Explicabilité de séries temporelles : étude de cas sur des données de sécurité urbaine

Matthieu Delahaye*, Lina Fahed*
Florent Castagnino**, Philippe Lenca*

*IMT Atlantique, Lab-STICC, UMR CNRS 6285, F-29238 Brest, France
**IMT Atlantique, LEMNA, F-44307 Nantes, France
prénom.nom@imt-atlantique.fr

Résumé. Les méthodes d'explicabilité en intelligence artificielle (XAI) sont devenues essentielles pour mieux comprendre les modèles complexes notamment dans les domaines sensibles tels que la sécurité urbaine. Dans cet article, nous étudions un jeu de données public gouvernemental sur les crimes et délits commis en France et mettons en application une méthode d'explicabilité à base de shapelets, des sous-séquences distinctives qui ressortent les informations pertinentes de séries temporelles d'infractions. Après avoir transformé le problème pour une tâche de classification supervisée, nous cherchons à faire apparaître des motifs caractéristiques de la criminalité. Néanmoins, la qualité et la disponibilité des données représentent un défi majeur pour la performance des modèles et leur interprétation. Ces conclusions illustrent le besoin d'ouvrir davantage les ressources et les données afin de mieux comprendre et analyser les phénomènes de sécurité urbaine, un milieu encore trop peu accessible.

1 Introduction

Les modèles d'apprentissage automatique, en particulier ceux dits "boîtes noires", se sont imposés dans divers domaines grâce à leurs résultats remarquables. Cependant, leur complexité croissante, motivée par la recherche de performances, a entraîné un manque de transparence, limitant ainsi la compréhension de leurs utilisateurs. Face à cette problématique, le domaine de l'Intelligence Artificielle Explicable (XAI) a émergé, donnant naissance à diverses méthodes d'interprétation et d'explication des résultats des modèles (Adadi et Berrada, 2018).

L'explicabilité est un enjeu dans des domaines critiques tels que la défense, la médecine ou la sécurité, où les décisions ont des conséquences majeures. Dans ce contexte, les utilisateurs/contributeurs peuvent souhaiter appréhender les modèles d'apprentissage automatique d'un bout à l'autre, à la recherche d'une cohérence entre leurs entrées et sorties (Barredo Arrieta et al., 2020). Ce besoin de transparence est aussi influencé par le type de données traitées. Si la recherche en explicabilité s'est surtout orientée sur des données intuitives comme les images et le texte, les séries temporelles représentent un défi supplémentaire : leur interprétation n'est pas instinctive, un signal qui évolue dans le temps nous est bien moins familier que des images ou des mots (Rojat et al., 2021). Pourtant, les séries temporelles, qui représentent l'objet de notre étude, sont omniprésentes dès lors que la notion de temps entre en jeu.

Dans la littérature, nous distinguons plusieurs familles de méthodes d'XAI, notamment, les méthodes *post-hoc* qui expliquent les résultats ou la structure des modèles d'apprentissage, et les méthodes *ante-hoc* qui sont transparentes par conception (Guidotti et al., 2019). Une distinction est faite sur la nature du modèle d'apprentissage : les méthodes *agnostiques* sont adaptées à n'importe quel type de modèles indépendamment de son architecture, tandis que d'autres sont *spécifiques* à une famille de modèles. Par ailleurs, certaines méthodes d'XAI sont développées pour un type de données particulier alors que d'autres sont plus génériques.

Nous nous focalisons sur les méthodes d'XAI appliquées aux séries temporelles. L'approche des shapelets, introduite par Ye et Keogh (2009), se distingue par sa capacité à extraire des sous-séquences représentatives et discriminantes des séries temporelles. Elle offre la possibilité de fournir des explications sous forme de motifs directement issus des données brutes. L'algorithme *Learning Shapelets* de Grabocka et al. (2014) améliore cette approche en intégrant un cadre d'apprentissage qui évite de devoir parcourir l'ensemble des sous-séquences au sein des séries, tout en combinant précision et interprétabilité.

Nous appliquons l'algorithme *Learning Shapelets* à un domaine à fort impact social et particulièrement sensible, à savoir la sécurité urbaine, qui nécessite des modèles à la fois performants et interprétables. **La contribution de cet article réside dans le traitement étendu de données de criminalité et de délinquance et l'identification des motifs temporels, de type shapelets, les plus représentatifs afin d'obtenir de nouvelles clés de compréhension du domaine complexe de la sécurité urbaine.**

Dans la section 2, nous passons en revue les travaux principaux de l'état de l'art sur les approches d'XAI pour les séries temporelles. Nous introduisons notre cas d'étude dans le section 3, puis, dans la section 4, nous présentons l'application du modèle d'XAI retenu. Nous discutons des résultats de notre étude dans la section 5. Enfin, dans la section 6, nous résumons les apports de l'étude et présentons les perspectives de recherche futures.

2 État de l'art

Nous présentons ci-dessous une étude de travaux en XAI portants sur les séries temporelles ainsi que des définitions inspirées de l'état de l'art.

Selon Rojat et al. (2021), l'explication devient nécessaire lorsque la tâche à accomplir dépasse les capacités d'un modèle simple et interprétable, tout en étant trop importante pour être confiée à un modèle opaque dont les décisions seraient difficiles à justifier et à accepter. Pour répondre à ce besoin, l'XAI cherche à fournir des explications des modèles et/ou des résultats afin de permettre aux utilisateurs de se les approprier au mieux.

L'explicabilité des séries temporelles pose des défis importants en raison de leur nature séquentielle et non intuitive, ainsi que, par des relations temporelles complexes qui existent entre chaque point de mesure de la série (Rojat et al., 2021). Certaines méthodes, utilisées principalement sur des données intuitives (*e.g.* texte, images), ont pu être réadaptées pour des séries temporelles (Ge et al., 2018; Vinayavekhin et al., 2018; Kashiparekh et al., 2019). Par ailleurs, d'autres approches se sont développées pour répondre directement aux défis des séries temporelles notamment en identifiant des motifs et des tendances au sein des séries (Lin et al., 2007; Ye et Keogh, 2009). Nous explorons diverses méthodes d'XAI appliquées aux séries temporelles (voir définition 2.1), en débutant par celles développées sur d'autres types de données, puis par celles conçues pour les séries temporelles.

Définition 2.1 Une **série temporelle** $T = \{x_1, x_2, \ldots, x_M\} \in \mathbb{R}^{M \times D}$ est un ensemble ordonné de *M* observations réelles de dimension *D*. Un jeu de données $J = \{T_1, T_2, \ldots, T_N\}$ est une collection de *N* séries temporelles où $J \in \mathbb{R}^{N \times M \times D}$. Lorsque $D = 1$ les séries temporelles sont dites univariées tandis que lorsque $D > 1$ les séries sont dites multivariées.

À partir de modèles de classification tels que les *Convolutional Neural Network (CNN)*, les *Recurrent Neural Network (RNN)* ou les *Transformers*, diverses méthodes d'XAI se sont développées. Pour les CNN, citons la méthode de perturbation *ConvTimeNet* de Kashiparekh et al. (2019) qui se base sur l'*"occlusion sensitivity"*. En perturbant les données par occlusion, il est possible de retrouver le niveau de contribution des parties de l'image à sa classification par le modèle. Des modèles de traitement de texte tels que les *Long Short Term Memory (LSTM)* sont particulièrement efficaces en raison de leur capacité à traiter les données de manière séquentielle. Par exemple, l'étude de Ge et al. (2018) combine un CNN avec un LSTM, puis, un réseau de neurones feedforward est utilisé pour la classification, dont les poids de sa dernière couche indiquent l'importance de chaque observation de la série temporelle. Des mécanismes d'attention, utilisés dans les *transformers* notamment, sont aussi proposés (Vinayavekhin et al., 2018) afin de concentrer le modèle sur les parties importantes de la série temporelle. Cependant, différentes attentions peuvent aboutir au même résultat risquant de donner des explications trompeuses (Jain et Wallace, 2019). Des techniques comme *Local Interpretable Model-agnostic Explanation (LIME)* (Ribeiro et al., 2016) et *SHapley Additive Explanations (SHAP)* (Lundberg et Lee, 2017) peuvent être adaptées pour traiter les séries temporelles selon chaque observation réelle (voir définition 2.1). Cependant, prendre en compte chaque point de la série temporelle comme une variable présente un défi : l'hypothèse de l'indépendance des variables n'est pas respectée dû aux observations temporellement adjacentes de la série (Watson, 2022). Ces méthodes présentent donc plusieurs limites pour les séries temporelles compliquant l'interprétation de leurs résultats.

D'autres approches comme l'explicabilité par l'exemple (Ming et al., 2019) ou par contrefactuels (Delaney et al., 2021), ont été développées, mais présentent des contraintes : (i) elles sont basées sur l'intuition humaine ce qui est parfois mis en difficulté par la nature même des séries temporelles et (ii) la capacité à fournir un exemple interprétable par l'utilisateur dépend fortement de son niveau de compréhension (Theissler et al., 2022).

Les méthodes *features-based* (Ito et Chakraborty, 2019) permettent d'extraire des caractéristiques à partir des séries temporelles brutes de manière à en déduire un sens ou améliorer la précision du modèle. Néanmoins, ces méthodes ne sont pas forcément adaptées aux utilisateurs cibles et dépendent aussi de la capacité d'interprétabilité des caractéristiques.

En comparaison, des travaux sur des méthodes d'XAI conçues spécialement pour identifier les éléments distinctifs des séries temporelles suscitent beaucoup d'intérêt. Nous allons présenter deux exemples de ces méthodes.

Méthode SAX : La méthode *Symbolic Aggregate approXimation* (SAX) (Lin et al., 2007) discrétise une série temporelle en une suite de lettres. Des symboles (*i.e.* lettres) issus de la discrétisation sont assignés à chacun des segments et forment des mots. Ces mots représentent des motifs dont nous pouvons analyser la fréquence afin d'en tirer une interprétation.

Shapelets : Les méthodes d'extraction de shapelets introduites par Ye et Keogh (2009) ainsi que celle de Grabocka et al. (2014) ont retenu notre attention.

Définition 2.2 Un **shapelet** de taille L est une séquence ordonnées de valeurs. Sémantiquement, le shapelet a une valeur informative importante car il doit permettre de séparer facilement les classes au sein du jeu de données. Les K shapelets les plus informatifs sont notés $S \in \mathbb{R}^{K \times L}$.

Dans la méthode initiale de Ye et Keogh (2009), des sous-séquences des séries temporelles sont d'abord récupérées, et définies comme possible *candidat* à appartenir à S puis, la distance entre chaque *candidat* et toutes les sous-séquences de même taille dans chaque série est mesurée. La distance minimale, caractéristique de la meilleure correspondance entre une série et un shapelet, devient la distance finale entre l'intégralité de chacune des séries temporelles et chacun des candidats. Les shapelets finaux sont les *candidats* qui maximisent le gain d'information de séparation des classes (voir définition 2.2). L'algorithme *Learning Shapelets* de Grabocka et al. (2014) intègre une fonction d'apprentissage pour définir les shapelets, évitant ainsi le traitement l'ensemble des sous-séquences ce qui réduit significativement les temps de calcul. De par ses qualités de performance et d'interprétabilité, la méthode *Learning Shapelets* est, pour nous, la plus appropriée pour notre cas d'étude.

3 Cas d'étude : sécurité urbaine

Notre étude dans le domaine de la sécurité urbaine s'inscrit dans le cadre d'un projet ANR [1] dont un objectif est d'aider les services de police et de gendarmerie à accepter davantage les outils d'intelligence artificielle en leur apportant une compréhension des phénomènes de criminalité et de délinquance à l'aide de modèles d'apprentissage automatique explicable. Ce travail est mené avec des experts en sciences sociales travaillant sur les phénomènes de criminalité.

En 1972, le gouvernement français met en place l'"État 4001", un outil de mesures statistiques mensuelles des crimes et des délits [2]. Ces mesures correspondent aux plaintes et signalements enregistrés par les services de police et de gendarmerie nationales à travers les départements français selon 107 catégories d'infractions entre 1996 et 2022, distribuées via un jeu de données public. Dans la suite de cette section, nous décrivons ce jeu de données ainsi que son adaptation à une tâche de classification supervisée.

3.1 Description du jeu de données

Dans le jeu de données de l'État 4001 [3], les 107 catégories de crimes et de délits sont transformées en 10 variables : *Homicides, Vols, Cambriolages/violation de domicile, Crimes/délits sexuels, Infractions liées aux stupéfiants, Dégradations/destructions de biens volontaires, Violences, Falsifications et contrefaçons, Délits Économiques et Financiers* et *Irrégularités de Main-d'œuvre*. Les données sont séquencées pour chaque année en 12 mois et par département selon les 10 variables et forment un ensemble de 2193 séries temporelles dans un jeu de données $J \in \mathbb{R}^{2193 \times 12 \times 10}$ (voir définition 2.1).

1. https://anr.fr/Projet-ANR-21-CE26-0023
2. https://mobile.interieur.gouv.fr/Interstats/Sources-et-methodes-statistiques/Glossaire/Etat-4001
3. https://www.data.gouv.fr/fr/datasets/chiffres-departementaux-mensuels-relatifs-aux-crimes-et-delits-enregistres-par-les-services-de-police-et-de-gendarmerie-depuis-janvier-1996/

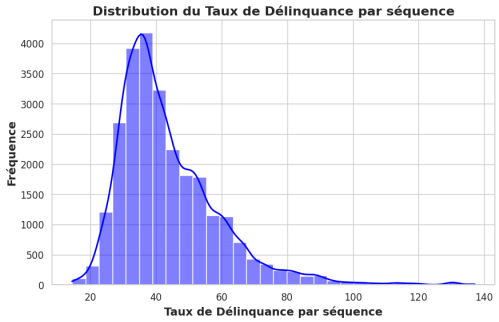


FIG. 1 – Distribution du taux de délinquance pour l'ensemble des séries temporelles.

3.2 Adaptation pour une tâche de classification

Après concertation avec les experts en sciences sociales travaillant sur la criminalité, nous souhaitons obtenir une prédiction du niveau de délinquance pour chaque département. L'adaptation du jeu de données (création d'une variable cible) pour une tâche de classification supervisée permet de répondre à l'objectif d'explicabilité par l'identification de motifs temporels distinctifs liés avec le niveau de criminalité. Nous mesurons le taux de délinquance de chacune des séries temporelles, qui correspond au rapport entre, la somme des valeurs de chaque catégorie de délits/crimes, et de la population du département sur l'année de chaque série, le tout multiplié par un facteur 1000. Les séries temporelles représentent des séquences annuelles, nous prenons la somme des catégories des 12 observations mensuelles de chaque série afin d'avoir un résultat à l'année.

La distribution du taux de délinquance des séries temporelles (voir figure 1), est asymétrique avec une longue queue vers la droite qui témoigne d'un grand nombre de valeur élevées éloignées de la moyenne. L'hypothèse de normalité est rejetée par les tests de Shapiro Wilk et de Kolmogorov-Smirnov.

Afin de définir les classes pour labelliser nos séries temporelles, nous avons étudié l'adaptabilité de deux méthodes de discrétisation du taux de délinquance et une méthode ensembliste :

— *Répartition par intervalles égaux : n* intervalles de même taille sont définis par rapport à l'entendue des valeurs de notre variable cible *i.e.* taux de délinquance, puis, une étiquette est assignée en fonction de chaque intervalle. Cependant, la distribution du taux de délinquance n'étant pas symétrique, la répartition par intervalles égaux risque de former des classes déséquilibrées et biaiser les résultats de la classification.

— *Clustering par K-Means* : assigne une étiquette pour un nombre k de clusters. Néanmoins, les clusters sont définis à partir de l'ensemble des variables d'intérêts, donc sur un espace multidimensionnel, ce qui complexifie leur interprétation.

— *Répartition par quantiles :* le jeu de données est découpé en n parties proportionnelles, avec le même nombre de séries temporelles par partie, chaque partie se voit assigner un label. Cette répartition s'adapte à la distribution du taux de délinquance afin de produire des classes équilibrées.

La différence de proportion de séries temporelles selon chaque méthode de labellisation est mise en évidence figure 2 pour une répartition en 4 classes. Les méthodes de répartition par

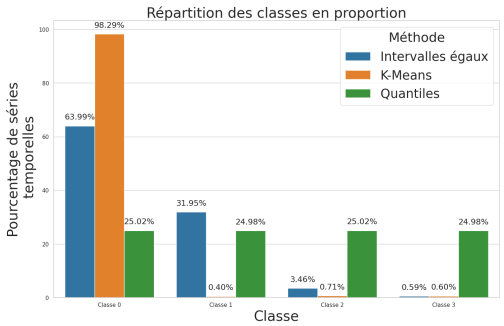


FIG. 2 – Comparaison de la proportion de séries temporelles de chaque classe selon trois méthodes de labellisation.

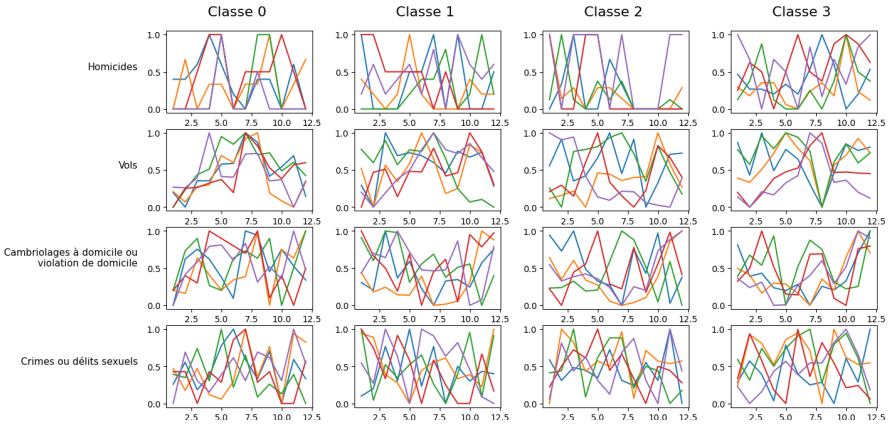


FIG. 3 – Représentation des séries temporelles propres à 4 classes selon différentes variables.

intervalles égaux et d'utilisation de K-Means entraînent un déséquilibre important des classes comparé à la répartition par quantiles dont la répartition est similaire pour chaque classe. De plus, avec la répartition par quantiles, chaque classe est définie selon des seuils fixes (*i.e.* valeur min et max par classe) aisément compréhensibles par l'utilisateur. Grâce à son adaptation à toute distribution et sa facilité d'explication, la répartition par quantiles est l'approche la plus adaptée à notre cas d'étude. Dans la suite de l'étude, les séries temporelles seront réparties selon 4 classes, ce qui revient à diviser en quartiles. Cela permet une analyse comparative entre les classes facilitant l'interprétation.

Le jeu de données est découpé en un jeu d'entraînement (80%), validation (10%) et de test (10%). Comme nous cherchons à comparer les formes relatives des séries temporelles, les séries sont standardisées indépendamment sur une moyenne nulle et un écart-type unitaire en premier lieu sur le jeu d'entraînement puis sur le jeu de test, puis normalisées sur la plage [0,1] afin de simplifier les calculs de distances et l'interprétation globale.

Dans la figure 3, pour chaque classe, les séries sont représentées selon différentes variables. Ainsi, nous remarquons que la série temporelle rouge de la classe 2 a un pic marqué sur chaque

variable au mois de mai, significatif d'une hausse de criminalité. Néanmoins, aucune tendance ne se distingue sur les variables, c'est-à-dire que nous ne retrouvons pas de motif spécifique à une catégorie. Cela peut être dû à la taille réduite des séries temporelles étudiées (*i.e.* 12 observations). La répartition des classes étant faite par quartiles, nous pouvons interpréter graduellement chaque classe par niveau de délinquance : la classe 0 étant caractéristique du niveau de délinquance le plus faible. Maintenant que les données sont labellisées et pré-traitées, nous pouvons appliquer l'algorithme retenu *Learning Shapelets*.

4 Déploiement du modèle

Nous présentons ici le déploiement de l'algorithme *Learning Shapelets* puis l'entraînement d'un modèle de classification supervisée sur les données transformées par l'algorithme.

Learning Shapelets : *Learning Shapelets* de Grabocka et al. (2014) est un algorithme d'apprentissage basé sur la méthode des shapelets de Ye et Keogh (2009). Les shapelets sont déterminés (voir définition 2.2) à partir d'une fonction objectif. Nous utilisons le package Python *tslearn* (Tavenard et al., 2020) pour déployer l'algorithme de *Learning Shapelets*.

Définition 4.1 Distance entre un shapelet et une série temporelle : La distance entre la n-ième série temporelle T_n et le k-ième shapelet S_k correspond à la distance minimale $M_{i,k}$ entre S_k et les différents segments i extraits de T_n, afin de calculer la ressemblance entre un shapelet et le segment de la série qui est le plus proche de ce dernier (Grabocka et al., 2014).

Définition 4.2 Transformation de l'espace de données : Les distances minimales aux shapelets permettent de transformer les séries temporelles $T \in \mathbb{R}^{N \times M}$ en une nouvelle représentation $M \in \mathbb{R}^{N \times K}$ (Lines et al., 2012; Grabocka et al., 2014)

Fonctionnement de *Learning Shapelets* : Tout d'abord, un algorithme K-Means est appliqué à l'ensemble des segments des séries temporelles brutes. Les centroïdes représentent les shapelets initiaux. Un nombre de shapelets par taille est déterminé par l'algorithme *Learning Shapelets*, en fonction du nombre de séries temporelles et de classes.

Pour chaque shapelet, la distance euclidienne minimale à tous les segments de chaque série temporelle est mesurée (voir définition 4.1). La fonction *min* n'étant pas différentiable, elle est remplacée par une fonction différentiable approximée *soft-min*, afin d'utiliser un algorithme d'optimisation pour le modèle supervisé. Par la suite, les séries temporelles sont transformées comme présenté définition 4.2, puis, une régression logistique est appliquée sur le nouvel espace afin de classifier les séries. La fonction de perte associée est une *log-loss* régularisée. Un algorithme d'optimisation met à jour les valeurs des shapelets et des poids du modèle linéaire. Pour la suite de l'étude, nous avons choisi l'algorithme Adam (Kingma, 2014), plus performant que la descente de gradient stochastique.

Récupération des attributs : L'algorithme est déployé pour chaque variable indépendamment, ainsi, les shapelets appris correspondent à une seule variable. Les espaces transformés propres à chaque variable sont concaténés de manière à former un seul et même jeu de données transformé pour l'ensemble des variables. En complément, pour chaque variable, nous

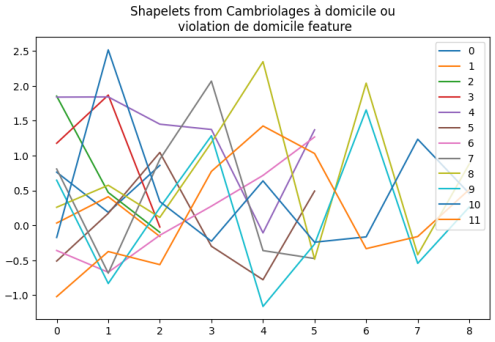


FIG. 4 – Ensemble des 12 shapelets de la variable “Cambriolages à domicile ou violation de domicile” résultant de l’algorithme *Learning Shapelets* : 4 shapelets de taille 3, 4 shapelets de taille 6 et 4 shapelets de taille 9.

récupérons la forme de chaque shapelet, les paramètres de la dernière couche d’apprentissage du modèle responsable de la classification et la localisation de la sous-séquence dont la distance est minimale avec le shapelet. Ces attributs vont nous aider à interpréter les résultats par la suite. L’ensemble des 12 shapelets générés sur la variable “Cambriolages à domicile ou violation de domicile” sont visibles sur la figure 4.

Classification à partir de l’espace transformé : Nous constatons que *Learning Shapelets* n’arrive pas à séparer nettement les classes entre elles, les séries temporelles transformées forment un ensemble de points qui se superposent sans groupes distincts (voir figure 5). Pour cette raison, et après avoir étudié plusieurs modèles de classification, nous optons pour le modèle *Random Forest* (Breiman, 2001).

5 Résultats et discussions

Dans cette section, nous décrivons les shapelets obtenus, les résultats de classification du modèle ainsi que les interprétations qui en découlent.

Limites de décision : *Learning Shapelets* génère des shapelets de façon à ce que : (i) leurs distances avec les séries temporelles soient séparées linéairement en fonction de leur classe, (ii) les distances avec les séries temporelles de chaque classe soient regroupées ensemble et éloignées de celles des séries des autres classes. Pour chaque variable, nous pouvons visualiser ces concepts en utilisant les paramètres de la couche de classification de *Learning Shapelets*.

