability of communities in dynamic graphs using IP and port as node. The green box shows where most attacks are located

In Table 1, we additionally observe that our model with dynamic graph community metrics is slightly better than the one with only static graphs metrics for every performance metrics, with in particular an F-score of **0.812**. However, this model only uses local stability. Nevertheless we can observe in Figure 1 that there are still unnoticed interesting behaviours in regard to global stability

4.3 Scalability of metrics extraction

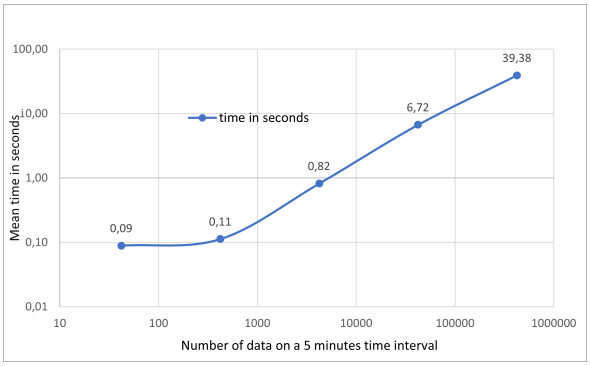


FIGURE 2 – Graph community metrics extraction time depending on the amount of data

As can be observed on figure 2, the graph community metrics extraction is sublinear in time complexity . The tendency of the time spent curve seems stable, with about 717% time increase for 900% data increase on average if we do not take the smallest sample in account. The speed-up observed of **25.52%** is assumed to be due to the amount of nodes in the graph not scaling as fast as the number of edges.

5 Conclusions

In this paper, we discussed on the importance of FE and FS when working with unsupervised detection algorithms. Graph based approaches are well suited for attack detection problems. As such, we propose two new metrics for dynamic graph communities with the local and global stability and obtained a F-score of 0.812. Along others graph community metrics, in particular density and externality, we used them as input features for anomaly detection using the Isolation Forest algorithm and obtained encouraging results using a scalable approach. However while the approach is able to detect most of scan and dos attacks, it is unable to detect the other types of attacks (nerisbotnet, spam) in the dataset. Due to the ever-changeability and diversity of attacks in the data of our application case, we do not think it is possible to be able to detect every current type of attacks or new ones that will come to be. However we hope to be able to enhance the trustability in the positive detection in the future.

Références

[1] A. Abou Rida, R. Amhaz, and P. Parrend. *Anomaly Detection on Static and Dynamic Graphs using Graph Convolutional Neural Networks, chapter -*, page 23. Studies in Computational Intelligence Series. Springer, 2022.

[2] Siddharth Bhatia, Bryan Hooi, Minji Yoon, Kijung Shin, and Christos Faloutsos. Midas : Microcluster-based detector of anomalies in edge streams. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3242–3249, 2020.

[3] Xavier Larriva-Novo, Víctor A. Villagrá, Mario Vega-Barbas, Diego Rivera, and Mario Sanz Rodrigo. An iot-focused intrusion detection system approach based on preprocessing characterization for cybersecurity datasets. *Sensors*, 21(2), 2021.

[4] Gabriel Maciá-Fernández, José Camacho, Roberto Magán-Carrión, Pedro García-Teodoro, and Roberto Therón. Ugr'16 : A new dataset for the evaluation of cyclostationarity-based network idss. *Computers & Security*, 73 :411–424, 2018.

[5] J. Navarro, A. Deruyver, and P. Parrend. A systematic survey on multi-step attack detection. *Computers and Security*, page 102, 2018.

[6] William Robertson, Giovanni Vigna, Christopher Krügel, and Richard Kemmerer. Using generalization and characterization techniques in the anomaly-based detection of web attacks. In *NDSS*, 01 2006.

[7] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, MDS '12, New York, NY, USA, 2012. Association for Computing Machinery.

[8] Tommaso Zoppi, Andrea Ceccarelli, Tommaso Capecchi, and Andrea Bondavalli. Unsupervised anomaly detectors to detect intrusions in the current threat landscape. *ACM/IMS Trans. Data Sci.*, 2(2), apr 2021.

Structural and spectral analysis of dynamic graphs for attack detection

Majed Jaber[1,2], Nicolas Boutry[2], Pierre Parrend[1,2]
[1] ICube - Laboratoire, des sciences de l'ingénieur, de l'informatique et de l'imagerie, UMR 7357, Université de Strasbourg, CNRS, 67000, Strasbourg, France
[2] Laboratoire de Recherche, de L'EPITA (LRE), 14-16 rue Voltaire, 94270 Le Kremlin-Bicêtre, France

May 5, 2023

Abstract

At this time, cyberattacks represent a constant threat. Many approaches exist for detecting suspicious behaviors, but very few of them seem to benefit from the huge potential of mathematical approaches like spectral graph analysis, which are able to extract topological features of a graph using its Laplacian spectrum. For this reason, we consider our network as a dynamic graph composed of nodes (representing the devices) and of edges (representing the requests), and we compute its Laplacian spectrum across time. Since an important change of topology inducing an important change in the spectrum, we propose new detectors using information on spectrum dynamics for detecting advanced cyberattacks. The first evaluations show that the approach is promising.

Keywords

Cybersecurity, anomaly detection, spectral graph analysis, Laplacian spectrum, graph topology.

1 Introduction

The frequency of cyberattacks has been rising dramatically over the years, with an increasing number of threat actors from nation-states and hacker groups that are getting involved in such activities. The shortage of skilled personal needed to counter these threats grew at the same time. Skill shortage leads to an increased need for automation of cybersecurity analysis, and thus more expressive and powerful models for the detection of cyberattacks. To efficiently ease cybersecurity risks, we need advanced solutions that allow us to relate and analyze connections on a practical scale. Defenders usually depend on lists : alerts and logs from software tools, and thus supporting heterogeneous data sources and formats.

Attackers can find a weakness in the network and exploit it to gain access to more devices. Graph data representation and graph analysis models grow as a promising approach to support analysis, detection and reaction capability, providing a high level of transparency with respect to the origin of alerts, and of explainability to help the security analysis react to identified malicious actions. Graphs nowadays

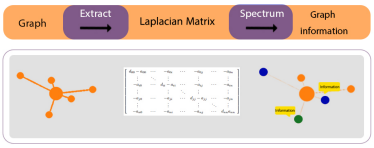


FIGURE 1 – Information extraction using spectral graph analysis

are concepts that have been widely used in various applications especially in the cybersecurity domain [1–3] Using graphs, we can cover cybersecurity patterns and detect anomalies and threats on networks [4, 5]. Usually, we model a network as a graph $G = (V, E)$ whose nodes V are the devices and whose edges E are the *i.e.* the communications between nodes.

The proposed approach in this paper involves extracting the spectrum of the Laplacian of dynamic graphs and analyzing its evolution for the purpose of detecting cyberattacks (see Fig. 1). The goal is to detect topological patterns using spectral graph analysis, to allow us to identify cyberattacks types.

2 State-of-the-Art

Let us present a brief overview of the current state-of-the-Art in matter of anomaly detection and cyberattacks. For *anomaly detection* [6], such approaches are *statistical ones* [7] and ML-based ones [8]. The shortage of labeled data in network security poses a challenge in training classifiers effectively, and the limited existing labeled data may not be applicable to other contexts as noted by [9]. However, graph-based machine learning techniques are expected to have a considerable impact on the development of next-generation cybersecurity systems. One such technique is walk-based sampling, which involves sampling graph-structured data by traversing through the graph using walks. This approach has been investigated in previous studies [10,11] that introduce the *DeepWalk* and *node2vec* methods, respectively. Deep learning has gained significant attention in the field of graph data, with Graph Convolutional

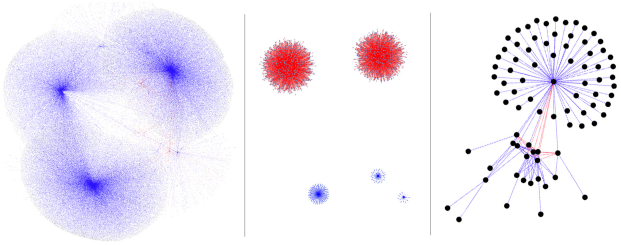


FIGURE 2 – IP-IP graph representation for Ton-IoT, IoT Healthcare Security, and Bot-IoT data sets

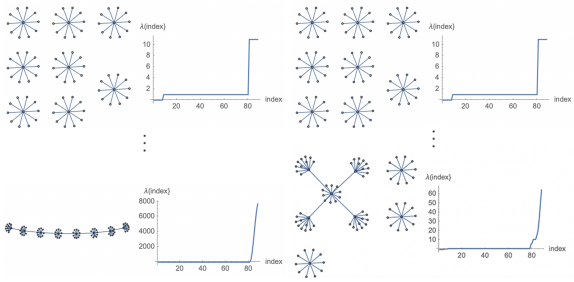


FIGURE 3 – Two different scenarios. On the left side, Scenario 1 : many interconnections depict that a threat is occurring in the network. On the right side, Scenario 2 : a « normal » scenario is depicted with no excessive connections or huge packets.

Networks [12, 13] (GCN) being the best choice for graph data learning tasks. Another approach is Graph Attention Networks [14], which utilize the self-attention mechanism to encode hidden representations of each node.

Spectral graph analysis is a useful tool for extracting topological properties from graphs [15]. The Laplacian spectrum can provide features such as the number of connected components, the bipartiteness [16], and the robustness [17] to edge rewiring. Spectral analysis has been applied to cybersecurity, such as in change detection in TCP packet transport [18], in forensic evidence analysis [19], in complexity reduction of graphs [20], in various attacks identification [21], in clustering evolving graphs [22], and in anomaly detection. Techniques like power spectral density [23], diffusion and spectral methods, dictionary learning, and hypothesis testing have been used in these approaches.

3 Proposed metrics

In order to effectively evaluate changes in the spectra resulting from different datasets Fig. 2, it is necessary to consider a range of metrics. By observing the spectrum at various timestamps and tracking the graph's evolution, it becomes possible to identify different factors that can impact the spectrum. Examples of such factors include flooding of packets, node connectivity, and degree of nodes. To facilitate the evaluation of such changes, we propose four metrics.

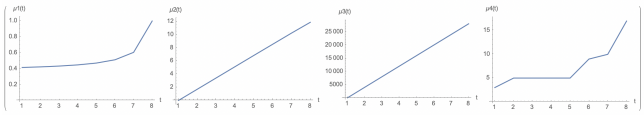


FIGURE 4 – From left to right, the four dynamic metrics μ_1 to μ_4 in Scenario 1.

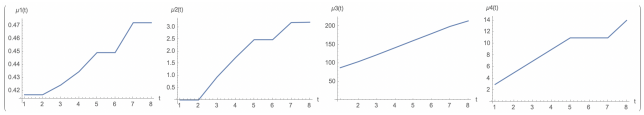


FIGURE 5 – From left to right, the four dynamic metrics μ_1 to μ_4 in Scenario 2.

The first metric, μ_1, called the *connectedness*, is based on the number of connected components in the graph. This metric is useful for evaluating the overall connectivity of the network, and can provide insights into potential areas of weakness or vulnerability.

The second metric, μ_2, called the *flood value*, takes into account both interconnections and the weights of connecting edges. This metric is designed to quantify the flooding events on the network which can have a significant impact on network performance and security.

The third metric, μ_3, called the *wiringness*, is primarily influenced by the degrees of the nodes of the graph. Nodes with a high degree of connectivity indicate potential areas of interest for further analysis.

Finally, the fourth metric, μ_4, called the *asymmetry*, measures the cardinality of identical patterns in the network. The variation of the number of patterns in the network can be a sign of potential threat.

Thanks to these metrics, it becomes possible to gain valuable insights into potential security threats or other issues impacting network performance.

4 Experiments and observations

Consider a graph G that represents a network of interconnected devices, where the nodes V represent devices and edges E represent the connections between these devices, with weights W corresponding to the amount of data transmitted. In Scenario 1 (see Figure 3), we propose to start with a graph consisting of $\mathcal{N} = 8$ separate connected components and gradually connecting the central nodes, as if a *threat* is occurring. It has been observed that most of the traffic information is stored in the first and last $\mathcal{N}$ values of the spectrum. In Scenario 2 (see Figure 3), we start with $\mathcal{N} = 8$ disconnected star graphs and gradually establish connections between non-central client nodes and central server nodes or between two central server nodes. The corresponding edge in the adjacency matrix is assigned a weight of 10 to represent the connection. This scenario is considered *normal*. Let us recall that the bigger the metric, the bigger the risk of a threat. If we observe the dynamic metrics (see Figures 4, 5) , we understand that, compared to a normal case, the threatening case is easily detected.

5 Conclusion and future works

As we have observed through many experiments (not depicted in this paper due to a lack of space), strong changes in the topology of the network due to threats imply strong variations in our metrics ; threats can then be detected more easily and in a more explainable way. The next phase of our research involves applying these metrics to real-world datasets. However, working with real-world datasets presents numerous challenges, particularly when attempting to detect changes in the behavior of large graphs. To address these challenges, we plan to apply explainable machine learning algorithms to detect threats, allowing us to provide detailed warnings to network administrators when malicious activity is detected.

Références

[1] V. Vlachos, Y. C. Stamatiou, P. Tzamalis, S. Nikoletseas, and K. Chantzi, "A social network analysis tool for uncovering cybersecurity threats," in *6th International Symposium for ICS & SCADA Cyber Security Research 2019 6*, 2019, pp. 97–106.

[2] F. Böhm, F. Menges, and G. Pernul, "Graph-based visual analytics for cyber threat intelligence," *Cybersecurity*, vol. 1, no. 1, pp. 1–19, 2018.

[3] H. Karimipour and H. Leung, "Relaxation-based anomaly detection in cyber-physical systems using ensemble kalman filter," *IET Cyber-Physical Systems : Theory & Applications*, vol. 5, no. 1, pp. 49–58, 2020.

[4] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description : a survey," *Data mining and knowledge discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[5] D. Sensarma and S. S. Sarma, "A survey on different graph based anomaly detection techniques," *Indian J Sci Technol*, vol. 8, no. 31, pp. 1–7, 2015.

[6] N. M. Adams and N. A. Heard, *Dynamic networks and cyber-security*. World Scientific, 2016, vol. 1.

[7] M. Ahmed, A. N. Mahmood, and M. R. Islam, "A survey of anomaly detection techniques in financial domain," *Future Generation Computer Systems*, vol. 55, pp. 278–288, 2016.

[8] D. K. Bhattacharyya and J. K. Kalita, *Network anomaly detection : A machine learning perspective*. Chapman and Hall/CRC, 2019.

[9] B. Bowman and H. H. Huang, "Towards next-generation cybersecurity with graph ai," *ACM SIGOPS Operating Systems Review*, vol. 55, no. 1, pp. 61–67, 2021.

[10] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk : Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.

[11] A. Grover and J. Leskovec, "node2vec : Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[12] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *Ieee Access*, vol. 5, pp. 21 954–21 961, 2017.

[13] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune : an ensemble of autoencoders for online network intrusion detection," *arXiv preprint arXiv :1802.09089*, 2018.

[14] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv :1710.10903*, 2017.

[15] F. R. Chung, *Spectral graph theory*. American Mathematical Soc., 1997, vol. 92.

[16] F. Bauer and J. Jost, "Bipartite and neighborhood graphs and the spectrum of the normalized graph laplacian," *arXiv preprint arXiv :0910.3118*, 2009.

[17] S. de Lange, M. de Reus, and M. Van Den Heuvel, "The laplacian spectrum of neural networks," *Frontiers in computational neuroscience*, vol. 7, p. 189, 2014.

[18] C.-M. Cheng, H. Kung, and K.-S. Tan, "Use of spectral analysis in defense against dos attacks," in *Global Telecommunications Conference, 2002. GLOBECOM'02. IEEE*, vol. 3. IEEE, 2002, pp. 2143–2148.

[19] W. Wang and T. E. Daniels, "Diffusion and graph spectral methods for network forensic analysis," in *Proceedings of the 2006 workshop on New security paradigms*, 2006, pp. 99–106.

[20] P.-Y. Chen, S. Choudhury, and A. O. Hero, "Multi-centrality graph spectral decompositions and their application to cyber intrusion detection," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4553–4557.

[21] X. Ying, X. Wu, and D. Barbara, "Spectrum based fraud detection in social networks," in *Proceedings of the 17th ACM conference on Computer and communications security*, 2010, pp. 747–749.

[22] C. C. Bilgin and B. Yener, "Dynamic network evolution : Models, clustering, anomaly detection," *IEEE Networks*, vol. 1, 2006.

[23] Y. Chen and K. Hwang, "Collaborative detection and filtering of shrew ddos attacks using spectral analysis," *Journal of Parallel and Distributed Computing*, vol. 66, no. 9, pp. 1137–1151, 2006.

Formalisation de la classification multi-Labels en flux et son application en cybersécurité : une étude approfondie

X. WANG[1], F. Meyer[2], P. Kuntz[3]

[1] Centre de recherche et d'innovation de Talan
[2] Orange Labs
[3] Laboratoire des sciences du numérique de Nantes

5 mai 2023

Résumé

Un nombre croissant d'applications actuelles confronte les algorithmes de classification multi-labels à un défi majeur : celui d'apprendre des modèles avec des ressources de calcul et de stockage limitées et à partir de données en flux intégrant des changements de distribution de données au fil du temps. Dans cet article, nous proposons un tour d'horizon de cette problématique avec un attention particulière pour des applications dans le domaine de la cybersécurité.

Mots-clés

Classification Multi-Labels, Flux de données, Dérive Conceptuelle, Cybersécurité.

Abstract

Due to the ever-increasing number of current applications, multi-labels classification algorithms are facing a major challenge : their capacity for learning models from streaming data including changes in distributions over time, while constantly coming up against limited computational and storage resources. In this article, we offer an in-depth analysis of this problem which will serve as a basis for a proposal of algorithms for applications in the cybersecurity domain.

Keywords

Classification Multi-Labels, Data Stream, Concept Drift.

1 Introduction

La classification multi-labels a connu un fort développement cette dernière décennie stimulé par l'essor des besoins applicatifs qui nécessitent de prendre en compte des relations de dépendance entre les labels et qui ne peuvent donc plus s'appuyer sur les algorithmes éprouvés de la classification mono-label sous l'hypothèse d'indépendance. Par exemple, pour l'annotation de textes, de nombreuses labels coexistant peuvent être liés entre eux - un article sur l'actualité concernant une conférence sur le changement climatique peut être étiqueté à la fois par les labels « politique » et « environnement »[1]. En exploitant la corrélation entre ces labels, des travaux ont montré que la classification multi-labels améliore grandement la précision de la prédiction pour la tâche d'annotation des textes. Au cours des dernières décennies, de nombreuses applications (e.g. annotation de textes [2], d'images [3] ou de menaces d'intrusion dans les réseaux [4]) ont donc stimulé la recherche sur les techniques de classification multi-labels.

Cependant, les modèles classiques sont aujourd'hui confrontés à deux défis. Leur complexité computationnelle se heurte à l'augmentation du nombre de labels pouvant être pris en compte dans certaines applications. Les jeux de données peuvent en contenir des dizaines de milliers [9]. De plus, la majorité des modèles s'appuie sur une disponibilité de l'ensemble des données pour leur traitement. Or, ces données peuvent être produites de manière continu et en grande quantité. Les ressources de calcul et de stockage pouvant être limitées par des contraintes techniques et des coûts énergétiques [6], leur traitement en grand nombre et leur stockage permanent ne sont plus toujours possible. S'ajoute à ces évolutions la nécessité de prendre en compte dans les modèles les changements de distribution des données qui peuvent apparaître au cours du temps, ce qu'on appelle la dérive conceptuelle [8].

Ces nouveaux besoins motivent la recherche sur des techniques de classification multi-labels en flux (en anglais Multi-Label Stream Classification ou MLSC) comme en témoigne les études et les travaux récents menées dans ce domaine [12]. Dans cette revue de la littérature, nous avons constaté que la recherche sur ce problème n'a fait qu'effleurer le sujet. Dans cet article, nous proposons un aperçu via une formalisation détaillée du problème de classification multi-labels en flux.

La classification multi-labels en flux revêt une importance considérable dans le domaine de la cybersécurité, notamment dans la détection d'URL malveillantes. Les URL malveillantes concernent des contenus non sollicités tels que le spam et le phishing, attirant ainsi des utilisateurs peu méfiants qui deviennent victimes d'escroqueries, subissant des pertes financières, des vols d'informations privées et des installations de logiciels malveillants. Ces activités malveillantes engendrent chaque année des pertes financières s'élevant à plusieurs milliards de euros. Un système capable de détecter rapidement ces URL malveillantes et d'appliquer les modifications nécessaires peut jouer un rôle cru-

cial dans la lutte contre un large éventail de menaces en matière de cybersécurité. Cependant, actuellement, peu de modèles se concentrent spécifiquement sur ce problèmes. C'est pourquoi, nous proposons de porter une attention particulière sur ce sujet en fin d'article.
Dans la section suivante, nous définissons le problème en détaillant les caractéristiques et les contraintes liées à la MLSC. La dernière section clôture l'article en détaillant les travaux en cours traitant du problème de détection d'URLs malveillantes.