L’analyse de l’importance de caractéristiques du *Random Forest* permet d’identifier les shapelets les plus influents pour une variable. La figure 5 présente les deux shapelets les plus significatifs : shapelets 5 et 6 (représentés en haut à gauche de la figure) de la variable “Vols”. Ils sont de même taille mais capturent une tendance différente. Les séries temporelles de chaque classe sont représentées sur des graphiques au milieu et en bas gauche de la figure 5. Bien

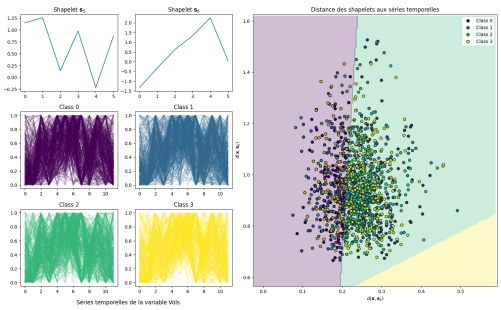


FIG. 5 – Distances entre les séries temporelles et les shapelets 5 *(abscisses)* et 6 *(ordonnées)* correspondant à la variable “Vols”.

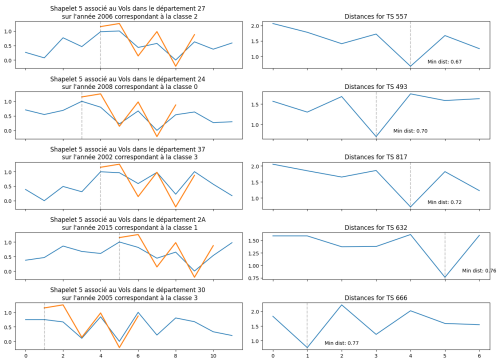


FIG. 6 – Localisation des séries temporelles dont les distances sont les plus proches avec le cinquième shapelet de la variable “Vols”.

qu’une tendance d’observations élevées entre les mois de mai et de juillet se démarque pour l’ensemble des séries, nous ne distinguons pas de forme particulière au sein des classes.

Le graphique à droite de la figure 5 permet de visualiser les séries temporelles transformées selon leurs distances aux deux shapelets (*i.e.* shapelet 5 en abscisses et 6 en ordonnées). À l’aide des paramètres de la couche de classification, nous pouvons tracer les limites de décision linéaires visibles sur le graphique à droite de la figure 5. Ces paramètres rendent compte de l’importance des distances de chaque shapelet avec l’ensemble des séries. Ils reflètent la contribution de chaque shapelet à la séparation des classes dans l’espace transformé ce qui donne une piste d’interprétation sur l’influence des shapelets identifiés. Cependant, les points de la classe 0 se retrouvent à la fois dans les limites de la classe 0 et celles de la classe 2. Le phénomène est similaire pour les points de la classe 2. De plus, aucun point de la classe 3 ne se retrouve dans sa limite de décision. Les données se regroupent en un ensemble de points, avec des séries de chaque classe qui se superposent, ce qui complique la séparation linéaire.

Localiser les correspondances des shapelets avec les séries temporelles : La figure 6 illustre le shapelet 5 associé à la variable “Vols”, le shapelet dont le score d’importance est

le plus élevé dans le *Random Forest*. Ce shapelet, de taille 6, traduit une tendance sur une période 6 mois. Les graphiques à gauche de la figure 6 représentent les 5 séries temporelles (en bleu) dont les distances sont les plus courtes avec le shapelet étudié (en orange). La forme du shapelet est globalement similaire à celle des segments des séries temporelles représentées. Les graphiques à droite correspondent à la distance entre le shapelet et chaque segment de la série temporelle étudiée. Par exemple, pour la première série en haut du graphique, la distance est minimale au cinquième point, (mesuré sur le cinquième segment de la série temporelle). Ce point est marqué par une ligne grise pointillée sur les deux graphes de chaque série.

Par ailleurs, pour les quatre premières séries temporelles, la distance est minimale sur des segments proches : 3, 4 et 5, couvrant la période de avril à novembre. Nous cherchons à savoir si cette période est significative et caractérisée par d'autres informations : année, département et classe de la série temporelle. Les résultats sont peu concluants : les années et les départements sont dispersés. Nous manquons d'informations pour déterminer la période ou le lieu représentatif de ce shapelet. Plus étonnement, aucune classe ne se distingue alors que l'objectif de *Learning Shapelets* est justement de déterminer des shapelets caractéristiques d'une classe. Nous pouvons conclure que les shapelets appris par l'algorithme ne ressortent pas d'informations distinctives des séries temporelles, ce qui complique l'interprétation des motifs obtenus. Le modèle semble être très indécis sur le choix de la classe (accuracy = 0,300). La matrice de confusion montre qu'il classifie presque uniformément les séries sur chacune des classes. Les shapelets générés n'ont donc pas séparé les classes efficacement.

6 Discussion et conclusion

Nous avons étudié un domaine à fort impact social, à savoir, la sécurité urbaine, dans l'objectif de mieux comprendre les phénomènes de criminalité et leurs tendances en appliquant un algorithme d'XAI à base de shapelets sur des données publiques. Nous avons adapté les données à une tâche de classification supervisée en introduisant une variable cible afin de rendre l'interprétation des shapelets plus accessible, notamment aux sociologues. Nous soulignons l'importance de rendre les données de sécurité urbaine accessibles et complètes, tout en favorisant le développement de méthodes d'explicabilité adaptées à d'autres contextes critiques.

Le principal défi de notre étude réside dans la quantité et la qualité des données disponibles. Le jeu de données provenant de l'"État 4001" représente uniquement les plaintes déposées, négligeant la délinquance non signalée. De plus, malgré la publication de données publiques, les ressources fournies sont limitées car très agrégées ce qui complique leur interprétation. L'autre limite de notre étude est la segmentation des séries temporelles par année qui fait perdre la dépendance temporelle entre les séries et empêche l'étude des tendances à long terme.

Pour répondre à ces défis, il est possible d'utiliser des jeux de données plus élaborés pour fiabiliser les résultats tout en conservant une analyse des tendances sur la durée, tel que la base de données couplant les mesures de l'"État 4001" avec des enquêtes de victimation [4] réalisées auprès des ménages. Cependant, les mesures sont disponibles par année uniquement ce qui contraint l'analyse précise des phénomènes de criminalité. Une solution est de s'intéresser aux pays dont la politique d'accessibilité est plus ouverte. La police de Toronto [5], par exemple,

4. Enquête Vécu et Ressenti en matière de Sécurité (VRS) : https ://mobile.interieur.gouv.fr/Interstats/L-enquete-Vecu-et-ressenti-en-matiere-de-securite-VRS

5. Données sur les infractions à Toronto : https ://data.torontopolice.on.ca/search ?collection=Dataset&q=crime

publie des données horaires sur la commission des infractions au sein de la ville, permettant des analyses à granularité temporelle variable.

Nous envisageons le déploiement de la méthode de classification non-supervisée par *unsupervised shapelet* (Zakaria et al., 2012) qui permet de s'affranchir de la dépendance à une classe. Afin d'apporter une compréhension plus globale au modèle, la détermination de shapelets sur des séries multivariées (Medico et al., 2021) ou bien l'intégration de connaissances au modèle (Wan et al., 2024) sont des perspectives judicieuses.

Références

Adadi, A. et M. Berrada (2018). Peeking inside the black-box : A survey on explainable artificial intelligence (XAI). IEEE Access 6, 52138–52160.

Barredo Arrieta, A., N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, et F. Herrera (2020). Explainable artificial intelligence (XAI) : Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion 58, 82–115.

Breiman, L. (2001). Random forests. Machine Learning 45, 5–32.

Delaney, E., D. Greene, et M. T. Keane (2021). Instance-based counterfactual explanations for time series classification. In International conference on case-based reasoning, pp. 32–47.

Ge, W., J.-W. Huh, Y. R. Park, J. H. Lee, Y. Kim, et A. Turchin (2018). An interpretable ICU mortality prediction model based on logistic regression and recurrent neural networks with LSTM units. AMIA Symposium 2018, 460–469.

Grabocka, J., N. Schilling, M. Wistuba, et L. Schmidt-Thieme (2014). Learning time-series shapelets. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 392–401.

Guidotti, R., A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, et D. Pedreschi (2019). A survey of methods for explaining black box models. ACM Computing Surveys 51(5), 1–42.

Ito, H. et B. Chakraborty (2019). A proposal for shape aware feature extraction for time series classification. In 2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST), pp. 1–6.

Jain, S. et B. C. Wallace (2019). Attention is not Explanation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1, pp. 3543–3556.

Kashiparekh, K., J. Narwariya, P. Malhotra, L. Vig, et G. Shroff (2019). Convtimenet : A pre-trained deep convolutional neural network for time series classification. In 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–8.

Kingma, D. P. (2014). Adam : A method for stochastic optimization. arXiv preprint arXiv :1412.6980.

Lin, J., E. Keogh, L. Wei, et S. Lonardi (2007). Experiencing SAX : A novel symbolic representation of time series. Data Min. Knowl. Discov. 15, 107–144.

Lines, J., L. M. Davis, J. Hills, et A. Bagnall (2012). A shapelet transform for time series classification. In Proceedings of the 18th ACM SIGKDD, pp. 289–297.

Lundberg, S. M. et S.-I. Lee (2017). A unified approach to interpreting model predictions. In Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 4768–4777.

Medico, R., J. Ruyssinck, D. Deschrijver, et T. Dhaene (2021). Learning multivariate shapelets with multi-layer neural networks for interpretable time-series classification. Advances in Data Analysis and Classification 15(4), 911–936.

Ming, Y., P. Xu, H. Qu, et L. Ren (2019). Interpretable and steerable sequence learning via prototypes. In Proceedings of the 25th ACM SIGKDD, pp. 903–913.

Ribeiro, M. T., S. Singh, et C. Guestrin (2016). "Why Should I Trust You ?" : Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD, pp. 1135–1144.

Rojat, T., R. Puget, D. Filliat, J. Del Ser, R. Gelin, et N. Díaz-Rodríguez (2021). Explainable artificial intelligence (XAI) on TimeSeries data : A survey. arXiv preprint arXiv :2104.00950.

Tavenard, R., J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, K. Kolar, et E. Woods (2020). Tslearn, a machine learning toolkit for time series data. Journal of Machine Learning Research 21(118), 1–6.

Theissler, A., F. Spinnato, U. Schlegel, et R. Guidotti (2022). Explainable AI for time series classification : A review, taxonomy and research directions. IEEE Access 10, 100700–100724.

Vinayavekhin, P., S. Chaudhury, A. Munawar, D. J. Agravante, G. De Magistris, D. Kimura, et R. Tachibana (2018). Focusing on what is relevant : Time-series learning and understanding using attention. In 2018 24th Int. Conf. on Pattern Recognition (ICPR), pp. 2624–2629.

Wan, X., L. Cen, X. Chen, Y. Xie, et W. Gui (2024). Prior knowledge-augmented unsupervised shapelet learning for unknown abnormal working condition discovery in industrial process. Advanced Engineering Informatics 60, 102429.

Watson, D. S. (2022). Conceptual challenges for interpretable machine learning. Synthese 200(2), 65.

Ye, L. et E. Keogh (2009). Time series shapelets : a new primitive for data mining. In Proceedings of the 15th ACM SIGKDD, pp. 947–956.

Zakaria, J., A. Mueen, et E. Keogh (2012). Clustering time series using unsupervised-shapelets. In 2012 IEEE 12th International Conference on Data Mining, pp. 785–794.

Summary

Explainability methods in artificial intelligence (XAI) have become essential for better understanding complex patterns, particularly in sensitive areas such as urban security. In this article, we study a governmental public dataset on crimes committed in France and implement an explainability method based on shapelets, distinctive subsequences that extract relevant information from time series of offenses. After transforming the problem for a supervised classification task, we aim to reveal patterns characteristic of crime. Nevertheless, data quality and availability represent a major challenge for model performance and interpretation. These findings illustrate the need to further open up resources and data in order to better understand and analyze urban security phenomena, an area that is not sufficiently accessible.

Perspectives for Direct Interpretability in Multi-Agent Deep Reinforcement Learning

Abstract. Multi-Agent Deep Reinforcement Learning (MADRL) was proven efficient in solving complex problems in robotics or games, yet most of the trained models are hard to interpret. While learning intrinsically interpretable models remains a prominent approach, its scalability and flexibility are limited in handling complex tasks or multi-agent dynamics. This paper advocates for direct interpretability, generating post hoc explanations directly from trained models, as a versatile and scalable alternative, offering insights into agents' behaviour, emergent phenomena, and biases without altering models' architectures. We explore modern methods, including relevance backpropagation, knowledge edition, model steering, activation patching, sparse autoencoders and circuit discovery, to highlight their applicability to single-agent, multi-agent, and training process challenges. By addressing MADRL interpretability, we propose directions aiming to advance active topics such as team identification, swarm coordination and sample efficiency.

1 Introduction

The increasing complexity of agents trained by Reinforcement Learning (RL) has raised significant safety and ethical concerns (Mitelut et al., 2023; Shavit et al., 2023; Vishwanath et al., 2024). These considerations are even more crucial when training multiple agents based on Deep Neural Networks (DNNs), commonly referred to as black boxes, i.e., in Multi-Agent Deep Reinforcement Learning (Chelarescu, 2021). MADRL enables solving more complex problems through cooperation or opponent modelling (Hernandez-Leal et al., 2018; Gronauer and Diepold, 2021; Wong et al., 2021), and finds applications in robotics (Orr and Dutta, 2023), video games (Vinyals et al., 2019) or even health (Shaik et al., 2023). Recent advancements, such as pre-trained world models (Reed et al., 2022; Yang et al., 2023; Alonso et al.; Bruce et al., 2024) and the integration of Large Language Models (LLMs), as standalone agents (Wang et al., 2023) or within Multi-Agent Systems (MAS) (Wu et al., 2023; Li et al., 2024; Han et al., 2024), further exacerbate the interpretability challenge. While the field of eXplainable RL (XRL) is growing by the year (Heuillet et al., 2020; Milani et al., 2023; Qing et al., 2022; Hickling et al., 2022; Bekkemoen, 2023), with one of the first dedicated workshops organised at the first RL Conference edition (Kohler et al., 2024), interpretability is anecdotal in MADRL (Heuillet et al., 2021; Wang et al., 2021; Milani et al., 2022; Zabounidis et al., 2023; Mahjoub

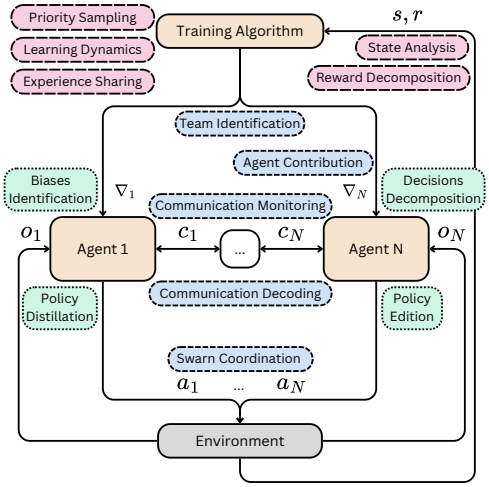


FIG. 1 – *Visual taxonomy of MADRL challenges that could benefit from direct interpretability methods. In green (dots) challenges related to a single agent, in blue (short dashes) to multiple agents and in red (long dashes) to the training process.*

et al., 2023; Khlifi et al., 2023). Yet, as we expose in Section 3, interpretability could help advance specific challenges in MADRL, such as team identification, swarm coordination and sample efficiency.

Existing efforts in agent interpretability predominantly focus on intrinsically interpretable models (Heuillet et al., 2020; Milani et al., 2023; Qing et al., 2022; Hickling et al., 2022; Bekkemoen, 2023), emphasising simplicity in architecture to make systems inherently understandable (Chattopadhyay et al., 2022; Rodriguez et al., 2024). However, these approaches often need to be revised for large and performant systems where expressiveness, scalability and flexibility are essential (Rudin et al., 2021). We thus propose to focus on direct interpretability, i.e., methods that are post-hoc, applicable after training, and generate explanations directly from DNNs. This class of methods enables probing complex systems without constraining their design or needing to extract an interpretable model. Inspired by modern interpretability methods (Zou et al., 2023; Cunningham et al., 2023; Dunefsky et al., 2024; Katz et al., 2024), and new XRL approaches (Levin and Chockler, 2023; Seong and Shim, 2024; Lange et al., 2024), we decided to anticipate the adoption of explainability in the expanding field of MADRL and advocate for a more systematic use and engagement with modern direct interpretability methods. We list our contributions as follows:

- Arguments to engage with direct interpretability methods.
- A taxonomy to position direct interpretability in MADRL.

In this article, we first present an initial background about the systems of study and the methods advocated. Then, we propose a simple taxonomy to position modern interpretability methods in the MADRL framework. Finally, we outline the limitations of some current works while proposing alternative ideas tracks.

2 Background

2.1 Multi-Agent Deep Reinforcement Learning

A typical system consists of the following components: agents, an environment, and a training algorithm, as depicted in Figure 2. Formally, we consider a system with N agents, each indexed by $i \in \{1, \ldots, N\}$. At each time step, the agent i is presented with an observation o_i and produces an action a_i. For the sake of generality, we included a possible communication channel c_i, seeing that it is increasingly used (Zhu et al., 2022). In principle, we can extend the definition of communication to include the most common MADRL methods like parameter sharing (Gupta et al., 2017; Chu and Ye, 2017), which can be seen as a form of latent space communication. Finally, the training algorithm provides feedback ∇_i to each agent.

Training algorithms in MADRL can be centralized, decentralized, or hybrid. Centralized training uses the joint action $a = (a_1, ..., a_N)$ and the state s, which can be understood as an observation augmented by information at training time (Lambrechts et al., 2023), and consists of applying classical RL to multi-agent problems like for AplhaStar (Mathieu et al., 2023). While decentralized training restricts each agent to local observations o_i, possibly including a local reward r_i, see IDQN (Tampuu et al., 2015) or IPPO (Yu et al., 2021). Hybrid approaches, such as centralized training with decentralized execution, leverage global information during training but allow agents to act independently using only local observations during execution, see VDN (Sunehag et al., 2017), QMIX (Rashid et al., 2018), MADPG (Lowe et al., 2017) or MAPPO (Yu et al., 2021). Here, we consider agents based on DNNs; therefore, the feedbacks ∇_i are gradients of a loss ℓ. Depending on the training algorithm, this loss can be a function of the reward r, the state s, the actions a_i, the observations o_i and the communications c_i. For simplicity, we didn't include those dependencies in Figure 2.

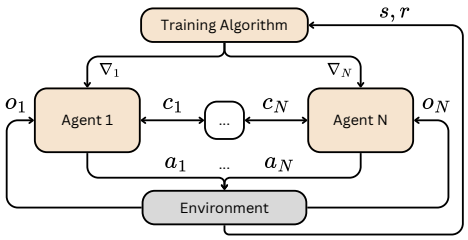


FIG. 2 – *Schema of a simplified view of MADRL systems. At each time step, the agent i receives the initial observation o_i, complemented by potential communications c_i and produces an action a_i. The agent learns throughout training by the means of gradients ∇_i.*

2.2 Direct Interpretability of DNNs

We now present an overview of the modern methods widely used to interpret DNNs in Computer Vision (CV) and Natural Language Processing (NLP). As these domains heavily relied on pre-trained models (Simonyan and Zisserman, 2014; He et al., 2015; Radford and

Narasimhan, 2018), direct post-hoc methods have dominated the research landscape, providing key hindsight without altering models' architectures.

Feature importance. Typical methods used in CV to understand convolutional networks involve visualising important pixels, i.e. saliency maps, (Zeiler and Fergus, 2013; Selvaraju et al., 2016). Other methods compute importance by perturbing the input (Covert et al., 2020), using the gradients (Radford et al., 2015; Selvaraju et al., 2016; Shrikumar et al., 2016; Smilkov et al., 2017) or locally decomposing relevance (Montavon et al., 2015; Bach et al., 2015). Recent works in NLP focus on the Transformer architecture and its attention mechanism (Vaswani et al., 2017), providing token-level insights (Wiegreffe and Pinter, 2019; Achtibat et al., 2024).

Prototypes: a class of methods that creates explanations based on characteristic samples. In CV, it is common to analyse neurons using activation maximisation to create pre-images (Mahendran and Vedaldi, 2015), or find related images (Chen et al., 2020). Prototypes can be of various forms like perturbed images (Ribeiro et al., 2018), cropped images (Dreyer et al., 2023a) or latent space vector (Alain and Bengio, 2018; Kim et al., 2018). Recent works based on sparse autoencoders were able to elicit interpretable features in LLMs, i.e., prototypes (Cunningham et al., 2023).

Latent manipulation: techniques that further extend the interpretability of concepts and features by exploring the internal representations learned by models. These methods were introduced in CV with (Kim et al., 2018), later derived as the field of representation engineering (Zou et al., 2023). Such latent features enable locating, editing, erasing or decoding models' knowledge (Meng et al., 2022; Belrose et al., 2023b; Ghandeharioun et al., 2024), but causally modify or analyse the produced outputs (Rimsky et al., 2023; Kram'ar et al., 2024).

Circuit analysis: provides a more granular understanding of model internals by examining pathways and dependencies between models' components, usually neurons or attention heads. Circuits were first discovered in CNNs (Olah et al., 2020) before being formalised for Transformers (Elhage et al., 2021). These circuits revealed peculiar models' components that learned precise mechanisms like induction (Olsson et al., 2022). Using specific datasets, relevant circuits can be automatically discovered (Conmy et al., 2023). More recent works focus on larger models' components at the layer scale (Dunefsky et al., 2024).

3 Advocating Direct Interpretability

Direct methods offer a significant advantage in their applicability to models during and after training, enabling developers to analyse and interpret complex systems without requiring architectural changes. This flexibility makes them particularly suitable for MADRL systems compared to intrinsic methods that might be challenging to scale with several agents. Figure 1 outlines speculative research directions and methodologies that can enhance systems understanding at different levels, from individual agents to the overall training process.

3.1 Single-Agent Challenges

To understand agents trained using MADRL, we can study each agent independently. Methods drawn from XRL and general interpretability are thus directly applicable to tackle single-agent challenges.

Biases identification:eliciting models' biases learned during training. In order to debug those "Clever Hans"[1], it is possible to use feature importance techniques, described in Section 2.2. Previous work (Lapuschkin et al., 2019) showed that this debugging could be semi-automated by combining LRP (Bach et al., 2015) with spectral clustering (von Luxburg, 2007). While these methods are relatively established in XRL, further improvements tailored to MADRL could offer more context-specific explanations, e.g., by comparing different agents' perceptions.

Policy distillation, converting a model into a simpler one, can be achieved by training a new smaller model (Rusu et al., 2015), or by extracting intrinsically interpretable models (Ross et al., 2010; Bastani et al., 2018), even for MADRL (Milani et al., 2022). Yet, these distillation methods are computationally expensive. Recent works proposed network compression based on interpretability, using weights relevance (Yeom et al., 2019) or circuit analysis (Pochinkov and Schoots, 2024).

Decision decomposition: could be achieved by internally decomposing an agent's decision into functional modules or representations. This methodology was proven efficient to elicit the algorithms behind certain capabilities, like addition or modular addition (Quirke and Barez, 2023; Nanda et al., 2023). Future work could focus on extracting different circuits using ACDC (Conmy et al., 2023) to analyse simple shared actor-critic architectures, e.g., to extract the actor subnetwork.

Policy edition,an essential aspect to regain control over DNNs. Indeed, being able to edit a trained policy is essential to remove biases, unwanted associations or dangerous behaviours without needing to retrain the model. In this respect, direct interpretability is perfectly suited for the task with methods leveraging CAVs (Dreyer et al., 2023b) or causal tracing (Meng et al., 2022).

3.2 Multi-Agent Challenges

Interpretability could be a powerful tool for automating the oversight of systems involving multiple agents. Indeed, such systems become more complex through inter-agent interactions, coordination strategies, and emergent behaviours.

Team identification: grouping together agents with similar roles or policies. This is particularly interesting to reduce the complexity of MAS by having fewer agents to train or could be an avenue to extend the mean-field framework (Yang et al., 2018). Previous work showed that selective parameter-sharing can be based on latent spaces (Christianos et al., 2021). Further improvements could consider dynamic teams throughout learning by analysing mixing networks (Rashid et al., 2018), e.g., by partitioning the positive weights using NMF (Paatero and Tapper, 1994), or using other prototype methods like SAE (Cunningham et al., 2023).

Agent contribution,or agent credit assignment, is a well-known challenge introduced by MAS. Shapley values theoretically give the individual agent contributions (Shapley, 1988), and thus can be computed using SHAP or equivalent methods (Lundberg and Lee, 2017; Heuillet et al., 2021; Wang et al., 2021). Yet, as it can be expensive to compute, it might be beneficial to explore other versatile methods like LRP (Bach et al., 2015), e.g., by designing specific relevance propagation rules.

[1]Cognitive bias that was learned due to spurious correlations, see (Lapuschkin et al., 2019).

Communication monitoring In settings with natural language communication between agents, leveraging LLMs or pre-trained models can enable a seamless integration (Zhu et al., 2022). Yet, these models are highly opaque and would benefit from interpretability, offering an avenue to supervise and interpret conversations. Applications could make use of feature importance methods, like AttnLRP (Achtibat et al., 2024), to spot key information used in the agent prediction.

Communication decoding For learned communication analyses, it becomes harder and might be reduced to finding patterns or comparing and aligning latent spaces to spot similar messages between agents. In order to uncover how agents derive meaning from these interactions, causal interventions might yields interesting hindsights (Kram'ar et al., 2024).

Swarm coordination: an inherent challenge of MAS that becomes increasingly complex as the number of agents scales. Fortunately, modern direct interpretability offers means to control models using methods from representation engineering (Zou et al., 2023), like activation steering (Rimsky et al., 2023). The latter method has proven useful to control an agent's policy by favouring different goals (Mini et al., 2023). Further application to MADRL could improve swarm coordination by enhancing traits like cooperativeness or better distributing goals among agents, e.g., by alternating resource collection among sites and agents.

3.3 Training Process Challenges

Training multiple agents simultaneously demands more computing power and can lead to learning instabilities. Therefore, it is crucial to better understand the training process of MADRL at different levels by improving learning efficiency and ensuring robustness.

State analysis. In order to model complex environments, one can train world models (Bruce et al., 2024), later used by an agent (Hafner et al., 2023). The condensed latent representation obtained can be analysed (Ivanitskiy et al., 2023) with tools like the tuned lens (Belrose et al., 2023a). This framework offers a better view of the transition function, which could help guide the agents towards unbiased training if analysed thoroughly.

Reward decomposition: often achieved by learning separate value functions aggregated afterwards (van Seijen et al., 2017; Juozapaitis et al., 2019). To avoid arbitrary decompositions, one could rely on local backpropagation methods like LRP or CRP (Bach et al., 2015; Achtibat et al., 2022), enabling the discovery of concepts that can later clarify the influence of the reward on the learning process of a policy. Further improvements could consider generating an adaptative curriculum (Jiang et al., 2020), prioritizing the concepts to learn.

Priority sampling, a staple method in RL that improves sample efficiency (Schaul et al., 2015). Also, in RL, interpretability was proven efficient to prioritize the important pixels for a visual policy by means of a consistency loss (Bertoin et al., 2022). Such a framework could be extended to compute importance over multiple inputs, creating a metric for better eliciting shared critical training samples.

Learning dynamics: trying to understand the agents throughout training, e.g., by observing the trained policies. Yet, it becomes more complicated as the number of agents scales and requires automated methods beyond observing policies. A widely used method to detect learned concepts in a model is to train linear probes (Alain and Bengio, 2018), which gave valuable insights for the analysis of AlphaZero networks (McGrath et al., 2022). By monitoring each agent, it would be possible to gain a more nuanced understanding of the swarm development and track the emergence or disappearance of certain capabilities.

Experience sharing: a method introduced to scale MADRL by improving sample efficiency (Christianos et al., 2020). Further improvements shared the data selectively according to exploration metrics (Gerstgrasser et al., 2023). Yet, this framework is missing a key point: you might want to select agents that share their experience similarly to parameter sharing (Christianos et al., 2021). A naive method could be to cluster experiences based on some latent representation of the different agents, enabling efficient knowledge sharing (Zou et al., 2023).

4 Discussion

4.1 Post-Hoc Interpretability in Deep RL

Post hoc interpretability in Deep Reinforcement Learning (DRL) is an increasingly important field, with methods such as saliency maps already being used to visualize agent behaviour (Greydanus et al., 2017), debug learned concepts (Jaderberg et al., 2018; Hilton et al., 2020), and inform sampling strategies to improve efficiency (Bertoin et al., 2022). Other approaches analyse agent behaviour by querying interaction data (Sequeira et al., 2019) or by visualising pattern prototypes (Ragodos et al., 2022; Alicioğlu and Sun, 2024). More extensive efforts have focused on interpreting well-known chess engines like AlphaZero (McGrath et al., 2022; Lovering et al., 2022; Hammersborg and Strümke, 2023; Schut et al., 2023; Jenner et al., 2024a; Poupart, 2024) and Stockfish (Pálsson and Björnsson), providing valuable insights into learned strategies. Ongoing efforts are also focused on exposing the key mechanisms behind planning, especially with games as a testbed (Jenner et al., 2024b; Taufeeque et al., 2024; Guei et al., 2024; Chung et al., 2024).

Other post hoc methods, like policy distillation into interpretable models, often referred to as model extraction, have also been a central focus. Techniques such as DAGGER (Ross et al., 2010) and VIPER (Bastani et al., 2018) leverage imitation learning to simplify policies. However, these methods struggle to scale effectively when applied globally to complex models, limiting their applicability to large-scale systems.

4.2 Interpretability in MADRL

Interpretability in MADRL is an evolving field with several promising approaches. Shapley values have been widely applied to analyse individual agent contributions, providing a robust theoretical framework for evaluating each agent's influence on team performance (Heuillet et al., 2021; Wang et al., 2021; Mahjoub et al., 2023). Diversity measures of agent policies have also emerged as a valuable tool for understanding agent behaviour, revealing distinctions between individual strategies and their roles in collective dynamics (Khlifi et al., 2023).

Similarly to XRL, policy extraction techniques, such as VIPER (Bastani et al., 2018), have been extended to leverage MADRL training to distil interpretable policies from complex models (Milani et al., 2022). Furthermore, predicting high-level concepts instead of actions offers a novel pathway to intrinsically interpretable models, aligning model outputs with human-understandable abstractions (Zabounidis et al., 2023). These advancements highlight the growing potential of interpretability methods in uncovering insights into multi-agent behaviour and learning processes.

4.3 Limits of Intrinsically Interpretable Models

Intrinsically interpretable models, whether obtained by design or post hoc extraction, have long been a dominant paradigm in agent interpretability research, relying on predefined, transparent model architectures. Design frameworks like XAg (Rodriguez and Thangarajah, 2024), concept bottlenecks (Poeta et al., 2023), learning skills with decision trees (Wen et al., 2024), or learning modularised agents (Cloud et al., 2024), aim to embed interpretability directly into model structures. However, such approaches face challenges in scalability and flexibility, particularly in multi-agent settings or with complex DRL models like the latest pre-trained world models (Reed et al., 2022; Yang et al., 2023; Alonso et al.; Bruce et al., 2024). The rigidity of design-based interpretability often compromises performance and fails to capture emergent behaviours, highlighting the need for alternative approaches that can adapt to the complexity and scale of modern systems.

5 Perspectives

5.1 MADRL Should Leverage Direct Interpretability

Engaging and expanding interpretability is an opportunity to address existing challenges in MADRL. Direct approaches are particularly well-suited for analysing communication dynamics, coordination strategies, and emergent behaviours in MAS. Graph-based analysis, for instance, could provide insights into inter-agent interactions, while feature importance techniques can identify biases and ensure fairness in decision-making. By systematically exploring and applying scalable direct methods to trained models, researchers can better address the inherent complexities of MADRL, enabling the development of more transparent, robust, and accountable systems for real-world applications.

Although previous calls to action are prone to integrate interpretability beforehand (Rodriguez and Thangarajah, 2024), this paper claims that the interpretation of models post hoc is highly valuable. Direct interpretability offers greater flexibility, particularly for existing models where architectural modifications are impractical.

5.2 Robust Evaluation Protocols

As repeatedly outlined, direct post-hoc methods are easily actionable and scalable. However, their adoption requires acknowledging and addressing limitations such as the inherent shortcomings of saliency maps (Adebayo et al., 2018; Bilodeau et al., 2022), counterfactual explanations (Laugel et al., 2019), or other interpretability illusions (Bolukbasi et al., 2021; Friedman et al., 2023,?). In fact, these methods often generate metrics with limited predictive power, and thus, claims should be reasonable.

A key priority is the development of robust evaluation protocols for direct methods. Given the absence of ground-truth explanations, reliable metrics and standardized evaluation frameworks must be established to assess the quality and utility of these methods (Gill et al., 2020; Madsen et al., 2021; Amorim et al., 2023; Hedström et al., 2022; Wei et al., 2024; Huang et al., 2024; Chaudhary and Geiger, 2024). Advancing evaluation thoroughly, e.g., by evaluating out of distribution, is especially important to develop scalable, effective, and actionable interpretability solutions.

6 Conclusion

We outlined that direct interpretability might be vital for addressing the challenges of scalability and complexity in modern MADRL. It enables the analysis of trained models without imposing architectural constraints, providing critical insights into agent behaviour, emergent dynamics, and biases. Advancing these methods will ensure scalable oversight of these systems, which is a precious desideratum for real-world applications. However, challenges such as explanation illusions, lack of robust evaluation metrics, and difficulty disentangling causal effects should be considered and tackled.

References

Achtibat, R., M. Dreyer, I. Eisenbraun, S. Bosse, T. Wiegand, W. Samek, and S. Lapuschkin (2022). From attribution maps to human-understandable explanations through concept relevance propagation. *Nature Machine Intelligence 5*, 1006 – 1019.

Achtibat, R., S. M. V. Hatefi, M. Dreyer, A. Jain, T. Wiegand, S. Lapuschkin, and W. Samek (2024). Attnlrp: Attention-aware layer-wise relevance propagation for transformers. *ArXiv abs/2402.05602*.

Adebayo, J., J. Gilmer, M. Muelly, I. J. Goodfellow, M. Hardt, and B. Kim (2018). Sanity checks for saliency maps. In *Neural Information Processing Systems*.

Alain, G. and Y. Bengio (2018). Understanding intermediate layers using linear classifier probes.

Alicioğlu, G. and B. Sun (2024). Use bag-of-patterns approach to explore learned behaviors of reinforcement learning. In *xAI*.

Alonso, E., A. Jelley, V. Micheli, A. Kanervisto, A. Storkey, T. Pearce, and F. Fleuret. Diffusion for world modeling: Visual details matter in atari. In *Thirty-eighth Conference on Neural Information Processing Systems*.

Amorim, J. P., P. H. Abreu, J. A. M. Santos, and H. Müller (2023). Evaluating post-hoc interpretability with intrinsic interpretability. *ArXiv abs/2305.03002*.

Bach, S., A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE 10*.

Bastani, O., Y. Pu, and A. Solar-Lezama (2018). Verifiable reinforcement learning via policy extraction. In *Neural Information Processing Systems*.

Bekkemoen, Y. (2023). Explainable reinforcement learning (xrl): a systematic literature review and taxonomy. *Machine Learning 113*, 355–441.

Belrose, N., Z. Furman, L. Smith, D. Halawi, I. V. Ostrovsky, L. McKinney, S. Biderman, and J. Steinhardt (2023a). Eliciting latent predictions from transformers with the tuned lens. *ArXiv abs/2303.08112*.

Belrose, N., D. Schneider-Joseph, S. Ravfogel, R. Cotterell, E. Raff, and S. Biderman (2023b). Leace: Perfect linear concept erasure in closed form.

Bertoin, D., A. Zouitine, M. Zouitine, and E. Rachelson (2022). Look where you look! saliency-guided q-networks for generalization in visual reinforcement learning. In *Neural Information Processing Systems*.

Bilodeau, B., N. Jaques, P. W. Koh, and B. Kim (2022). Impossibility theorems for feature attribution. *Proceedings of the National Academy of Sciences of the United States of America 121*.

Bolukbasi, T., A. Pearce, A. Yuan, A. Coenen, E. Reif, F. Vi'egas, and M. Wattenberg (2021). An interpretability illusion for bert. *ArXiv abs/2104.07143*.

Bruce, J., M. D. Dennis, A. Edwards, J. Parker-Holder, Y. Shi, E. Hughes, M. Lai, A. Mavalankar, R. Steigerwald, C. Apps, Y. Aytar, S. Bechtle, F. M. P. Behbahani, S. Chan, N. M. O. Heess, L. Gonzalez, S. Osindero, S. Ozair, S. Reed, J. Zhang, K. Zolna, J. Clune, N. de Freitas, S. Singh, and T. Rocktaschel (2024). Genie: Generative interactive environments. *ArXiv abs/2402.15391*.

Chattopadhyay, A., S. Slocum, B. D. Haeffele, R. Vidal, and D. Geman (2022). Interpretable by design: Learning predictors by composing interpretable queries. *IEEE Transactions on Pattern Analysis and Machine Intelligence 45*, 7430–7443.

Chaudhary, M. and A. Geiger (2024). Evaluating open-source sparse autoencoders on disentangling factual knowledge in gpt-2 small. *ArXiv abs/2409.04478*.

Chelarescu, P. C. (2021). Deception in social learning: A multi-agent reinforcement learning perspective. *ArXiv abs/2106.05402*.

Chen, Z., Y. Bei, and C. Rudin (2020). Concept whitening for interpretable image recognition. *Nature Machine Intelligence 2*, 772 – 782.

Christianos, F., G. Papoudakis, A. Rahman, and S. V. Albrecht (2021). Scaling multi-agent reinforcement learning with selective parameter sharing. *ArXiv abs/2102.07475*.

Christianos, F., L. Schäfer, and S. V. Albrecht (2020). Shared experience actor-critic for multi-agent reinforcement learning. *ArXiv abs/2006.07169*.

Chu, X. and H. Ye (2017). Parameter sharing deep deterministic policy gradient for cooperative multi-agent reinforcement learning. *ArXiv abs/1710.00336*.

Chung, S., S. Niekum, and D. Krueger (2024). Predicting future actions of reinforcement learning agents. *ArXiv abs/2410.22459*.

Cloud, A., J. Goldman-Wetzler, E. Wybitul, J. Miller, and A. M. Turner (2024). Gradient routing: Masking gradients to localize computation in neural networks. *ArXiv abs/2410.04332*.

Conmy, A., A. N. Mavor-Parker, A. Lynch, S. Heimersheim, and A. Garriga-Alonso (2023). Towards automated circuit discovery for mechanistic interpretability.

Covert, I., S. M. Lundberg, and S.-I. Lee (2020). Explaining by removing: A unified framework for model explanation. *J. Mach. Learn. Res. 22*, 209:1–209:90.

Cunningham, H., A. Ewart, L. Riggs, R. Huben, and L. Sharkey (2023). Sparse autoencoders find highly interpretable features in language models. *ArXiv abs/2309.08600*.

Dreyer, M., R. Achtibat, W. Samek, and S. Lapuschkin (2023a). Understanding the (extra-)ordinary: Validating deep model decisions with prototypical concept-based explanations. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*

(CVPRW), 3491–3501.

Dreyer, M., F. Pahde, C. J. Anders, W. Samek, and S. Lapuschkin (2023b). From hope to safety: Unlearning biases of deep models via gradient penalization in latent space. In *AAAI Conference on Artificial Intelligence*.

Dunefsky, J., P. Chlenski, and N. Nanda (2024). Transcoders find interpretable llm feature circuits. *ArXiv abs/2406.11944*.

Elhage, N., N. Nanda, C. Olsson, T. Henighan, N. Joseph, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, et al. (2021). A mathematical framework for transformer circuits. *Transformer Circuits Thread 1*(1), 12.

Friedman, D., A. K. Lampinen, L. Dixon, D. Chen, and A. Ghandeharioun (2023). Interpretability illusions in the generalization of simplified models. *ArXiv abs/2312.03656*.

Gerstgrasser, M., T. Danino, and S. Keren (2023). Selectively sharing experiences improves multi-agent reinforcement learning. *ArXiv abs/2311.00865*.

Ghandeharioun, A., A. Caciularu, A. Pearce, L. Dixon, and M. Geva (2024). Patchscopes: A unifying framework for inspecting hidden representations of language models. *ArXiv abs/2401.06102*.

Gill, N., P. Hall, K. Montgomery, and N. Schmidt (2020). A responsible machine learning workflow with focus on interpretable models, post-hoc explanation, and discrimination testing. *Inf. 11*, 137.

Greydanus, S., A. Koul, J. Dodge, and A. Fern (2017). Visualizing and understanding atari agents. *ArXiv abs/1711.00138*.

Gronauer, S. and K. Diepold (2021). Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review 55*, 895 – 943.

Guei, H., Y.-R. Ju, W.-Y. Chen, and T.-R. Wu (2024). Interpreting the learned model in muzero planning.

Gupta, J. K., M. Egorov, and M. J. Kochenderfer (2017). Cooperative multi-agent control using deep reinforcement learning. In *AAMAS Workshops*.

Hafner, D., J. Pazukonis, J. Ba, and T. P. Lillicrap (2023). Mastering diverse domains through world models. *ArXiv abs/2301.04104*.

Hammersborg, P. and I. Strümke (2023). Information based explanation methods for deep learning agents–with applications on large open-source chess models. *arXiv preprint arXiv:2309.09702*.

Han, S., Q. Zhang, Y. Yao, W. Jin, Z. Xu, and C. He (2024). Llm multi-agent systems: Challenges and open problems. *ArXiv abs/2402.03578*.

He, K., X. Zhang, S. Ren, and J. Sun (2015). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Hedström, A., L. Weber, D. Bareeva, F. Motzkus, W. Samek, S. Lapuschkin, and M. M.-C. Höhne (2022). Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations. *ArXiv abs/2202.06861*.

Hernandez-Leal, P., B. Kartal, and M. E. Taylor (2018). A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems 33*, 750 – 797.

Heuillet, A., F. Couthouis, and N. D. Rodríguez (2020). Explainability in deep reinforcement learning. *Knowl. Based Syst. 214*, 106685.

Heuillet, A., F. Couthouis, and N. D. Rodríguez (2021). Collective explainable ai: Explaining cooperative strategies and agent contribution in multiagent reinforcement learning with shapley values. *IEEE Computational Intelligence Magazine 17*, 59–71.

Hickling, T., A. Zenati, N. Aouf, and P. Spencer (2022). Explainability in deep reinforcement learning: A review into current methods and applications. *ACM Computing Surveys 56*, 1 – 35.

Hilton, J., N. Cammarata, S. Carter, G. Goh, and C. Olah (2020). Understanding rl vision.

Huang, J., Z. Wu, C. Potts, M. Geva, and A. Geiger (2024). Ravel: Evaluating interpretability methods on disentangling language model representations. *ArXiv abs/2402.17700*.

Ivanitskiy, M. I., A. F. Spies, T. Rauker, G. Corlouer, C. Mathwin, L. Quirke, C. Rager, R. Shah, D. Valentine, C. G. D. Behn, K. Inoue, and S. W. Fung (2023). Structured world representations in maze-solving transformers. *ArXiv abs/2312.02566*.

Jaderberg, M., W. M. Czarnecki, I. Dunning, L. Marris, G. Lever, A. G. Castañeda, C. Beattie, N. C. Rabinowitz, A. S. Morcos, A. Ruderman, N. Sonnerat, T. Green, L. Deason, J. Z. Leibo, D. Silver, D. Hassabis, K. Kavukcuoglu, and T. Graepel (2018). Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science 364*, 859 – 865.

Jenner, E., S. Kapur, V. Georgiev, C. Allen, S. Emmons, and S. Russell (2024a). Evidence of learned look-ahead in a chess-playing neural network.

Jenner, E., S. Kapur, V. Georgiev, C. Allen, S. Emmons, and S. Russell (2024b). Evidence of learned look-ahead in a chess-playing neural network. *ArXiv abs/2406.00877*.

Jiang, M., E. Grefenstette, and T. Rocktäschel (2020). Prioritized level replay. In *International Conference on Machine Learning*.

Juozapaitis, Z., A. Koul, A. Fern, M. Erwig, and F. Doshi-Velez (2019). Explainable reinforcement learning via reward decomposition.

Katz, S., Y. Belinkov, M. Geva, and L. Wolf (2024). Backward lens: Projecting language model gradients into the vocabulary space. *ArXiv abs/2402.12865*.

Khlifi, W., S. Singh, O. Mahjoub, R. de Kock, A. Vall, R. Gorsane, and A. Pretorius (2023). On diagnostics for understanding agent training behaviour in cooperative marl. *ArXiv abs/2312.08468*.

Kim, B., M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres (2018). Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav).

Kohler, H., Q. Delfosse, P. Festor, and P. Preux (2024). Towards a research community in interpretable reinforcement learning: the interppol workshop. *ArXiv abs/2404.10906*.

Kram'ar, J., T. Lieberum, R. Shah, and N. Nanda (2024). Atp*: An efficient and scalable method for localizing llm behaviour to components. *ArXiv abs/2403.00745*.

Lambrechts, G., A. Bolland, and D. Ernst (2023). Informed pomdp: Leveraging additional information in model-based rl. In *RLC*.

Lange, M., R. C. Engelhardt, W. Konen, and L. Wiskott (2024). Interpretable brain-inspired representations improve rl performance on visual navigation tasks. *ArXiv abs/2402.12067*.

Lapuschkin, S., S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller (2019). Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications 10*.

Laugel, T., M.-J. Lesot, C. Marsala, X. Renard, and M. Detyniecki (2019). The dangers of post-hoc interpretability: Unjustified counterfactual explanations. In *International Joint Conference on Artificial Intelligence*.

Levin, M. and H. Chockler (2023). Clustered policy decision ranking. *ArXiv abs/2311.12970*.

Li, X., S. Wang, S. Zeng, Y. Wu, and Y. Yang (2024). A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*.

Lovering, C., J. Forde, G. Konidaris, E. Pavlick, and M. Littman (2022). Evaluation beyond task performance: Analyzing concepts in alphazero in hex. *Advances in Neural Information Processing Systems 35*, 25992–26006.

Lowe, R., Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. *ArXiv abs/1706.02275*.

Lundberg, S. M. and S.-I. Lee (2017). A unified approach to interpreting model predictions. In *Neural Information Processing Systems*.

Madsen, A., S. Reddy, and A. P. S. Chandar (2021). Post-hoc interpretability for neural nlp: A survey. *ACM Computing Surveys 55*, 1 – 42.

Mahendran, A. and A. Vedaldi (2015). Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision 120*, 233–255.

Mahjoub, O., R. de Kock, S. Singh, W. Khlifi, A. Vall, K. ab Tessera, and A. Pretorius (2023). Efficiently quantifying individual agent importance in cooperative marl. *ArXiv abs/2312.08466*.

Mathieu, M., S. Ozair, S. Srinivasan, C. Gulcehre, S. Zhang, R. Jiang, T. L. Paine, R. Powell, K. Zolna, J. Schrittwieser, D. Choi, P. Georgiev, D. Toyama, A. Huang, R. Ring, I. Babuschkin, T. Ewalds, M. Bordbar, S. Henderson, S. G. Colmenarejo, A. van den Oord, W. M. Czarnecki, N. de Freitas, and O. Vinyals (2023). Alphastar unplugged: Large-scale offline reinforcement learning. *ArXiv abs/2308.03526*.

McGrath, T., A. Kapishnikov, N. Tomaš ev, A. Pearce, M. Wattenberg, D. Hassabis, B. Kim, U. Paquet, and V. Kramnik (2022). Acquisition of chess knowledge in AlphaZero. *Proceedings of the National Academy of Sciences 119*(47).

Meng, K., D. Bau, A. Andonian, and Y. Belinkov (2022). Locating and editing factual associations in gpt. In *Neural Information Processing Systems*.

Milani, S., N. Topin, M. Veloso, and F. Fang (2023). Explainable reinforcement learning: A survey and comparative review. *ACM Computing Surveys 56*, 1 – 36.

Milani, S., Z. Zhang, N. Topin, Z. R. Shi, C. A. Kamhoua, E. E. Papalexakis, and F. Fang (2022). Maviper: Learning decision tree policies for interpretable multi-agent reinforcement learning. In *ECML/PKDD*.

Mini, U., P. Grietzer, M. Sharma, A. Meek, M. S. MacDiarmid, and A. M. Turner (2023). Understanding and controlling a maze-solving policy network. *ArXiv abs/2310.08043*.

Mitelut, C., B. Smith, and P. Vamplew (2023). Intent-aligned ai systems deplete human agency: the need for agency foundations research in ai safety.

Montavon, G., S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller (2015). Explaining nonlinear classification decisions with deep taylor decomposition. *ArXiv abs/1512.02479*.

Nanda, N., L. Chan, T. Lieberum, J. Smith, and J. Steinhardt (2023). Progress measures for grokking via mechanistic interpretability. *ArXiv abs/2301.05217*.

Olah, C., N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter (2020). Zoom in: An introduction to circuits.

Olsson, C., N. Elhage, N. Nanda, N. Joseph, N. Dassarma, T. Henighan, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, S. Johnston, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. B. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah (2022). In-context learning and induction heads. *ArXiv abs/2209.11895*.

Orr, J. and A. Dutta (2023). Multi-agent deep reinforcement learning for multi-robot applications: A survey. *Sensors (Basel, Switzerland) 23*.

Paatero, P. and U. Tapper (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values†. *Environmetrics 5*, 111–126.

Pálsson, A. and Y. Björnsson. Unveiling concepts learned by a world-class chess-playing agent.

Pochinkov, N. and N. Schoots (2024). Dissecting language models: Machine unlearning via selective pruning. *ArXiv abs/2403.01267*.

Poeta, E., G. Ciravegna, E. Pastor, T. Cerquitelli, and E. Baralis (2023). Concept-based explainable artificial intelligence: A survey. *ArXiv abs/2312.12936*.