2 Définition du problème

Nous considérons la classification supervisée multi-label comme un problème de prédiction consistant, pour un exemple donné et une collection de labels considérés, à lui associer un sous-ensemble de labels le caractérisant. Chaque exemple est associé à au moins un label. Dans le contexte de l'apprentissage en flux, nous considérons un flux de données infini où chaque exemple arrive à une grande vitesse et selon une distribution de probabilité inconnue. La tâche de l'apprentissage en flux de données consiste à générer un nouveau modèle à chaque instant à partir du modèle précédent et du nouvel exemple avant l'arrivée du prochain exemple. Le nouveau modèle doit pouvoir donner une prédiction associée aux labels pour le prochain exemple.

2.1 Caractéristiques du problème

Dans ce qui suit, nous nous concentrerons d'abord sur les caractéristiques des données multi-labels, puis nous décrirons les caractéristiques des données en flux.

Les caractéristiques de données multi-labels. Trois caractéristiques sont présentes dans les données multi-labels : la volumétrie de l'espace des labels, la présence de corrélations entre labels et le déséquilibre dans la distribution des labels [10].

- **Espace de sortie à grande dimension.** Le nombre de labels prédéterminés est noté l. Un jeu de données multi-labels peut avoir 2^l combinaisons de labels possibles. Plus le nombre de labels augmente, plus le volume de l'espace de sortie croît exponentiellement et des données peuvent se retrouver « isolées » dans l'espace de recherche. Cela est problématique pour les méthodes qui nécessitent un nombre significatif de données pour l'apprentissage.
- **La présence de corrélations entre labels.** Les labels ne sont généralement pas indépendants les uns des autres ; ils sont corrélés et peuvent apparaître conjointement à des fréquences différentes. Par exemple, les articles de journaux sont plus susceptibles d'être associés à la fois aux catégories "science" et "environnement" qu'aux catégories "environnement" et "sport". Dans une base de données de films, les labels "famille" et "guerre" n'apparaîtront sans doute que très rarement ensemble.
- **Le déséquilibre dans la distribution des labels.** Des déséquilibres peuvent apparaître à deux niveaux : dans la distribution marginale des labels ou dans leurs distributions jointes. Par exemple pour un jeu d'images d'animaux domestiques, la combinaison de labels "chat" et "chien" apparaît plus souvent que la combinaison de labels, "chat" et "serpent" ; de même le label "chat" peut apparaître beaucoup plus souvent que le label "serpent".

Les caractéristiques de données en flux. Dans le contexte du problème en flux, outre les caractères soutenu et sans fin de la génération des données à traiter, une autre caractéristique majeure est **la dérive conceptuelle** : la distribution de données peut évoluer d'une manière imprévue sur le flux [11]. Ce phénomène peut se produire à la suite de changements dans l'environnement. Un exemple typique est la manière dont les périodes de croissance des cultures sont actuellement modifiées en réponse au changement climatique. Un autre exemple est celui en cybersécurité, où les contenus des URLs malveillantes changent constamment et deviennent plus difficiles à identifier à mesure que les techniques évoluent.

Les données étant multi-labels, il convient également de noter que la distribution des labels change en permanence. Par exemple, les principaux sujets d'actualité que les gens suivent changent d'un jour à l'autre : la fréquence du label "économie" augmente considérablement pendant la période d'inflation et la fréquence du label "féminisme" augmente considérablement autour du 8 mars. Non seulement la distribution marginale des labels peut changer, mais la distribution jointe des labels peut également changer. Par exemple, lorsque le climat change, la relation entre la politique et l'environnement est plus étroite que la relation entre la politique et l'économie.

2.2 Contraintes du problème

En réalité, la quantité de mémoire disponible pour le stockage des données et la capacité de calcul des algorithmes sont limitées par le matériel physique. Dans le même temps, les données qu'ils doivent traiter sont continuellement générées dans de nombreuses applications. A titre illustratif, chaque seconde, des milliers de requêtes URLs sont réalisées sur internet et 300 heures de vidéo par minute sont mises en ligne sur Youtube.

Par conséquent, afin de répondre aux exigences posées par les contraintes imposées par le MLSC, un algorithme doit non seulement être capable d'apprendre chaque donnée entrante en utilisant une quantité limitée de mémoire, mais en même temps son temps d'apprentissage doit être contrôlé dans un intervalle de temps très limité. De plus, les modèles doivent non seulement accumuler les informations au cours du temps lorsque la distribution de donnée est stable afin de prédire plus précisément, mais aussi s'adapter rapidement lorsque une dérive conceptuelle se produit.

3 Travaux antérieurs

La formalisation du problème MLSC permet de cerner les caractéristiques et les contraintes du problèmme de détection des URLs malveillantes dans le domaine de la cybersécurité.

Les URLs (*Uniform Resource Locator*) sont utilisées pour référencer des ressources sur Internet. Chaque URL possède une structure spécifique (e.g, le protocole, le domaine, le port, le chemin, etc). Les attaquants tentent souvent de modifier un ou plusieurs éléments de la structure de l'URL pour inciter les utilisateurs à accéder à des URL malveillantes. Ces URLs malveillantes redirigent les utilisateurs vers des ressources ou des pages où le pirate peut exécuter du code sur l'ordinateur de l'utilisateur, rediriger les utilisateurs vers des sites web indésirables, des sites web malveillants ou d'autres sites d'hameçonnage, ou télécharger des logiciels malveillants [13]. Ces URLs menant vers des sites web malveillants sont une menace courante et sérieuse pour la cybersécurité.

Ces attaques informatiques sont parfois de grande ampleur et se propagent très rapidement. Si un type d'attaque se prépare, plusieurs URLs et plusieurs sites peuvent être concernés. Des alertes peuvent être générées par des organismes de lutte contre les cyberattaques, mais ces alertes ne concernent qu'une portion des URLs malveillantes. L'enjeu est donc alors d'apprendre très rapidement les motifs discriminants associés à ces URLs, pour être capable de reconnaître et d'alerter dans les instants qui suivent toute URL potentiellement liée à une attaque en cours.

Pour ce type d'application, un mécanisme d'apprentissage en flux peut s'avérer être pertinent. En effet la capacité de réactivité et de généralisation en temps réel, à partir de quelques exemples d'URLs dangereuses, aux autres URLs dynamiquement générées correspondant à la même attaque est cruciale. Un système qui apprend d'une manière traditionnelle, chaque semaine ou même chaque nuit n'est pas assez réactif et peut passer à côté d'URLs malveillantes qui ont plusieurs heures ou plusieurs jours pour se propager puis être consultées par les clients. Par ailleurs la qualification des types de menaces, non exclusive, est importante ; certaines menaces sont corrélées et pourraient être mieux identifiées via une analyse multi-labels. D'autres menaces peuvent être moins importantes, ou correspondre à une « zone grise » où le blocage systématique doit être remplacé par un message indiquant aux utilisateurs de prendre des précautions.

4 Conclusion

Dans cet article, nous avons présenté une formalisation des caractéristiques et des contraintes de la classification multi-labels en flux. Nous avons mis en évidence la nécessité pour les modèles de prendre en compte la grande dimension de l'espace de sortie, la présence de corrélations entre les labels, le déséquilibre dans la distribution des labels, ainsi que l'adaptation des modèles aux changements de la distribution de données avec des ressources limitées. Cette analyse permet d'avoir une vision plus globale de l'enjeu du problème et contribue à orienter le développement de nouvelles approches pour aborder ce problème complexe. Enfin, nous avons complété l'article par une ouverture sur les applications spécifiques au domaine de la cybersécurité, et plus spécifiquement dans la détection d'URL malveillantes. En perspective, nous nous intéresserons plus spécifiquement à la représentation à utiliser pour structurer des URLs et aux modèles de classification multi-labels en flux pour répondre à cette problèmatique.

Références

[1] Lang, Ken. "Newsweeder : Learning to filter netnews." Machine learning proceedings 1995. Morgan Kaufmann, 1995. 331-339.

[2] Chalkidis, Ilias, et al. Large-scale multi-label text classification on EU legislation. arXiv preprint arXiv :1906.02192 (2019).

[3] Liu, Yang, et al. "SVM based multi-label learning with missing labels for image annotation." *Pattern Recognition* 78 (2018) : 307-317.

[4] Almusawi, A. and Amintoosi, H. (2018). DNS tunneling detection method based on multilabel support vector machine. *Security and Communication Networks*, 2018, 1-9.

[5] Tarekegn, A. N., et al. (2021). A review of methods for imbalanced multi-label classification. *Pattern Recognition*, 118, 107965.

[6] Domingos, Pedro, and Geoff Hulten. Mining high-speed data streams. *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2000.

[7] Zheng, Xiulin, et al. A survey on multi-label data stream classification. *IEEE Access* 8 (2019) : 1249-1275.

[8] Gama, João, et al. A survey on concept drift adaptation. *ACM computing surveys (CSUR)* 46.4 (2014) : 1-37.

[9] Weston, Jason, Samy Bengio, and Nicolas Usunier. Wsabie : Scaling up to large vocabulary image annotation. (2011).

[10] Tsoumakas, Grigorios, and Ioannis Katakis. Multi-label classification : An overview. *International Journal of Data Warehousing and Mining (IJDWM)* 3.3 (2007) : 1-13.

[11] Lu, Jie, et al. Learning under concept drift : A review. *IEEE transactions on knowledge and data engineering* 31.12 (2018) : 2346-2363.

[12] Zheng, Xiulin, et al. "A survey on multi-label data stream classification." IEEE Access 8 (2019) : 1249-1275.

[13] Do Xuan, Cho, Hoa Dinh Nguyen, and Victor Nikolaevich Tisenko. "Malicious URL detection based on machine learning." International Journal of Advanced Computer Science and Applications 11.1 (2020).

De l'Organisation des Systèmes Multi-Agents de Cyber-defense

Julien Soulé[1,2], Jean-Paul Jamont[2], Michel Occello[2], Paul Théron[3], Louis-Marie Traonouez[2]

[1] Thales Land and Air Systems, BU IAS, Rennes, France
[2] Univ. Grenoble Alpe, Grenoble INP, LCIS, Valence, France
[3] AICA IWG, La Guillermie, France

{julien.soule, jean-paul.jamont, michel.occello}@lcis.grenoble-inp.fr
paul.theron@orange.fr
louis-marie.traonouez@thalesgroup.com

Résumé

Cet article présente un travail de thèse s'intéressant aux systèmes multi-agents de cyber-défense vus comme un ensemble d'entités autonomes coopérantes et déployables au plus près des points sensibles d'un environnement en réseau. L'aspect organisationnel de ces systèmes est central dans la prise en compte des besoins de cyber-défense. La modélisation proposée dans l'article fournit un cadre d'étude pour appréhender son impact dans un environnement de déploiement attaquable.

Mots-clés

cyber-défense, système multi-agents, organisation

Abstract

This article presents a PhD work focusing on multi-agent cyber-defense systems seen as a set of cooperating and deployable autonomous entities as close as possible to the sensitive points of a networked environment. The organizational aspect of these systems is central to take cyber defense needs into account. The modeling proposed in the article provides a study framework to apprehend its impact in an attackable deployment environment.

Keywords

cyber-defense, multi-agent system, organization

1 Introduction

Le développement de l'« Internet of Things » et de l'« Internet of Battle Things » a entrainé une augmentation de la surface d'attaque des systèmes en réseau. Tenant compte de ce contexte, le groupe de travail « AICA IWG » [1] développe des travaux sur les agents AICA (Autonomous Intelligent Cyber-defence Agent). Un agent est par définition une entité autonome capable de percevoir son environnement local grâce à des capteurs, et d'agir sur cet environnement à l'aide d'effecteurs [7]. L'agent AICA doit pouvoir être déployé sur un système hôte pour détecter, identifier et caractériser des anomalies/attaques, élaborer et piloter l'exécution de contre-mesures et dialoguer avec l'extérieur. À cette fin, il est conçu comme proactif, discret et capable d'apprendre. L'agent AICA peut être conçu comme un Système Multi-Agent (SMA). Le paradigme multi-agent offre des moyens de gérer l'ouverture, le passage à l'échelle et l'autonomie du système hôte en déléguant différents aspects de la cyber-défense à différents agents. L'agent AICA est alors un système collectif décentralisé et distribué d'agents cyber-défenseurs déployés au plus près des composants du système [5].

Notre problématique consiste à définir l'organisation du SMA qui permettrait de répondre à des besoins de cyber-défense compte tenu des contraintes d'un environnement de déploiement en réseau.

2 SMA de Cyber-défense

Nous appelons **cyber-défense** l'ensemble des activités entreprises lorsqu'une cyber-attaque est détectée et qu'il est nécessaire de réagir [10]. Ces activités sont décrites dans le cadre du « P3R3 Resilience Engineering Framework » [9] et sont regroupées en trois fonctions de cyber-défense :

R1 - Detect and alarm : détection des cyber-attaques et déclenchement des mécanismes de réponse ;

R2 - Respond and restore : mise en œuvre et suivi des réponses apportées aux cyber-attaques et à la restauration des niveaux de services/activités minimaux. La gestion de la crise provoquée par l'attaque est au cœur de cette fonction ;

R3 - Recover and rebound : rétablissement des parties endommagées du système à défendre et traitement final des conséquences. Ce point inclut une phase d'apprentissage permettant l'amélioration du système de cyber-défense.

Nous nommons **objectifs de cyber-défense**, tous les objectifs impliquant la mise en œuvre d'une ou plusieurs des fonctions de cyber-défense.

Dans un **SMA de cyber-défense**, plusieurs agents atteignent un objectif global de cyber-défense par le comportement collectif résultant de la réalisation de sous-objectifs individuels et/ou de mécanismes locaux [4]. Des exemples de tels sous-objectifs pourraient être la détection des intru-

1. Ce groupe de travail (voir `https://www.aica-iwg.org/`) s'appuie sur les résultats du *Research Task Group IST-152* de l'OTAN qui a travaillé sur le concept des « Intelligent, Autonomous and Trusted Agents for Cyber Defense and Resilience ».

sions, la mise en œuvre d'un plan de récupération, la restauration d'une image, la redirection des ports…

Prenant appui sur un rapprochement des notions de SMA et cyber-défense dans la littérature, nous avons considéré chacun des travaux selon les fonctions de cyber-défense qu'il couvre. Nous avons constaté que la plupart des objectifs de cyber-défense des SMA se concentrent principalement sur la détection d'anomalies et d'intrusions (plus de 50% des travaux de notre revue complète se focalisent sur la fonction R1).

Pour chacun de ces mêmes travaux, nous nous sommes aussi intéressés aux caractéristiques principales de l'organisation et de l'environnement de déploiement. Nous constatons qu'indépendamment des objectifs de cyber-défense, l'organisation centralisée et/ou hiérarchique est la plus répandue parmi les SMA de cyber-défense étudiés.

La centralisation des données acquises de l'environnement, en un seul point, favorise de meilleures performances pour l'analyse de la situation globale et le contrôle du système de cyber-défense. Ces types d'organisation semblent moins facilement s'appliquer pour des réseaux dynamiques, mais sont répandus sur des systèmes de taille moyenne avec des contraintes connues [11].

Les organisations alternatives identifiées comme décentralisées prennent davantage en compte l'incertitude des cyber-attaques en laissant l'autonomie aux agents de s'organiser sans atteindre d'organisation définies a priori. Cependant, elles restent peu établies en tant que solution de cyber-défense.

Cette revue a permis d'identifier de premiers mécanismes sous-jacents à un SMA de cyber-défense. Cependant, la diversité (des objectifs, des environnements, des architectures d'agents, des protocoles d'interaction…) des SMA de cyber-défense disponibles rend l'appréciation générale des organisations difficile sans cadre commun. Il apparaît nécessaire d'avoir un modèle permettant de modéliser le système hôte sur lequel sont déployés des agents attaquants et défenseurs.

3 Établissement d'une modélisation

Environnement de déploiement des agents Reprenant un cas d'usage de l'AICA [10], nous nous intéressons à un environnement réseau constitué de *nœuds* sur lesquels des *agents* d'attaque et de défense peuvent être déployés pour les observer et agir. Ces nœuds peuvent être décrits par un ensemble de *propriétés* liées aux processus, au système de fichier, au système d'exploitation, à l'architecture matérielle, etc. Les *observations* et *actions* des agents sont conditionnées par leurs propriétés propres (dont les propriétés connues par eux) et une éventuelle non-certitude. Par exemple, la lecture d'un fichier donné ou une redirection des ports peut nécessiter un niveau de privilège élevé ; ou encore, la réception de données d'un capteur physique n'est pas assurée en tout temps. Chaque agent appliquant une/des action(s), modifie les propriétés d'un ou plusieurs nœuds. Cela change l'état de l'environnement induisant un éloignement/rapprochement des agents de leur(s) objectif(s).

Vers une modélisation de l'environnement Les caractéristiques de cet environnement de déploiement l'inscrivent comme un cas spécifique d'un « Partially Observable Stochastic Game » (POSG) et plus spécifiquement d'un « Decentralized Partially Observable Markov Decision Process » (Dec-POMDP). Les POSGs et les Dec-POMDPs sont tous les deux des cadres de modélisation mathématique de problèmes de prise de décision dans lesquels les agents interagissent entre eux et dans un environnement stochastique [2]. Dans un POSG, un groupe d'agents interagit avec un environnement stochastique et partiellement observable. Chaque agent agit en fonction de ses propres observations et d'une politique locale. Les agents peuvent avoir des objectifs différents et le jeu est généralement supposé non coopératif [8]. Dans un Dec-POMDP, les agents doivent coordonner leurs actions pour atteindre un objectif commun en étant capables de communiquer [3].

Modélisation Dec-POMDP Notre modèle Dec-POMDP intègre la notion de propriétés de nœud modifiables par des actions qu'appliquent les agents. Adoptant le jeu séquentiel simple du modèle « Agent Environment Cycle » [8], dans notre modèle, chaque itération se déroule de la façon suivante : i) Un agent choisit une action à partir des observations précédentes (propriétés connues) selon une fonction de comportement ; ii) L'environnement est mis à jour par une fonction de transition dépendant de l'état précédent et de l'action prise par l'agent (changement de propriétés une fois la pré-condition satisfaite) ; iii) Une observation est renvoyée à l'agent en se basant sur l'état actuel (propriétés connues de l'agent) et l'action associée selon une fonction d'observation. Une récompense basée sur l'évaluation des métriques recueillies pour cet état courant est également envoyée à l'agent.

Nous posons les éléments relatifs aux propriétés des nœuds, agents et actions de l'environnement suivants :

$Ag = \{ag_1, .., ag_{|Ag|}\}$: L'ensemble des agents

$P_j = \{p_1, .., p_{|P_j|}\}$: L'ensemble des propriétés du nœud j ($j \in N$). Par exemple, les identifiants des processus en cours d'exécution, les fichiers disponibles dans un dossier, le type de système d'exploitation, etc.

$P = \{P_1, .., P_{|P|}\}$: L'ensemble des propriétés de tous les nœuds.

$Kb : P \times Ag \rightarrow P_Ag, P_{Ag} \subset P$: Donne les propriétés connues par un agent.

$Action : \mathcal{P}(P) \rightarrow P$: L'ensemble des relations qui associent une pré-condition de propriétés à un à un ensemble de propriétés nouvelles. Par exemple, les propriétés « l'agent X est root », « l'agent X accède à Vim » et « l'agent X connaît l'emplacement du fichier .bashrc » forment une pré-condition pour y associer les propriétés précédentes en plus de la propriété « fichier .bashrc est modifié par agent X ».

$Metrics : P \rightarrow \mathbb{R}^n$: Donne les métriques associées à un ensemble de propriétés. Par exemple, le nombre de nœuds encore actifs, le nombre de déplacements latéraux, etc.

Reprenant la description formelle d'un Dec-POMDP [6], nous proposons le modèle suivant :

$S = \{s_1, ..s_{|S|}, s_i \subseteq P\}$: L'espace des ensembles de propriétés possibles.
$A_i = \{a_i^1, .., a_i^{|A_i|}\}$ $with$ $a_i^k \in Action$ $(with\ k \in 1, .., |A_i|)$: L'ensemble des actions pour l'agent i.
$T(s, a, s') = \mathbb{P}(s'|s, a)$: La probabilité de transition d'un état ; et $\mathbb{P}(s'|s, a) = 0$ si s' ne satisfait pas la pré-condition de a.
$R : S \times A \to N = Eval \circ Metrics \circ Next$: La fonction de récompense avec $Eval : \mathbb{R}^n \to \mathbb{R}$, associant les métriques à une récompense ; et $Next : S \times A \to S$, donnant l'état induit par une action.
$\Omega_i \subset Im(Kb) \subset P$: L'ensemble des observations pour l'agent i. Par exemple, le contenu d'un fichier, la sortie de logs d'une commande, le résultat d'un scan des ports, etc.
$O(s', a, o) = \mathbb{P}(o|s', a)$: La probabilité qu'un agent observe un ensemble de propriétés. Avec $\mathbb{P}(o|s', a) = 1$ si l'état s' contient les propriétés de o. Par exemple, un agent joue l'action « l'agent X lit un fichier de log", il résulte un nouvel état dont une propriété appartenant à la connaissance de l'agent X est « le contenu du fichier de log 'abc' est connu de l'agent X". L'observation « le contenu du fichier de log est 'abc' » sera donc retourné à l'agent X.