Poupart, Y. (2024). Contrastive sparse autoencoders for interpreting planning of chess-playing agents. *ArXiv abs/2406.04028*.

Qing, Y., S. Liu, J. Song, and M. Song (2022). A survey on explainable reinforcement learning: Concepts, algorithms, challenges. *ArXiv abs/2211.06665*.

Quirke, P. and F. Barez (2023). Understanding addition in transformers. *ArXiv abs/2310.13121*.

Radford, A., L. Metz, and S. Chintala (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR abs/1511.06434*.

Radford, A. and K. Narasimhan (2018). Improving language understanding by generative pre-training.

Ragodos, R. J., T. Wang, Q. Lin, and X. Zhou (2022). Protox: Explaining a reinforcement learning agent via prototyping. *ArXiv abs/2211.03162*.

Rashid, T., M. Samvelyan, C. S. D. Witt, G. Farquhar, J. N. Foerster, and S. Whiteson (2018). Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. *ArXiv abs/1803.11485*.

Reed, S., K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron,

M. Giménez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. D. Edwards, N. M. O. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and N. de Freitas (2022). A generalist agent. *ArXiv abs/2205.06175*.

Ribeiro, M. T., S. Singh, and C. Guestrin (2018). Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence*.

Rimsky, N., N. Gabrieli, J. Schulz, M. Tong, E. Hubinger, and A. M. Turner (2023). Steering llama 2 via contrastive activation addition.

Rodriguez, S. and J. Thangarajah (2024). Explainable agents (xag) by design. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '24, Richland, SC, pp. 2712–2716. International Foundation for Autonomous Agents and Multiagent Systems.

Rodriguez, S., J. Thangarajah, and A. Davey (2024). Design patterns for explainable agents (xag). In *Adaptive Agents and Multi-Agent Systems*.

Ross, S., G. J. Gordon, and J. A. Bagnell (2010). A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*.

Rudin, C., C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong (2021). Interpretable machine learning: Fundamental principles and 10 grand challenges. *ArXiv abs/2103.11251*.

Rusu, A. A., S. G. Colmenarejo, Çaglar Gülçehre, G. Desjardins, J. Kirkpatrick, R. Pascanu, V. Mnih, K. Kavukcuoglu, and R. Hadsell (2015). Policy distillation. *CoRR abs/1511.06295*.

Schaul, T., J. Quan, I. Antonoglou, and D. Silver (2015). Prioritized experience replay. *CoRR abs/1511.05952*.

Schut, L., N. Tomasev, T. McGrath, D. Hassabis, U. Paquet, and B. Kim (2023). Bridging the human-ai knowledge gap: Concept discovery and transfer in alphazero.

Selvaraju, R. R., A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra (2016). Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision 128*, 336 – 359.

Seong, H. and D. H. Shim (2024). Self-supervised interpretable end-to-end learning via latent functional modularity. In *International Conference on Machine Learning*.

Sequeira, P., E. Yeh, and M. T. Gervasio (2019). Interestingness elements for explainable reinforcement learning through introspection. In *IUI Workshops*.

Shaik, T. B., X. Tao, H. Xie, L. Li, J. Yong, and H. Dai (2023). Adaptive multi-agent deep reinforcement learning for timely healthcare interventions.

Shapley, L. S. (1988). A value for n-person games.

Shavit, Y., S. Agarwal, M. Brundage, C. authors, S. Adler, C. O'Keefe, R. Campbell, T. Lee, P. Mishkin, T. Eloundou, A. Hickey, K. Slama, L. Ahmad, P. McMillan, A. Beutel, A. Passos, and D. G. Robinson (2023). Practices for governing agentic ai systems.

Shrikumar, A., P. Greenside, A. Shcherbina, and A. Kundaje (2016). Not just a black box: Learning important features through propagating activation differences. *ArXiv abs/1605.01713*.

Simonyan, K. and A. Zisserman (2014). Very deep convolutional networks for large-scale image recognition. *CoRR abs/1409.1556*.

Smilkov, D., N. Thorat, B. Kim, F. B. Viégas, and M. Wattenberg (2017). Smoothgrad: removing noise by adding noise. *ArXiv abs/1706.03825*.

Sunehag, P., G. Lever, A. Gruslys, W. M. Czarnecki, V. F. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, and T. Graepel (2017). Value-decomposition networks for cooperative multi-agent learning. *ArXiv abs/1706.05296*.

Tampuu, A., T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus, J. Aru, J. Aru, and R. Vicente (2015). Multiagent cooperation and competition with deep reinforcement learning. *PLoS ONE 12*.

Taufeeque, M., P. Quirke, M. Li, C. Cundy, A. D. Tucker, A. Gleave, and A. Garriga-Alonso (2024). Planning in a recurrent neural network that plays sokoban.

van Seijen, H., M. Fatemi, R. Laroche, J. Romoff, T. Barnes, and J. Tsang (2017). Hybrid reward architecture for reinforcement learning. *ArXiv abs/1706.04208*.

Vaswani, A., N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). Attention is all you need. In *Neural Information Processing Systems*.

Vinyals, O., I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. P. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature 575*, 350 – 354.

Vishwanath, A., L. A. Dennis, and M. Slavkovik (2024). Reinforcement learning and machine ethics:a systematic review. *ArXiv abs/2407.02425*.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing 17*, 395–416.

Wang, J., Y. Zhang, Y. Gu, and T.-K. Kim (2021). Shaq: Incorporating shapley value theory into multi-agent q-learning. In *Neural Information Processing Systems*.

Wang, L., C. Ma, X. Feng, Z. Zhang, H. ran Yang, J. Zhang, Z.-Y. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, and J. rong Wen (2023). A survey on large language model based autonomous agents. *ArXiv abs/2308.11432*.

Wei, J., H. Turb'e, and G. Mengaldo (2024). Revisiting the robustness of post-hoc interpretability methods. *ArXiv abs/2407.19683*.

Wen, Y., S. Li, R. Zuo, L. Yuan, H. Mao, and P. Liu (2024). Skilltree: Explainable skill-based deep reinforcement learning for long-horizon control tasks.

Wiegreffe, S. and Y. Pinter (2019). Attention is not not explanation. In *Conference on Empirical Methods in Natural Language Processing*.

Wong, A., T. Bäck, A. V. Kononova, and A. Plaat (2021). Multiagent deep reinforcement learning: Challenges and directions towards human-like approaches. *ArXiv abs/2106.15691*.

Wu, Q., G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu, A. H.

Awadallah, R. W. White, D. Burger, and C. Wang (2023). Autogen: Enabling next-gen llm applications via multi-agent conversation.

Yang, S., O. Nachum, Y. Du, J. Wei, P. Abbeel, and D. Schuurmans (2023). Foundation models for decision making: Problems, methods, and opportunities. *ArXiv abs/2303.04129*.

Yang, Y., R. Luo, M. Li, M. Zhou, W. Zhang, and J. Wang (2018). Mean field multi-agent reinforcement learning. *ArXiv abs/1802.05438*.

Yeom, S.-K., P. Seegerer, S. Lapuschkin, S. Wiedemann, K.-R. Müller, and W. Samek (2019). Pruning by explaining: A novel criterion for deep neural network pruning. *ArXiv abs/1912.08881*.

Yu, C., A. Velu, E. Vinitsky, Y. Wang, A. M. Bayen, and Y. Wu (2021). The surprising effectiveness of ppo in cooperative multi-agent games. In *Neural Information Processing Systems*.

Zabounidis, R., J. Campbell, S. Stepputtis, D. Hughes, and K. P. Sycara (2023). Concept learning for interpretable multi-agent reinforcement learning. *ArXiv abs/2302.12232*.

Zeiler, M. D. and R. Fergus (2013). Visualizing and understanding convolutional networks. *ArXiv abs/1311.2901*.

Zhu, C., M. M. Dastani, and S. Wang (2022). A survey of multi-agent deep reinforcement learning with communication. *Autonomous Agents and Multi-Agent Systems 38*, 1–48.

Zou, A., L. Phan, S. Chen, J. Campbell, P. Guo, R. Ren, A. Pan, X. Yin, M. Mazeika, A.-K. Dombrowski, S. Goel, N. Li, M. J. Byun, Z. Wang, A. Mallen, S. Basart, S. Koyejo, D. Song, M. Fredrikson, J. Z. Kolter, and D. Hendrycks (2023). Representation engineering: A top-down approach to ai transparency.

Résumé

L'Apprentissage par Renforcement Profond Multi-Agent (MADRL) s'est révélé efficace pour résoudre des problèmes complexes en robotique ou dans les jeux, mais la plupart des modèles entraînés restent difficiles à interpréter. Alors que l'apprentissage de modèles intrinsèquement interprétables demeure une approche prometteuse, sa scalabilité et sa flexibilité sont limitées pour gérer des tâches complexes ou des dynamiques multi-agents. Cet article plaide en faveur de l'interprétabilité directe comme une alternative polyvalente et évolutive, offrant des perspectives sur le comportement des agents, les phénomènes émergents et les biais, sans modifier les architectures des modèles. Nous explorons des méthodes modernes, notamment la rétropropagation de pertinence, l'édition des connaissances, le pilotage des modèles, le collage d'activations, les autoencodeurs sparses et la découverte de circuits, pour mettre en lumière leur applicabilité aux défis des agents uniques, des systèmes multi-agents et des processus d'entraînement. En abordant de manière fiable et pragmatique l'interprétabilité du MADRL, nous proposons des orientations visant à faire progresser des sujets actifs tels que l'identification d'équipes, la coordination d'essaims et l'efficacité de l'échantillonnage.

Vers une couverture partielle d'explications de réseaux neuronaux par des réseaux d'approximation

Mathieu Brassart*, Laurent Simon*

*LaBRI, Univ. Bordeaux, France

Résumé. Les avancées rapides en intelligence artificielle (IA) et en apprentissage automatique ont souligné la nécessité de décisions explicables et vérifiables. Cependant, les réseaux neuronaux profonds et les grandes forêts aléatoires résistent aux méthodes traditionnelles de vérification, agissant comme des "boîtes noires" opaques sur lesquelles la vérification formelle ne peut être appliquée. Cela soulève des problèmes significatifs de sécurité, de fiabilité et de prévisibilité pour ces systèmes d'IA. Cet article s'intérresse à l'application de la vérification formelle aux réseaux neuronaux sans compromettre leur précision globale, mais en ne couvrant uniquement qu'une partie d'entre elles. Nous proposons une méthode pour approximer des réseaux complexes par des modèles plus simples et vérifiables qui expliquent ou valident les décisions. Nos résultats montrent que, pour des problèmes bien connus, plus de 94% des décisions du système peuvent être vérifiées formellement, augmentant ainsi la confiance sans réduire les performances.

1 Introduction

La diffusion rapide, dans tous les pans de la société, de modèles d'Intelligence Artificielle basés sur l'Apprentissage Automatique ont mis en évidence le besoin crucial d'expliquer et de vérifier les décisions de ces systèmes. Beaucoup de travaux ont déjà été proposés dans cet objectif, par exemple en se basant sur les valeurs SHAP den Broeck et al. (2021); Nohara et al. (2022), les arbres de décision Audemard et al. (2022); Shih et al. (2018); Chan et Darwiche (2012); Tang et al. (2023); Zhang et al. (2023); Learning (2019), les circuits booléens Choi et al. (2017); Shi et al. (2020), ou d'autres approches Leofante et al. (2018); Goulet et al. (2021); Ignatiev et al. (2018) plus spécifiques.

Malheureusement, la quête de précision des modèles actuels, appuyée par de gigantesques quantités de calculs disponibles, a permis la production de modèles d'une complexité très largement hors de porté des techniques de vérification traditionnelles. Les modèles actuels sont ainsi souvent qualifiés de "boîtes noires" (tels que les réseaux neuronaux profonds et même les forêts aléatoires avec de nombreux arbres). Par construction, ces systèmes ne peuvent fournir ni explication concise ni preuve de leurs décisions, rendant la preuve de propriétés formelles impraticable. La vérification formelle reste cependant une méthode privilégiée pour prouver ou réfuter la correction d'un modèle d'IA par rapport à une spécification donnée, ce qui est crucial pour le déploiement de modèles complexes d'IA dans des applications critiques.

Dans cet article, nous proposons d'étudier jusqu'où la vérification formelle peut encore être appliquée aux réseaux neuronaux en ne sacrifiant pas sur la précision du modèle dans sa globalité mais en ne couvrant qu'une partie (identifiée) de ses décisions. Le modèle est capable d'identifier les décisions "prouvées" de celles qui comportent encore un certain risque. Notre objectif est de pouvoir prouver formellement de grands réseaux neuronaux en les limitant avec des réseaux neuronaux beaucoup plus simples qui sont suffisants pour expliquer ou prouver un sous-ensemble de décisions prises par l'ensemble du système. Notre approche s'appuie sur l'approche de Choi et al. (2017); Shi et al. (2020) qui considère les réseaux neuronaux comme des circuits booléens afin de pouvoir prouver des propriétés sur eux, grâce aux réseaux neuronaux binarisés Courbariaux et al. (2016). Nous montrons que, sur des problèmes bien connus, il est possible de couvrir plus de 94% des décisions prises par le système, sans perte de performances, avec des décisions qui peuvent être vérifiées formellement. De plus, notre approche permet d'identifier quand une décision n'a pas été vérifiée formellement, et peut donc être soumise à d'autres mesures de contrôle. Ce travail est encore préliminaire et fait état de la possibilité de construire un système accessible aux méthodes de preuves actuelles.

Dans la section suivante, nous examinons la capacité des méthodes actuelles à vérifier formellement des réseaux neuronaux. Sur la base de ces observations, nous introduisons, dans la section 3, la notion d'approximateurs encadrant les décisions globales du système. Dans la section 4, nous présentons les résultats de notre étude expérimentale, qui démontre que les performances du modèle global ne sont pas compromises par notre modèle, tout en permettant à une proportion significative de décisions d'être accessibles aux techniques de vérification formelle. Nous concluons ensuite.

2 Estimation de la taille des réseaux de neurones accessibles à la vérification formelle

Le domaine de la vérification formelle des réseaux neuronaux est un domaine de plus en plus important dans les domaines de l'intelligence artificielle et de l'apprentissage automatique et de nombreux travaux font déjà état d'avancées significatives. L'approche sur laquelle nous basons notre travail considère les réseaux neuronaux comme un type particulier de circuits booléens Choi et al. (2017); Shi et al. (2020) afin d'appliquer des méthodes de vérification formelle (FM) classiques. Cependant, avant de pouvoir appliquer l'une d'elles, le modèle d'IA doit être exprimé dans le bon formalisme et surtout d'une taille raisonnable. Pour rester générique, **nous faisons l'hypothèse qu'un réseau de neurone pouvant être exprimé sous forme de Diagrammes Binaires de Décisions (OBDD) est un réseau de neurones qui peut être vérifié par les méthodes actuelles.** Nous avons donc cherché à évaluer la taille des réseaux neuronaux qui peuvent être exprimés sous sous cette forme. Les OBDD présentent de très bonnes propriétés computationnelles dans la Carte de Compilation des Connaissances Darwiche et Marquis (2002). Une fois exprimé en BDD, le modèle peut offrir des algorithmes polynomiaux pour la vérification et l'explication, voire plus. Pour cela, nous utilisons une approximation binaire d'un réseau neuronal (Réseau Neuronal Binarisé, BNN Courbariaux et al. (2016)) pour le réécrire en OBDD. Comme nous allons le voir (cela a déjà été souligné dans Choi et al. (2017) par exemple), il est très difficile de réécrire un BNN en un OBDD, ce qui

Algorithm 1 – combine_obdd, combiner une liste de listes de obdd représentant un réseau neuronal en un seul obdd

```
prev_layer ← layers_obdd[0]
p_vars ← prev_layer[0].vars
for each layer of layers_obdd[1:] do
    res_layer ← []
    for each obdd of layer do
        res_bdd ← obdd.compose(obdd.vars, prev_layer)
        res_bdd.vars ← p_vars
        append(res_layer, res_bdd)
    prev_layer ← res_layer
return prev_layer
```

limite drastiquement l'approche globale. Nous avons donc cherché à évaluer la difficulté de cette tâche en fonction de la taille du réseau neuronal.

2.1 Évaluation de la difficulté de la compilation des réseaux neuronaux

Les réseaux neuronaux utilisés pour la compilation sont des réseaux avec des entrées et sorties binaires, et des poids et biais flottants. Nous utilisons une couche d'activation entre chaque couche entièrement connectée qui effectue la fonction de seuil 1. Cela garantit que chaque entrée et sortie de neurone est binaire, soit 1 soit 0.

$$f(x) = 1 \texttt{ si } x \geq 0 \texttt{ sinon } 0 \tag{1}$$

Pendant la phase d'entraînement, les couches d'activation fonctionnent comme une activation sigmoïde, facilitant ainsi un processus d'entraînement plus rapide. La méthode de compilation suit les étapes suivantes : premièrement, chaque couche est traitée séquentiellement, en considérant un neurone à la fois, en se basant sur l'algorithme de Chan et Darwiche (2012). Chaque neurone est alors converti en un OBDD qui est exprimé en termes des sorties de la couche précédente, ou des caractéristiques d'entrée pour la première couche. Ensuite, en remplaçant les sorties de la couche précédente par le BDD correspondant, un seul BDD est généré, représentant la sortie du réseau neuronal en termes des caractéristiques d'entrée (voir Algorithme **??**).

Par soucis de simplicité, il est possible d'illustrer cet algorithme à l'aide de l'exemple du réseau neuronal binarisé, Figure 1. Dans cet exemple, nous pouvons exprimer successivement, les sorties (h_1, h_2, h_3, h_4) de la couche 1 avec les entrées (i_1, i_2, i_3), et, la sortie (o_1) de la couche 2 avec les sorties (h_1, h_2, h_3, h_4) de la couche 1. À la fin de l'algorithme, un seul BDD sera construit représentant la sortie en fonction des entrées.

Malheureusement, cette méthode est rapidment limitée par la taille du réseau neuronal et le nombre de *features*. La complexité temporelle est $O(n2^{\frac{n}{2}})$ pour convertir une fonction de seuil linéaire et $O(n!n2^{\frac{n}{2}})$ pour le BDD minimum, selon Tang et al. (2023).

Les tableaux 1 et 2 illustrent la croissance exponentielle du coût computationnel nécessaire pour réécrire un réseau neuronal binaire en fonction de sa taille.

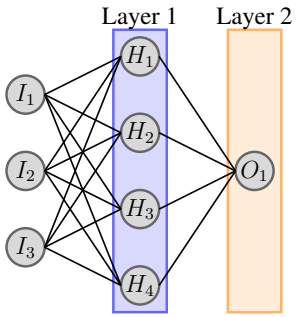


FIG. 1 – Exemple de réseau neuronal binaire

TAB. 1 – Temps pris en fonction de la taille de l'entrée, pour compiler le réseau neuronal en un ODD. Le réseau neuronal est composé de couches entièrement connectées de taille (Taille d'entrée, 10) et (10, 1)

Taille d'entrée	5	10	15	20	25
Temps moyen (s)	0.018	0.18	3.54	102.5	2693
Nombre de runs	1000	500	250	100	10

En conclusion de cette étude préliminaire, il peut être remarqué que, malgré un certain nombre d'approches précédentes, il est encore très difficile de compiler un réseau neuronal en un OBDD dont la taille puisse, en pratique, permettre une vérification formelle efficace. Les tailles des réseaux neuronaux accessibles qui sont considérées dans cette expérience préliminaire montrent qu'il est probablement vain d'imaginer vérifier des réseaux neuronaux de tailles courantes avec ces approches.

3 Couverture d'un modèle de réseau neuronal par des réseaux d'approximation

Dans cette section, nous proposons de surmonter partiellement la limitation forte mise en avant précédemment grâce à l'introduction d'une couche logique reliant plusieurs réseaux neuronaux permettant de forcer les décisions du modèle global par des réseaux plus petits,

TAB. 2 – Temps pris en fonction de la taille de la couche cachée pour compiler le réseau neuronal en un ODD. Le réseau neuronal est composé de couches entièrement connectées de taille (15, Taille de la couche cachée) et (Taille de la couche cachée, 1)

Taille de la couche cachée	5	10	15	20	25
Temps moyen (s)	0.80	3.39	18.40	115.3	956.8
Nombre de runs	500	250	100	50	10

chaque NN indépendant pouvant alors agir comme une raison ou une explication indépendante pour la décision globale.

Afin d'ajuster et de tester différents scénarios pour les mécanismes de rétropropagation, la fonction de perte et l'entraînement des différents NN dans le modèle global, nous nous concentrons dans cette section sur l'entraînement sur un exemple tabulaire simple et bien connu : le jeu de données "Diabetes".

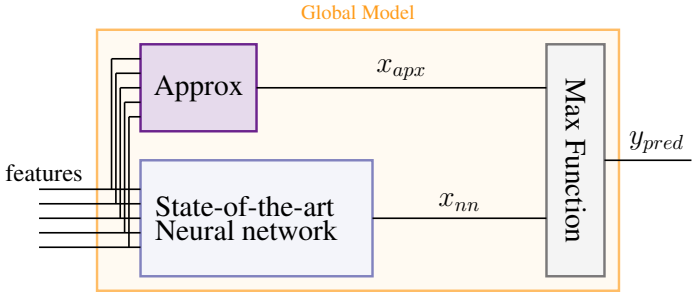


FIG. 2 – Modèle avec un réseau neuronal et un réseau d'approximation

Notre idée repose sur le remplacement d'un seul grand réseau neuronal (le réseau original avec lequel nous allons nous comparer), par un réseau *Global* qui contient plusieurs réseeaux, comme illustré Figure 2. Ce réseau global produit le maximum entre un petit réseau neuronal (appelé réseau d'approximation) et le réseau neuronal principal, qui a typiquement la même taille que le réseau neuronal unique original avec lequel se comparer. La prédiction unique y_{pred} est alors « câblée » pour être le maximum entre x_{apx} et x_{nn}. Notons que nous considérons ici la fonction max comme un moyen d'accumuler les décisions car, dans ce premier travail, nous cherchons à expliquer les décisions positives (typiquement lorsque le modèle répond "oui" (ou "1")). L'extension aux décisions 0 / 1 est un travail en cours.

L'architecture de notre système permet de considérer chaque approximateur comme une formule logique pouvant produire une explication pour la décision finale du Modèle Global, dès lors que cet approximateur décide "1". Cette façon d'accumuler les sous réseaux permet de décomposer les raisons logiques d'une décision positive de manière efficace (il est possible ainsi de vérifier formellement et indépendamment les décisions de tous les approximateurs). Il est important à ce stade de noter qu'il n'est pas attendu que tous les approximateurs couvrent la complexité et la précision du modèle initial de référence. Nous anticipons que les approximateurs traiteront les décisions simples, tandis que le grand réseau neuronal (au sein du Modèle Global) traitera les cas plus complexes. Clairement, toutes les instances de décisions positives ne seront pas générées par les approximateurs, mais la proportion de décisions positives attribuables aux approximateurs reste à mesurer. Il est également intéressant d'étudier comment le Modèle Global peut être entraîné de manière à maximiser la couverture des décisions positives par les approximateurs.

Notre intuition est que nous observerons un effet similaire à celui du principe de Pareto, selon lequel 80% de la complexité des décisions est due à seulement 20% des cas. Nous anticipons qu'un grand nombre de décisions relativement simples seront prises, couvertes par un approximateur pour lequel la vérification formelle sera applicable.

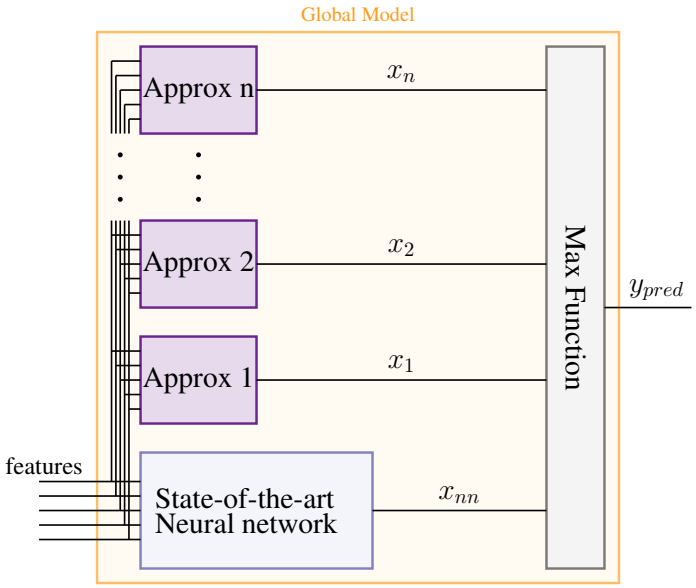


FIG. 3 – Modèle avec un réseau neuronal et plusieurs réseaux d'approximation

3.1 Couche logique pour l'agrégation des décisions

La couche logique effectue l'opération max entre les prédictions des différents sous-modèles (dernière couche sur les figures). Cette fonction d'agrégation permet d'incorporer plusieurs réseaux neuronaux binaires afin d'améliorer les performances globales. Étant donné que la prédiction globale est positive dès qu'une des prédictions des réseaux neuronaux est positive, la précision sur la classe positive et le rappel sur la classe négative pour chaque réseau séparé sont les métriques les plus cruciales pour évaluer les performances du système global. L'entraînement doit permettre de couvrir autant de cas que possible avec chaque réseau neuronal couvrant seulement une partie de la classe positive, tout en minimisant les erreurs sur les "1". Cette approche est similaire aux méthodes d'apprentissage ensembliste et de bagging Breiman (1996), mais avec un mécanisme de vote différent.