4 Vers une implémentation

Parmi les simulateurs que nous avons identifiés pour implémenter le modèle Dec-POMDP, aucun ne permet de couvrir à la fois la prise en compte d'un environnement cyber multi-agent selon le modèle Dec-POMDP et le besoin d'accessibilité du code (code ouvert) permettant de façon simple l'implémentation des agents attaquants et défenseurs. Cependant, nous avons identifié le framework Python « PettingZoo », conçu pour permettre l'implémentation d'un Dec-POMDP [8]. Il fournit un framework où le concepteur dispose d'outils pour faciliter la mise en place de l'espace des observations, des actions, de la gestion des agents à chaque tour et des récompenses associées [8].
Le développement de notre modèle avec PettingZoo, permet de proposer le simulateur « Multi Cyber Agent Simulator » (MCAS) [1]. Un aperçu de l'interface de ce simulateur est présenté Figure 1. En l'état actuel du développement, ce simulateur permet de charger/sauvegarder un environnement, de lancer l'exécution des agents de cet environnement en mode tour par tour via le terminal (en bas à droite de la Figure 1). Il permet aussi d'afficher les propriétés des nœuds de l'environnement sous format *json* (partie gauche de la Figure 1) et visualiser l'environnement sous forme d'un graphe et l'affichage des métriques.

5 Conclusion

Un SMA de cyber-défense déployé sur un système hôte en réseau permettrait de relever les défis liés à la complexité et la rapidité de cyber-attaques. Une première étude bibliographique donne un aperçu des liens entre l'environnement de déploiement, les objectifs et l'organisation adoptée par le concepteur du SMA de cyber-défense. Montrant des limites pour une compréhension générale, nous proposons une modélisation sous la forme d'un Dec-POMDP fournissant un cadre théorique général pour notre problématique et bénéficiant de plusieurs approches algorithmiques de résolution établies. La mise en œuvre de ce modèle prend la forme d'un simulateur pensé pour être utilisé dans le futur avec un protocole expérimental visant à évaluer et tirer des recommandations sur l'organisation d'un SMA de cyber-défense.

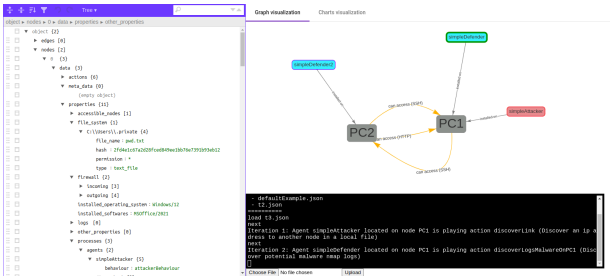


FIGURE 1 – *Aperçu de l'interface du simulateur*

Références

[1] Multi cyber agent simulator. `https://github.com/julien6/MCAS`. Accessed : 2023-03-07.

[2] Beynier, Aurélie et al. DEC-MDP / DEC-POMDP. In *Markov Decision Processes in Artificial Intelligence*, pages 277–313. 2010.

[3] Daniel S. Bernstein et al. The complexity of decentralized control of markov decision processes. *CoRR*, abs/1301.3836, 2013.

[4] J.-P. Jamont and M. Occello. Meeting the challenges of decentralised embedded applications using multi-agent systems. int. journal of agent-oriented software engineering 5 (1), 22–68, 2015.

[5] A. Kott and P. Théron. Doers, not watchers : Intelligent autonomous agents are a path to cyber resilience. *IEEE Secur. Priv.*, 18(3) :62–66, 2020.

[6] F. A. Oliehoek and C. Amato. *A Concise Introduction to Decentralized POMDPs*. Springer Briefs in Intelligent Systems. Springer, 2016.

[7] S. Russell and P. Norvig. A modern, agent-oriented approach to introductory artificial intelligence. *Acm Sigart Bulletin*, 6(2) :24–26, 1995.

[8] Terry, J. K et al. Pettingzoo : Gym for multi-agent reinforcement learning, 2020.

[9] P. Theron. Ict resilience as dynamic process and cumulative aptitude. *Critical Information Infrastructure Protection and Resilience in the ICT Sector*, 3 :1–35, 01 2013.

[10] P. Theron, N. Evans, M. Drasar, and A. Guarino. Autonomous Intelligent Cyber Defence Agent Prototype 2021 - Project Report, Dec. 2021.

[11] E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, and M. Fischer. Taxonomy and survey of collaborative intrusion detection. *ACM Computing Surveys (CSUR)*, 47(4) :1–33, 2015.

Dans quelle mesure les simulations informatiques de l'activité humaine sont-elles réalistes ?

J-B. Ly[1,2], Q. Reynaud[3], C. Le Bail[1], M. Schumann[2], V. Boccara[1], N. Sabouret[1],

[1] Université Paris-Saclay, CNRS, LISN, France
[2] EDF R&D, France
[3] QRCI, France

Résumé

Cet article s'intéresse à la notion de réalisme des simulations informatiques de l'activité humaine. Nous proposons une liste de propriétés de l'activité humaine basée sur l'ergonomie comme base pour qualifier le réalisme de ces simulations. Nous proposons une critique des approches informatiques de simulation de l'activité humaine à partir de ces propriétés. Nous proposons d'étudier ces propriétés sous le prisme des données afin d'identifier celles qui contribuent au réalisme de la simulation de l'activité humaine. Enfin, nous proposons un début d'approche hybride prenant en compte des facteurs impactant l'activité humaine, permettant de dépasser les limites des autres approches de l'état de l'art.

Mots-clés

Simulation de l'activité humaine, réalisme d'une simulation, ergonomie.

Abstract

This article focuses on the notion of realism of computer simulations of human activity. We propose a list of properties of human activity based on ergonomics as a basis to qualify the realism of these simulations. We propose a critique of computer approaches to simulating human activity based on these properties. We propose to study these properties under the prism of the data to identify those which contribute to the realism of the simulation of human activity. Finally, we propose the beginning of a hybrid approach considering factors impacting human activity, allowing to overcome the limits of other approaches of the state of the art.

Keywords

Human activity simulation, realistic simulation, ergonomics.

1 Introduction

En informatique, la simulation consiste à « reproduire artificiellement [...] un système, un phénomène, à l'aide [...] d'un programme informatique [...] » [1]. La simulation informatique de l'activité humaine est utilisée à différentes fins dont deux que nous retenons particulièrement : 1) prédire et analyser un phénomène dépendant de l'activité humaine sur un temps long (*e.g.*, la consommation d'énergie) [1, 41, 35], 2) analyser des phénomènes difficilement reproductibles en conditions réelles comme l'évacuation d'un bâtiment lors d'un incendie [34] ou la propagation de virus due aux interactions sociales [14]. La simulation informatique de l'activité permet d'étudier ces phénomènes pour se préparer aux crises. Les interactions entre les individus ainsi qu'entre ces derniers et leur environnement sont simulées. Quel que soit l'usage ou la finalité de la simulation, respecter les propriétés de l'activité humaine dans la simulation est incontournable pour tirer des conclusions fiables et de qualité. Ceci nécessite toutefois de se doter d'une définition robuste de ce qu'on entend par le réalisme d'une simulation informatique de l'activité humaine.

Pour répondre à cette question de recherche, nous proposons d'investiguer les apports de l'ergonomie, dont un des concepts centraux est celui d'activité humaine. Selon l'International Ergonomics Association, "L'ergonomie (ou les facteurs humains) est la discipline scientifique qui vise la compréhension fondamentale des interactions entre les humains et les autres composantes d'un système [...]. L'ergonomie est une discipline orientée vers les systèmes, qui s'étend aujourd'hui à tous les aspects de l'activité humaine." [2].

Tout d'abord, nous présentons les apports de l'ergonomie pour appréhender les propriétés de l'activité humaine. Puis nous présentons différentes approches informatiques pour simuler l'activité humaine. Ensuite nous discutons des limites des différentes approches informatiques qui visent à simuler les propriétés de l'activité humaine, et des différents types de données alimentant ces modèles de simulation informatique. Enfin, nous proposons une nouvelle approche permettant de dépasser les limites identifiées en couplant un modèle informatique hybride d'activité humaine avec une démarche de recueil de données.

1. https://www.cnrtl.fr

2. https://iea.cc/

2 L'activité humaine en ergonomie

Comment définir l'activité humaine ? Pour répondre à cette interrogation, nous allons identifier des propriétés de l'activité humaine telles qu'elles sont appréhendées en ergonomie, en vue de définir ce que pourrait être le réalisme de la simulation informatique de l'activité humaine.

2.1 L'activité humaine

2.1.1 Différentes visions de l'activité humaine

L'activité définie dans [13] possède les propriétés suivantes :

- « L'activité est saisie au travers des comportements "visibles" » ([[13], p. 48]). L'activité et le comportement se distinguent : le comportement est "ce qu'on voit" [25]. Les gestes, les postures, les regards et les communications entre individus sont des exemples [13]. Le comportement n'est qu'une partie visible de l'activité.
- L'activité peut être expliquée par des logiques d'action. Autrement dit, les enchaînements de décision ne sont pas anodins et sont sous-tendus par des raisonnements et, plus largement, des processus cognitifs.
- Les humains ne sont pas impassibles. Leurs processus cognitifs sont impactés par leur "vécu", leurs expériences et leurs émotions.
- Les humains sont des êtres biologiques. Leur activité et leur processus biologiques sont liés. Par exemple, l'ergonome peut interpréter les comportements en analysant leurs mesures physiologiques comme le rythme cardiaque.

Les émotions sont aussi une composante importante de l'activité humaine [10]. La vision de l'activité dans [15] rejoint celle dans [13] en disant que l'activité humaine repose sur la cognition car elle "mobilise et construit des savoirs". Dans le contexte des formations des adultes, nous dégageons à partir des travaux dans [16] les propriétés suivantes de l'activité humaine :

- L'autonomie des individus : chacun peut évoluer tout en étant indépendant de facteurs extérieurs.
- L'activité est collective : c'est leur capacité de s'organiser ensemble. L'activité collective émerge à partir des activités individuelles.
- Le couplage individu-environnement : chaque individu possède sa propre vision de son environnement.
- Le vécu et l'expérience de l'individu.

Dans le cadre de l'activité domestique, les propriétés théoriques de l'activité présentées dans [24] rejoignent celles dans [16] : l'autonomie des systèmes vivants traduite par l'activité individuelle des acteurs, l'activité collective, et le couplage asymétrique de l'acteur avec son environnement. « Cette interaction est asymétrique car c'est l'acteur qui définit ce qui est significatif/pertinent pour lui dans son environnement. » ([[24], p. 7]. Néanmoins la dimension émotionnelle des individus n'y figure pas. Les auteurs de l'article [24] prolongent la définition de l'activité humaine par des analyses empiriques sur l'activité humaine avec les travaux de l'article [18]. Ils ont permis d'identifier plusieurs propriétés de l'activité réelle des personnes des ménages :

- La dynamique individuelle caractérisée par la flexibilité des activités. Les individus peuvent arrêter leur activité en cours pour la déplacer plus tard, l'interrompre totalement... Par exemple, un individu faisant des travaux chez soi, peut les interrompre et les reprendre le lendemain. L'activité de l'individu possède des variabilités structurées autour d'une routine quotidienne.
- La dynamique collective caractérisée par l'articulation de plusieurs dynamiques individuelles. Les individus se coordonnent, ils interagissent ensemble et forment cette dynamique collective. Ils peuvent participer à la même activité.
- Le couplage de l'activité à l'environnement : la disposition des éléments composant l'environnement impacte l'activité humaine. L'agencement des pièces dans une maison impacte l'activité des individus.

Les travaux empiriques présentés dans [18] et la définition de l'activité humaine dans [16] et dans [24] ont alors en commun les activités individuelles et collectives, et la prise en compte de l'environnement. Le caractère empirique rejoint le comportement visible et capturable [13]. Enfin, la notion de variabilité est importante en ergonomie. Les états des personnes, les objets composant l'environnement peuvent changer au cours du temps et impacter l'activité sous la forme de "variabilités de situations" [11]. Un déterminant ici, est un élément, un facteur, impactant l'activité. Le déterminant est la cause de ces variabilités. Les déterminants peuvent être externes, propres à la situation, comme l'organisation, l'environnement etc ; ou alors internes, propres à l'Homme, comme le sexe, l'âge, l'expérience, les états instantanés (physique, psychique, etc.), etc. L'auteur dans [3] reprend cette différence entre les déterminants internes et externes.

2.1.2 Les propriétés de l'activité humaine retenues

Bien qu'il existe un socle de base partagé au sein des théories de l'activité mobilisées en ergonomie [12], les auteurs se centrent sur certaines propriétés de l'activité humaine selon leur objet de recherche. Dans notre cas, nous proposons une liste de propriétés de l'activité humaine qu'il nous semble essentielle dans un objectif de réalisme de la simulation de l'activité humaine.

- L'activité est individuelle : elle est caractérisée par des variabilités et des régularités à différentes échelles temporelle et spatiale.
- L'activité est collective : les individus sont capables de s'organiser et d'interagir ensemble. Ainsi l'articulation des activités individuelles forment une activité collective. Autrement dit, l'activité collective émerge à partir des activités individuelles.
- Le couplage individu-environnement : les individus évoluent dans un environnement, dont chacun a sa propre vision. L'individu interagit avec son environ-

nement.
- L'autonomie des individus : les individus sont capables d'évoluer en relative indépendance des caractéristiques de l'environnement dans lequel ils évoluent.
- La cognition et les émotions : l'activité dépend pour partie des processus mentaux des individus, et de leurs émotions.

Comment pouvons-nous qualifier le réalisme de la simulation de l'activité humaine au regard de ses propriétés que nous venons de définir ?

2.2 Le réalisme de la simulation de l'activité humaine

Les principes théoriques et les analyses empiriques des auteurs dans [24] leur « [...] ont permis de situer le réalisme de la simulation par rapport à l'activité humaine. » ([[24], p. 15]). Dans ce cas, les principes théoriques et empiriques de l'activité humaine qu'ils ont établis sont tous représentés dans leurs simulations. Cependant, nous pouvons identifier des limites au réalisme des simulations basées sur ces propriétés. Par exemple, leurs simulations manquent de variabilités temporelles. Les simulations produisent des activités globalement similaires d'un jour à l'autre. De plus, les émotions ne sont pas simulées. Pour autant, ils considèrent leurs simulations comme réalistes.
Ainsi dans la même idée, il est possible d'avoir une simulation de l'activité humaine réaliste sans pour autant respecter toutes les propriétés de l'activité humaine. À la place, nous pourrions parler plutôt de simuler de manière réaliste une ou plusieurs propriétés de l'activité humaine que nous aurions choisies.

Comment pouvons-nous les traduire dans le cadre de la simulation de l'activité humaine ? Autrement dit, quelles sont les propriétés non plus de l'activité humaine, mais de sa simulation ?

- L'activité est individuelle : les simulations doivent pouvoir se situer à l'échelle d'un individu. De plus, l'activité individuelle doit aussi bien posséder des variabilités que des régularités.
- L'activité est collective : les individus simulés doivent pouvoir interagir et se coordonner entre-eux. Une activité collective doit donc pouvoir émerger à partir de plusieurs activités individuelles.
- Le couplage individu-environnement : les activités individuelle et collective doivent être structurées dans un environnement. Les interactions entre les individus et leur environnement doivent pouvoir être prises en compte, ce qui signifie que les facteurs environnementaux doivent être représentés et impacter l'activité humaine et inversement.
- L'autonomie des individus : La capacité d'adaptation des individus dans leur environnement doit pouvoir être simulée. La simulation doit prendre en compte la réactivité des individus.
- La cognition et les émotions : représenter les processus mentaux et cognitifs basiques, des émotions simples et leurs impacts sur les activités simulées. Les changements émotionnels doivent pouvoir être simulés.

Nous venons de dégager plusieurs propriétés de l'activité humaine simulée. Nous pouvons alors nous demander si les simulations informatiques de l'activité humaine existantes respectent ces propriétés, et lesquelles ? Intéressons-nous alors aux approches informatiques simulant l'activité humaine.

3 Les simulations informatiques de l'activité humaine

Il existe différentes façons de modéliser l'activité humaine : les approches non-centrées agent (déterministes ou statistiques), et les approches multi-agents (classiques, couplées aux statistiques ou basées sur la simulation participative). Cette dernière se centre sur la simulation d'individus qui entrent en interaction, ce qui permet de produire des phénomènes émergents que l'on peut ainsi observer et dont on peut analyser les processus d'émergence.

3.1 Les approches non-centrées agent

3.1.1 Les approches déterministes

Les approches déterministes ont deux caractéristiques principales.
La première est l'immuabilité des simulations de l'activité humaine avec des mêmes paramètres d'entrée.
La seconde concerne la simulation en elle-même, l'activité générée durant la simulation est répétitive et manque alors de variabilités. Ces limites sont rencontrées par exemple parmi les simulations informatiques urbaines, il est possible d'utiliser des profils déterministes d'occupation des résidences [9]. Il existe aussi plusieurs modèles déterministes pour générer les emplois du temps représentant l'activité humaine [21]. Un emploi du temps est décrit par un enchaînement d'activités que font les individus durant une période de temps donnée. Les approches déterministes sont ainsi efficaces lorsque que l'on souhaite simuler une activité humaine statique. Elles le sont en revanche moins lorsque que l'on souhaite comme nous, simuler aussi bien des régularités que des variabilités à l'échelle individuelle.

3.1.2 Les approches statistiques

Les approches statistiques sont des approches stochastiques. Chaque simulation exécutée diffère l'une de l'autre, et, chaque simulation varie elle-même au cours du temps, produisant des variabilités inter- et intra-simulation.
Parmi ces approches, le processus de Bernoulli génère des évènements qui sont indépendants des uns des autres [47], contrairement aux chaînes de Markov [32]. Ces dernières sont utilisées par exemple pour générer des probabilités d'activation des activités [29], et des emplois du temps d'individus simulés [31]. L'activité courante choisie par l'agent va dépendre des activités choisies précédemment.
Bien que ces approches permettent de simuler l'activité humaine tout en intégrant des variations entre les simu-

lations, la coordination et la réactivité des individus ainsi que leur évolution dans un environnement sont complexes à reproduire [38].
L'approche formulée dans [29] présente des facteurs d'influence, dont quelques-uns issus de l'environnement, mais ne les intègrent pas dans leur modèle de simulation. Ils sont plutôt traités en pré-simulation. Ainsi si l'environnement change, il faut traiter à nouveau les données issues de l'environnement avant d'exécuter de nouvelles simulations. L'activité simulée ne s'adapte donc pas de façon réactive aux changements de l'environnement.
L'approche exprimée dans [27] repose sur une modélisation statistique des comportements des ouvertures et fermetures de fenêtre dans un logement en prenant en compte des facteurs physiques environnementaux, mais la prise de décision collective n'est pas prise en compte.
Dans [48] la modélisation des interactions entre les émotions est suggérée car elles sont une composante importante de la cognition et elles sont une réponse à des stimuli de l'environnement. Ainsi les auteurs ont modélisé les émotions avec les chaînes de Markov, où chaque état représente une émotion. L'approche employée dans [33] simule aussi les changements émotionnels, en se basant sur des automates finis. Mais les interactions entre les individus ne sont pas simulées.

En conclusion, les approches statistiques, en particulier les chaînes de Markov, permettent de simuler un enchaînement d'évènements dépendants des précédents. Leur caractère stochastique permet d'obtenir des simulations non-déterministes intégrant des variabilités inter- et intra-simulations. Elles peuvent aussi prendre en compte des facteurs d'influence comme ceux issus de l'environnement. Néanmoins ces approches ne visent pas à reproduire des interactions réactives entre les individus et leur environnement. Les propriétés de l'activité simulée d'activité collective et d'autonomie des individus ne sont pas simulées.
Pour prendre en compte la réactivité des agents face à l'environnement et l'aspect social entre les individus, il faut se diriger vers les approches multi-agents.

3.2 Les approches multi-agents

3.2.1 Les approches classiques

Le caractère stochastique des systèmes multi-agents se décrit par deux critères [46]. Premièrement, l'agent agit sur son environnement au cours de la simulation, ce qui signifie que l'environnement peut être modifié en fonction des actions de l'agent. L'environnement n'est alors pas figé. Deuxièmement, les décisions de l'agent en interaction avec son environnement ne sont pas déterministes.
En plus de leur autonomie et de la prise en compte de leur environnement, les coordinations entre agents sont une caractéristique forte des systèmes multi-agents. Les interactions notamment spatiales entre les agents sont au cœur de la simulation. Dans le milieu urbain, les déplacements humains sont reproduits pour simuler leur impact sur la propagation de virus dans un environnement [44, 42, 20], mais aussi pour étudier l'activité des piétons [45]. L'avantage ici, est la reproduction des activités collectives émergentes à partir des comportements individuels articulés ensemble.

Le modèle BDI (Belief-Desire-Intention) reprend ces deux points sur l'environnement : l'agent possède des croyances sous forme d'informations sur son environnement et sur les autres agents du même environnement, et chaque agent possède des objectifs et prend ses propres décisions pour les atteindre [7].
Ce modèle permet de prendre en compte la cognition et les émotions au sein de l'activité des agents dans l'objectif de rendre les simulations plus réalistes d'un point de vue social, par exemple dans le cadre d'une évacuation d'un bâtiment incendié [6]. Le modèle EBDI (Emotion, Belief, Desire, Intention) proposé dans [26] intègre les émotions dans le modèle BDI.

Il existe aussi d'autres approches cognitives, comme celles présentées dans la review [37]. Les approches décrites dans cette dernière sont des modèles multi-agents prenant en compte la notion de confiance et de réputation entre les individus. Néanmoins la plupart des modélisations de la cognition de la review ne sont pas ou difficilement implémentables à cause de leur complexité [37].

Les approches classiques multi-agents peuvent considérer toutes les propriétés de l'activité humaine simulée. Néanmoins il peut être compliqué de calibrer les régularités et variabilités de l'activité des agents. Pour cela, il existe des approches mixant les approches multi-agents et les approches statistiques.

3.2.2 Les simulations multi-agents couplées aux statistiques

Cette hybridation permet une calibration de l'activité des agents grâce à des données statistiques pour l'échelle macroscopique, tout en ayant une activité autonome et coopérative à l'échelle microscopique. Le modèle proposé dans [38] génère statistiquement les emplois du temps d'agents en fonction des typologies des individus et de leurs journées. Dans [1], des modèles probabilistes sont utilisées pour simuler les différents états d'un occupant de foyer tout en ayant des agents réactifs.

Ce couplage des statistiques avec le système multi-agent possède plusieurs avantages. Tout d'abord, elle permet aussi bien de simuler l'échelle microscopique grâce à l'approche centré sur l'agent, et à l'échelle macroscopique avec des données statistiques. Ainsi pour vérifier la validité des résultats à une large échelle, les données simulées peuvent être comparées aux données réelles, grâce à l'agrégation des données individuelles simulées [38]. Néanmoins, les données simulées peuvent ne pas être réalistes, dans la mesure où elles peuvent être proches des statistiques, mais éloignées des phénomènes opérant à l'échelle individuelle.

En revanche, nous ne sommes pas assurés que la calibration à l'échelle microscopique est comparable à la réalité. Comment pouvons-nous nous assurer que l'activité individuelle simulée l'est de manière réaliste ?

3.2.3 Les simulations multi-agents participatives

Les simulations participatives sont un moyen de répondre à ce questionnement. Ce sont des simulations multi-agents où les agents peuvent être contrôlés par des personnes réelles, notamment au travers d'IHM dédiées [19]. Cette méthode est exploitée afin de valider qualitativement à l'échelle microscopique leur modèle multi-agent en confrontant les simulations à la vie de personnes dont la vie quotidienne est simulée sur une semaine [23]. Les auteurs précisent que les participants ont pu effectivement reproduire des moments de la vie réelle. Néanmoins, les participants se concentraient sur les régularités de leur vie quotidienne et non pas sur sa variabilité. Les participants peuvent alors omettre les variabilités durant la simulation.

Outre la validation de la simulation, un autre objectif de la simulation participative est décrit dans [2]. Il s'agit de la construction de nouvelles idées à partir de réflexions collectives grâce à la simulation participative. Cette méthode participative est utilisée pour stimuler l'apprentissage social [28].
En faisant participer des humains dans les simulations, les relations sociales complexes sont alors intégrées dans la simulation comme la réputation et la confiance [19]. De plus, grâce aux traces de la simulation enregistrées [19], des analyses peuvent être faites sur les simulations afin de comprendre les règles régissant les interactions entre les humains [4]. Comme les systèmes multi-agents classiques, des activités collectives peuvent émerger.

Enfin, il est possible de combiner cette participation avec un aspect ludique afin d'éveiller des réflexions sur la réduction de la consommation d'énergie à l'échelle d'un ensemble de ménages [5]. D'ailleurs, la simulation participative est comparée avec les jeux de rôles dans [8]. Mais cet aspect ludique peut être un obstacle au réalisme car les participants pourraient ne pas reproduire la même activité dans la réalité sans le cadre ludique.

En conclusion, les approches multi-agents ont le principal intérêt de jouir de l'autonomie des agents. Ils sont capables de réagir et de s'adapter par rapport à leur environnement et par rapport aux autres agents. Mais l'activité peut être difficile à calibrer. Pour cela, il est possible de les combiner avec les approches statistiques. Une autre façon est de remplacer le contrôle virtuel des agents par le contrôle humain grâce à la simulation participative qui ne nécessite pas de données statistiques.

3.3 Synthèse

Nous venons de présenter plusieurs approches informatiques permettant de simuler l'activité humaine. Chacune de ces méthodes simule différemment l'activité humaine. Certaines propriétés de l'activité humaine simulée sont prises en compte, d'autres non, en fonction de l'approche employée.
Toutefois, bien que certaines propriétés de l'activité humaine simulée soient modélisées par ces approches informatiques, elles le sont de différentes manières. Nous allons voir quelles sont les approches informatiques les plus à même de les simuler.

4 Discussions

4.1 Comparaisons des approches informatiques en vue de simuler une activité humaine réaliste

Nous pouvons désormais lier les approches à leurs propriétés en indiquant le degré d'adaptation au réalisme de la simulation de l'activité humaine.
Pour cela, nous comparons la capacité des différentes approches informatiques à simuler les propriétés de l'activité humaine simulée. Nous notons cette capacité sur une échelle de 0 à 2 dans le tableau 1, allant du plus faible niveau d'adaptation au plus élevé.

Les approches non-centrées agent, ne peuvent en général simuler l'activité collective car elles ne simulent pas les interactions entre les individus. De plus, elles ne simulent pas non plus la réactivité des agents, et donc leur caractère autonome.
Les approches déterministes non-centrées agent peuvent simuler l'activité humaine à l'échelle individuelle. La dualité régularités/variabilités n'est pas simulée car les variabilités ne sont pas prises en compte. Un environnement est dynamique, il peut changer au cours du temps, ainsi il ne peut être pris en compte par une approche déterministe. Inversement pour les mêmes raisons, les approches statistiques et probabilistes simulent la balance régularités/variabilités et peuvent prendre en compte l'environnement et son couplage avec l'individu. Les changements émotionnels nécessitent des variations et peuvent être des réponses à notre environnement, donc seules les approches non déterministes peuvent les prendre en compte.
Néanmoins, dans ces approches, l'activité de l'individu n'est pas le centre de la simulation, contrairement justement aux approches centrées agent, qui elles ont des agents représentant des individus. Les agents sont réactifs, ils s'adaptent à leur environnement. Inversement, l'environnement est impacté par l'activité des agents. Grâce à leur réactivité et adaptabilité, les agents sont donc autonomes. Le modèle BDI permet de reproduire des processus cognitifs basiques. Nous pouvons imaginer aussi un couplage des méthodes probabilistes modélisant ces processus cognitifs basiques, combinées avec les systèmes multi-agents sans passer par le modèle BDI. Les agents virtuels sont capables d'interagir ensemble et de se coordonner spatialement. L'activité collective émerge alors à partir des activités individuelles. Cependant les simulations participatives avec des vrais humains améliorent cette facette puisqu'ils

	Activité individuelle	Régularités / Variabilités	Activité collective	Individu - Environnement	Autonomie	Cognition - Emotion
Déterministes non-centrées agent	1	0	0	0	0	0
Statistiques, probabilistes non-centrées agent	1	1	0	1	0	1
SMA (BDI)	2	1	1	2	2	2
SMA-statistiques	2	1	1	2	2	1
SMA participatives	2	1	2	2	2	2

TABLE 1 – Les approches informatiques et leurs propriétés de l'activité humaine
0 : mauvaise adaptation au réalisme ; 1 : adaptation passable ; 2 : bonne adaptation

peuvent communiquer ensemble réellement. Le problème reste la prise en compte des régularités et des variabilités puisque les statistiques ne permettent que de simuler des données moyennes, et les vrais individus des simulations participatives peuvent parfois ne se focaliser que sur les régularités de leur activité [24].

En conclusion, les systèmes multi-agents sont les plus adaptés pour simuler les propriétés de l'activité humaine. Cependant, nous remarquons qu'aucune approche n'arrive à simuler entièrement ces propriétés. Par exemple, les régularités et les variabilités ne sont pas simulables entièrement pour toutes les approches.
Par ailleurs, se pose la question des données nécessaires pour alimenter les modèles notamment les systèmes multi-agents, qui doivent les utiliser pour initialiser et calibrer l'activité des agents. Les données statistiques purement quantitatives et agrégées ne sont probablement pas suffisantes. Qu'en est-il des données qualitatives et/ou individuelles ?

4.2 Les données pour les propriétés de l'activité humaine simulée

Pour chaque propriété de l'activité humaine simulée, des données sont nécessaires et impactent la capacité de l'approche à simuler l'activité humaine.

- L'activité est individuelle : les données peuvent être qualitatives. Par exemple, l'activité des personnes peut être filmée et analysée ensuite [18]. Elles peuvent aussi être quantitatives, dans [39] des données de eye-tracking sont exploitées pour analyser l'activité des pilotes d'avion. Un autre moyen est d'utiliser les enquêtes emploi du temps pour simuler statistiquement l'activité à l'échelle individuelle [22]. Ce sont des journaux de bord où les individus écrivent leurs activités durant quelques jours. Il est aussi possible d'utiliser une application mobile en guise de journal de bord pour récolter les mêmes types de données [17].
- L'activité est collective : elle utilise des données individuelles et agrégées. Par exemple, des données agrégées d'enquête emploi du temps sont utilisées dans [38] pour valider les simulations à l'échelle macroscopique.
- Le couplage individu-environnement : L'environnement peut être généré à partir de données géographiques [43].Les interactions entre les individus et leur environnement peuvent être filmées pour ensuite être analysées et reproduites [24].
- L'autonomie des individus : les données sont qualitatives et peuvent utiliser les mêmes données que celles exploitées par l'activité individuelle [24].
- La cognition et les émotions : les données peuvent être aussi bien quantitatives puisque les émotions peuvent être modélisées stochastiquement. Mais elles peuvent aussi exploiter des données qualitatives et les convertir en données quantitatives. Par exemple, pour simuler les émotions faciales des agents conversationnels, il est possible de filmer des acteurs et d'utiliser des modèles statistiques pour modéliser et simuler les émotions [36].

Ainsi pour une même propriété, il existe plusieurs types de données que ce soit quantitatives que qualitatives. Des mêmes données peuvent être exploitées pour différentes propriétés de l'activité humaine.
Toutefois, à notre connaissance, aucune source de donnée ne donne une vision sur l'équilibre entre les régularités et les variabilités de la propriété de l'activité individuelle. Par exemple, les enquêtes emploi du temps de l'INSEE ne donnent qu'un jour de semaine et qu'un jour de week-end au maximum par individu. Nous ne pouvons donc pas connaître les variabilités entre les différents jours de la semaine des individus.
Nous proposons donc une nouvelle approche couplant un nouveau modèle informatique et un nouveau recueil de données, permettant d'obtenir cet équilibre entre les régularités et les variabilités de l'activité.

4.3 Proposition d'une nouvelle approche

Ces notions de régularités et de variabilités au sein de l'activité impactent plusieurs propriétés à la fois : le couplage individu-environnement pour l'aspect spatial, mais aussi l'individu lui-même puisque son activité individuelle est concernée. De plus, malgré l'utilisation des données citées, ces limites ne sont pas dépassées.
Toutefois, nous avons mentionné la notion de déterminant

étant à la base des variabilités de l'activité. Nous faisons l'hypothèse que la prise en compte des déterminants de l'activité humaine rendrait la simulation de l'activité humaine plus réaliste en résolvant les limites associées à l'équilibre entre la routine et la variabilité. Nous proposons donc d'intégrer cette notion de déterminant dans les modèles informatiques.
Les systèmes multi-agents sont les modèles les plus adaptés, et nous proposons de repartir de ces modèles puisque notre tableau 1 nous montre qu'ils permettent de simuler de la manière la plus réaliste l'activité humaine.
Plus particulièrement, les simulations multi-agents participatives permettent un meilleur réalisme de la simulation de l'activité humaine. Nous pourrions donc repartir de cette approche afin de prendre en compte les déterminants.

Néanmoins la simulation participative est coûteuse en temps pour les participants. À long terme, il s'agirait d'avoir des agents logiciels. Dans ce cas, une idée serait d'utiliser un modèle hybride BDI-statistiques intégrant les déterminants de l'activité humaine, afin d'avoir des agents émotionnels calibrés avec des méthodes statistiques.
Les travaux décrits dans [40] et [30] proposent ce couplage hybride statistiques à base de règles, mais ils sont focalisés sur la simulation de dialogues. D'autres aspects de la simulation de l'activité humaine comme par exemple, le déplacement des agents dans leur environnement, ne sont pas visés.

La simulation participative aurait donc comme première utilité de récolter de nouvelles données qualitatives pour analyser les activités de vraies personnes, afin de nous aider à modéliser nos agents hybrides BDI-statistiques réactifs aux déterminants de l'activité humaine.
Un autre type de recueil de données est celui présenté dans [17], où une application mobile fait office de journal de bord. Un recueil de données qualitatives et quantitatives de ce type pourrait être directement adapté aux finalités d'un modèle informatique de l'activité.

5 Conclusion

Nous avons défini des propriétés de l'activité humaine sous le prisme de l'ergonomie. Puis nous avons présenté plusieurs types d'approches informatiques pour simuler l'activité humaine, ce qui nous a permis de conclure que les approches multi-agents sont les plus adaptées pour simuler les propriétés de l'activité humaine. Cependant aucune approche actuelle ne valide entièrement toutes les propriétés, de plus les régularités et variabilités de l'activité humaine ne sont pas simulées de la meilleure façon pour toutes les approches mentionnées.
En perspective, nous proposons donc de développer une nouvelle approche multi-agent hybride intégrant la prise en compte des déterminants de l'activité humaine et une nouvelle source de données.

6 Bibliographie

Références

[1] Fatima ABDALLAH, Shadi BASURRA et Mohamed Medhat GABER. "A Hybrid Agent-Based and Probabilistic Model for Fine-Grained Behavioural Energy Waste Simulation". en. In : *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*. Boston, MA : IEEE, nov. 2017, p. 991-995.

[2] Nicolas BECU. "Les courants d'influence et la pratique de la simulation participative : contours, design et contributions aux changements sociétaux et organisationnels dans les territoires". fr. In : (2020), p. 270.

[3] Laurent VAN BELLEGHEM. "Faut-il repenser le « schéma à 5 carrés » pour analyser les besoins de convergence du travail contemporain ?" fr. In : *SELF* (2017), p. 5.

[4] Matthew BERLAND et William RAND. "Participatory Simulation as a Tool for Agent-based Simulation". en. In : *Proceedings of the International Conference on Agents and Artificial Intelligence*. Porto, Portugal : SciTePress - Science, 2009, p. 553-557.

[5] Anaïs BERRY et al. "De la simulation à la simulation participative : apporter une vision intégrée de la rénovation urbaine aux aménageurs publics, à travers une démarche expérimentale : From simulation towards participatory simulation : providing an integrated vision to public planners of urban renewal, through an experimental approach". fr. In : *netcom* 34-1/2 (jan. 2020).

[6] M BOURGAIS, P TAILLANDIER et Laurent VERCOUTER. "Cognition, émotions et relations sociales pour la simulation multi-agent". fr. In : *JFSMA* (juill. 2017).

[7] Michael BRATMAN. *Intention, Plans, and Practical Reason*. Cambridge : Cambridge, MA : Harvard University Press. 1987.

[8] Jean-Pierre BRIOT, Paul GUYOT et Marta IRVING. "Participatory Simulation for Collective Management of Protected Areas for Biodiversity Conservation and Social Inclusion". en. In : 2007, p. 6.

[9] Giuseppina BUTTITTA et Donal P. FINN. "A high-temporal resolution residential building occupancy model to generate high-temporal resolution heating load profiles of occupancy-integrated archetypes". en. In : *Energy and Buildings* 206 (jan. 2020), p. 109577.

[10] Béatrice CAHOUR et Alain LANCRY. "Émotions et activités professionnelles et quotidiennes :" fr. In : *Le travail humain* Vol. 74.2 (juin 2011), p. 97-106.

[11] François DANIELLOU. "L'ergonomie dans la conduite de projets de conception de systèmes de travail". In : *Ergonomie*. 2004, p. 359-373.

[12] François DANIELLOU * et Pierre RABARDEL. "Activity-oriented approaches to ergonomics : some traditions and communities". en. In : *Theoretical Issues in Ergonomics Science* 6.5 (sept. 2005), p. 353-357.

[13] Françoise DARSES et Maurice DE MONTMOLLIN. *L'ergonomie*. La Découverte. Repères. 2012.

[14] Alexis DROGOUL et al. "Designing social simulation to (seriously) support decision-making : COMOKIT, an agent-based modelling toolkit to analyse and compare the impacts of public health interventions against COVID-19". In : *Frontiers in Public Health* (2020).

[15] Marc DURAND. "Theureau, J. Le cours d'action. L'enaction et l'expérience". fr. In : *activites* 13.1 (mars 2016).

[16] Marc DURAND. "Un programme de recherche technologique en formation des adultes : Une approche enactive de l'activité humaine et l'accompagnement de son apprentissage/développement". fr. In : *educationdidactique* 2-3 (déc. 2008), p. 97-121.

[17] Philipp GRÜNEWALD et al. "What we do matters – a time-use app to capture energy relevant activities". en. In : *ECEEE Summer Study Proceedings* (2017), p. 9.

[18] Julien GUIBOURDENCHE et al. "Analyse de contextes d'activité domestique pour la conception de systèmes diffus énergétiquement efficients". fr. In : *activites* 12.1 (avr. 2015).

[19] Paul GUYOT. "Simulations multi-agents participatives : Faire interagir agents et humains pour explorer, modéliser et reproduire les comportements collectifs". fr. Thèse de doct. 2006.

[20] Jürgen HACKL et Thibaut DUBERNET. "Epidemic Spreading in Urban Areas Using Agent-Based Transportation Models". en. In : *Future Internet* 11.4 (avr. 2019), p. 92.

[21] Gabriel HAPPLE, Jimeno A. FONSECA et Arno SCHLUETER. "A review on occupant behavior in urban building energy models". en. In : *Energy and Buildings* 174 (sept. 2018), p. 276-292.

[22] Yvon HARADJI. "Multi-agent simulation of human activity : a concretization in ergonomics of the technological "course of action" research program". en. In : *activites* 18-1 (avr. 2021).

[23] Yvon HARADJI. "Simulation multi-agent de l'activité humaine : une concrétisation en ergonomie du programme de recherche technologique « cours d'action »". fr. In : *activites* 18-1 (avr. 2021).

[24] Yvon HARADJI et al. "De la modélisation de l'activité humaine à la modélisation pour la simulation sociale : entre réalisme et fécondité technologique". fr. In : *activites* 15.1 (avr. 2018).

[25] François HUBAULT. "A quoi sert l'analyse de l'activité en ergonomie ?" fr. In : (1995), p. 14.

[26] Hong JIANG, Jose M. VIDAL et Michael N. HUHNS. "EBDI : an architecture for emotional agents". en. In : *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. Honolulu Hawaii : ACM, mai 2007, p. 1-3.

[27] Rory V. JONES et al. "Stochastic behavioural models of occupants' main bedroom window operation for UK residential buildings". en. In : *Building and Environment* 118 (juin 2017), p. 144-158.

[28] Christophe LE PAGE. "Simulation multi-agent interactive : engager des populations locales dans la modélisation des socio-écosystèmes pour stimuler l'apprentissage social". In : (2017).

[29] Yuanmeng LI, Yohei YAMAGUCHI et Yoshiyuki SHIMODA. "Impact of the pre-simulation process of occupant behaviour modelling for residential energy demand simulations". en. In : *Journal of Building Performance Simulation* 15.3 (mai 2022), p. 287-306.

[30] Bing LIU et al. *Dialogue Learning with Human Teaching and Feedback in End-to-End Trainable Task-Oriented Dialogue Systems*. en. arXiv :1804.06512 [cs]. Avr. 2018.

[31] Diba MALEKPOUR KOUPAEI, Kristen S. CETIN et Ulrike PASSE. "Stochastic residential occupancy schedules based on the American Time-Use Survey". en. In : *Science and Technology for the Built Environment* 28.6 (juill. 2022), p. 776-790.

[32] Jeetika MALIK et al. "Ten questions concerning agent-based modeling of occupant behavior for energy and environmental performance of buildings". en. In : *Building and Environment* 217 (juin 2022), p. 109016.