Comme illustré Figure 3, plusieurs stratégies peuvent être employées pour l'apprentissage des différents composants et distribuer le gradient en arrière à travers la couche logique introduite ci-dessus. Nous avons expérimenté deux méthodes distinctes : l'une qui transmet le gradient uniquement à la valeur maximale (colonne MAX), et l'autre qui distribue le gradient à toutes les entrées (colonne ALL).

Il est également possible d'entraîner séquentiellement ou simultanément tous les composants du modèle global, ce qui peut avoir un impact sur les performances du modèle :

— En *séquence*, nous apprenons d'abord un approximateur, puis gelons ses poids, apprenons un autre approximateur et ainsi de suite, jusqu'à ce que tous les approximateurs aient été entraînés. Nous entraînons ensuite le grand réseau neuronal après tous les approximateurs.

— En apprentissage *simultané*, tous les composants sont entraînés en même temps.

— Il est également possible d'entraîner tous les composants *séparément* sans interférence entre eux, puis de les attacher ensemble en utilisant la couche logique. Cette méthode n'est utilisée que dans la section 4 car l'expérience actuelle n'est pas pertinente pour ce type d'entraînement.

Intuitivement, on peut s'attendre à ce que l'approche d'apprentissage séquentiel produise des résultats optimaux. En effet, on peut imaginer que l'approximateur initial se concentrera sur les cas les plus simples, permettant ainsi au deuxième approximateur de traiter d'autres cas simples dans une région différente de l'espace des caractéristiques. Enfin, le réseau neuronal principal sera chargé de résoudre les cas plus complexes, nécessitant l'intervention de plusieurs réseaux neuronaux.

Le Tableau 3 rapporte plusieurs mesures pour chacune des combinaisons ci-dessus.

— "Couverture du modèle" est la précision du modèle global sur les "1",
— "Couverture de l'approximateur" est la précision en considérant uniquement l'approximateur,
— "Couverture relative" est le pourcentage de "1" donné par le modèle global qui sont donnés par l'approximateur ($\frac{\text{Approx coverage}}{\text{Model coverage}}$),
— "Couverture du réseau neuronal" est la précision du grand réseau neuronal (une faible valeur n'est pas nécessairement une mauvaise valeur, car le réseau neuronal vise à considérer uniquement les cas difficiles),
— "Approximateur classifiant incorrectement des zéros" est le pourcentage de faux positifs pour la classe 1,
— "Précision globale du modèle" est la précision sur le modèle global en considérant les deux classes.

Nous pouvons observer d'abord que la méthode ALL est la meilleure. Mais il est assez surprenant de mesurer que la méthode Simultanée offre de si bons résultats. La différence n'est visible que sur les zéros incorrectement classifiés par l'approximateur. Cette mesure est néenmoins importante car, si un approximateur classifie incorrectement un 0 en 1, alors le grand réseau neuronal ne pourra pas corriger cette décision, quelle que soit sa taille.

TAB. 3 – Comparaison des métriques sur la classe 1, pour les deux méthodes de passage d'un gradient à travers la couche max. Pour cette expérience, un modèle composé de 3 approx. et 1 réseau neuronal a été utilisé (jeu de données Diabetes, 100 exécutions, 500 époques, p=2)

Méthode d'entraînement	Séquentiel		Simultané	
Méthode de gradient	ALL	MAX	ALL	MAX
Couverture du modèle	0.81	0.66	0.81	0.70
Couverture de l'approximateur	0.72	0.66	0.73	0.24
Couverture relative	0.89	1.00	0.89	0.34
Couverture du réseau neuronal	0.63	0.00	0.34	0.50
Approximateur classifiant incorrectement des zéros	0.27	0.26	0.34	0.56
Précision globale du modèle	0.73	0.70	0.73	0.71

TAB. 4 – Comparaison des résultats obtenus avec différentes pénalités appliquées à la classe 1 (jeu de données Diabetes, 100 exécutions, 500 époques, entraînement séquentiel avec 1 approx.)

Pénalité	F1-score	Précision		Rappel	
		Classe 0	Classe 1	Classe 0	Classe 1
p = 1 (BCE)	0.69	0.73	0.69	0.67	0.73
p = 2	0.70	0.69	0.73	0.74	0.68
p = 5	0.73	0.70	0.79	0.82	0.65
p = 10	0.62	0.65	0.57	0.88	0.46
p = 15	0.48	0.58	0.30	0.94	0.24
p = 20	0.37	0.52	0.09	0.98	0.07
p = 25	0.34	0.51	0.01	1.00	0.01

3.2 Fonction de perte

Afin d'optimiser le processus d'entraînement, nous avons utilisé une gamme de fonctions de perte *ad hoc*, dans une étude préliminaire, dans le but d'améliorer les performances du modèle global. Au cours de cette phase, nous avons utilisé des fonctions spécialisées qui priorisent une étiquette spécifique, dans le but de minimiser le nombre d'étiquettes positives mal classées.

La fonction qui offre les meilleurs résultts est une entropie croisée binaire pondérée 2, où p est une pénalité sur la classe 1. Lorsque l'étiquette attendue est 0 et que l'étiquette prédite est 1, la perte est p fois plus grande que le cas inverse.

$$L = -(p \cdot (1-y) \cdot \log(1-x) + y \cdot \log x) \tag{2}$$

L'utilisation de cette fonction de perte a donné de meilleurs résultats par rapport à l'entropie croisée binaire, en particulier en termes de rappel pour la classe 0 et de précision pour la classe 1. Cependant, cette approche a conduit à une diminution de la précision pour la classe 0 et à une diminution du rappel pour la classe 1.

Une augmentation de la pénalité peut entraîner une confiance accrue dans la classe 1 mais au détriment de moins de cas couverts (une valeur extrême peut même faire en sorte que le réseau neuronal ne prédise qu'une seule classe, dans ce cas $p = 25$ finit généralement par entraîner un modèle ne prédisant que la classe 0). Une valeur adéquate est nécessaire pour trouver l'équilibre entre précision et couverture. Une étude détaillée de l'effet des valeurs de p est rapportée dans les Tableaux 4 et 5. Une valeur de $p = 2$ semble bien équilibrée et un bon point de départ, et peut être ajustée selon les besoins.

4 Expérimentation sur des jeux de données supplémentaires

Dans cette section, nous avons utilisé plusieurs jeux de données qui présentent des données tabulaires et fournissent un problème de classification binaire. Ces jeux de données contiennent des features qui peuvent être soit booléennes, flottantes ou catégorielles (ces features sont binarisées dans le cadre de notre approche). L'objectif est d'entraîner un modèle global qui offre

Pénalité	F1-score	Précision		Rappel	
		Classe 0	Classe 1	Classe 0	Classe 1
p = 1 (BCE)	0.69	0.75	0.68	0.59	0.80
p = 2	0.72	0.77	0.71	0.65	0.81
p = 5	0.72	0.69	0.79	0.82	0.64
p = 10	0.66	0.67	0.65	0.86	0.53
p = 15	0.49	0.58	0.31	0.94	0.25
p = 20	0.37	0.52	0.06	0.99	0.05
p = 25	0.34	0.50	0.01	1.00	0.02

TAB. 5 – Comparaison des résultats obtenus avec différentes pénalités appliquées à la classe 1 (jeu de données Diabetes, 100 exécutions, 500 époques, entraînement séquentiel avec 3 approx.)

Dataset	nombre de features	features binarisées	nombre de classes	f1-score nn
Compas	13	18	2	0.66
Diabetes	8	29	2	0.76
Titanic	7	16	2	0.85
Adult	14	113	2	0.81

TAB. 6 – Présentation des jeux de données, introduisant le nombre de features avant et après encodage (binarisation) et le F1-score en utilisant un réseau de neurones adapté. Les caractéristiques (features) prétraitées sont la binarisation des caractéristiques originales.

TAB. 7 – Rapport des meilleures performances connues sur l'ensemble des problèmes que nous comparons à notre approche. (a) et (b) sont des méthodes tirées des meilleures solutions sur Kaggle. (c) est notre méthode (performance du modèle global).

Dataset	F1-Sc.	Précis.		Rapp.	
		C0	C1	C0	C1
Compas	0.65	0.64	0.68	0.71	0.61
Diabet.	0.73	0.73	0.74	0.76	0.71
Titanic	0.80	0.77	0.84	0.85	0.75
Adult	0.80	0.82	0.79	0.78	0.83

(a) RNN (100 exécutions, 500 époques)

Dataset	F1-Sc.	Précis.		Rapp.	
		C0	C1	C0	C1
Compas	0.66	0.65	0.68	0.71	0.62
Diabet.	0.75	0.75	0.77	0.77	0.74
Titanic	0.80	0.78	0.82	0.84	0.76
Adult	0.82	0.83	0.81	0.81	0.83

(b) Forêt aléatoire (100 estimateurs)

Dataset	F1-Sc.	Précis.		Rapp.	
		C0	C1	C0	C1
Compas	0.65	0.66	0.66	0.65	0.66
Diabet.	0.72	0.77	0.71	0.65	0.81
Titanic	0.77	0.79	0.77	0.75	0.79
Adult	0.78	0.87	0.73	0.66	0.90

(c) Notre méthode (100 exécutions, 500 époques, 3 approx, entraînement séquentiel, p=2)

TAB. 8 – Comparaison de plusieurs processus d'entraînement, pour un modèle contenant 1 ou 3 approx. et un réseau de neurones, évaluant la performance sur la classe 1. (Jeu de données Adult, p=1.2, moyenne de 100 exécutions, 500 époques par exécution)

Modèle	1 Approx. + NN			3 Approx. + NN		
Méthode d'entraînement	Seq.	Simult.	Sépar.	Seq.	Simult.	Sépar.
Couverture du modèle	0.88	0.83	0.92	0.91	0.84	0.97
Approx. couverture	0.83	0.29	0.84	0.88	0.41	0.96
Couverture relative	0.94	0.34	0.91	0.97	0.49	0.99
Approx. zéros mal classifiés	0.26	0.58	0.26	0.28	0.44	0.32
Précision du modèle	0.77	0.80	0.78	0.77	0.80	0.75

TAB. 9 – Comparaison de plusieurs processus d'entraînement, pour un modèle contenant 1 ou 3 approx. et un réseau de neurones, évaluant la performance sur la classe 1. (Jeu de données Compas, p=1.2, moyenne de 100 exécutions, 500 époques par exécution)

Modèle	1 Approx. + NN			3 Approx. + NN		
Méthode d'entraînement	Seq.	Simult.	Sépar.	Seq.	Simult.	Sépar.
Couverture du modèle	0.62	0.58	0.65	0.70	0.60	0.70
Approx. couverture	0.47	0.34	0.47	0.61	0.40	0.61
Couverture relative	0.75	0.57	0.72	0.87	0.64	0.87
Approx. zéros mal classifiés	0.28	0.46	0.28	0.36	0.45	0.33
Précision du modèle	0.66	0.66	0.66	0.63	0.66	0.65

les meilleures performances, tout en étant capable de produire des approximateurs d'une taille suffisemment petite pour vérifier formellement autant de décisions que possible. Pour évaluer la qualité du modèle global, nous rapportons ici le meilleur modèle que nous avons trouvé pour un ensemble de problèmes simples. Une recherche exhaustive sur Kaggle et d'autres référentiels de machine learning a été menée pour identifier les solutions les plus prometteuses. Les solutions optimales, qui utilisaient des réseaux de neurones ou des forêts aléatoires, ont été sélectionnées pour comparaison (voir Tableau 7). L'objectif était de développer un modèle global qui pourrait atteindre les meilleures performances possibles.

Le Tableau 7 démontre que notre approche donne de très bons résultats. Notre méthode atteint des F1-scores proches de ceux des deux autres méthodes mais, comme nous le verrons, elle a l'avantage de pouvoir donner une explication pour / vérifier formellement une grande partie des prédictions positives (75% dans le pire jeu de données étudié 9 et 94% dans le meilleur cas étudié 8). Un problème restant est le rappel sur la classe 1 et la précision sur la classe 0 qui sont généralement inférieurs aux autres méthodes. Un autre problème encore à résoudre concerne l'encodage et la discrétisation des caractéristiques non binaires. À l'heure actuelle, la méthode actuelle peut conduire à trop de caractéristiques pour compiler des réseaux de neurones binaires (cf. Tableau 6). De plus, seul le réseau de neurones binaire nécessite des caractéristiques binaires mais pour des raisons de simplicité, à la fois le réseau de neurones et les approx. sont entraînés avec des jeux de données binarisés. Une solution pourrait être de mettre en œuvre une couche précédant les approx. qui gérerait automatiquement le processus de binarisation. Nous travaillons dans cette direction.

TAB. 10 – Comparaison de plusieurs processus d'entraînement, pour un modèle contenant 1 ou 3 approx. et un réseau de neurones, évaluant la performance sur la classe 1. (Jeu de données Titanic, p=1.2, moyenne de 100 exécutions, 500 époques par exécution)

Modèle	1 Approx. + NN			3 Approx. + NN		
Méthode d'entraînement	Seq.	Simult.	Sépar.	Seq.	Simult.	Sépar.
Couverture du modèle	0.75	0.75	0.78	0.78	0.79	0.82
Approx. couverture	0.60	0.56	0.61	0.74	0.68	0.76
Couverture relative	0.80	0.75	0.78	0.95	0.86	0.93
Approx. zéros mal classifiés	0.15	0.13	0.16	0.23	0.22	0.26
Précision du modèle	0.78	0.79	0.77	0.76	0.77	0.74

TAB. 11 – Comparaison de plusieurs processus d'entraînement, pour un modèle contenant 1 ou 3 approx. et un réseau de neurones, évaluant la performance sur la classe 1. (Jeu de données Diabetes, p=1.2, moyenne de 100 exécutions, 500 époques par exécution)

Modèle	1 Approx. + NN			3 Approx. + NN		
Méthode d'entraînement	Seq.	Simult.	Sépar.	Seq.	Simult.	Sépar.
Couverture du modèle	0.79	0.77	0.82	0.81	0.84	0.91
Approx. couverture	0.69	0.66	0.68	0.79	0.79	0.89
Couverture relative	0.87	0.86	0.83	0.97	0.93	0.98
Approx. zéros mal classifiés	0.32	0.22	0.31	0.26	0.30	0.38
Précision du modèle	0.71	0.74	0.71	0.70	0.73	0.67

Notre dernière observation concerne la manière dont les différents réseaux de neurones sont entraînés. Les tableaux 8, 9, 10 et 11 montrent que l'entraînement séquentiel et l'entraînement séparé sont les meilleures méthodes pour couvrir le plus de cas avec des réseaux approx., et entre ces deux méthodes, l'entraînement séquentiel semble donner légèrement moins de mauvaises classifications et de couverture que l'entraînement séparé. Quant à la méthode d'entraînement simultané, elle conduit généralement à beaucoup plus de mauvaises classifications de zéros et à une couverture plus faible. De plus, en utilisant la méthode séquentielle ou séparée, l'entraînement peut être facilement poursuivi à une date ultérieure en ajoutant de nouveaux réseaux approx., un autre avantage est la capacité de changer la fonction de perte pour chaque partie du modèle global contrairement à la méthode simultanée qui ne permet pas une mise en œuvre facile de ce comportement.

5 Conclusion

L'objectif de la méthode que nous proposons est de permettre la vérification formelle des cas simples décidés par des réseaux neuronaux, qui peuvent être décidés par des sous-réseaux de taille raisonnables. Nous démontrons qu'il est possible d'intégrer un petit réseau neuronal dans un modèle de réseau neuronal plus grand de manière à garantir que si l'un des petits réseaux neuronaux décide "Oui" pour une décision, le modèle global décidera également "Oui". Nous démontrons que, même sur des exemples non triviaux, de très petits réseaux neuronaux

(NN) peuvent capturer une proportion élevée de décisions pour lesquelles la vérification formelle ou l'explication peut être effectuée efficacement.

Dans le pire des cas, nous montrons que plus de 75% des décisions peuvent être vérifiées formellement (plus de 94% sur d'autres exemples). Cela implique que, sans perte de performance, notre système peut prendre de manière autonome des décisions sûres et fiables dans la grande majorité des cas. De plus, le système est capable d'identifier les instances où une décision est non sécurisée. Cela se produit lorsque tous les approximateurs indiquent une décision de "0", tandis que le grand réseau neuronal indique une décision de "1". Dans de tels cas, la décision peut être présentée à un humain (ou à toute autre méthode) pour vérification. Nous travaillons actuellement à l'extension de cette approche pour limiter à la fois les décisions "1" et "0" et pour les problèmes de classification.

Références

Audemard, G., S. Bellart, L. Bounia, F. Koriche, J.-M. Lagniez, et P. Marquis (2022). On the Explanatory Power of Boolean Decision Trees. *Data and Knowledge Engineering 142*, 102088.

Breiman, L. (1996). Bagging predictors. *Machine learning 24*, 123–140.

Chan, H. et A. Darwiche (2012). Reasoning about bayesian network classifiers.

Choi, A., W. Shi, A. Shih, et A. Darwiche (2017). Compiling neural networks into tractable boolean circuits. *intelligence*.

Courbariaux, M., I. Hubara, D. Soudry, R. El-Yaniv, et Y. Bengio (2016). Binarized neural networks : Training deep neural networks with weights and activations constrained to +1 or -1.

Darwiche, A. et P. Marquis (2002). A knowledge compilation map. *Journal of Artificial Intelligence Research 17*, 229–264.

den Broeck, G. V., A. Lykov, M. Schleich, et D. Suciu (2021). On the tractability of shap explanations.

Goulet, J.-A., L. H. Nguyen, et S. Amiri (2021). Tractable approximate gaussian inference for bayesian neural networks.

Ignatiev, A., N. Narodytska, et J. Marques-Silva (2018). Abduction-based explanations for machine learning models.

Learning, A.-S. (2019). Verifying binarized neural networks. In *Theory and Applications of Satisfiability Testing–SAT 2019 : 22nd International Conference, SAT 2019, Lisbon, Portugal, July 9–12, 2019, Proceedings*, Volume 11628, pp. 354. Springer.

Leofante, F., N. Narodytska, L. Pulina, et A. Tacchella (2018). Automated verification of neural networks : Advances, challenges and perspectives.

Nohara, Y., K. Matsumoto, H. Soejima, et N. Nakashima (2022). Explanation of machine learning models using shapley additive explanation and application for real data in hospital. *Computer Methods and Programs in Biomedicine 214*, 106584.

Shi, W., A. Shih, A. Darwiche, et A. Choi (2020). On tractable representations of binary neural networks.

Shih, A., A. Choi, et A. Darwiche (2018). A symbolic approach to explaining bayesian network classifiers.

Tang, Y., K. Hatano, et E. Takimoto (2023). Boosting-based construction of bdds for linear threshold functions and its application to verification of neural networks.

Zhang, Y., Z. Zhao, G. Chen, F. Song, et T. Chen (2023). Precise quantitative analysis of binarized neural networks : a bdd-based approach. *ACM Transactions on Software Engineering and Methodology 32*(3), 1–51.

Summary

The rapid advancements in AI and ML have emphasised the need for explainable and verifiable decision-making. However, Deep Neural Networks and large random forests resist traditional verification, functioning as opaque "black boxes" on which formal verification cannot be applied. This casts serious problems on these AI systems' safety, reliability, and predictability. This paper investigates the extent to which formal verification can be applied to neural networks without sacrificing accuracy. We present a method for approximating complex networks with simpler, verifiable models that explain or validate decisions. Our results demonstrate that, for well-known problems, over 94% of the system's decisions can be formally verified, enhancing trustworthiness without diminishing performance.

Reconcilier performance et explicabilité dans la classification du cancer du sein : Une approche basée sur la carte auto-organisatrice (SOM)

1 Introduction

Un débat persistant oppose deux communautés dans le domaine de la science des données et de la santé : la communauté du machine learning, centrée sur la performance des modèles, et la communauté médicale, qui privilégie l'explicabilité des résultats en lien avec les connaissances médicales. D'un côté, les algorithmes de machine learning, optimisés pour la précision, produisent des modèles souvent perçus comme des "boîtes noires", ce qui limite leur adoption dans des contextes cliniques sensibles (Ribeiro et al. (2016)). De l'autre, les méthodes favorisant l'explicabilité manquent parfois de la performance nécessaire à des prédictions robustes, ce qui peut réduire leur pertinence dans les situations critiques Doshi-Velez et Kim (2017).

Pour combler ce fossé, nous proposons l'utilisation des cartes auto-organisatrices(Kohonen, 1982) (SOM) appliquées au jeu de données Wisconsin Breast Cancer (William Wolberg (1993)). Ce case study a été choisi en raison de son importance dans le domaine de la classification médicale et de son utilisation fréquente dans la littérature pour comparer les méthodes de classification des tumeurs bénignes et malignes. Il a permis à de nombreuses études de développer et comparer des modèles performants, tels que les machines à vecteurs de support (SVM), les réseaux de neurones et les forêts aléatoires. Bien que ces approches atteignent des performances élevées, elles mettent principalement l'accent sur des métriques quantitatives (précision, F1-score) sans offrir une interprétation claire des résultats pour les cliniciens. Par exemple, les réseaux de neurones atteignent souvent des précisions supérieures à 98 %, mais leur caractère basé sur la detection d'une corrélation globale (sur tout le jeu de données) limite leur utilisation dans des contextes nécessitant une compréhension approfondie des décisions Modi et Ghanchi (2016); Benbrahim et al. (2020); Rumbe et Youh (2010).

Cette approche offre un compromis en permettant non seulement de préserver la performances en classification (attribution du label du neurone de la carte), mais aussi de visualiser et interpréter les relations complexes entre les différentes caractéristiques des tumeurs, répondant ainsi aux attentes des deux communautés.

2 Présentation du jeu de données

Le jeu de données Wisconsin Breast Cancer est une référence largement utilisée dans les domaines de la recherche médicale et de l'apprentissage automatique. Il comprend 569 obser-

vations issues de patients, avec des mesures extraites d'images numérisées de masses mammaires. Ces mesures permettent de caractériser différentes propriétés des noyaux cellulaires présents dans les images, fournissant ainsi un riche ensemble de données pour la classification des tumeurs.

2.1 Caractéristiques du jeu de données

Le jeu de données comporte 30 caractéristiques numériques dérivées de 10 attributs fondamentaux mesurés sur les noyaux cellulaires :

— Rayon,
— Texture,
— Périmètre,
— Surface,
— Douceur (Smoothness),
— Compacité (Compactness),
— Concavité,
— Points concaves,
— Symétrie,
— Dimension fractale.

Pour chaque attribut, trois statistiques sont calculées :

— Valeur moyenne (mean) : moyenne des observations dans une image,
— Erreur standard (error) : écart-type des observations,
— Valeur extrême (worst) : observation la plus extrême pour chaque image.

Ces statistiques offrent une vision complète des propriétés locales et globales des tumeurs analysées.

2.2 Classe cible

La variable cible est binaire et distingue deux classes :

— Tumeurs bénignes : généralement associées à des valeurs faibles pour la majorité des caractéristiques.
— Tumeurs malignes : caractérisées par des valeurs plus élevées et des variations plus importantes, reflétant une hétérogénéité accrue due aux différents stades de développement.

3 Méthodologie

Le jeu de données Wisconsin Breast Cancer contient 569 échantillons et 30 caractéristiques extraites de l'analyse des noyaux cellulaires. Nous avons appliqué une carte auto-organisatrice (SOM) de 15x15 neurones (soit 225 neurones au total) pour projeter ces données multidimensionnelles dans un espace bidimensionnel tout en conservant leur structure topologique. Cette taille a été choisie comme un compromis entre la capacité de capturer des détails fins et une représentation lisible compte tenu du nombre d'observations.

Après l'apprentissage compétitif, où chaque observation est associée au neurone le plus proche (BMU), nous avons visualisé les résultats via :

— Une carte des distances pour identifier les regroupements similaires.
— La projection des variables pour analyser la distribution des caractéristiques.
— Un clustering hiérarchique (CAH) pour regrouper les neurones en clusters homogènes.

Les paramètres, tels que la fonction de voisinage gaussienne et un critère de convergence basé sur la stabilisation des poids, ont été optimisés pour garantir une représentation précise et interprétable des données.

4 Analyse de la carte des distances

la carte des distances permet de capturer les relations topologiques des données en mettant en lumière les similarités et les divergences des observations. Cette visualisation est un outil essentiel pour comprendre la structure des données et interpréter les regroupements identifiés par la SOM. La carte des distances (Figure 1) illustre les distances moyennes normalisées entre chaque neurone et ses voisins dans la carte SOM. La couleur de chaque neurone représente cette distance moyenne, permettant de visualiser les regroupements et les dissimilarités des données.

L'analyse du gradient de couleur révèle des motifs distincts :

— Zones cohésives (cellules claires) : situées en haut à droite de la carte, ces régions indiquent des distances faibles, reflétant une forte similarité entre les observations associées. Ces zones correspondent principalement aux cellules bénignes, caractérisées par une régularité et une homogénéité accrues.
— Zones dispersées (cellules sombres) : localisées dans le bas et le côté gauche de la carte, elles présentent des distances élevées, indiquant une plus grande hétérogénéité dans les observations. Ces régions sont associées aux cellules malignes, qui affichent des variations importantes liées aux stades et chemins de leur développement.

Cette organisation spatiale met en évidence les différences structurelles entre les cellules bénignes et malignes. La classification binaire des échantillons est également bien alignée avec cette organisation : les cellules bénignes (◯) sont regroupées dans des zones bien définies, tandis que les cellules malignes (◇) lignes and Benign se répartissent dans des zones plus variées.

5 Analyse des cartes des caractéristiques

Afin d'étudier la relation entre les différentes caractéristiques, nous avons projeté chaque variable du jeu de données sur la carte SOM (Figure 3). Cette projection permet d'analyser la structure de chaque caractéristique ainsi que les relations entre elles.

5.1 Structuration des caractéristiques

La carte montre une structuration claire (gradient monotone) pour la majorité des caractéristiques, représentée par des gradients bien définis. Toutefois, certaines caractéristiques, comme l'erreur de texture, présentent des structures moins organisées avec plusieurs composantes.

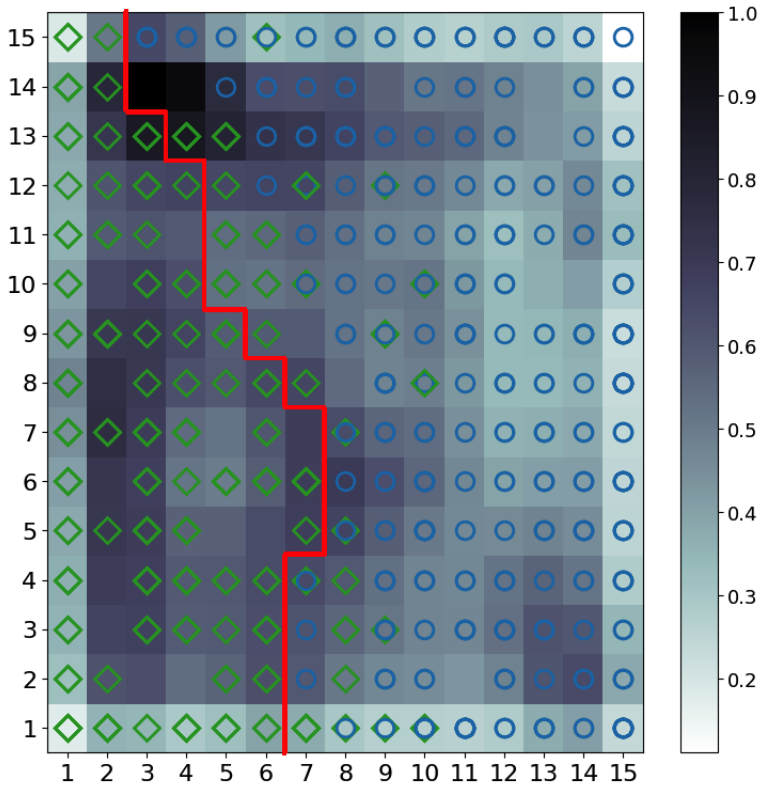


FIG. 1 – *Matrice des distances pour la carte SOM du jeu de données sur le cancer du sein : la couleur de chaque neurone représente la distance moyenne entre celui-ci et ses voisins. La forme à l'intérieur de chaque neurone indique le type de tumeur (◇ : Maligne et ◯ : Bénigne). La ligne rouge sépare les neurones en deux classes, en utilisant la classification hiérarchique ascendante.*

Les caractéristiques peuvent être regroupées en trois sous-ensembles principaux en fonction de leur structuration sur la carte :

1. Premier sous-ensemble : Les variables comme la surface (Area), le rayon (Radius) et le périmètre (Perimeter) suivent une structuration le long de la première diagonale.
2. Deuxième sous-ensemble : Les caractéristiques telles que la douceur (Smoothness), la symétrie (Symmetry) et la dimension fractale (Fractal Dimension) montrent une structuration le long de la deuxième diagonale.
3. Troisième sous-ensemble : Les caractéristiques comme la texture (Texture), la compacité (Compactness) et les points concaves (Concave Points) varient principalement le long de l'axe horizontal.

5.2 Corrélations intra-caractéristiques (valeurs moyennes, erreurs et extrêmes)

La carte révèle trois comportements distincts :

1. Premier pattern : Les caractéristiques comme le rayon, le périmètre, la surface, la douceur et la dimension fractale montrent un gradient bien structuré. Cela reflète une forte corrélation entre les valeurs moyennes (mean) et les valeurs extrêmes (worst).
2. Deuxième pattern : Pour la compacité, la concavité et les points concaves, deux composantes distinctes d'erreur sont observées, menant à des variations dans les valeurs extrêmes.
3. Troisième pattern : La texture et la symétrie présentent trois comportements différents : certains alignés avec les moyennes, et d'autres déviant fortement.

5.3 Corrélations inter-caractéristiques

La carte SOM met également en évidence trois groupes principaux de caractéristiques :

1. Groupe 1 : Le rayon, le périmètre et la surface, représentant des mesures liées à la taille des cellules.
2. Groupe 2 : La douceur, la symétrie et la dimension fractale, décrivant les attributs de forme des cellules.
3. Groupe 3 : La texture, la compacité, la concavité et les points concaves, caractérisant la composition des cellules.

Ces observations concordent avec les résultats d'un clustering hiérarchique basé sur les corrélations des caractéristiques, renforçant la pertinence des regroupements identifiés [1].

De manière générale, la bonne structuration de la cartes sur la majorité des variables permet d'identifier des sous-groupes de patients ayant des caractéristiques similaires, facilitant ainsi la tâche de classification. Pour certaines autres variables, comme la texture ou la symétrie, les projections peuvent présenter des variations plus complexes(non détectables par les méthodes de classification classiques - SVM, DT, MLP, etc-), reflétant des comportements plus variés ou des relations non linéaires entre ces variables et le diagnostic.

1. https ://scikit-learn.org/stable/auto_examples/inspection/plot_permutation_importance_multicollinear.html

6 Clustering hiérarchique des tumeurs du sein

L'analyse de la matrice des distances (Figure 1) confirme la pertinence de la carte SOM pour représenter la structure topologique des données. En nous basant sur les distances entre les neurones de la carte SOM, nous avons appliqué une classification ascendante hiérarchique (CAH) afin de regrouper les neurones en clusters homogènes.

6.1 Regroupement en deux clusters principaux

La première étape a consisté à diviser la carte en deux clusters principaux :

— Cluster 1 : Composé de 75 neurones, il regroupe principalement des échantillons caractérisés par des valeurs élevées pour la majorité des caractéristiques. Ces neurones sont associés à des tumeurs malignes, qui présentent une forte hétérogénéité en raison des différents stades de développement.

— Cluster 2 : Constitué de 150 neurones, il regroupe des observations plus homogènes avec des valeurs généralement faibles pour les caractéristiques. Ce cluster est majoritairement associé aux tumeurs bénignes.

Les neurones situés à la frontière entre ces deux clusters contiennent souvent un mélange de tumeurs bénignes et malignes, mais sont dominés par des échantillons bénins. Cela reflète la complexité de certaines observations où des cellules malignes peuvent coexister avec des cellules bénignes dans un même échantillon.

6.2 Refinement en sous-clusters

Pour affiner l'analyse, nous avons divisé les deux clusters principaux en un total de 11 sous-clusters (Figure 2). Parmi ces sous-clusters :

— 5 sous-clusters sont associés aux tumeurs malignes. Ces groupes présentent des caractéristiques hétérogènes et des valeurs extrêmes pour certaines variables, comme la compacité, la concavité et les points concaves.

— 6 sous-clusters sont liés aux tumeurs bénignes. Ces groupes sont globalement homogènes, avec des valeurs plus faibles pour la majorité des caractéristiques .

Chaque sous-cluster reflète des profils spécifiques, permettant une meilleure compréhension des propriétés des tumeurs et une classification plus détaillée.

7 Discussion

L'identification de 11 sous-clusters distincts via la carte SOM et le clustering hiérarchique offre une analyse détaillée de la variabilité des tumeurs bénignes et malignes. Chaque sous-cluster reflète des profils spécifiques, soulignant des distinctions fines entre les observations. Ces résultats sont en adéquation avec les propriétés biologiques connues des tumeurs mammaires.

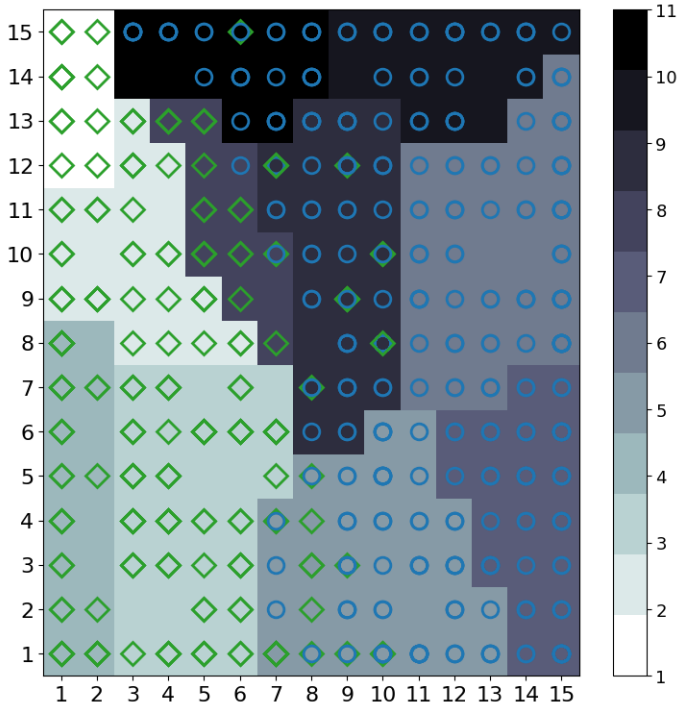


FIG. 2 – *Hierarchical clustering of the map into eleven sub-clusters.*

7.1 Sous-clusters associés aux tumeurs malignes

Les 5 sous-clusters malins mettent en évidence une hétérogénéité notable, reflétant les variations entre les stades de développement et les sous-types tumoraux.

Sous-cluster 1 : Caractérisé par des valeurs élevées pour la compacité, la concavité et les points concaves, mais des valeurs relativement modérées pour les dimensions cellulaires (rayon, périmètre, surface). Ces observations sont cohérentes avec des tumeurs malignes à croissance modérée mais avec une déstructuration importante des bords cellulaires, un phénomène observé dans les carcinomes canalaires in situ, (Schnitt (2003)).

Sous-cluster 2 : Présente des valeurs élevées pour les mesures de compacité et de concavité, mais aussi une texture élevée. Ce sous-cluster reflète potentiellement des tumeurs présentant des caractéristiques mixées, combinant un désordre cellulaire localisé et une hétérogénéité texturale globale, fréquente dans les lésions précancéreuses Hanby et walker (2004).

Sous-cluster 3 : Caractérisé par des dimensions élevées (rayon, périmètre, surface) avec une texture et une symétrie modérées. Ces profils sont associés à des tumeurs invasives où une croissance rapide des cellules favorise des tailles accrues, comme dans les carcinomes infiltrants de haut grade (Rakha et al. (2008)).

Sous-cluster 4 : Présente les valeurs les plus extrêmes pour les dimensions cellulaires (rayon, périmètre, surface) et des caractéristiques marquées de compacité et de concavité. Ces profils correspondent souvent à des carcinomes agressifs, associés à un haut potentiel métastatique et une prolifération rapide (Salamat (2010)).

Un sous-cluster spécifique **(Sous-cluster 8)** a attiré notre attention, particulierement intéressant il regroupe des échantillons ayant des moyennes similaires à celles des tumeurs bénignes, mais des valeurs extrêmes pour certaines variables comme la texture et la symétrie. Cela reflète des échantillons où des cellules malignes coexistent avec des cellules bénignes, un phénomène clinique observé dans les lésions hétérogènes , (Page et Dupont (1988)). Ce profil est particulièrement utile pour détecter des cas à risque élevé mal classifiés par des approches globales.

7.2 Sous-clusters associés aux tumeurs bénignes

Les 6 sous-clusters bénins présentent une homogénéité globale plus forte, reflétant des caractéristiques similaires à celles des tissus normaux.

Sous-cluster 5 : Présente des valeurs modérées pour la texture et des dimensions cellulaires moyennes (rayon, périmètre, surface). Ce profil correspond à des adénofibromes typiques, qui sont des lésions bénignes les plus fréquentes chez les femmes jeunes (Guray et Sahin (2006)).

Sous-cluster 6 : Caractérisé par les plus faibles valeurs pour la texture. Ces observations sont associées à des lésions fibrotiques ou des masses à faible densité cellulaire, comme les modifications fibro-kystiques (Collins et al. (2015)).

Sous-cluster 7 : Montre de faibles valeurs pour les dimensions cellulaires (rayon, périmètre, surface) et la compacité. Ce sous-cluster est représentatif des lésions bénignes à faible prolifération, telles que les hyperplasies ductales non atypiques (ELSTON et ELLIS (1991)).

Sous-cluster 9 : Considéré comme "moyen" parmi les clusters bénins, il présente des valeurs proches de la moyenne pour toutes les caractéristiques. Ce profil peut être lié à des masses bénignes communes, souvent observées lors de dépistages de routine.

(Sous-cluster 10) : Regroupe des échantillons présentant une douceur et une symétrie élevées, mais des dimensions cellulaires relativement faibles. Ces observations correspondent à des tumeurs fibreuses, souvent associées à une faible probabilité de transformation maligne (Hoda et Patel (2018)).

Sous-cluster 11 : Présente des valeurs élevées pour les erreurs de texture, de compacité et de concavité. Cela reflète des échantillons bénins avec une variabilité intra-tumorale significative, pouvant être liés à des processus inflammatoires ou à des changements hormonaux (Tavassoli (1999)).

7.3 Contributions cliniques des clusters

Les clusters malins permettent de distinguer des stades et des sous-types tumoraux importants pour la prise en charge clinique. Par exemple :

Les clusters associés aux grandes dimensions cellulaires (3 et 4) pourraient indiquer des tumeurs nécessitant une intervention rapide. Les clusters présentant des caractéristiques mixtes (comme le sous-cluster 8) sont particulièrement pertinents pour identifier des cas de lésions complexes souvent mal détectées. Pour les clusters bénins, la différenciation fine aide à confirmer le diagnostic de lésions non malignes et à éviter des traitements inutiles. Les clusters comme le 10 (douceur et symétrie élevées) peuvent indiquer des lésions stables à surveiller sans biopsie invasive.

7.4 Limites et perspectives

Bien que ces 11 clusters offrent une interprétation détaillée des données, reliant des observations quantitatives à des propriétés cliniques connues, leur validation clinique reste essentielle. des futures études pourraient confirmer/valider le nombre 11 de sous-clusters ainsi qu'inclure des études sur des données histopathologiques ou des résultats de biopsies pour renforcer la robustesse de l'approche proposée.

8 Conclusion

Ce travail met en lumière un aspect souvent négligé par les approches classiques de machine learning : certaines caractéristiques, bien que peu corrélées globalement avec les résultats (par exemple la texture ou la symétrie des cellules), peuvent avoir une importance locale cruciale, notamment dans la classification des tumeurs malignes à des stades avancés. Le SOM permet de capturer ces subtilités et de les rendre visibles pour une analyse approfondie. aussi elle montre qu'il est possible de trouver un équilibre entre la performance des modèles et leur explicabilité. En explorant les données via un SOM, nous avons non seulement amélioré la classification des tumeurs bénignes et malignes, mais aussi fourni des explications visuelles claires, ce qui répond aux besoins des deux communautés. Cette approche hybride est une voie prometteuse pour faciliter l'adoption des outils de machine learning dans les environnements médicaux, où la transparence et l'explicabilité sont essentielles.

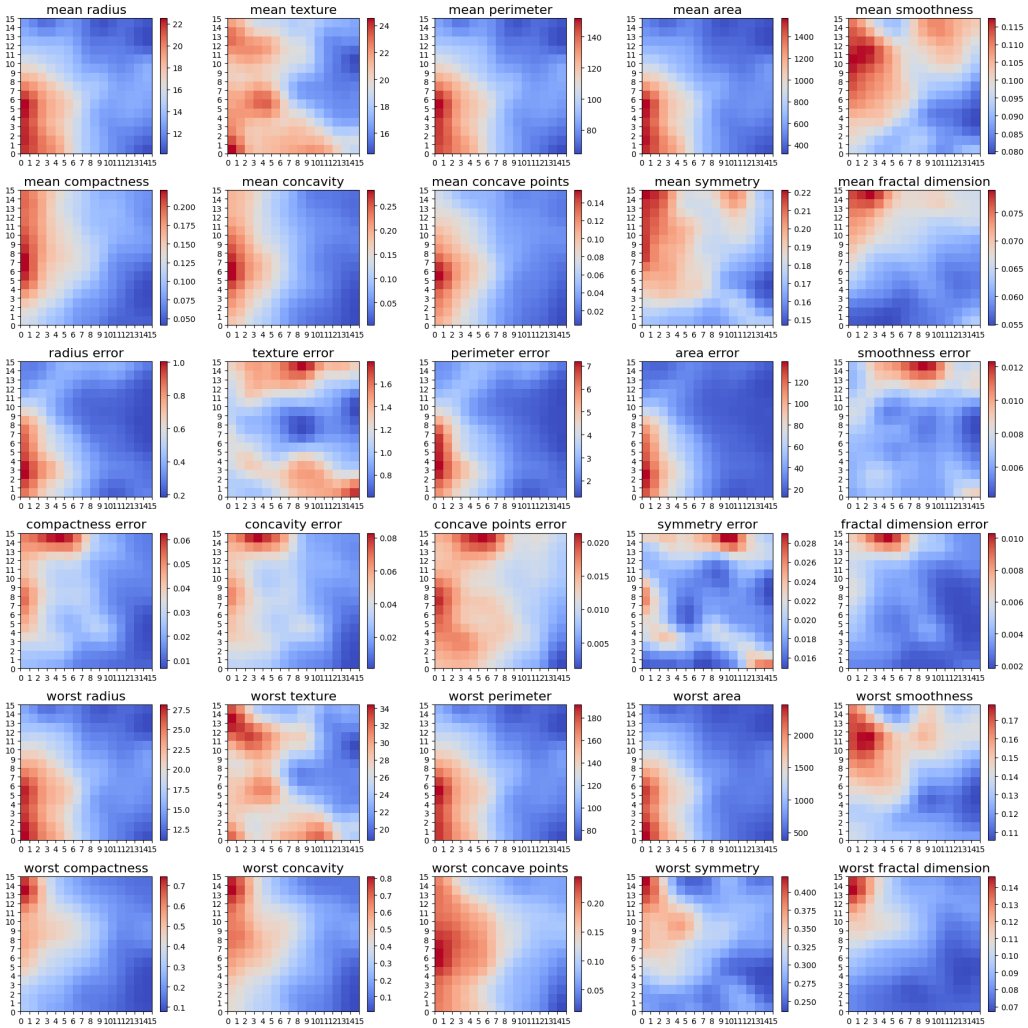


FIG. 3 – *La projection de la carte SOM(15x15) sur les 30 variables de la base Wisconsin Breast Cancer dataset*

Références

Benbrahim, H., H. Hachimi, et A. Amine (2020). *Comparative Study of Machine Learning Algorithms Using the Breast Cancer Dataset*, pp. 83–91. Springer International Publishing, doi:10.1007/978-3-030-36664-3_10.

Hoda, S. A. et A. Patel (2018). Rosai and ackerman's surgical pathology. *American Journal of Clinical Pathology 149*(6), 548–548, doi:10.1093/ajcp/aqy016.

Doshi-Velez, F. et B. Kim (2017). Towards a rigorous science of interpretable machine learning. *arXiv : Machine Learning*.

Ribeiro, M. T., S. Singh, et C. Guestrin (2016). "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, pp. 1135–1144. Association for Computing Machinery, doi:10.1145/2939672.2939778.

Modi, N. et K. Ghanchi (2016). A comparative analysis of feature selection methods and associated machine learning algorithms on wisconsin breast cancer dataset (wbcd). In S. C. Satapathy, A. Joshi, N. Modi, et N. Pathak (Eds.), *Proceedings of International Conference on ICT for Sustainable Development*, Singapore, pp. 215–224. Springer Singapore.

Collins, L. C., S. A. Aroner, J. L. Connolly, G. A. Colditz, S. J. Schnitt, et R. M. Tamimi (2015). Breast cancer risk by extent and type of atypical hyperplasia : An update from the <scp>n</scp>urses' <scp>h</scp>ealth <scp>s</scp>tudies. *Cancer 122*(4), 515–520, doi:10.1002/cncr.29775.

Salamat, M. S. (2010). Robbins and cotran : Pathologic basis of disease, 8th edition. *Journal of Neuropathology amp; Experimental Neurology 69*(2), 214–214, doi:10.1097/nen.0b013e3181cd8dbc.

Rumbe, G. et H. Youh (2010). Comparative study of classification techniques on breast cancer fna biopsy data. *International Journal of Interactive Multimedia and Artificial Intelligence 1*(3), 5–12, doi:10.9781/ijimai.2010.131.

Rakha, E. A., J. S. Reis-Filho, et I. O. Ellis (2008). Basal-like breast cancer : A critical review. *Journal of Clinical Oncology 26*(15), 2568–2581, doi:10.1200/jco.2007.13.1748.

Guray, M. et A. A. Sahin (2006). Benign breast diseases : Classification, diagnosis, and management. *The Oncologist 11*(5), 435–449, doi:10.1634/theoncologist.11-5-435.

Hanby, A. M. et C. walker (2004). Tavassoli fa, devilee p : Pathology and genetics : Tumours of the breast and female genital organs. who classification of tumours series - volume iv. lyon, france : Iarc press. *Breast Cancer Research 6*(3), 133, doi:10.1186/bcr788.

Schnitt, S. J. (2003). Benign breast disease and breast cancer risk : Morphology and beyond. *The American Journal of Surgical Pathology 27*(6), 836–841, doi:10.1097/00000478-200306000-00017.
en

Tavassoli, F. A. (1999). *Pathology of the breast* (2 ed.). Stamford, CT : Appleton & Lange.

William Wolberg, O. M. (1993). Breast cancer wisconsin (diagnostic). doi:10.24432/C5DW2B.

ELSTON, C. et I. ELLIS (1991). pathological prognostic factors in breast cancer. i. the value of

histological grade in breast cancer : experience from a large study with long-term follow-up. *Histopathology 19*(5), 403–410, doi:10.1111/j.1365-2559.1991.tb00229.x.

Page, D. L. et W. D. Dupont (1988). Histopathologic risk factors for breast cancer in women with benign breast disease. *Seminars in Surgical Oncology 4*(4), 213–217, doi:10.1002/ssu.2980040403.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics 43*(1), 59–69, doi:10.1007/bf00337288.

Summary

This study employs self-organizing maps (SOM) to classify benign and malignant tumors in the Wisconsin Breast Cancer dataset. It demonstrates that SOM enhances classification while providing an explainable visualization of key tumor characteristics.

Prise en compte des propriétés FATES en MLOps: perspectives et ambitions

Mireille Blay-Fornarino*, Jean-Michel Bruel**
Sébastien Mosser****
Frédéric Precioso***,*

*Université de Toulouse / CNRS-IRIT
**Laboratoire CNRS I3S
***INRIA Nice
****McMaster University
contact@fates-mlops.org

Résumé. Le mouvement MLOps reprend les objectifs DevOps de réduction des écarts entre les équipes de développement et d'opérations en intégrant la collaboration avec les équipes de *data scientists* et les phases liées à la construction et le déploiement des modèles de *Machine Learning* (ML). Le projet ANR FATES-MLOps a pour ambition d'étudier les propriétés extra-fonctionnelles telles que l'équité, la responsabilité, la transparence et la sécurité, regroupées en anglais sous l'acronyme FATES. En nous appuyant et en affinant les concepts et outils éprouvés du génie logiciel, nous souhaitons proposer une approche systématique et outillée pour la prise en compte de ces propriétés fondamentales dans le cycle de vie d'un logiciel développé en suivant une approche MLOps. Les verrous technologiques portent sur la formalisation et la mesure de ces propriétés en fonction des contextes et leur prise en charge systématique dans le processus MLOps par des mécanismes et algorithmes adaptés. Cela implique l'analyse et la conception des workflows de construction des modèles, les processus d'intégration et de déploiement, ainsi que la justification du respect de ces propriétés.