[33] Qing-mei MENG et Wei-guo WU. "Artificial emotional model based on finite state machine". en. In : *J. Cent. South Univ. Technol.* 15.5 (oct. 2008), p. 694-699.

[34] Farid MIRAHADI, Brenda MCCABE et Arash SHAHI. "IFC-centric performance-based evaluation of building evacuations using fire dynamics simulation and agent-based modeling". en. In : *Automation in Construction* 101 (mai 2019), p. 1-16.

[35] Fernanda P. MOTA et al. "A persuasive multi-agent simulator to improve electrical energy consumption". en. In : *Journal of Simulation* 17.1 (jan. 2023), p. 17-31.

[36] Magalie OCHS et al. “Vers des Agents Conversationnels Animés dotés d’émotions et d’attitudes sociales”. fr. In : *Journal d’Interaction Personne-Système* Volume 3, Issue 2, Special...Special Issue "the best... (Sept. 2015), p. 1282.

[37] Isaac PINYOL et Jordi SABATER-MIR. “Computational trust and reputation models for open multi-agent systems : a review”. en. In : *Artif Intell Rev* 40.1 (juin 2013), p. 1-25.

[38] Quentin REYNAUD et al. “Using Time Use Surveys in Multi Agent based Simulations of Human Activity :” en. In : *Proceedings of the 9th International Conference on Agents and Artificial Intelligence*. Porto, Portugal : SCITEPRESS - Science et Technology Publications, 2017, p. 67-77.

[39] Nadine B. SARTER, Randall J. MUMAW et Christopher D. WICKENS. “Pilots’ Monitoring Strategies and Performance on Automated Flight Decks : An Empirical Study Combining Behavioral and Eye-Tracking Data”. en. In : *Hum Factors* 49.3 (juin 2007), p. 347-357.

[40] Jost SCHATZMANN et al. “Agenda-based user simulation for bootstrapping a POMDP dialogue system”. en. In : *Human Language Technologies 2007 : The Conference of the North American Chapter of the Association for Computational Linguistics ; Companion Volume, Short Papers on XX - NAACL ’07*. Rochester, New York : Association for Computational Linguistics, 2007, p. 149-152.

[41] Mathieu SCHUMANN et al. “Multi-Agent based simulation of human activity for building and urban scale assessment of residential load curves and energy use”. en. In : *2021 Building Simulation Conference* (sept. 2021).

[42] Mohammad SHANAA et Sherief ABDALLAH. “Agent-based simulation for COVID-19 outbreak within a semi-closed environment”. In : *2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*. Riyadh, Saudi Arabia : IEEE, nov. 2020, p. 231-236.

[43] Patrick TAILLANDIER et al. “Des données géographiques à la simulation à base d’agents : application de la plate-forme GAMA”. fr. In : *cybergeo* (mars 2014).

[44] Agnieszka TRUSZKOWSKA et al. “High-Resolution Agent-Based Modeling of COVID-19 Spreading in a Small Town”. en. In : *Adv. Theory Simul.* 4.3 (mars 2021), p. 2000277.

[45] Giuseppe VIZZARI, Luca CROCIANI et Stefania BANDINI. “An agent-based model for plausible wayfinding in pedestrian simulation”. en. In : *Engineering Applications of Artificial Intelligence* 87 (jan. 2020), p. 103241.

[46] Janusz WOJTUSIAK, Tobias WARDEN et Otthein HERZOG. “Machine learning in agent-based stochastic simulation : Inferential theory and evaluation in transportation logistics”. en. In : *Computers & Mathematics with Applications* 64.12 (déc. 2012), p. 3658-3665.

[47] Da YAN et al. “Occupant behavior modeling for building performance simulation : Current state and future challenges”. en. In : *Energy and Buildings* 107 (nov. 2015), p. 264-278.

[48] Dong-mei YU et al. “Homogeneous Markov Chain for Modeling Emotional Interactions”. en. In : *2008 10th International Conference on Advanced Communication Technology*. ISSN : 1738-9445. Gangwon-Do, South Korea : IEEE, fév. 2008, p. 265-269.

Optimisation de matériaux et dispositifs pour l'énergie à partir de concepts d'intelligence artificielle pour small data

K. KHOUSSA[1,2,3], Y.-A. CHAPUIS[1,2], N. LACHICHE[1,3]

[1]Laboratoire ICube, UMR CNRS et Université de Strasbourg

[2]MaCEPV (Équipe Matériaux et composants électroniques et photovoltaïque)

[3]SDC (Équipe Sciences des données et des connaissances)

Résumé

L'optimisation des procédés de fabrication des matériaux organiques pour l'énergie photovoltaïque est un processus qui manque de transparence sur la chimie et la physique qui agissent sur les matériaux et qui peuvent ou non conduire à une optimisation des performances du dispositif qui en dérive. Ceci est particulièrement pertinent pour les dispositifs photovoltaïques organiques (OPV) qui dépendent de relations très complexes entre les structures chimiques et les propriétés photovoltaïques. Aujourd'hui, l'intelligence artificielle (IA) peut saisir la complexité d'un dispositif et construire des modèles qui peuvent prédire l'efficacité de conversion. Des approches fondées sur les petits jeux de données ont été développées au cours de la dernière décennie. En particulier, de nouvelles approches telles que le design of experiments (DoE) combiné avec l'analyse de l'apprentissage artificiel permettent à l'expérimentateur d'utiliser les ressources rares plus efficacement, avec une probabilité plus élevée d'arriver à un véritable optimum. Dans ce travail, nous discuterons de l'utilisation de méthodes ML basées sur DoE pour l'optimisation des dispositifs OPV. Aussi, nous élargirons notre exploration des méthodes d'IA pour les petites données en utilisant des concepts d'apprentissage actif afin de surmonter les limites du DoE.

Mots-clés

Intelligence artificielle, apprentissage artificiel, plan d'expériences, cellules photovoltaïques organiques, petits jeux de données, apprentissage actif.

Abstract

The optimization of materials manufacturing processes for photovoltaic energy is a process that lacks complete transparency of chemistry and physics that relies on constraints imposed by the user that may or may not lead to an overall optimum. This is particularly relevant for organic photovoltaic devices (OPV) which depend on a very complex relationship between chemical structures and photovoltaic properties. Today, artificial intelligence (AI) can grasp the complexity of a device and build models that can effectively predict and achieve optimal conversion efficiency. Small data approaches have grown significantly over the past decade. Recently, new approaches such as the Design of experiments (DoE) combined with machine-learning analysis allow the experimentalist to use scarce resources more efficiently, with a higher probability of achieving a true optimum. In this work, we will discuss the use of DoE-based ML methods for optimizing OPV devices. In addition, we will expand our exploration of AI methods for small data using active learning concepts to overcome DoE limitations.

Keywords

Artificial intelligence, machine learning, design of experiments (DoE), organic photovoltaic (OPV) cells, small data, active learning.

1. Introduction

L'intelligence artificielle (IA) est aujourd'hui présente dans de nombreux domaines des sciences, tels que la physique, la chimie, la biologie, la santé, et bien d'autres. On l'emploie également de plus en plus dans tous les secteurs de l'industrie, tels que l'automobile, la construction aéronautique, les semi-conducteurs, l'énergie et bien d'autres encore [1]. Cet intérêt pour l'IA s'explique en grand partie par les progrès dans le traitement des données massives, aussi appelé *big data*, lesquelles permettent une nouvelle approche pour analyser les processus industriels mais pose également le problème de la redondance de l'information, c'est-à-dire de l'information qui n'est pas strictement nécessaire pour définir les modèles de prédiction en toute confiance [1]-[2]. Il existe une alternative au *big data* par la collecte d'ensembles de données plus petits mais plus riches et plus sures en information. Cette approche, plus connue sous l'appellation *small data*, représente un vecteur d'innovation récent et prometteur mais encore complexe à traiter en IA [3]. Par exemple, la production de semi-conducteurs nécessite d'énormes quantités de données afin de pouvoir contrôler, améliorer et gérer la complexité des procédés de fabrication, d'autant plus si l'on souhaite accompagner les faibles latences des marchés. Ainsi, le concept de *small data* s'avère comme une solution d'avenir, bien que ce dernier reste une voie de recherche complexe de l'IA, et encore peu diffusée dans la littérature [4]. Récemment, des recherches en *small data* ont montré que la combinaison d'un simple plan d'expériences (*design of experiments - DoE*) avec un concept classique d'apprentissage artificiel (AA) (*machine learning - ML*) permettait d'accélérer l'optimisation de dispositifs à base de matériaux complexes pour l'énergie [2]. Cette méthode prometteuse reste encore à explorer, sachant que les performances prédictives des modèles d'apprentissage restent encore très en dessous des exigences industrielles [5]–[7]

Dans cet article, nous proposons un état de l'art sur les concepts de *small data* employés dans l'optimisation des matériaux et dispositif dérivés. Nous nous intéresserons

particulièrement au domaine des matériaux organiques pour l'énergie photovoltaïque (*organic photovoltaic - OPV*). En effet, l'organique présente la particularité de produire des cellules photovoltaïques par voie chimique ou liquide, ce qui le rend très attractif en termes d'intégration et de coût de production. Par contre, la complexité chimique de ces matériaux impose encore aujourd'hui un traitement long, délicat et nécessitant de nombreuses variables. Dans ce contexte, l'IA peut s'avérer comme une solution pour accélérer les développements dans cette technologie, laquelle se base sur des jeux de données réduits propres au *small data*.

2. Contexte scientifique

2.1. Matériaux : *OPV*

Tout comme leurs parents inorganiques, les cellules OPV utilisent l'effet photovoltaïque pour transformer l'énergie lumineuse en électricité. Par contre, les matériaux organiques ont l'avantage de produire des dispositifs beaucoup plus légers, flexibles, transparents, respectueux de l'environnement, et s'intégrant avec des substrats très variés. Ils sont également moins chers à produire grâce aux technologies de fabrication développées dans l'industrie de l'électronique flexible (e.g. OLED, etc.) [4]. Par contre, l'OPV reste encore aujourd'hui peu bénéfique en termes de rendement énergétique, durée de vie, stabilité, dégradation par exposition à l'oxygène, et dépendance à l'énergie solaire [5], [8], [9]. Pourtant, récemment, grâce à la découverte de nouveaux matériaux, l'OPV semble avoir rompu de nombreuses barrières technologiques telles que le rendement énergétique et de stabilité, ce qui les placent aujourd'hui en compétition directe avec les solutions PV classiques à base de matériaux inorganiques [5], [9].

Technologiquement, les matériaux OPV sont constitués de matériaux donneurs d'électrons et accepteurs d'électrons plutôt que de jonctions classiques *pn* semi-conductrices. Les cellules OPV sont ainsi fabriquées à partir d'une couche active contenant un polymère actif donneur d'électron et un accepteur d'électrons à base de molécules fullerènes ou non fullerènes. On parle de cellules solaires en polymère à hétérojonction en vrac ou BHJ (bulk heterojunction) [2]. La figure 1 présente la structure par couches d'une cellule OPV à BHJ.

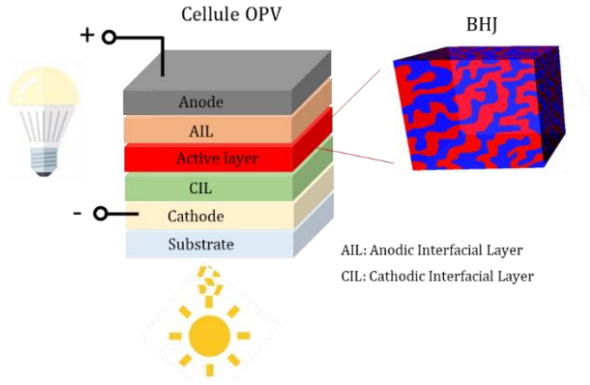


Figure 1 : Structure d'une cellule OPV à BHJ pour la conversion électrique.

Les BHJ nécessitent un contrôle sensible de la morphologie des matériaux à l'échelle nanométrique. Les variables importantes comprennent les matériaux, les solvants et le rapport pondéral donneur-accepteur, ce qui constitue toute la complexité chimique de la couche active photovoltaïque. De plus, la cellule OPV doit aussi prendre en compte la conception et l'adaptation des couches d'interface et des électrodes pour compléter le dispositif photovoltaïque [9]–[11]. Chacune des couches supplémentaires de la cellule OPV va entraîner de nouvelles variables expérimentales qui vont d'autant plus contraindre le travail d'optimisation de ces dispositifs énergétiques.

Récemment, de nouveaux matériaux ont été développés à partir d'accepteur non fullerène (*NFA*), ce qui a permis une progression spectaculaire des performances des dispositifs OPV. Dans la littérature, il a déjà été enregistré des rendements énergétiques supérieurs à 18% pour des durées de vie d'au moins 10 ans [9]. Ces résultats s'approchent et même dépassent les performances des cellules classiques PVs à couche minces à base de silicium. Cependant, pour poursuivre ces progrès, l'OPV doit à présent élaborer et optimiser une variété de matériaux type NFA, ce qui nécessitera de lourds investissements de recherche et de développement, aussi bien académiques qu'industriels. C'est dans ce contexte que l'IA et le *small data* offrent au secteur photovoltaïque une opportunité d'accélérer la découverte et d'optimiser les matériaux et dispositifs OPV [2].

2.2. Approche expérimentale : *DoE*

En matériaux, la plupart des découvertes ont été obtenues de manière empirique, généralement par le biais d'une approche expérimentation de type « *Edisonienne* » ou dite « à une variable à la fois » (*One factor at a time - OFAT*) [2]. Pourtant, cette méthode a de nombreux défauts car les caractéristiques des systèmes basés sur les matériaux ne sont ni simples, ni non corrélées, ce qui entraîne une quantité exhaustive du nombre de variables à expérimenter. En effet, il faut savoir que la modification d'une variable expérimentale peut avoir de multiples effets imprévus en raison de l'inter-connectivité entre leurs propriétés des matériaux. Généralement, on utilise une approche expérimentale basée sur les principes de DoE, lesquels englobent de nombreux calculs statistiques et permettent une analyse multi-variables. On peut donc tester et optimiser plusieurs variables simultanément, accélérant ainsi le processus de découverte et d'optimisation tout en économisant du temps et de précieuses ressources. Dans ce domaine, les travaux de Fisher [12], Box et Wilson [13] font depuis longtemps références dans le domaine de l'approche par *DoE*. Plus récemment, on mentionnera la méthode de Taguchi [14].

Bien que la méthode par DoE ait déjà fait ses preuves, cette dernière reste fondamentalement dépendante de l'expérimentateur, malgré le semblant d'automatisation. En effet, si le choix des DoEs se base sur des règles strictes d'échantillonnage, qui permettent d'explorer un grand nombre de paramètres dans un espace multidimensionnel, la méthode utilise un volume de données où l'expérimentateur est tenu de sélectionner les facteurs d'entrée qui doivent être inclus dans les expériences [12], [15]. Il est donc nécessaire d'avoir une bonne connaissance du processus d'élaboration du matériau ou du dispositif dérivé.

2.3. IA : Concept de *small data*

Il existe un intérêt croissant de l'emploi des concepts d'IA dans la recherche des matériaux et des dispositifs dérivés. Cependant, pour des raisons de coûts matériel et de développement, les jeux de données produits dans l'étude des matériaux sont généralement plus petits et plus diversifiés comparativement à d'autres domaines[3], [5], [16]. Si la réduction de la taille des jeux de données peut sembler être un désavantage dans l'élaboration des modèles d'IA, il faut aussi considérer les avantages, comme la fiabilité, l'accessibilité, la compréhension et l'exploitation des données [8]. Par définition, les modèles d'IA appliqués aux matériaux font souvent références au concept de *small data*. Dans l'approche de l'IA pour les matériaux, si le concept de *small data* reste encore peu développé et présent dans la littérature, il commence tout de même à bénéficier des développements dans le domaine général de l'IA , et l'on voit de plus en plus d'articles faisant mention de ce thème [3], [16].

Récemment, une étude publiée par Zhanh et al. [3] présente des travaux permettant d'analyser de manière exhaustive l'interaction entre la disponibilité des données sur les matériaux et la capacité prédictive des modèles d'AA. Dans un premier temps, les auteurs donnent une représentation des erreurs de précision prédictive en fonction de la taille des jeux de données qu'ils ont collectées dans la littérature, comme le reprend la figure 2.

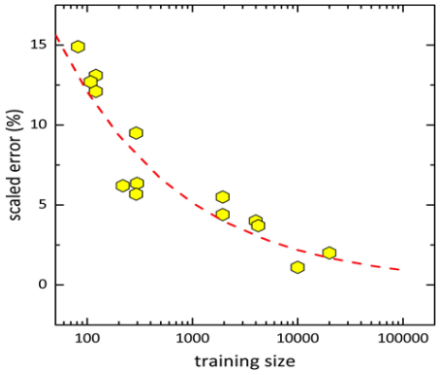


Figure 2 : Représentation des erreurs de précision prédictive en fonction de la taille des jeux de données, comme collectées dans la littérature [3].

Par la suite, une stratégie pour améliorer la précision prédictive des modèles classiques d'AA appliquée à des petits jeux de données pour études de matériaux est proposée. Les auteurs analysent l'interaction fondamentale entre la disponibilité des données sur les matériaux et la capacité prédictive des modèles d'AA. Leurs travaux révèlent que l'augmentation de la précision se fait au prix d'un degré de liberté (*degree of freedom - DoF*) du modèle. Pour remédier à ce problème, les auteurs proposent d'incorporer dans l'espace des caractéristiques un coefficient appelé « estimation brute de la propriété » (*crude estimation of property - CEP*). Puis, ils présentent trois cas d'étude de prédiction sur des matériaux : (i) bandes interdites de semiconducteurs; (ii) conductivité thermique; (iii) module d'élasticité des zéolithes. A travers ces exemples, les auteurs valident leur stratégie en démontrant que le CEP a effectivement amélioré la précision prédictive des modèles d'AA[2], [4].

3. Approche par apprentissage artificiel

3.1. Combinaison DoE-AA

Récemment, l'équipe du professeur J. Buriak de l'Université d'Alberta (Canada) a proposé de combiner les données de plan d'expériences (DoE) avec un modèle d'AA pour optimiser les performances de rendement d'un matériau pour OPV [2]. Un premier DoE est choisi pour initier le premier modèle d'AA à partir d'un jeu de seulement 16 conditions expérimentales. Puis, après visualisation des prédictions de la cible par cartographie (*mapping*), le DoE est affiné et le modèle peut être optimisé autant de cycles que nécessaire (boucle d'affinement). Le flot de la méthode appelée DoE-AA est illustrée sur la figure 3.

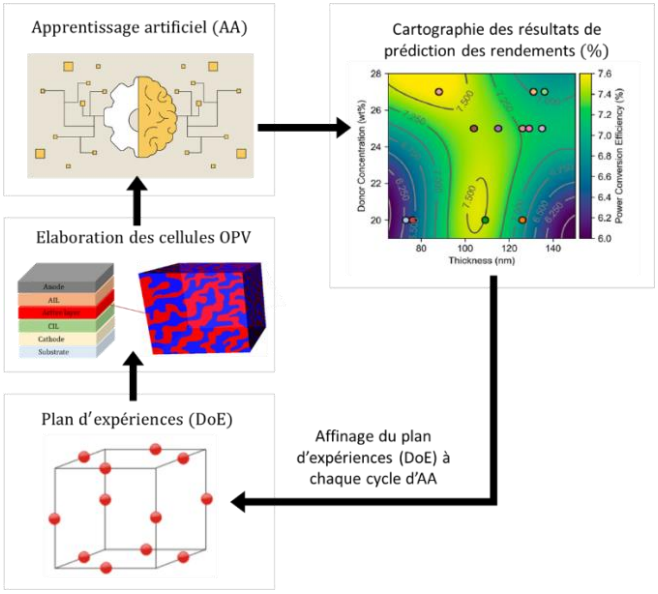


Figure 3 : Flot de la méthode DoE-AA pour l'optimisation des performances de rendement des matériaux pour cellules OPV

La méthode DoE-AA ne permet pas d'automatiser entièrement le processus d'IA mais il réduit significativement la dépendance à la décision de l'expérimentateur. En effet, si le choix du premier plan d'expériences (DoE) reste à l'initiative de l'expérimentateur, les autres cycles de décision sont fortement assistés par l'aide cartographique, laquelle ne nécessite pas intrinsèquement de connaissance en matériaux OPV. A noter que l'intérêt de cette méthode ne se limite pas aux matériaux OPV puisque plusieurs travaux de recherche s'en sont depuis inspirées pour diverses applications, telles que les électrodes transparentes [1]–[5], [8]–[11], [17], [18].