1 Motivations

Le mouvement MLOps [1] reprend les objectifs du mouvement DevOps (Kim et al., 2016), qui est né de la nécessité de réduire les écarts entre les équipes de développement et d'opérations, en y intégrant la collaboration avec les équipes de *data scientists* (DS) et les phases liées à la construction des modèles de *Machine Learning* (ML) (Testi et al., 2022). Ainsi, mener un projet qui intègre du ML en suivant une démarche MLOps implique l'automatisation, l'intégration et la surveillance à toutes les étapes de la construction d'un système ML, y compris l'entraînement, l'intégration, les tests, la publication, le déploiement et la gestion de l'infrastructure. Cette systématisation des processus de construction des modèles de ML s'accompagne d'une exigence sur la qualité des systèmes logiciels produits. Cependant, cette qualité

1. https://ml-ops.org/

reste à définir, étudier, formaliser, mesurer, notamment dans le contexte des systèmes intégrant du ML. La surveillance continue des modèles ML est cruciale pour garantir leur performance et leur qualité dans des contextes réels, notamment leur adaptation quand les données évoluent. Le mouvement international pour passer du “Model-centric AI" au “Data-Centric AI" met en exergue la nécessité de vérifier et justifier que les systèmes respectent notamment les propriétés FATES tout au long du processus de construction de systèmes intégrant du ML. Nées en 2014 de l'initiative FAT/ML [2], les propriétés d'équité (*Fairness*), responsabilité (*Accountability*), et de transparence (*Transparency*), ont été complétées par l'éthique (*Ethics*) pour donner le groupe de recherche Microsoft FATE [3], puis plus récemment par la sécurité et la sûreté (*Security*/*Safety*) pour donner les propriétés FATES, et le mouvement *Data for Good* [4] de l'Université de Columbia. Pour répondre à ces exigences, plusieurs algorithmes ont été développés pour aborder les propriétés à différents degrés (F, T, et S), tandis que d'autres propriétés reposent plus sur l'engagement (A et E). Les régulations internationales et la société en général exigent, de manière croissante, de la transparence et des responsabilités de la part des “développeurs" de systèmes utilisant ces modèles. Les états s'attachent aujourd'hui à proposer des cadres pour aider les organisations et les individus “à favoriser la conception, le développement, le déploiement et l'utilisation responsables des systèmes d'IA au fil du temps" (Tabassi, 2023; Garrido et al., 2023). Actuellement, il n'existe pas, à notre connaissance, d'étude systématique ni de support pour guider les scientifiques et/ou les ingénieurs en ML sur des indicateurs permettant le suivi des propriétés FATES. Il impacte l'ensemble du cycle de vie du logiciel de manière variable, en fonction des problèmes et des avancées dans le domaine. En s'appuyant et en affinant les concepts et outils du génie logiciel, notre projet FATES-MLOps [5] a pour ambition d'étudier les propriétés FATES, de proposer une démarche outillée systématique pour la prise en compte de ces propriétés fondamentales dans le cycle de vie d'un logiciel développé en suivant une approche MLOps. Le verrou principal auquel ce projet s'attaque donc est le suivant : **Peut-on inclure de manière systématique et évolutive la justification d'une construction et d'une exploitation FATES d'un système logiciel intégrant du ML ?**

Dans la section 2 nous dressons l'état de l'art couvrant le domaine du projet, à savoir les propriétés FATES, les outils existants, les aspects variabilité et justification. Dans la section 3 nous donnons les grandes lignes de nos contributions futures en la matière. Enfin, dans la section 4 nous dressons les actions à plus ou moins long termes qui nous permettrons d'atteindre nos objectifs.

2 État de l'art

Nous abordons l'état de l'art selon deux axes. D'une part les propriétés FATES en mettant l'accent sur les points à vérifier et d'autre part les outils qui doivent être adaptés pour aider à prendre en charge ces propriétés dans un processus MLOps.

2. https://www.fatml.org
3. https://www.microsoft.com/en-us/research/theme/fate
4. https://datascience.columbia.edu/news/2018/data-for-good-fates-elaborated
5. Projet ANR-24-IAS2-0002-01.

2.1 Les propriétés FATES

Les propriétés FATES se recoupent. Pour garantir l'équité, il est essentiel de pouvoir expliquer le modèle, d'assurer la fiabilité des données utilisées et de surveiller les dérives potentielles lors de l'exploitation du modèle. Sans transparence, la responsabilité devient plus complexe à définir. Dans cette section, nous présentons ces propriétés dans l'ordre de l'acronyme FATES, en mettant en évidence les mécanismes et algorithmes, lorsqu'ils existent, à intégrer dans un processus MLOps pour garantir ces propriétés.

Fairness/Équité

La recherche sur l'équité dans l'apprentissage automatique vise à garantir l'impartialité des décisions ou prédictions des modèles construits (Dorleon et al., 2023). Définir formellement l'équité est un domaine de recherche actif, impliquant des spécialistes en mathématiques, en informatique, en sciences sociales, et des juristes. Les biais peuvent apparaître à diverses étapes d'un processus ML (Suresh et Guttag, 2021). Des algorithmes sont proposés pour atténuer les biais, notamment le débiaisement des données lors de la collecte et l'analyse des modèles de ML (Feldman et al., 2015). Pour détecter les biais, de nombreuses métriques sont proposées, récemment Wachter et al. (2021) proposent la disparité démographique conditionnelle (CDD) comme référence statistique pour évaluer la discrimination potentielle dans les systèmes automatisés. Breck et al. (2017) identifient différentes formes de tests pour détecter ces dérives. Plusieurs travaux sur les *Large Language Models* (LLMs) mettent en avant une combinaison de ces approches (Brown et al., 2020; Ferrara, 2023).

Accountability/Responsabilités

Aujourd'hui, la nécessité pour les producteurs de systèmes logiciels d'assumer la responsabilité des choix effectués est largement discutée tant les parties prenantes sont nombreuses et impactantes à des niveaux différents [6]. La responsabilité signifie que la manière dont un résultat d'un modèle a été obtenu grâce à un système de bout en bout, est compréhensible/explicable, est vérifiable et est reproductible [7]. Le versionnement des modèles comprenant les informations sur les données d'apprentissage, les résultats des tests, ainsi que des environnements de calcul, est un moyen courant pour garantir la traçabilité. Des frameworks comme MLFlows [8] et les approches par conteneurs sont utilisés dans le contexte MLOps pour améliorer cette traçabilité qui reste cependant à renforcer (Chen et al., 2020). La réutilisation des modèles pré-entraînés est devenue indispensable dans la construction de nouveaux modèles, en particulier pour les approches basées sur les LLMs. En explicitant les dépendances à la version de ces modèles, la traçabilité est renforcée, mais la question de l'introspection des modèles, notamment en ce qui concerne les données utilisées pendant la phase de pré-entraînement, devient plus forte (Liu et al., 2023).

6. Air Canada, cf. `https://intelligence-artificielle.com/chatbot-air-canada-hallucine/`.
7. La CNIL en donne une définition différente, dans ce projet, nous nous limitons à la définition donnée ici : `https://www.cnil.fr/fr/developpement-des-systemes-dia-les-recommandations-de-la-cnil-pour-respecter-le-rgpd`.
8. `https://mlflow.org/`

Transparency/Transparence

L'IA explicable (XAI) est un champ de recherches très intenses visant à rendre les décisions des modèles d'IA compréhensibles par les humains, même si la pleine explication des modèles reste un défi (Cugny et al., 2022). Les algorithmes d'explication peuvent être classés en méthodes ante-hoc, nécessitant l'accès aux mécanismes internes du modèle, et en méthodes post-hoc n'accédant qu'aux prédictions du modèle (Lopardo et al., 2023, 2024). En production, l'utilisation d'algorithmes post-hoc est privilégiée. L'utilisation d'architectures à base d'événements est une solution pour atteindre le double objectif d'indépendance, permettant des traitements de surveillance adaptés, et une montée en charge (Klaise et al., 2020). Des défis persistent dans la surveillance et l'explication des modèles déployés, des solutions sont déjà disponibles pour en relever certains (Wang et al., 2024), mais déterminer les solutions techniques à partir des spécifications d'un problème reste une difficulté majeure qui entrave la production de systèmes d'IA transparents (Mill et al., 2024).

Ethics/Éthique

La question de l'éthique est intrinsèquement liée à la philosophie, et déterminer si un système est éthique ou acceptable dépend souvent du point de vue adopté, qui peut varier d'un individu à un autre, voire d'un contexte à un autre. En conséquence, évaluer l'éthique d'un système peut se situer en amont du projet. Dans le cadre de ce projet, nous nous inscrirons dans la logique des *Responsible AI Licences* (RAIL) Contractor et al. (2022).

Safety & Security/Sécurité

La prise en compte de la sécurité est une préoccupation largement documentée, y compris récemment dans le cadre du DevOps, par un focus appelé DevSecOps (Enoiu et al., 2023; Nigmatullin et al., 2022). Les spécificités de la sécurité dans MLOps vont surtout concerner la *privacy*. Les utilisateurs de solutions basées ML sont légitimement en interrogation du devenir des données (où sont stockées les données, qui y a accès, etc.). Nous utiliserons plus particulièrement l'exemple prégnant de l'anonymisation des données. L'autre facette de la sécurité en français (au sens de la *safety* en anglais), par exemple qui est responsable en cas de problème de sécurité, concerne plus les propriétés de responsabilité (*Accountability*) et sera donc abordée dans cette propriété. Enfin, la sécurité doit également garantir que les modèles de ML sont robustes face aux attaques et ne peuvent pas être utilisés à des fins malveillantes. Ce dernier point, bien que très actuel avec l'injection de codes malveillants par les prompts dans les LLMs, ne sera pas abordé parce qu'il pourrait représenter un projet à lui seul.

Contextualisation

Les propriétés FATES sont contextuelles au système logiciel. On ne recherche pas à garantir les mêmes propriétés, ou du moins pas à un même degré selon l'usage et le domaine (critique ou non, impliquant l'humain ou non, spécifique ou général, etc.). Ces propriétés sont invasives dans l'ensemble du cycle de vie d'un logiciel, par exemples, dans l'analyse du problème (est-il éthique d'aborder cette question ?), dans la collecte des données (est-ce que les données collectées sont équitables ?), dans les choix des modèles (que sait-on des décisions prises par les modèles produits ?), dans les traitements opérés sur les données pour apprendre (est-ce que les

choix pour améliorer l'efficacité de l'entraînement sont pris en responsabilité ?), dans les traitements réalisés en exploitation (est-ce que les données utilisées pour renforcer l'apprentissage ne brise pas l'équité du modèle ?), etc. La surveillance de ces propriétés évolue en fonction des connaissances que nous avons des systèmes de ML et des cas réels observés. Par exemple, si certains types de biais sont connus et des algorithmes ont été définis pour pallier ces biais, d'autres tels que la production d'exemples adversaires sont proposés chaque jour. Cette évolution est si forte que le document de référence produit par le NIST[9] en matière des risques liés à l'IA, est conçu comme un document vivant (Tabassi, 2023). Dans Sculley et al. (2015) et Breck et al. (2017), les auteurs mettent en évidence différents facteurs de risque spécifiques au ML à prendre en compte dans la conception du système. Bien que nous ayons choisi un angle d'attaque différent, l'étude des propriétés FATES adresse différents éléments de dettes, dont ceux dits liés aux changements dans le monde extérieur tels que le monitoring et le test, le choix de métriques, mais aussi la gestion du processus mise à jour et de reconstruction des modèles.

2.2 Les outils en support au FATES MLOps

Il existe de nombreux algorithmes et outils qui visent à mesurer et garantir les propriétés FATES et ceux-ci sont en plein développements rapides et concurrents. Nous abordons, dans notre projet, la question davantage d'un point de vue Génie Logiciel et intégrateur.

Le test et la surveillance

Dans Breck et al. (2017), les auteurs mettent en exergue la difficulté de formuler des tests spécifiques, puisque le comportement réel d'un modèle de prédiction donné est difficile à spécifier a priori. En comparant l'entraînement d'un modèle à de la compilation, ils proposent différentes approches du test complémentaires où la source est à la fois le code et les données d'entraînement. Même si ces tests ne sont pas liés aux propriétés FATES, il est intéressant de reprendre certaines d'entre elles comme les exigences de méta-niveaux pour réduire les biais ou les contrôles de *privacy*.

Les environnements

L'un des grands défis du MLOps dans le contexte du suivi des propriétés FATES est de concevoir des systèmes qui intègrent à la fois les bons composants pour adapter les données, de la surveillance adaptée des déploiements, déclenchent des alertes, assurent un versionnement et une traçabilité des modèles. Actuellement, plusieurs outils d'automatisation du *machine learning* sont disponibles, tels que MLFlow (Zaharia et al., 2018), SageMaker[10] et Kubeflow[11]. Cependant, à notre connaissance, aucun de ces outils n'aborde explicitement la question du support à la vérification des propriétés FATES, que ce soit lors de la production des modèles ou de la vérification des composants de surveillance de ces propriétés. En intégrant des solutions de surveillance des propriétés FATES dans des piles logicielles telles qu'Hugging Face

9. *National Institute of Standards and Technology, U.S. Department of Commerce.*
10. `https://aws.amazon.com/sagemaker/`
11. `https://www.kubeflow.org/`

et Langchain, nous visons à répondre aux besoins croissants de la communauté en matière de contrôle qualité et de fiabilité des systèmes intégrant du ML.

2.3 Des exigences aux justifications

Construire des systèmes efficaces de science des données est d'autant plus difficile que les solutions ML disponibles ne cesse de croître (Zaharia et al., 2018). Pour aider les DS à sélectionner des pipelines cohérents en fonction de leur problème, nous avons appréhendé cette diversité sous la forme d'une ligne de produit (Amraoui et al., 2022), que nous exploitons pour identifier des solutions à réutiliser (Brault et al., 2023) et avons modélisé un méta-modèle de pipeline de ML pour les prendre en charge dans un contexte DevOps (Benni et al., 2019). Sur la base de ces travaux préliminaires, nous visons à étendre notre approche pour intégrer spécifiquement les propriétés FATES dans les workflows de ML (Vasudevan et Kenthapadi, 2020), en capturant la variabilité des algorithmes avec la logique des *feature models*.

Dans les contextes critiques, la documentation joue un rôle essentiel dans l'accréditation des produits en établissant la confiance dans leur processus de développement et de conception. L'objectif est d'élaborer des justifications pour rassurer sur la gestion appropriée du processus de développement et le respect des normes. Justifier qu'un système logiciel respecte les propriétés FATES [12], rejoint le même objectif. Polacsek et al. (2018) ont introduit les diagrammes de justification (JD), en conformité avec l'IEC 62304 pour organiser les éléments contribuant à la justification d'un résultat. Dans Duffau et al. (2018), nous avons étendu et appliqué ces diagrammes à l'élaboration de dispositifs médicaux critiques, puis plus récemment, à la justification de pipelines DevOps à grande échelle (Mosser et al., 2023). Nous proposons de poursuivre ces travaux dans le cadre du MLOps.

3 Contribution

3.1 Approche/Solution et plus-value scientifique

Dans le domaine dynamique et en constante évolution du ML, et plus spécifiquement dans le cadre d'une approche responsable des systèmes intégrant des LLMs, les recherches menées par les juristes, philosophes et sociologues revêtent une importance capitale. Cependant, pour que ces avancées puissent profiter à un large éventail d'acteurs, y compris les entreprises de taille plus modeste, il est essentiel de rendre les concepts et les pratiques accessibles et intégrables dans les processus de développement logiciel. C'est précisément l'objectif de notre projet, qui s'attaque de manière pragmatique à un problème complexe, embrassant de multiples dimensions (cf. Figure 1). En alignant les développements algorithmiques sur les besoins concrets de la production logicielle, notre ambition est de fournir à la communauté scientifique des principes, des théories et des outils favorisant une approche systématique de l'évaluation et de la traçabilité des propriétés FATES, de la phase d'analyse jusqu'à la mise en exploitation. Dans cette démarche, nous nous engageons également à mettre en évidence les limites ainsi que les progrès réalisés dans ce domaine.

12. Cf. le début de formalisation Microsoft Responsible AI Standard, v2 GENERAL REQUIREMENTS disponible ici : `https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RE5cmFl?culture=fr-fr&country=fr`.

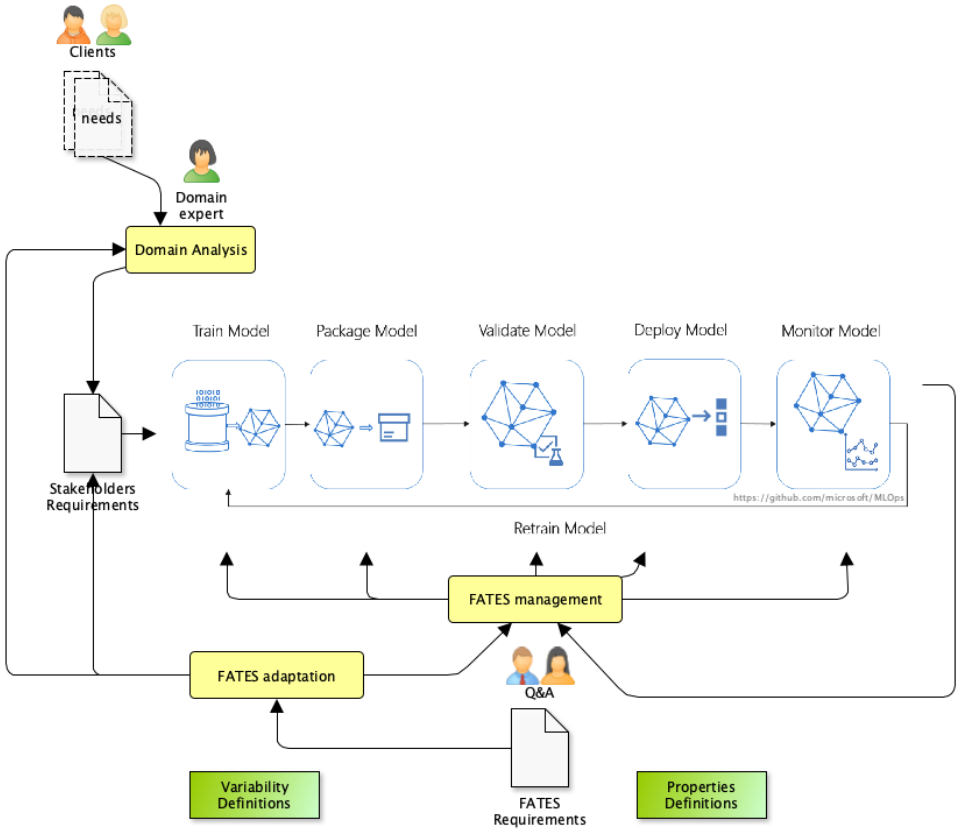


FIG. 1 – *Vue d'ensemble du projet FATES-MLOps*

3.2 Sorties attendues et mesure objective de qualité

Les sorties attendues du projet sont directement liées aux cas d'usage envisagés, avec d'une part un ChatBot en Wolof, dont une évaluation tout au long du développement sera réalisée, et d'autre part des composants intégrés à StarCoder et évalués sur la génération des codes. Nous évaluerons par exemple si les codes générés par StarCoder avec nos propriétés FATES ont récupéré, eux aussi, des propriétés FATES. La caractérisation des propriétés FATES dans leur globalité servira de guidelines pour les implémentations FATES futures. Nous produirons des exemples de prise en compte des propriétés FATES et de leur justification dans un processus d'intégration et de déploiement continu. Nous produirons des éléments de mesures sur les exigences et la qualité des propriétés en utilisant au moins des métriques de l'état de l'art. Nous distribuerons en open source les artefacts logiciels de définition, de mesure, d'intégration d'algorithmes, de trace qui seront évalués à la fois dans la diversité des mécanismes pris en compte et des applications considérées.

3.3 Démarche scientifique

Complémentarité et principes généraux

Nous présentons à présent les bases de notre démarche, qui repose sur une approche transversale des propriétés FATES : de leur analyse en tant qu'exigences à leur surveillance dans les applications intégrant des composants de ML. Cette approche nécessite une collaboration étroite entre les chercheurs en génie logiciel et en sciences des données, collaboration déjà existante et fructueuse (Benni et al., 2019; Brault et al., 2023). Pour atténuer les risques inhérents à un domaine aussi dynamique et aux multiples applications, nous adopterons une approche itérative, cohérente avec MLOps, en enrichissant progressivement les processus d'analyse avec de nouvelles propriétés et mécanismes. L'opérationnalisation de ces propriétés sera ainsi au cœur de notre démarche. En intégrant les propriétés FATES dès la phase d'analyse d'un problème et en les suivant tout au long du cycle de vie du logiciel, notre projet vise à améliorer les développements en IA, ainsi que les systèmes qui incorporent ces technologies. Notre approche se distingue en ce sens que nous ne cherchons pas à développer un nouveau framework, mais plutôt à mener une étude approfondie pour comprendre les interdépendances entre les propriétés, les outils et les objectifs. Nous proposerons des versions outillées des points abordés, tout en reconnaissant que nous ne pourrons pas couvrir tout l'espace des propriétés FATES, qui est très vaste.

Qualification/modélisation/formalisation des propriétés FATES

Nous visons à formaliser les propriétés FATES pour guider leur analyse et faciliter leur intégration dans le processus de développement en tenant compte des exigences qui portent sur un système donné. Nous établirons des relations logiques entre les exigences et les algorithmes/recommandations existants, afin de guider le choix des composants algorithmiques à intégrer dans le développement, qu'il s'agisse des codes d'entraînements, des chaînes d'intégration continue, déploiement ou des workflows tels que définis par Langchain. En nous basant d'une part sur la formalisation des propriétés et d'autres parts sur les algorithmes existants ou recommandations, nous visons à guider l'analyse des propriétés FATES et à faciliter leur intégration dans le processus de développement. Nous ne développerons pas d'algorithmes, ni ne proposerons de nouvelles recommandations. Nous nous plaçons en aval de ces recherches ; nous nous focaliserons sur leur exploitation systématique dans le développement des applications.

Intégration dans le Processus MLOps

Dans la mesure des artefacts logiciels dont nous disposons, nous analyserons les différentes étapes du processus de développement pour intégrer et vérifier la présence de composants utiles au suivi et au respect des propriétés FATES. Nous nous concentrerons sur les workflows d'entraînement des modèles de ML (e.g., pour sur-échantillonner les groupes sous-représentés), les compositions de pipelines dans LangChain (e.g., pour introduire une étape de débiaisage sur les données de renforcement) et des workflows de CI/CD (e.g., pour déployer un modèle d'explication parallèle au système ou un système de journalisation des événements). Nous développerons des justifications automatiques pour suivre et documenter les compromis et les vérifications effectués. Cette étape intégrera le développement de COTS (*Components*

Off-The-Shelf) réutilisables, indépendants et composables, permettant à d'autres travaux de minimiser leur effort d'intégration de ces bonnes propriétés. Nous mettrons en œuvre notre approche dans des applications MLOps, qui serviront également de démonstrateur. En résumé, notre approche vise à formaliser les propriétés FATES, à les intégrer de manière systématique dans le processus de développement, et à les appliquer dans des applications MLOps réelles pour garantir des systèmes plus responsables et fiables.

4 Conclusion

Les applications de l'IA ont un impact important dans la société. Dans son ambition de souveraineté et de compétitivité, la France lance de nombreux investissements et chantiers autour de l'IA. Nous sommes persuadés qu'elle se doit d'être exemplaire dans ces efforts en investissant également dans la maîtrise des propriétés FATES afin de minimiser les biais et les risques en matière de déploiement de du Machine Learning. En effet, de l'introduction de biais de recrutement chez Amazon lors de l'automatisation de la lecture des CVs, à la non-reconnaissance de personnes racisées par les algorithmes de détection de piétons de Tesla, la non-prise en compte des propriétés FATES lors du développement de produits basés sur de l'IA conduit inévitablement à des situations dramatiques. Par ses applications pratiques, ce projet a pour ambition de démontrer concrètement le ratio coût-bénéfice de la prise en compte systématique des propriétés FATES lors de la production de logiciels : coût de la documentation, possibilité d'automatisation, impact sur les processus de développement. Les applications visées couvrent différents domaines (Génération de code, Agent conversationnel pour langues sous-représentées, Santé mentale). Par ses contributions fondamentales, le projet proposera un cadre conceptuel et outillé permettant de supporter les ingénieurs logiciels lors de la mise en place de chaînes de production de nouveaux produits logiciels. L'ambition est ici de fournir des modèles réutilisables et open-source à la communauté, en se reposant sur l'expertise préexistante au sein du consortium, sur la publication et maintenance de logiciels "open-source" et de jeux de données "open-data". La compagnie Hugging Face, qui soutient le projet, indique un intérêt tout particulier sur le transfert technologique de ces résultats fondamentaux et appliqués à leur propre chaîne de production et d'entraînement de LLMs. Si une exploitation industrielle est hors du périmètre de ce projet (100% académique), l'intérêt d'un acteur majeur du domaine pour valider les résultats obtenus est un atout supplémentaire à la validité, la pérennité des résultats en dehors du projet lui-même, et surtout à sa visibilité qui bénéficiera du rôle central que joue Hugging Face au sein de la communauté. Conscients de l'ampleur de la tâche qui relève de la prise en compte des propriétés FATES et du besoin profond et impérieux de cette prise en compte, que ce soit dans l'industrie, mais aussi dans le monde académique, nous souhaitons que ce projet soit un démonstrateur et une illustration concrète que non seulement c'est possible, mais extrêmement bénéfique. Nous anticipons des impacts à court, moyen et long terme, chaque étape étant associée à des livrables spécifiques et présentant des risques et des opportunités croissants. À court terme, nous apportons (i) une compréhension améliorée des risques et des solutions FATES (par exemple l'amélioration des futurs générateurs de code comme Starcoder contre les biais), et (ii) une démocratisation du FATES-MLOps (accessibilité des codes et artefacts en open-source). À moyen terme, la réalisation concrète d'un chatbot "FATES" en Wolof constituera une vitrine pour une diffusion plus large des principes FATES. Enfin à long terme, via la formalisation des propriétés FATES, leur alignement sur les

normes existantes, la prise en compte de leur évolution et leur opérationnalisation pour guider leur intégration dans les processus MLOps, nous fournirons des outils précieux pour les *Data Scientists* / *ML Engineers* de demain.