3.2. Etude des biais en AA et solutions

La méthode DoE-AA décrite dans la section précédente ouvre la voie de l'automatisation des processus d'optimisation des matériaux à partir de petits jeux de données. Cependant, la dépendance à l'utilisateur reste un point critique de la méthode. Dans l'article de review de Zhao et al. [2] les auteurs analysent plusieurs études combinant petites tailles de données

expérimentales et modèles d'AA pour l'optimisation de matériaux OPV. Ces études montrent que, généralement, les ensembles de données expérimentaux sont quelque peu biaisé, cela créant des difficultés pour évaluer objectivement les performances des modèles d'AA. Si les biais en IA sont un problème bien réel, c'est encore plus vrai dans l'élaboration et la caractérisation des matériaux, comme le montre notre étude en OPV.

Finalement, pour remédier aux biais et améliorer les modèles d'AA, les auteurs proposent un cycle de travail en quatre étapes combinant découverte, optimisation expérimentale et concepts d'AA: (1) Génération des données expérimentales en automatisant la procédure; (2) Sélection de descripteurs; (3) Analyse par AA; (4) Découverte des conditions expérimentales. Ce flot de travail est présenté sur la figure 4.

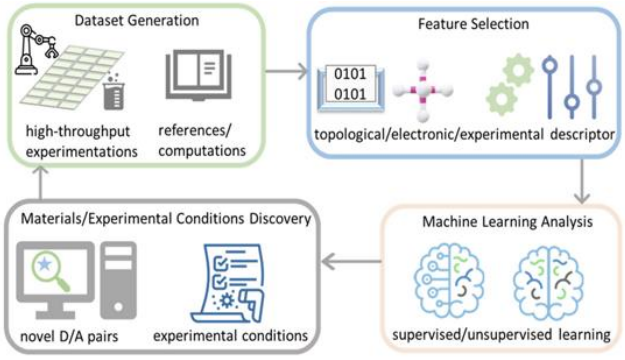


Figure 4 : Cycle de travail en quatre étapes combinant découverte, optimisation expérimentale et concepts d'AA [5]

Le cycle de travail proposé par Zhao et al. [5] se base sur l'élaboration d'un ensemble de données valides comme une condition préalable à toute méthodologie impliquant les concepts d'AA. Ces données sont liées à l'évaluation de la synthétisabilité des matériaux OPV et au choix des protocoles de fabrication de ces matériaux, ce qui est très coûteux en termes de matériel et temps de développement. Pour remédier à cette problématique de coût, le flot propose de combiner l'automatisation de la synthèse des matériaux avec un criblage virtuel assisté par AA avec une caractérisation à haut débit, et ceci en engageant de faibles tailles de données (de 70 à ~1800 données) [10]. On retiendra également la notion de laboratoire « autonome », ou « plate-formes robotisées » introduit pour réduire la dépendance du processus à la reproductivité des expériences. Finalement, ce cycle ouvre une voie de développement très prometteuse.

Un exemple d'application de cette approche par étape expérimentale automatisée est rapporté dans l'article de Du et al. [10]. Dans ce travail, les auteurs prennent en compte jusqu'à 10 variables expérimentales dans l'élaboration de cellules OPV (e.g. ratio donneur:accepteur, concentration, vitesse de dépôt, additif, solvant, température et temps de recuit, interfaces, procédés des interfaces) pour produire et caractériser automatiquement en moins de 24 heures plus de 100 cellules OPV à base de matériaux NFA de type PM6:Y6 (matériau référence pour l'OPV depuis 2019). La figure 5 présente la plate-forme auto-développée appelée AMANDA Line One pour la fabrication et la caractérisation automatique à haut débit de cellules OPV NFA. Les données expérimentales sont ensuite traitées par un modèle d'AA, lequel va permettre a d'identifier les critères les plus impactant dans l'optimisation des performances de rendement des cellules OPV.

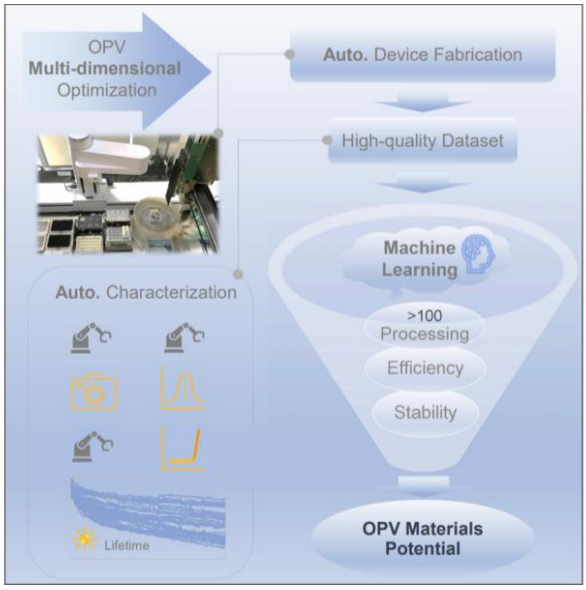


Figure 5 : Plate-forme auto-développée appelée AMANDA Line One pour la fabrication et la caractérisation automatique à haut débit de cellules OPV NFA [10].

3.3. Modèles d'AA pour small data

Dans la littérature, il n'existe pas, à notre connaissance, d'articles traitant de la pertinence de certains modèles d'AA pour le traitement de jeux de données de petites tailles. Par contre, lorsque nous manquons de donnés, il est courant que les modèles d'AA entraînent des problèmes, tels que :

- la précision de prédiction (*prediction accuracy*) ;
- la vitesse d'apprentissage (*training speed*) ;
- le surapprentissage (*overfitting*) :
- l'incapacité à valider le modèle (*inability to valide the model*)

Par conséquent, le choix de l'algorithme d'AA devra être toujours chercher à répondre positivement à ces critères.

Dans cette revue scientifique, nous avons relevé que pour gérer la faible quantité de données expérimentales dans le domaine des matériaux OPV, la plupart des auteurs développait des stratégies accompagnant les modèles d'AA, comme la combinaison de modèle d'AAA avec un plan d'expérience (DoE) ou un critère d'évaluation (CEP). Un autre point que nous avons relevé est l'emploi d'algorithmes d'AA avancés, tels que le Support vector machine (SVM) et le Gaussian process regressor (GPR).

Dans les sections suivantes, nous décrirons brièvement les algorithmes d'AA SVR et GPR, en mettant en avant leur adéquation avec les jeux de donnés réduits.

3.3.1. Support Vector Regression (SVR)

En général, le Support vector machine (SVM) est un

algorithme d'apprentissage supervisé linéaire qui est largement utilisé dans diverses industries, médecine, énergie, etc. pour résoudre les tâches de régression et de classification [16]. Cependant, en raison de la possibilité d'utiliser différents noyaux (*kernels*), cette méthode est également utilisée pour résoudre des tâches non linéaires, ce qui . Dans le cas de données quantitatives, comme employées dans l'étude de matériaux OPV, on parlera de l'algorithme Support vector regression (SVR). Les principaux avantages de l'algorithme SVR sont énuméré ci-dessous :

- fournir un travail efficace avec les petits jeux de données (*small data*) ;
- afficher de bons résultats lorsqu'il travaille dans un espace de dimensions supérieures ;
- produire une résolution sans ambiguïté.

Une série de fonctions mathématiques sont utilisées dans SVR pour convertir les données qu'il reçoit en entrée, appelé *Kernel*, dans le formulaire requis. Des prédictions non linéaires peuvent être faites dans le modèle créé par ces fonctions du kernel. Dans la littérature nous pouvons trouver pas mal de type de kernel mais dans notre cas avec les petits jeux de données, nous pouvons mentionner : les kernels polynomiales et FBR et linéaire. La relation (1) est utilisée pour le processus de normalisation:

$$y = \frac{x_i - x_{min}}{x_{max} - x_{min}} \qquad (1)$$

où : y est la valeur normalisée de x_i, x_{max} est la valeur maximale de x_i, et x_{min} est la valeur maximale de x_i.

Les relations de (2) à (4) sont utilisées pour le calcul des fonctions des kernels polynomial, RBF et linéaire :

Kernel polynomial :

$$k(x_i, x_j) = (x_i . x_j + 1)^d \qquad (2)$$

Kernel RBF :

$$k(x_i, x_j) = exp\left(-\gamma \|x_i - x_j\|^2\right) \qquad (3)$$

$$\text{où : } \gamma = \frac{1}{2\sigma^2} \quad \text{pour } \gamma > 0$$

Kernel linéaire :

$$k(x_i, x_j) = x_i^T x_j \quad (4)$$

Bien que l'algorithme SVR fonctionne efficacement avec les petits jeux de données, cette méthode n'est pas toujours très précise. La taille de l'ensemble de données, le bruit ou les valeurs aberrantes jouent également un rôle important pour son rendement [16].

3.3.2. Gaussian process regressor (GPR)

L'algorithme Gaussian process regressor (GPR) est basé sur la théorie des probabilités bayésienne et a des liens très étroits avec d'autres techniques de régression, comme la régression des crêtes du kernel (*KRR*) et la régression linéaire avec des fonctions radiales [19]. Les modèles de régression basés sur les processus gaussiens sont simples à mettre en œuvre, flexibles, entièrement probabilistes, et donc un outil puissant dans de nombreux domaines d'application [20]. Un processus gaussien est une sélection (peut-être infinie) de variables aléatoires pour lesquelles tout sous-ensemble fini de ces variables a une distribution gaussienne conjointe [21], [22]. Les variables sont généralement indexées par l'ensemble x, donc ensemble les variables $f(x)$ sont considérées comme une fonction (stochastique) sur l'ensemble d'index. Pour tout sous-ensemble fini d'indices $x_1, x_2, ..., x_n$, nous avons :

$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{bmatrix} \sim$$

$$\mathcal{N}\left(\begin{bmatrix} \mu(x_1) \\ \mu(x_2) \\ \vdots \\ \mu(x_n) \end{bmatrix}, \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & \dots & k(x_1,x_n) \\ k(x_2,x_1) & k(x_2,x_2) & \dots & k(x_2,x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n,x_1) & k(x_n,x_2) & \dots & k(x_n,x_n) \end{bmatrix}\right) \quad (5)$$

avec $\mu(x)$ est la fonction moyenne, et $k(x_i, x_j)$ est le kernel. Le kernel comme nous avons dit est une fonction semi-définie positive qui spécifie la matrice de covariance pour tout sous-ensemble fini de variables [23].

3.4. Performance de prédiction

Les mesures de rendement du modèle, comme le r, le coefficient de détermination (R^2), l'erreur racine moyenne carré (RMSE) , erreur quadratique moyenne (MSE), l'erreur moyenne absolue en pourcentage (MAPE) et l'erreur moyenne absolue (MAE) peuvent être utilisé pour les modèles de régression, ce qu'on peut les définir comme suit [5] :

$$r = \frac{\sum_{i=1}^{N}(R_i - \bar{R_i}) \times (P_i - \bar{P_i})}{\sqrt{\sum_{i=1}^{N}(R_i - \bar{R_i})^2 \times \sum_{i=1}^{N}(P_i - \bar{P_i})^2}} \qquad (6)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(R_i - \bar{P_i})^2}{N}} \qquad (7)$$

$$\text{MSE} = \frac{\sum_{i=1}^{N}(R_i - P_i)^2}{N} \quad (8)$$

$$\text{R}^2 = 1 - \frac{\text{MSE}}{\text{var}(R_i)} \quad (9)$$

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|P_i - R_i|}{|R_i|} \quad (10)$$

$$\text{MAE} = \frac{\sum_{i=1}^{N}|R_i - P_i|}{N} \quad (11)$$

avec :

- N : nombre de points de données dans l'ensemble de données.
- R_i et P_i : valeur réelle et prévue.
- $\bar{R_i}$ et $\bar{P_i}$: valeurs moyennes pour la valeur réelle et prévue, respectivement.
- $\text{var}(R_i)$: variance des données de l'échantillon.

4. Approche par apprentissage actif

Dans cet article, nous proposons d'étudier le concept d'apprentissage actif (*active learning - AL*), lequel peut présenter de nombreux avantages au traitement des petits jeux de données [24].

4.1. Notions d'AL

L'apprentissage actif ou AL, aussi connu sous le nom de conception expérimentale optimale ou apprentissage par requête, est un sous-domaine de l'AA et plus généralement de l'IA. Le concept d'AL a été introduit en 1988 par D. Angluin [25], et a initié de nombreuses études jusqu'à une synthèse plus complète par B. Settles [26] en 2009.

Le concept clé de l'AL est que si l'algorithme d'apprentissage est autorisé à choisir les données à partir desquelles il va apprendre, il sera plus performant avec moins de données annotées. Les systèmes d'AL tentent d'éliminer le manque de données « étiquetées » (ou « annotées ») en demandant des « requêtes » sous forme d'« instance non étiquetées » qui doivent être « annotées » par un oracle. Le figure 6 illustre le concept d'AL via l'ensemble des éléments de traitement :

- jeu de données « étiquetées » (*labeled training set* $\mathcal{L}$) ;
- algorithme d'AA (*machine learning model*) ;
- stratégie de requête (*select queries*) ;
- jeu de données « non-étiquetées » ou « pool » (*unlabeled pool* $\mathcal{U}$) ;
- oracle.

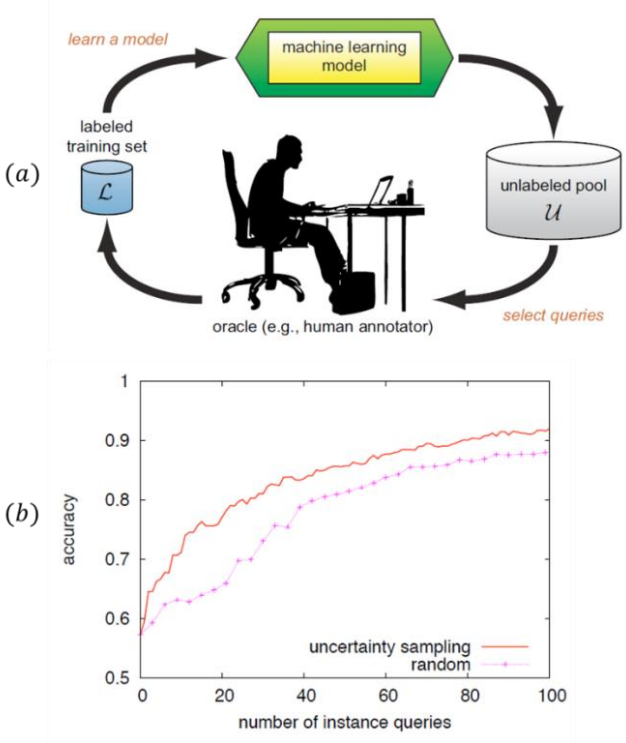


Figure 6 : (a) Flot de fonctionnement du système d'AL selon B. Settles. (b) Exemple de courbes d'apprentissage par AL montrant l'amélioration de la précision de prédiction d'un modèle quelconque en fonction du nombre de requêtes d'instance [26].

Il existe plusieurs scénarios dans lesquels l'algorithme peut interroger l'oracle, les trois principaux scénarios sont :

- la synthèse de requêtes par adhésion ;
- l'échantillonnage sélectif basé sur le flux ;
- l'AL basé sur un large jeu de données dites « non étiquetées » (*pool*).

Les systèmes d'AL doivent échantillonner des instances à partir d'un ensemble d'instances « non étiquetées » et utiliser une stratégie de requête pour décider s'il faut interroger un oracle (annotateur humain ou un appareil qui renvoie la véritable étiquette de l'instance) pour obtenir l'étiquette de l'instance ou pour l'abandonner.

En résumé, l'AL consiste à filtrer, au long d'itérations successives, les données les plus pertinentes à faire « étiqueter » ou annoter par un opérateur humain. Les nouvelles données annotées sont ajoutées aux précédentes pour construire un nouveau modèle qui sera lui-même utilisé pour affiner la sélection de nouvelles données à étiqueter. La figure 7 reprend cette démarche en formant un cycle d'apprentissage actif ou boucle d'Active Learning.

Comme le montre les exemples de la section précédente, l'approche par AL va permettre de réduire l'impact de l'étiquetage, et par conséquent de minimiser le nombre d'expériences pour obtenir le modèle d'apprentissage le plus précis.

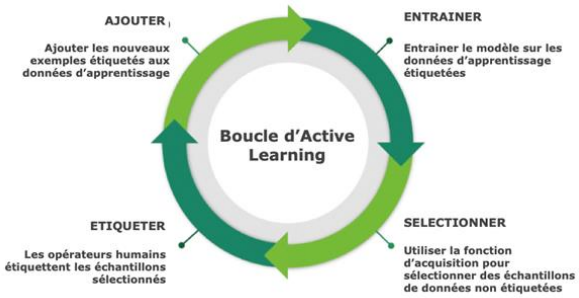


Figure 7 : Représentation de la boucle d'Active Learning [27]

4.2. Exemple d'application

Pour illustrer l'application de l'approche d'AL, nous présenterons les travaux de Lookman et al. [7], lesquels montrent l'utilisation des modèles et stratégies d'AL pour accélérer la découverte de nouveaux matériaux électroformables pour dispositifs piézoélectriques. Cette étude se base sur l'utilisation de données « non étiquetées », ou *pool*, correspondantes à près de 605 000 compositions d'un matériau électroformable qui pourra être appliqué au piézoélectrique. A chaque itération, 4 compositions du matériau seront extraites des données contenues dans le *pool*, *via* différentes stratégies de requête (exploitation, exploration, compromis entre les deux premiers et sélection aléatoire). Puis, un dispositif piézoélectrique sera élaboré et caractérisé pour chaque composition du matériau électroformable. Les résultats seront appelés « données d'itération », lesquelles seront ensuite « étiquetées » et intégrées au modèle d'AA. une nouvelle itération sera exécutée.

La boucle d'Active Learning et les résultats d'optimisation sont présentés sur la figure 8.

5. Conclusion

Cet article aborde le domaine de l'IA appliquée à l'optimisation de matériaux et dispositifs pour l'énergie, où la complexité et le coût élevé des processus restreignent le nombre d'expériences et de données disponibles. A travers une sélection dans la littérature récente, le thème combinant IA et petits jeux de données (*small data*) a été exploré dans le cas de matériaux pour le photovoltaïque organique (OPV) et de dispositif piézoélectrique. Deux approches d'IA ont été étudiées : (i) l'approche par AA ; (ii) l'approche par AL.

La première approche a permis de mettre en évidence l'amélioration des prédictions d'optimisation énergétique de matériaux OPV, en combinant algorithmes d'AA et méthodes classiques par plan d'expériences (DoE). On pourra retenir les travaux du professeur J. Buriak (2018), qui ont prouvé l'intérêt du concept, lequel fait aujourd'hui références. On notera que la tendance actuelle est le renforcement de cette approche par l'automatisation des procédés d'élaboration des matériaux OPV (Du et al., 2021), (Zhao et al., 2022).

La seconde approche, initiée par Yuan et al. (2018), propose le traitement de petits jeux de données par boucle d'AL dans le cas de cherche sur des matériaux pour dispositifs piézoélectriques. Ce concept semble très prometteur dans le domaine du *small data* car il permet de limiter le nombre d'expériences et d'améliorer les précisions de prédiction.

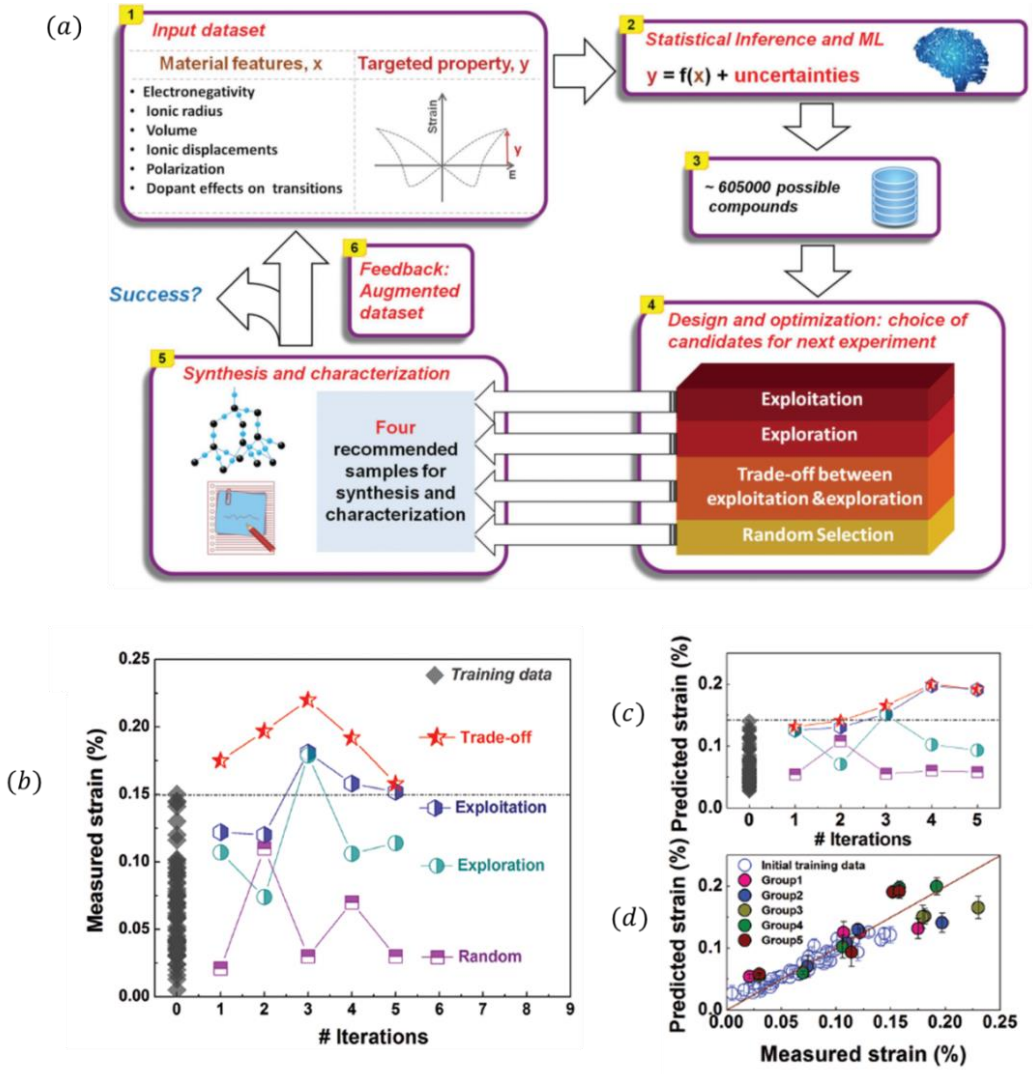


Figure 8 : (a) Boucle d'AL pour une découverte accélérée basée sur l'AA et une conception expérimentale optimale pour guider les expériences de manière itérative dans la recherche de piézoélectriques performants à grandes électrodéformations. Les composés sont synthétisés en suivant et comparant les prédictions de 4 stratégies (exploitation, exploration, compromis entre les deux premiers et sélection aléatoire). (b) Comparaison expérimentale des 4 méthodologies de conception montrant que le compromis entre l'exploration et l'exploitation fonctionne mieux à chaque itération que les autres stratégies pour trouver le composé avec les plus grandes électrodéformations. (c) Prédictions. (d) Les électrodéformations prédites et mesurées des nouveaux composés synthétisés sont en accord raisonnable et donner confiance dans la qualité du modèle d'inférence. [26]

Bibliographies

[1] L. Wei, X. Xu, Gurudayal, J. Bullock, and J. W. Ager, "Machine Learning Optimization of p-Type Transparent Conducting Films," *Chemistry of Materials*, vol. 31, no. 18, pp. 7340–7350, Sep. 2019, doi: 10.1021/acs.chemmater.9b01953.

[2] B. Cao *et al.*, "How to optimize materials and devices via design of experiments and machine learning: Demonstration using organic photovoltaics," *ACS Nano*, vol. 12, no. 8. American Chemical Society, pp. 7434–7444, Aug. 28, 2018. doi: 10.1021/acsnano.8b04726.

[3] Y. Zhang and C. Ling, "A strategy to apply machine learning to small datasets in materials science," *NPJ Comput Mater*, vol. 4, no. 1, Dec. 2018, doi: 10.1038/s41524-018-0081-z.

[4] L. Wei, X. Xu, Gurudayal, J. Bullock, and J. W. Ager, "Machine Learning Optimization of p-Type Transparent Conducting Films," *Chemistry of Materials*, vol. 31, no. 18, pp. 7340–7350, Sep. 2019, doi: 10.1021/acs.chemmater.9b01953.

[5] Z. Zhao, Y. Geng, A. Troisi, and H. Ma, "Performance Prediction and Experimental Optimization Assisted by Machine Learning for Organic Photovoltaics," *Advanced Intelligent Systems*, vol. 4, no. 6, p. 2100261, Jun. 2022, doi: 10.1002/aisy.202100261.

[6] S. Pruksawan, G. Lambard, S. Samitsu, K. Sodeyama, and M. Naito, "Prediction and optimization of epoxy adhesive strength from a small dataset through active learning," *Sci Technol Adv Mater*, vol. 20, no. 1, pp. 1010–1021, Dec. 2019, doi: 10.1080/14686996.2019.1673670.

[7] R. Yuan *et al.*, "Accelerated Discovery of Large Electrostrains in BaTiO3-Based Piezoelectrics Using Active Learning," *Advanced Materials*, vol. 30, no. 7, Feb. 2018, doi: 10.1002/adma.201702884.

[8] N. Meftahi, M. Klymenko, A. J. Christofferson, U. Bach, D. A. Winkler, and S. P. Russo, "Machine learning property prediction for organic photovoltaic devices," *NPJ Comput Mater*, vol. 6, no. 1, Dec. 2020, doi: 10.1038/s41524-020-00429-w.

[9] Y. Cui *et al.*, "Over 16% efficiency organic photovoltaic cells enabled by a chlorinated acceptor with increased open-circuit voltages," *Nat Commun*, vol. 10, no. 1, Dec. 2019, doi: 10.1038/s41467-019-10351-5.

[10] X. Du *et al.*, "Elucidating the Full Potential of OPV Materials Utilizing a High-Throughput Robot-Based Platform and Machine Learning," *Joule*, vol. 5, no. 2, pp. 495–506, Feb. 2021, doi: 10.1016/j.joule.2020.12.013.

[11] M.-H. Lee, "Performance and Matching Band Structure Analysis of Tandem Organic Solar Cells Using Machine Learning Approaches," *Energy Technology*, vol. 8, no. 3, p. 1900974, Mar. 2020, doi: 10.1002/ente.201900974.

[12] R. Fisher, "The Design of Experiments, ," *5th ed. (Oliver & Boyd, Oxford, 1949).*.

[13] G.E.P. Box and K.B. Wilson, "On the Experimental Attainment of Optimum Conditions," *Imperial Chemical Industries*, 1992.

[14] R. N. T. Kackar, " Off-line quality control, parameter design, and the Taguchi method.," *J. Qual. Technol.17, 176–188 (1985)*.

[15] D. Xue, P. V. Balachandran, J. Hogden, J. Theiler, D. Xue, and T. Lookman, "Accelerated search for materials with targeted properties by adaptive design," *Nat Commun*, vol. 7, Apr. 2016, doi: 10.1038/ncomms11241.

[16] I. Izonin, R. Tkachenko, M. Gregus, K. Zub, and N. Lotoshynska, "Input doubling method based on SVR with RBF kernel in clinical practice: Focus on small data," in *Procedia Computer Science*, Elsevier B.V., 2021, pp. 606–613. doi: 10.1016/j.procs.2021.03.075.

[17] S. Bhatti *et al.*, "Machine learning for accelerating the discovery of high performance low-cost solar cells: a systematic review," Dec. 2022, [Online]. Available: http://arxiv.org/abs/2212.13893

[18] A. Mahmood and J.-L. Wang, "A time and resource efficient machine learning assisted design of non-fullerene small

molecule acceptors for P3HT-based organic solar cells and green solvent selection," *J Mater Chem A Mater*, vol. 9, no. 28, pp. 15684–15695, 2021, doi: 10.1039/D1TA04742F.

[19] V. L. Deringer, A. P. Bartók, N. Bernstein, D. M. Wilkins, M. Ceriotti, and G. Csányi, "Gaussian Process Regression for Materials and Molecules," *Chemical Reviews*, vol. 121, no. 16. American Chemical Society, pp. 10073–10141, Aug. 25, 2021. doi: 10.1021/acs.chemrev.1c00022.

[20] J. Quiñonero, Q. Quiñonero-Candela, C. E. Rasmussen, and C. M. De, "A Unifying View of Sparse Approximate Gaussian Process Regression," 2005.

[21] E. Xing and @ Cmu, "School of Computer Science Probabilistic Graphical Models Learning one Learning one-node GM node GM Reading: Learning Graphical Models Given set of independent samples (assignments of random variables), find the best (the most likely?) Bayesian Network (both DAG and CPDs)," 2009.

[22] E. O. Pyzer-Knapp, G. N. Simm, and A. Aspuru Guzik, "A Bayesian approach to calibrating high-throughput virtual screening results and application to organic photovoltaic materials," *Mater Horiz*, vol. 3, no. 3, pp. 226–233, 2016, doi: 10.1039/C5MH00282F.

[23] D. Lizotte, T. Wang, M. Bowling, and D. Schuurmans, "Gaussian Process Regression for Optimization."

[24] T. Lookman, P. V. Balachandran, D. Xue, and R. Yuan, "Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design," *npj Computational Materials*, vol. 5, no. 1. Nature Publishing Group, Dec. 01, 2019. doi: 10.1038/s41524-019-0153-8.

[25] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active Learning with Statistical Models," 1996.

[26] B. Settles, "Computer Sciences Department Active Learning Literature Survey," 2009.

[27] Jean-Dominique Quinet, « L'Active Learning, une stratégie efficace pour diminuer le coût et le temps du travail préparatoire de vos données. » Onsi, 20, oct. 2020.

Replication and Extension of Schnappinger's Study on Human-level Ordinal Maintainability Prediction Based on Static Code Metrics

Sébastien Bertrand [1,2,4], Silvia Ciappelloni [4], Pierre-Alexandre Favier [1,2,3], Jean-Marc André [1,2,3]

[1] Université de Bordeaux
[2] IMS Laboratory
[3] ENSC, Bordeaux INP
[4] onepoint, Sud-Ouest

s.bertrand@groupeonepoint.com

Résumé

Dans le cadre d'un projet de recherche sur l'évaluation de la maintenabilité des logiciels en collaboration avec l'équipe de développement, nous avons voulu explorer les dissensions entre les développeurs et le facteur confondant de la taille. A cette fin, cette étude a reproduit et étendu une étude récente de Schnappinger et al. avec la partie publique de son jeu de données et les métriques extraites de l'outil basé sur les graphes Javanalyser. L'ensemble du processus de traitement a été automatisé, de l'extraction des métriques à l'entraînement des modèles d'apprentissage automatique. L'étude a été étendue en prédisant la maintenabilité continue pour prendre en compte les dissensions. Puis, tous les entraînements ont été dupliqués pour évaluer l'influence globale de la taille de la classe. Au final, l'étude originale a été reproduite avec succès. De plus, de bonnes performances ont été obtenues pour la prédiction de la maintenabilité continue. Enfin, la taille de la classe n'était pas suffisante pour une prédiction fine de la maintenabilité. Cette étude montre la nécessité d'explorer la nature de ce qui est mesuré par les métriques du code. Elle constitue également la première étape dans la construction d'un modèle de maintenabilité.

Mots-clés

Maintenabilité des logiciels, prédiction de la maintenabilité, classification ordinale, jugement d'expert, apprentissage automatique.

Abstract

As part of a research project concerning software maintainability assessment in collaboration with the development team, we wanted to explore dissensions between developers and the confounding effect of size. To this end, this study replicated and extended a recent study from Schnappinger et al. with the public part of its dataset and the metrics extracted from the graph-based tool Javanalyser. *The entire processing pipeline was automated, from metrics extraction to the training of machine learning models. The study was extended by predicting the continuous maintainability to take account of dissensions. Then, all experimental shots were duplicated to evaluate the overall influence of the class size. In the end, the original study was successfully replicated. Moreover, good performance was achieved on the continuous maintainability prediction. Finally, the class size was not sufficient for fine-grained maintainability prediction. This study shows the necessity to explore the nature of what is measured by code metrics, and is also the first step in the construction of a maintainability model.*

Keywords

Software Maintainability, Maintainability Prediction, Ordinal Classification, Expert Judgment, Machine Learning.

1 Introduction

According to the ISO 25010 [20], software maintainability is defined as the efficiency with which the development team can fix a defect within a software, or implement an evolution according to a change of the needs. Software maintenance represents an important part of software development costs in time [5, 3, 10, 33, 9]. Assessing the maintainability of a piece of software is therefore important to ensure the control of these costs.

Our research project concerns software maintainability assessment in collaboration with the development team. Our goal is to pinpoint maintainability problems within a *Java* program, while taking into account team preferences and potential disagreements between developers.

Most studies use static code metrics to predict software maintainability [1, 12]. Very often, they study the source code at the class-level, because it is both a convenient and sizeable way to talk about code [35]. One group of studies uses the number of changed code lines as a proxy for the maintenance effort [22, 25, 29, 31, 39, 40]. Another group of studies is based on the prediction of expert assessments [18, 19, 34, 38]. However, the relation between the number of changed code lines and maintainability is not established [35]. Kafura and Reddy even states that developers tend to avoid modification of complex part of the program during maintenance [21]. That is why, in this work, we focused on prediction of expert assessments.

A study from Schnappinger *et al.* [37] explore human-level maintainability prediction based on static code metrics, specifically the distinction between easy and hard-to-maintain code and a more fine-grained ordinal classification problem. This study is based on a recent high quality software maintainability dataset [35, 36], also built by Schnappinger *et al.* . This dataset targets *Java* classes and represents maintainability as an ordinal four-part scale percentage agreement. However, only the open-source part of the dataset is publicly available. Additionally, due to a very large number of existing tools [30, 2] and their lack of agreement [4, 28, 8], the extraction of metrics from the dataset is quite hazardous. In addition, the original study uses numerous tools, including unpublished ones [37]. So to rationalize the extraction of metrics, we chose to use *Javanalyser* [4] which we developed to extract metrics solely based on the structure of the source code. *Javanalyser*[1] is an open-source graph-based static code analysis tool that leverages declarative programming as formal definition of metrics. Moreover, the study only reports performance results, but the implementation to train the machine learning models is not available. Therefore, our first goal was to check we could independently replicate the findings of Schnappinger *et al.* with a full open-source setup available to the community, including *Javanalyser* for metrics extraction.

Then, the study from Schnappinger *et al.* models maintainability prediction as an ordinal classification problem. The maintainability category of a *Java* class is determined by taking the most probable among experts as the response variable. However, while building their dataset, Schnappinger *et al.* notes that disagreement between experts is frequent [35]. To be able to work in collaboration with development teams, we wanted to predict potential disagreement. This is why we chose to additionally study a continuous representation of software maintainability.

Finally, the class size in lines of code is highly correlated to many metrics [11, 13, 23, 26, 41] and is one of the most used metric for maintainability prediction [37]. From an engineering point of view, this is very surprising because poor maintainability seems intuitively more connected to a poorly built control flow or data flow. Moreover, modern integrated development environments make it easy to extract code from a class to a new one. Therefore, class size may be an unreliable measure to detect complex maintainability problems. So, we tried to assess its overall influence when predicting maintainability with this dataset.

In summary, the research questions replicated from the original study are :

- **RQ1** : How well can classifiers distinguish between easy and hard-to-maintain code ?
- **RQ2** : How good is the performance of the machine learning models considering an ordinal label ?

The extension part of our study is related to these two research questions :

- **RQ3** : How good is the performance of the machine learning models considering continuous maintainability ?
- **RQ4** : What is the overall influence of class size on maintainability prediction ?

In this study, we extracted a total of 33 metrics from the dataset with *Javanalyser*, and we replicated the study of Schnappinger *et al.* following the same overall method. Then we implemented an additional setup using regressors instead of classifiers to test performance on continuous maintainability. Finally, we duplicated the experiments with only the class size as input, to assess its overall influence. Lastly, we launched a total of almost 20 000 experimental shots, each consisting of 150 individual machine learning trainings. Along this study, we logged every training results to help potential further research. To allow for the definition and evaluation of new metrics and maintainability models, we open-sourced the implementation and the results of our experiments under the MIT licence (see section 3).

2 Method

This section begins by presenting the core replication of Schnappinger *et al.* 's study, that is the dataset and the base experimental setup. Then the extension is presented with the prediction of a continuous maintainability and the size experiments.

2.1 Replication

2.1.1 Dataset

The present study uses the public part of Schanappinger's dataset [35, 36]. Basically, it consists of a listing of 304 classes with their source code, drawn from 5 open-source projects : *ArgoUML*[2], *Art of Illusion*[3], *Diary Management*[4], *JUnit 4*[5], and *JSweet*[6]. Each *Java* class is labelled along 5 axis : readability, understandability, complexity, modularity and overall maintainability. Each *Java* class was assessed by several experts independently of its relation to other classes.

As the original study, this work focused on the overall maintainability. It is represented as a set of 4 probabilities corresponding to the evaluation of the experts on a 4-class Likert-scale. Each probability corresponds to the ratio over the total for this *Java* class. The original study refers to these probabilities with the labels *Class-A*, *Class-B*, *Class-C*, and *Class-D*, where *Class-A* is highly maintainable and *Class-D* is poorly maintainable. The table 1 shows that this dataset is heavily unbalanced, both in terms of maintainability labels distribution and project data points. For instance, *Diary Management* has only 11 data points, none with a *Class-D* maintainability assessment.

The extraction of metrics was done with *Javanalyser* [4]. Some minor modifications of the source code of *ArgoUML*

1. https://gitlab.com/onepoint/research/javanalyser

2. https://github.com/argouml-tigris-org/argouml

3. http://www.artofillusion.org/

4. https://sourceforge.net/projects/diarymanagement/

5. https://junit.org/junit4/

6. http://www.jsweet.org/

TABLE 1 – Dataset distribution

Project	Class-A	Class-B	Class-C	Class-D	**Total**
ArgoUML	34	25	11	4	74
Art of Illusion	10	20	25	18	73
Diary Management	7	2	2	0	11
JUnit 4	60	12	1	0	73
JSweet	63	5	2	3	73
Total	174	64	41	25	**304**

and *Art of Illusion* were necessary to make them parsable. Every modification is provided on the public repository as *diff* files. *Javanalyser* builds the graph of the code by folding and simplifying the abstract syntax tree, and then queries that graph to extract metrics. As a result, it computes a slightly different version of class size in lines of code, which is the number of statements (NOS), that is a formal count of statement nodes within the code graph. It is worth noting that even such a simple metric has many variants among metric tools [4], whereas *Javanalyser* implements formal definition of metrics.

2.1.2 Experimental setup from Schnappinger *et al.*

Here is presented a short version of the experimental setup of the study of Schnappinger *et al.* [37]. Their study presents two classification problems :

— A binary separation problem to distinguish between easy and hard-to-maintain code, where *Class-A* and *Class-B* are considered to be maintainable, whereas *Class-C* and *Class-D* are not ;

— An ordinal classification problem to predict the majority class assigned by the experts.

The original study preprocesses metrics (features) in three ways. First, oversampling is used to account for the unbalanced dataset. As Schnappinger *et al.* , k-means-SMOTE was used with the implementation described in [24] and set *k=2* due to the small dataset. Second, normalisation or standardisation of features is applied to potentially improve performance. Third, feature selection based on the mutual information between the metrics and the target is leveraged to select only a subset of available features. Each of these preprocessing techniques was implemented as a conditional setting and can be combined.

In their pre-study, Schnappinger *et al.* identify the six most promising algorithms for their setup [37] : Gradient Boosting [15], Ada-Boost [17], Extremely Randomized Trees [16], Logistic Regression, Random Forests [7] and the K-Nearest Neighbor classifier. Additionally, for the ordinal classification problem, they use 7 metamodels based on these base classifiers (or their regressor versions). One is the binary decomposition proposed by Frank and Hall [14]. The others are proposed by Schnappinger *et al.* : three chained binary classifiers, two probabilities classifiers and a rounded regressor classifier. For comparison, a baseline classifier that always predicts the majority class was implemented for each problem. All these classifiers were implemented using *scikit-learn* [32].

The original study uses many performance metrics to evaluate the models. The binary classification problem reports its results with the F-score, and the Area Under the receiver operating characteristic Curve (AUC) which is considered as a better choice for binary classification [6]. The ordinal classification problems uses micro-averaged accuracy (ACC), Cohen's Kappa (Cκ) and Matthews Correlation Coefficient (MCC). It also uses Mean-Square Error (MSE) by defining all intervals on the ordinal scale to be 1.

Every experiment was fully configured by a CSV file, which also stored the results. Each experiment was randomly sampled to explore the hyperparameters space without being too time-consuming. In total, $20,000$ experiments were sampled. Each experiment was repeated with 30 different random seeds. Like the original study, project-wise cross-validation was used, However the performance depends heavily on the chosen test project, because the dataset is heavily unbalanced between projects. That is why, a shuffled stratified 5-fold cross-validation was implemented for better statistical comparison. The reported results correspond to an average of these runs along the standard deviation.

2.2 Extension

2.2.1 Continuous maintainability

After the core replication, the problem of the continuous maintainability prediction was added. Continuous maintainability is defined as the expected value of each maintainability class. The scores ranging from 0 for *Class-A* to 3 for *Class-D* were assigned. This method allowed us to take into account disagreement between experts, for instance a continuous maintainability of 2.5 may correspond to half the experts assigning *Class-C* and the other half *Class-D*. Disagreement between experts could help to diagnose non-trivial maintainability problem.

For this problem, a baseline classifier that always predicts whole dataset expected maintainability was built, which is a maintainability of 0.75. As the other problems, the Mean-Square Error (MSE) was used for comparison. The models were also evaluated against the Mean Absolute Error (MAE), the Median Absolute Error (MedAE) and the R^2 score (R^2).

2.2.2 Size experiments

Finally, the overall influence of class size on maintainability prediction was explored. For that purpose, each experiment was duplicated and fed only the class size to the models. Moreover, as the variance with the project-wise cross-validation was quite important, the shuffled stratified 5-fold cross-validation was necessary for a better statistical com-

TABLE 2 – Results of the project-wise binary separation

Classifier	F-Score	AUC
Average expert	0.88	0.83
Baseline	0.87 ±0.14	0.50 ±0
Baseloc	0.92 ±0.07	0.67 ±0.17
Original study	0.91	0.82
This study	0.93 ±0.06	0.90 ±0.10

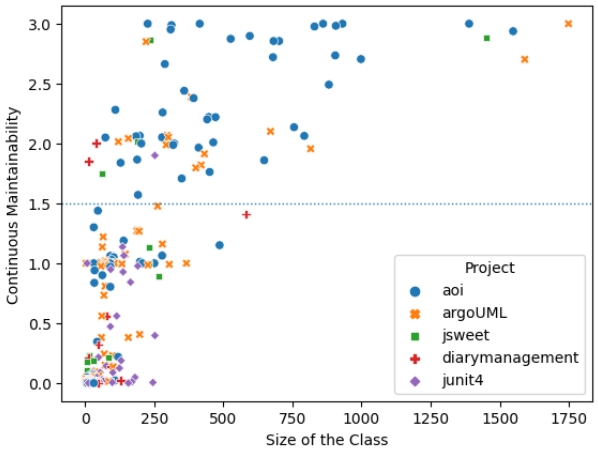


FIGURE 1 – The influence of size on maintainability

parison.
To truly challenge the prediction power of the trained models, some additional baseline classifier were implemented, hereafter referred to as *'baseloc'* classifiers. They adapt their prediction in function of the class size. For instance, for the binary separation problem, the baseloc classifier predicts good maintainability if the size is under 275 lines of code, otherwise it predicts poor maintainability.

3 Results

This section presents the best results of nearly 20 000 experimental shots, each consisting of 150 individual machine learning trainings, that is 5 folds (project-wise or stratified) times 30 seeds. The dataset, the metrics extracted, the processing code, and all the results are available under the free (as in freedom) MIT licence at `https://gitlab.com/onepoint/research/maintainability-dataset-analysis`.
All our results are presented with the notation *mean* ±*std*, *std* being the standard deviation. As the original study, the tables 2, 3, 4, 5 present the best score that was obtained by a classifier, that is two scores on the same row may not have been performed by the same classifier.

The table 2 presents the replication of the first problem from Schnappinger *et al.* 's study. The table 3 presents the replication of the ordinal classification. Like in the original paper, the measured performance depends heavily on the test project. That is why, k-fold cross validation results were included to overcome differences between projects.

Continuous maintainability ranges from 0 to 3, lower being better. As shown by the figure 1, the increase of class size clearly set a floor value for maintainability. The table 4 presents the results of continuous maintainability prediction. The performances of a perfect ordinal classifier is included for comparison with the ordinal classification problem. This problem was not tested by Schnappinger *et al.* 's study.

Finally, the table 5 presents the comparison of the performance of models when fed with only the class size *vs.* all metrics. This table is meant to be read vertically to compare baseline, baseloc, size-only and all-metrics classifiers for each problem. For simplicity, only the k-fold case is presented, the variability on the project-wise case being too large.

4 Discussion

This section begins by discussing class-level maintainability prediction problems. Then, disagreement between experts will be addressed as they are at the heart of these problems. Finally, the overall influence of class-size will shed a new light on the way forward.

4.1 Class-Level Maintainability Prediction

As shown by tables 2 and 3, the study of Schnappinger *et al.* [37] was successfully replicated with the metrics extracted by *Javanalyser*. On the binary separation problem, the best classifiers outperformed the baseline and baseloc classifiers, and an average expert. On the ordinal classification problem, the best classifiers outperformed the baseline and baseloc classifiers, and reached human-level performance. Moreover, concerning the continuous maintainability problem extension, the best classifiers outperformed the baseline and baseloc classifiers. There is no data concerning an average expert on this last problem. Thus, with respect to the first three research questions (**RQ1**, **RQ2**, **RQ3**), trained models performed well on binary separation, ordinal classification and continuous maintainability regression (respectively).
However, the addition of the standard deviation showed that the observed performance with project-wise cross-validation was highly unstable. As a reminder, the standard deviation needs to be tripled to give a 99% confidence interval. This was due to the unbalanced dataset (see table 1), indeed a k-fold cross-validation approach shows better deviations. In fact, oversampling cannot compensate for imbalance between projects when using project-wise cross-validation. Moreover, the table 3 shows that the results are very close to the original study with a k-fold cross-validation setting. The original study includes 9 projects within its dataset (5 open-source and 4 closed-source). To reach production-grade standard deviations, the training dataset would need *Java* classes drawn from more projects, until no differences are observed between project-wise and k-fold cross-validation.

4.2 Dissent Between Experts

The dataset is built with a four-part Likert scale to not overwhelm the experts [35]. However, when a class is globally maintainable, a minor problem could still be present

TABLE 3 – Results of the ordinal classification

Classifier	**ACC**	**MSE**	**Cκ**	**MCC**
Average expert	0.70	0.41	0.53	0.53
Project-wise				
Baseline	0.58 ±0.27	1.39 ±1.30	0 ±0	0 ±0
Baseloc	0.68 ±0.15	0.53 ±0.43	0.37 ±0.23	0.38 ±0.24
Original study	0.73	0.31	0.51	0.53
This study	0.74 ±0.12	0.39 ±0.32	0.46 ±0.18	0.47 ±0.18
K-Fold				
Baseline	0.57 ±0.01	1.49 ±0.03	0 ±0	0 ±0
Baseloc	0.73 ±0.05	0.36 ±0.08	0.54 ±0.08	0.54 ±0.08
Original study	0.75	0.30	0.60	0.60
This study	0.77 ±0.04	0.27 ±0.06	0.61 ±0.07	0.61 ±0.07

TABLE 4 – Results of the continuous maintainability regression

Classifier	**MSE**	**MAE**	**MedAE**	R^2
Project-wise				
Baseline	0.85 ±0.50	0.79 ±0.18	0.82 ±0.20	-0.67 ±0.66
Baseloc	0.39 ±0.25	0.43 ±0.15	0.24 ±0.07	0.21 ±0.46
This study	0.28 ±0.22	0.33 ±0.14	0.19 ±0.12	0.41 ±0.44
K-Fold				
Baseline	0.91 ±0.04	0.80 ±0.02	0.73 ±0.01	0 ±0.01
Baseloc	0.29 ±0.07	0.37 ±0.05	0.19 ±0.01	0.68 ±0.08
This study	0.18 ±0.05	0.26 ±0.04	0.10 ±0.04	0.80 ±0.06
Perfect ordinal	0.02	0.08	0.02	0.98

TABLE 5 – Size-Only vs All-Metrics k-fold results comparison

Classifier	**Binary sep.** (AUC)	**Ordinal class.** (MSE)	**Continuous maint.** (MSE)
Average expert	0.83	0.41	—
Baseline	0.50 ±0	1.49 ±0.03	0.91 ±0.04
Baseloc	0.84 ±0.05	0.36 ±0.08	0.29 ±0.07
Size-Only	0.95 ±0.03	0.35 ±0.08	0.27 ±0.06
All-Metrics	0.97 ±0.02	0.27 ±0.06	0.18 ±0.05

within it. For instance a method could be slightly too long and should be split in two. Such problems would lead to disagreement between expert, some assigning the *Class-A* and some the *Class-B*.

On a continuous scale, the expected maintainability of a class subject to disagreement is between two integers. This disagreement occurs quite often as shown by the figure 1. When building their dataset, Schnappinger *et al*. report disagreement between experts in 73.4% of the ratings [35]. By measuring the distance between ratings, they estimate that significant disagreement occurs in 17.2% of the cases, and strong disagreement occurs in 1.2% of the cases.

A perfect ordinal classifier would have a MSE of 0.02 on the continuous maintainability prediction (and reciprocally). This shows that these problems are not far apart. However, the MSE significantly improves from the ordinal classification problem to the continuous maintainability problem. This shows that predicting contentious cases is more effective than predicting the winner (the majority class). Despite this fact, a coarse binary separation to detect the most problematic classes is very effective. This can be seen on the figure 1 by drawing a horizontal line going through the middle of the continuous maintainability scale (1.5).

4.3 The Overall Influence of Class-Size

Size-only models show surprisingly effective performances (see table 5). In fact on the binary separation and the ordinal classification, size-only classifiers outperformed an average expert. On the binary separation problem, other metrics only marginally improve performances. Concerning the forth research question (**RQ4**), this confirms that class size is a very effective metric to predict maintainability.

There is no consensus on the confounding effect of class size [11, 13, 23]. Kitchenham argues that each line of code as a whole has the same probability of exhibiting a defect [23]. Nevertheless, table 5 shows other metrics help to improve performances on the ordinal classification and continuous maintainability problems. Thus, whatever the correlation with class size, other metrics tend to be useful for finer problems.

On that matter, Lemberger and Morel states that aggregation of metrics is not natural [27]. From that point of view, it is not obvious to compute the class-level cyclomatic complexity by summation of the complexity of its methods. It would clearly be misleading to define the intelligence quotient of a team by the sum of the intelligence quotients of its members. The definition of scale-invariant metrics would be a necessary step to assess the influence of each of them.

5 Conclusions

This study successfully replicated the study of Schnappinger *et al*. on human-level ordinal maintainability prediction [37]. However, the standard deviation remained very high for the project-wise ordinal classification problem. All the processing code, the dataset and the results are publicly available to allow further research. [7] The extension of the study with the continuous maintainability problem shows that all baseline and baseloc classifiers are outperformed by trained models. Finally, the analysis showed that models trained with only the class size are very efficient to detect coarse-grained maintainability problems and surprisingly outperformed an average expert. However, the class size is not enough when the problem is sufficiently complex, like the ordinal classification or the continuous maintainability. This study shows that there is a need to better design the maintainability prediction problem and to better define metrics in order to go beyond class-level analysis and to be able to pinpoint maintainability problems within classes.

Future works include building size-robust datasets with classes from many projects. Designing better scale-invariant metrics for static code analysis will be essential to go further in maintainability prediction and analysis. Then, executing controlled experiments to assert the individual influence of these metrics over the maintainability would be very insightful to complement the experimental data collected in real situations. In the end, such metrics should be part of a wider maintainability model.

Acknowledgments

We thank our collaborators at *onepoint* [8] for their insightful advices, in particular Alexandra Delmas, Jérôme Fillioux, Denis Maurel, Sylvain Métayer, and Guillaume Meurisse.

Références

[1] Hadeel Alsolai and Marc Roper. A systematic literature review of machine learning techniques for software maintainability prediction. *Information and Software Technology*, 119 :106214, March 2020.

[2] Luca Ardito, Riccardo Coppola, Luca Barbato, and Diego Verga. A Tool-Based Perspective on Software Code Maintainability Metrics : A Systematic Literature Review. *Scientific Programming*, 2020 :1–26, August 2020.

[3] Rajiv D. Banker, Srikant M. Datar, Chris F. Kemerer, and Dani Zweig. Software complexity and maintenance costs. *Communications of the ACM*, 36(11) :81–94, November 1993. https://doi.org/10.1145/163359.163375.

[4] Sébastien Bertrand, Pierre-Alexandre Favier, and Jean-Marc André. Building an operable graph representation of a Java program as a basis for automatic software maintainability analysis. In *EASE '22 : Proceedings of the International Conference on Evaluation and Assessment in Software Engineering 2022*, EASE 2022, pages 243–248, Gothenburg, Sweden, June 2022. Association for Computing Machinery. https://doi.org/10.1145/3530019.3534081.

7. https://gitlab.com/onepoint/research/maintainability-dataset-analysis

8. https://www.groupeonepoint.com/

[5] Barry W. Boehm, John R. Brown, and M. Lipow. Quantitative evaluation of software quality. In Raymond T. Yeh and C. V. Ramamoorthy, editors, *Proceedings of the 2nd International Conference on Software Engineering, San Francisco, California, USA, October 13-15, 1976*, pages 592–605. IEEE Computer Society, 1976. http://dl.acm.org/citation.cfm?id=807736.

[6] Andrew P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7) :1145–1159, July 1997. https://www.sciencedirect.com/science/article/pii/S0031320396001422.

[7] Leo Breiman. Random Forests. *Machine Learning*, 45(1) :5–32, October 2001. https://doi.org/10.1023/A:1010933404324.

[8] Dennis Breuker, Jacob Brunekreef, Jan Derriks, and Ahmed Nait Aicha. Reliability of software metrics tools. page 14, November 2009.

[9] Celia Chen, Reem Alfayez, Kamonphop Srisopha, Barry Boehm, and Lin Shi. Why Is It Important to Measure Maintainability and What Are the Best Ways to Do It ? In *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, pages 377–378, May 2017.

[10] Don Coleman, Bruce Lowther, and Paul Oman. The application of software maintainability models in industrial software systems. *Journal of Systems and Software*, 29(1) :3–16, April 1995.

[11] Khaled El Emam, S. Benlarbi, N. Goel, and S. N. Rai. The confounding effect of class size on the validity of object-oriented metrics. *IEEE Transactions on Software Engineering*, 27(7) :630–650, July 2001.

[12] Sara Elmidaoui, Laila Cheikhi, Ali Idri, and Alain Abran. Empirical Studies on Software Product Maintainability Prediction : A Systematic Mapping and Review. *e-Informatica Software Engineering Journal*, 13(1) :141–202, 2019.

[13] W.M. Evanco. Comments on "The confounding effect of class size on the validity of object-oriented metrics". *IEEE Transactions on Software Engineering*, 29(7) :670–672, July 2003.

[14] Eibe Frank and Mark Hall. A Simple Approach to Ordinal Classification. In Luc De Raedt and Peter Flach, editors, *Machine Learning : ECML 2001*, volume 2167, pages 145–156, Freiburg, Germany, 2001. Springer.

[15] Jerome H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4) :367–378, February 2002. https://www.sciencedirect.com/science/article/pii/S0167947301000652.

[16] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1) :3–42, April 2006. https://doi.org/10.1007/s10994-006-6226-1.

[17] Trevor Hastie, Saharon Rosset, Ji Zhu, and Hui Zou. Multi-class AdaBoost. *Statistics and Its Interface*, 2(3) :349–360, 2009. https://www.intlpress.com/site/pub/pages/journals/items/sii/content/vols/0002/0003/a008/abstract.php.

[18] Péter Hegedűs, Tibor Bakota, László Illés, Gergely Ladányi, Rudolf Ferenc, and Tibor Gyimóthy. Source Code Metrics and Maintainability : A Case Study. In Tai-hoon Kim, Hojjat Adeli, Haeng-kon Kim, Heaujo Kang, Kyung Jung Kim, Akingbehin Kiumi, and Byeong-Ho Kang, editors, *Software Engineering, Business Continuity, and Education*, Communications in Computer and Information Science, pages 272–284, Berlin, Heidelberg, 2011. Springer.

[19] Péter Hegedűs, Gergely Ladányi, István Siket, and Rudolf Ferenc. Towards Building Method Level Maintainability Models Based on Expert Evaluations. In Tai-hoon Kim, Carlos Ramos, Haeng-kon Kim, Akingbehin Kiumi, Sabah Mohammed, and Dominik Ślęzak, editors, *Computer Applications for Software Engineering, Disaster Recovery, and Business Continuity*, Communications in Computer and Information Science, pages 146–154, Berlin, Heidelberg, 2012. Springer.

[20] ISO/IEC. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. Standard ISO/IEC 25010 :2011, ISO/IEC, March 2011. https://www.iso.org/standard/35733.html.

[21] Dennis Kafura and Geereddy R. Reddy. The Use of Software Complexity Metrics in Software Maintenance. *IEEE Transactions on Software Engineering*, SE-13(3) :335–343, March 1987.

[22] Arvinder Kaur and Kamaldeep Kaur. Statistical comparison of modelling methods for software maintainability prediction. *International Journal of Software Engineering and Knowledge Engineering*, 23(06) :743–774, August 2013. https://www.worldscientific.com/doi/abs/10.1142/S0218194013500198.

[23] Barbara Kitchenham. What's up with software metrics ? - A preliminary mapping study. *Journal of Systems and Software*, 83(1) :37–51, 2010.

[24] György Kovács. Smote-variants : A python implementation of 85 minority oversampling techniques. *Neurocomputing*, 366 :352–354, November 2019. https://www.sciencedirect.com/science/article/pii/S0925231219311622.

[25] Lov Kumar, Debendra Kumar Naik, and Santanu Kumar Rath. Validating the Effecti-

veness of Object-Oriented Metrics for Predicting Maintainability. *Procedia Computer Science*, 57 :798–806, January 2015. https://www.sciencedirect.com/science/article/pii/S1877050915020086.

[26] Meir Manny Lehman, Dewayne E. Perry, and Juan F. Ramil. Implications of evolution metrics on software maintenance. In *1998 International Conference on Software Maintenance, ICSM 1998, Bethesda, Maryland, USA, November 16-19, 1998*, page 208. IEEE Computer Society, 1998.

[27] Pirmin Lemberger and Médéric Morel. Two Measures of Code Complexity. pages 195–206. January 2013.

[28] Valentina Lenarduzzi, Fabiano Pecorelli, Nyyti Saarimäki, Savanna Lujan, and Fabio Palomba. A Critical Comparison on Six Static Analysis Tools : Detection Agreement and Precision. *SSRN Electronic Journal*, 2022. https://www.ssrn.com/abstract=4044439.

[29] Wei Li and Sallie Henry. Object-Oriented Metrics that Predict Maintainability. *Journal of Systems and Software*, 23(2) :111–122, November 1993. http://www.sciencedirect.com/science/article/pii/016412129390077B.

[30] Rüdiger Lincke, Jonas Lundberg, and Welf Löwe. Comparing software metrics tools. In *Proceedings of the 2008 International Symposium on Software Testing and Analysis - ISSTA '08*, page 131, Seattle, WA, USA, 2008. ACM Press.

[31] Ruchika Malhotra and Kusum Lata. An empirical study on predictability of software maintainability using imbalanced data. *Software Quality Journal*, 28(4) :1581–1614, December 2020. https://doi.org/10.1007/s11219-020-09525-y.

[32] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn : Machine Learning in Python. *Journal of Machine Learning Research*, 12(85) :2825–2830, 2011. http://jmlr.org/papers/v12/pedregosa11a.html.

[33] Markus Pizka and Thomas Panas. Establishing Economic Effectiveness through Software Health-Management. May 2009. https://www.semanticscholar.org/paper/Establishing-Economic-Effectiveness-through-Pizka-Panas/744c02e1cce4aa3228ace764bb0d97029dd33303.

[34] Nicolino J. Pizzi, Arthur R. Summers, and Witold Pedrycz. Software Quality Prediction Using Median-Adjusted Class Labels. In *Proceedings of the 2002 International Joint Conference on Neural Networks*, volume 3, pages 2405–2409 vol.3, Honolulu, HI, USA, May 2002.

[35] Markus Schnappinger, Arnaud Fietzke, and Alexander Pretschner. Defining a Software Maintainability Dataset : Collecting, Aggregating and Analysing Expert Evaluations of Software Maintainability. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 278–289, Adelaide, Australia, September 2020. IEEE.

[36] Markus Schnappinger, Arnaud Fietzke, and Alexander Pretschner. A Software Maintainability Dataset. https://figshare.com/articles/dataset/A_Software_Maintainability_Dataset/12801215/3, 2020.

[37] Markus Schnappinger, Arnaud Fietzke, and Alexander Pretschner. Human-level Ordinal Maintainability Prediction Based on Static Code Metrics. In *EASE 2021 : Evaluation and Assessment in Software Engineering*, pages 160–169, Trondheim, Norway, June 2021. ACM.

[38] Markus Schnappinger, Mohd Hafeez Osman, Alexander Pretschner, and Arnaud Fietzke. Learning a classifier for prediction of maintainability based on static analysis tools. In *Proceedings of the 27th International Conference on Program Comprehension*, ICPC '19, pages 243–248, Montreal, Quebec, Canada, May 2019. IEEE Press. https://doi.org/10.1109/ICPC.2019.00043.

[39] Chikako van Koten and Andrew Gray. An application of Bayesian network for predicting object-oriented software maintainability. *Information and Software Technology*, 48(1) :59–67, January 2006. https://www.sciencedirect.com/science/article/pii/S0950584905000339.

[40] Yuming Zhou and Hareton Leung. Predicting object-oriented software maintainability using multivariate adaptive regression splines. *Journal of Systems and Software*, 80(8) :1349–1361, August 2007. https://www.sciencedirect.com/science/article/pii/S0164121206003372.

[41] Yuming Zhou, Baowen Xu, Hareton Leung, and Lin Chen. An in-depth study of the potentially confounding effect of class size in fault prediction. *ACM Transactions on Software Engineering and Methodology*, 23(1) :10 :1–10 :51, February 2014. https://doi.org/10.1145/2556777.