Cet publication est supportée par le projet ANR-24-IAS2-0002. Les auteurs remercient les autres membres du projet pour leur participation.

Références

Amraoui, Y. E., M. Blay-Fornarino, P. Collet, F. Precioso, et J. Muller (2022). Evolvable spl management with partial knowledge : an application to anomaly detection in time series. In *Proceedings of the 26th ACM International Systems and Software Product Line Conference - Volume A*, SPLC '22, New York, NY, USA, pp. 222–233. Association for Computing Machinery.

Benni, B., M. Blay-Fornarino, S. Mosser, F. Precioso, et G. Jungbluth (2019). When DevOps Meets Meta-Learning : A Portfolio to Rule them all. In *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, Munich, Germany, pp. 605–612. IEEE.

Brault, Y., Y. El Amraoui, M. Blay-Fornarino, P. Collet, F. Jaillet, et F. Precioso (2023). Taming the Diversity of Computational Notebooks. In *SPLC 2023 - 27th ACM International Systems and Software Product Line Conference*, SPLC '23 : Proceedings of the 27th ACM International Systems and Software Product Line Conference - Volume A, Tokyo, Japan, pp. 27–33. ACM.

Breck, E., S. Cai, E. Nielsen, M. Salib, et D. Sculley (2017). The ml test score : A rubric for ml production readiness and technical debt reduction. In *Proceedings of IEEE Big Data*.

Brown, T. B., B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, et D. Amodei (2020). Language models are few-shot learners.

Chen, A., A. Chow, A. Davidson, A. DCunha, A. Ghodsi, S. A. Hong, A. Konwinski, C. Mewald, S. Murching, T. Nykodym, P. Ogilvie, M. Parkhe, A. Singh, F. Xie, M. Zaharia, R. Zang, J. Zheng, et C. Zumar (2020). Developments in mlflow : A system to accelerate the machine learning lifecycle. In *Proceedings of the Fourth International Workshop on Data Management for End-to-End Machine Learning*, DEEM '20, New York, NY, USA. Association for Computing Machinery.

Contractor, D., D. McDuff, J. K. Haines, J. Lee, C. Hines, B. Hecht, N. Vincent, et H. Li (2022). Behavioral use licensing for responsible ai. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, New York, NY, USA, pp. 778–788. Association for Computing Machinery.

Cugny, R., J. Aligon, M. Chevalier, G. Roman Jimenez, et O. Teste (2022). Autoxai : A framework to automatically select the most adapted xai solution. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, New York, NY, USA, pp. 315–324. Association for Computing Machinery.

Dorleon, G., I. Megdiche, N. Bricon-Souf, et O. Teste (2023). FAPFID : A Fairness-Aware Approach for Protected Features and Imbalanced Data. *Transactions on Large-Scale Data- and Knowledge-Centered Systems 13840 (TLDKS)*, 107–125. Transactions on Large-Scale Data- and Knowledge-Centered Systems (TLDKS).

Duffau, C., T. Polacsek, et M. Blay-Fornarino (2018). Support of justification elicitation : Two industrial reports. In *Advanced Information Systems Engineering : 30th International Conference, CAiSE 2018, Tallinn, Estonia, June 11-15, 2018, Proceedings*, Berlin, Heidelberg, pp. 71–86. Springer-Verlag.

Enoiu, E., D. Truscan, A. Sadovykh, et W. Mallouli (2023). Veridevops software methodology : Security verification and validation for devops practices. In *ARES '23 : Proceedings of the 18th International Conference on Availability, Reliability and Security*, pp. 1–9.

Feldman, M., S. A. Friedler, J. Moeller, C. Scheidegger, et S. Venkatasubramanian (2015). Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 259–268.

Ferrara, E. (2023). Should chatgpt be biased ? challenges and risks of bias in large language models. *First Monday 28*(11).

Garrido, J. S., S. Tolan, I. H. Torres, D. F. Llorca, V. Charisi, E. G. Gutierrez, H. Junklewitz, R. Hamon, D. F. Yela, et C. Panigutti (2023). AI Watch : Artificial Intelligence Standardisation Landscape Update. (KJ-NA-31-343-EN-N (online)).

Kim, G., P. Debois, J. Willis, et J. Humble (2016). *The DevOps Handbook : How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press.

Klaise, J., A. V. Looveren, C. Cox, G. Vacanti, et A. Coca (2020). Monitoring and explainability of models in production.

Liu, Y., X. Chen, Y. Gao, Z. Su, F. Zhang, D. Zan, J.-G. Lou, P.-Y. Chen, et T.-Y. Ho (2023). Uncovering and quantifying social biases in code generation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, et S. Levine (Eds.), *Advances in Neural Information Processing Systems*, Volume 36, pp. 2368–2380. Curran Associates, Inc.

Lopardo, G., F. Precioso, et D. Garreau (2023). A sea of words : An in-depth analysis of anchors for text data.

Lopardo, G., F. Precioso, et D. Garreau (2024). Attention meets post-hoc interpretability : A mathematical perspective.

Mill, E., W. Garn, N. Ryman-Tubb, et C. Turner (2024). The sage framework for explaining context in explainable artificial intelligence. *Applied Artificial Intelligence 38*, e2318670.

Mosser, S., C. Pulgar, M. Blay-Fornarino, D. Patel, A. Loh, et J.-M. Bruel (2023). Yes, Configuring is Good, But Have You Ever Tried Justifying ? In *CONFLANG Workshop (co-located with SPLASH)*.

Nigmatullin, I., A. Sadovykh, N. Messe, S. Ebersold, et J.-M. Bruel (2022). Rqcode – towards object-oriented requirements in the software security domain. In *IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pp. 2–6.

Polacsek, T., S. Sharma, C. Cuiller, et V. Tuloup (2018). The need of diagrams based on toulmin schema application : an aeronautical case study. *EURO Journal on Decision Pro-*

cesses 6, 1–26.

Sculley, D., G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, et D. Dennison (2015). Hidden technical debt in machine learning systems. *NIPS*, 2494–2502.

Suresh, H. et J. Guttag (2021). A framework for understanding sources of harm throughout the machine learning life cycle. In *Proceedings of the 1st ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, EAAMO '21, New York, NY, USA. Association for Computing Machinery.
en

Tabassi, E. (2023). Artificial Intelligence Risk Management Framework (AI RMF 1.0).

Testi, M., M. Ballabio, E. Frontoni, G. Iannello, S. Moccia, P. Soda, et G. Vessio (2022). MLOps : A Taxonomy and a Methodology. *IEEE Access 10*, 63606–63618.

Vasudevan, S. et K. Kenthapadi (2020). Lift : A scalable framework for measuring fairness in ml applications. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, New York, NY, USA, pp. 2773–2780. Association for Computing Machinery.

Wachter, S., B. Mittelstadt, et C. Russell (2021). Why fairness cannot be automated : Bridging the gap between eu non-discrimination law and ai. *Computer Law & Security Review 41*, 105567.

Wang, Z., Y. Liu, A. Arumugam Thiruselvi, et A. Hamou-Lhadj (2024). Xaiport : A service framework for the early adoption of xai in ai model development. In *Proceedings of the 2024 ACM/IEEE 44th International Conference on Software Engineering : New Ideas and Emerging Results*, ICSE-NIER'24, New York, NY, USA, pp. 67–71. Association for Computing Machinery.

Zaharia, M. A., A. Chen, A. Davidson, A. Ghodsi, S. A. Hong, A. Konwinski, S. Murching, T. Nykodym, P. Ogilvie, M. Parkhe, F. Xie, et C. Zumar (2018). Accelerating the machine learning lifecycle with mlflow. *IEEE Data Eng. Bull. 41*, 39–45.

Summary

The MLOps movement adopts the DevOps objective of reducing the gaps between development and operations teams by integrating data scientist teams and Machine Learning (ML) models. In the FATES-MLOPs ANR project, we wish to apply and adapt good software engineering practices to strengthen both the overall quality of the ML model construction processes and the quality of the software systems produced, particularly in terms of extra-functional properties that will become crucial issues: Fairness, Accountability, Transparency, Ethics, and Security (FATES). The key concerns will tackle the study, formalization, measurement, and management of these properties throughout the continuous MLOps process. Indeed, more than traditional Key Performance Indicators (KPIs), such as precision and recall, are required to evaluate models' robustness in practical applications. Our project aims to study the FATES properties and, by refining proven software engineering concepts and tools, propose a systematic and tailored approach for considering those properties, particularly from the lens of ML Scientists or ML Engineers, throughout the lifecycle of the software developed following an MLOps approach.

MIIC-SR: From Complex Data to Structural Causal Models

Nadir Sella*, Arefe Asadi**
Myriam Tami**, Louis Verny*

*TotalEnergies OneTech
7-9 boulevard Thomas Gobert 91120 Palaiseau – France
nadir.sella@totalenergies.com
** MICS Laboratory, Ecole CentraleSupelec
3 rue Joliot-Curie 91192 Gif-sur-Yvette - France

Abstract. Designing effective interventions in complex systems requires precisely estimating causal relationships between variables and their mathematical equations. Estimating this set of mathematical equations from observational data with non-linear relationships presents a significant challenge. While methods have been introduced to estimate regression formulas from data, there has been limited focus on integrating Causal Networks with advanced regression algorithms. In this context, we propose a comprehensive framework that estimates a full Structural Causal Model from complex datasets, i.e., in the presence of nonlinear equations and collinearity. Our framework combines the strength of MIIC, a well-established causal discovery algorithm, and Genetic Programming Symbolic Regression (SR). We evaluated the performance of our framework, including comparisons with other causal discovery algorithms, such as the PC algorithm, and other SCM estimators like Generalized Linear Models (GLM). Our results demonstrate that the association of MIIC and SR shows significantly better results in every studied benchmark.

1 Introduction

To gain a deep understanding of a system, such as those found in industries and healthcare, one can use a causally oriented method to identify and characterize the causal relationships between the components of this system. These relationships can be represented by Structural Causal Models (SCM), which allow for counterfactual reasoning and evaluation of the impact of interventions or policy changes.

SCM estimation from data poses significant challenges, particularly when confronted with non-linear relationships and collinearity among variables. This requires robust performance in two separate tasks: causal discovery, which involves recovering and representing causal relationships through graphs, and structural causal modeling, which mathematically characterizes these relationships. Accomplishing both tasks becomes increasingly demanding, underlining the need for advanced techniques and methodologies.

The estimation of SCMs fundamentally diverges from traditional regression methodologies. While regression primarily aims to predict a quantitative value — such as using a highly correlated variable in a blood test to predict a patient's health status — SCM estimation focuses on identifying actionable variables and quantifying their collective impact on downstream variables. Traditional regression approaches do not account for causal directions, which may lead to downstream variables being employed to predict upstream causal features. However, in real life and the context of structural causal modeling downstream variables will not influence upstream variables.

To address the limitations of regression in causal contexts while still taking advantage of the robustness of the advanced techniques, we propose the MIIC-SR framework. It integrates the Multivariate Information-based Inductive Causation (MIIC) causal discovery algorithm Verny et al. (2017); Cabeli et al. (2020); Sella et al. (2018) with Genetic Programming-enhanced Symbolic Regression (SR) de Franca et al. (2023); Makke and Chawla (2024); Koza (1994) to estimate both linear and non-linear equations, even in the presence of collinearity. Our approach aims to construct an effective, interpretable data-driven simulator for complex systems, wherein each feature is a mathematical function of its causal parents. Experiments on synthetic data will show the accurate performances of our framework to reconstruct complete SCM, with a progressive level of complexity in the data.

In the following sections, we introduce the MIIC and SR algorithms, explain how the two methods work together, and propose a pipeline for testing the ability of our framework to perform the SCM estimation without making any assumption on independent and identically distributed datasets.

2 Material and Methods

2.1 Structural Causal Modeling Framework

Structural Causal Model (SCM) is a mathematical framework used to represent and analyze causal relationships between variables in a system. SCMs consist of a set of equations that define how each variable is determined by its direct causes, which are often represented in a directed acyclic graph (DAG). This graphical representation allows researchers to visualize the causal pathways and dependencies among variables. One of the key advantages of SCMs is their ability to make predictions about the effects of interventions and to estimate causal effects from observational data, by applying techniques such as do-calculus. By clearly specifying the mechanisms underlying causal relationships, SCMs provide a robust foundation for causal inference, helping researchers to make informed decisions based on the understanding of how different variables interact within complex systems. In this article, we use SCM to define variables, mathematical functions, and to generate data for benchmarks.

2.2 Causal discovery algorithms

For the causal discovery task, we compared multiple state-of-the-art and recently developed algorithms: DirectLingam Shimizu et al. (2011), ICALiNGAM Shimizu et al. (2006), PC algorithm Spirtes and Glymour (1991), NotearsNonlinear Zheng et al. (2020), GOLEM Ng et al. (2020) and MIIC Verny et al. (2017); Cabeli et al. (2020). In particular, MIIC has demonstrated

its efficacy in handling various data distributions without imposing any prior assumptions Cabeli et al. (2020).

2.3 Multivariate Information-based Induced Causation (MIIC) algorithm

The MIIC algorithm infers graphical networks from observational data and represents the set of direct and possibly causal associations between variables Verny et al. (2017). MIIC algorithm is a constraint-based approach based on an information theoretical framework, that takes advantage of scoring algorithms (like Bayesian network structure learning) by ranking the potential contributors in order to use the minimal separating set in the conditioning set. MIIC is robust in sampling noise and does not need any hyperparameter tuning. MIIC can evaluate mutual information even in the presence of a mixture of continuous and discrete type variables, without any assumption of the underlying distribution of the data Cabeli et al. (2020) and is available as an open source project Sella et al. (2018).

2.4 Symbolic Regression

Symbolic Regression (SR) Kronberger et al. (2024) is a computational method aimed at discovering interpretable mathematical equations from data without assuming any predefined functional form de Franca et al. (2023). The most common and successful algorithms for SR are based on Genetic Programming (GP), a metaheuristic optimization technique inspired by biological evolution mechanisms that allow for efficient convergence, as proposed by Koza Koza et al. (1989); Koza (1990, 1994). For SR estimation through genetic programming, we used **PySR** Cranmer (2023), implemented in Julia Bezanson et al. (2012), which allows fast and efficient executions.

2.5 MIIC-SR method

We introduce a robust framework leveraging MIIC and SR for full structural causal modeling with very few if any, assumptions or particular knowledge of the dataset. The method is divided into two steps:

1. Causal discovery: from a given dataset, we reconstruct the causal network using MIIC.
2. Symbolic regression: for each child node X_i in the graph, the parent nodes $\mathbf{Pa}(X_i)$ are taken as predictors to apply SR. If undirected edges remain in the graph, we treat each node as the parent of the other one (since neither the MIIC nor the PC-algorithm guarantees a Directed Acyclic Graph (DAG) as output Kalisch and Bühlmann (2007); Spirtes et al. (2001)).

2.6 Comparative approach

To assess our method's capability in accurately estimating the SCM, we compare MIIC in conjunction with SR with the results obtained from the application of the PC algorithm combined with SR. For the estimation of SCMs, we also compared classical Generalized Linear Models (GLM) as a baseline approach, allowing the estimation of each interaction between the predictor variables. We decided to apply the entire pipeline for SCM estimation only using

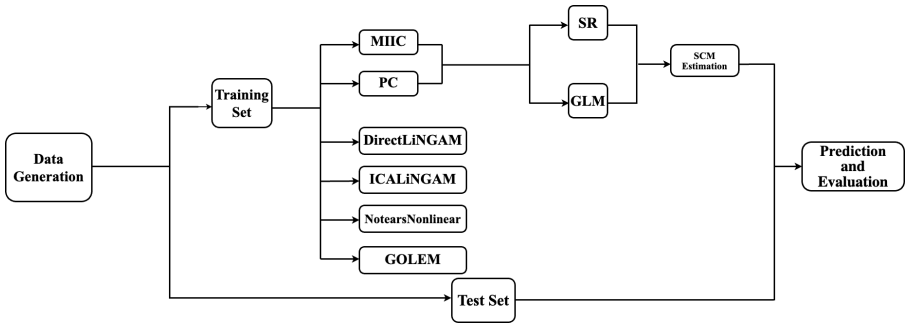


FIG. 1 – Evaluation pipeline of the different methods

MIIC and PC as causal discovery methods since they perform significantly better than other algorithms in our benchmarks (Supplementary Table S1). The benchmark pipeline resulted in four tested methods: MIIC-SR, PC-SR, MIIC-GLM, and PC-GLM. The experimental protocol is as follows: graphs and regression formulas are learned from a training data set. The estimated SCMs are then used to generate data using the predictors from a test set, generated using the same mathematical model. The complete pipeline is shown in Figure 1.

2.6.1 Synthetic data

We generated training and test datasets, each containing the same amount of samples (e.g. $n = 10,000$), based on a generative model represented by a graph G consisting of a set of nodes X, edges E, and corresponding SCM. The distribution laws for variables with no parents are specified in the tables.

1. **2 colliders graph**: a small graph with 5 nodes and 4 edges (Figure 2a), containing 2 colliders. In this graph, B has 2 parent nodes (Z and K) and one child node (A). In this benchmark the SCMs are estimated in two ways: once using causal discovery algorithms (MIIC and PC) as a first step, and once without using them, allowing SR to look for all possible nodes to estimate the regression equation. This network is used to show that regression algorithms alone may use wrong causal nodes to estimate the value of a given target node (Table 2b).

2. **Lingauss Graph**: a graph forming a collider (Figure 3a). The SCM, reported in Table 3d is inspired by Microsoft's CSuite benchmark Geffner et al. (2022).

3. **Symprod Simpson Graph**: a more complex graph also inspired by CSuite (Figure 4a). The SCMs, reported in Table 4d are mostly nonlinear.

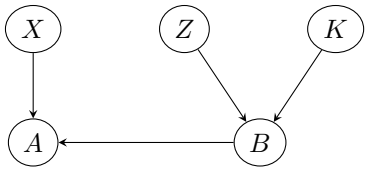
2.6.2 Performance evaluation

To assess the accuracy of SCM estimations, we first compared the identified SCM to the equation used to generate the data. Although comparing two SCMs can be challenging, classical predictive metrics, such as the Mean Squared Error (MSE), offer a straightforward comparison approach. The MSE is defined as $MSE = \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n}$ where $\hat{y} = f(\mathbf{Pa}(y))$, with f the

function estimated by SR, represents the predicted y values using $\mathbf{Pa}(y)$ as parent variables from the test set, as identified by causal discovery algorithms. The predicted values $\hat{y}$ are then compared against the actual y value in the test set. The corresponding result is a vector of MSEs, one for each variable that is described in the SCM.

3 Results

To demonstrate the importance of performing the causal discovery before any regression estimation, we built a simple network (Figure 2a), running our pipeline on it. The results show that MIIC and PC find the correct parent nodes for A and B, allowing SR to retrieve the exact SCM (Table 2b). In contrast, in the absence of causal discovery, the SR algorithm alone cannot find the correct predictors for B. In this scenario, SR considers X and A as parents for B while Z and K are instead the correct parent nodes. The SCM built using X and A to predict B is incorrect, as it suggests that we can affect B through A and X, which violates the SCM.



(a)

Equation	Method	SCM
$B = 2.5Z + 2.2K$	MIIC-SR	$B = 2.5Z + 2.2K$
	PC-SR	$B = 2.5Z + 2.2K$
	SR	$B = 0.322A - 0.713X$
$A = 2.21X + 3.11B$	MIIC-SR	$A = 2.21X + 3.11B$
	PC-SR	$A = 2.21X + 3.11B$
	SR	$A = 2.21X + 3.11B$
$X \sim \mathcal{N}(0, 1)$ $Z \sim \mathcal{N}(0, 1)$ $K \sim \mathcal{N}(0, 1)$		

(b)

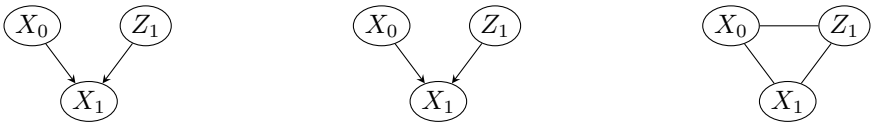
FIG. 2 – (a) Small network (b) Comparison between true SCM and estimated SCMs.

The second model is a three-node network (Figure 3a) corresponding to a collider. Our results show that the MIIC algorithm successfully identifies the underlying graphical causal model (Figure 3b), whereas PC fails to accurately identify the causal structure and reconstructs a complete graph with undirected edges (Figure 3c).

As mentioned in section 2.5 we chose to consider nodes connected by undirected edges as the parents of each other. For this reason, both GLM and SR correctly identify the structural causal model for X_1 (Table 3d). This choice helps methods that are not able to orient edges.

The third example is the "Symprod Simpson" network (Figure 4a). This network represents a more difficult causal discovery task, with a nonlinear SCM. The networks reconstructed by MIIC and PC algorithms are shown in Figure 4b and 4c respectively. While the MIIC algorithm correctly retrieves all directed edges, therefore solving the causal discovery task, the PC algorithm fails to capture the causal association $X_0 \to X_2$ and $X_1 \to X_2$, instead proposing the wrong edge $X_3 \to X_1$.

For node X_1 the PC mistakenly finds X_3 as a parent (Figure 4c). While the causal equation obtained by GLM includes every identified parent, SR correctly excludes X_3 from its estimation, finding the exact regression formula (Table 4d). On the other hand, since the MIIC algorithm identifies the correct nodes X_0 and Z_1 as parents of X_1, GLM and SR include them in their



(a) **(b)** **(c)**

True SCM	Method	Estimated SCM
$X_1 = \frac{1}{2}X_0 + \frac{\sqrt{3}}{2}Z_1$	MIIC-GLM	$X_1 = \frac{1}{2}X_0 + \frac{\sqrt{3}}{2}Z_1$
	MIIC-SR	$X_1 = \frac{1}{2}X_0 + \frac{\sqrt{3}}{2}Z_1$
	PC-GLM	$X_1 = \frac{1}{2}X_0 + \frac{\sqrt{3}}{2}Z_1$
	PC-SR	$X_1 = \frac{1}{2}X_0 + \frac{\sqrt{3}}{2}Z_1$
$X_0 \sim \mathcal{N}(0,1)$ $Z_1 \sim \mathcal{N}(0,1)$		

(d)

FIG. 3 – (a) Lingauss Graph (b) the graphs identified by MIIC (c) the graphs identified by PC (d) Comparison between true SCM and estimated SCMs. This graph is inspired by Microsoft's CSuite for causality benchmarking Geffner et al. (2022).

causal equations. Since the true SCM is not linear in X_1, GLM only finds the best linear approximation, whereas SR retrieves the exact formula.

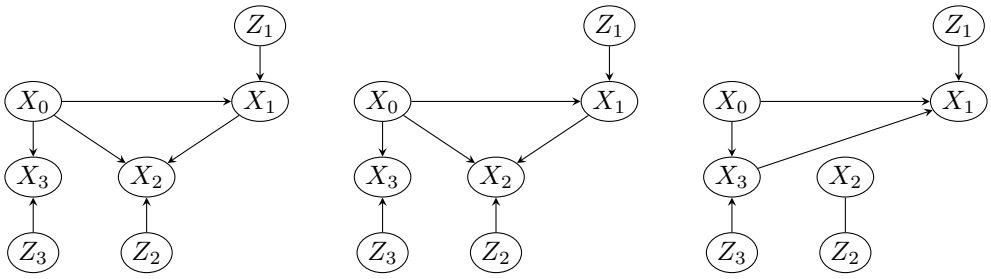
For node X_2, the PC algorithm identifies only one association with node Z_2, proposing an undirected edge (Figure 4c). Hence, PC-GLM and PC-SR methods include only Z_2 in the causal equations (Table 4d). In contrast, the MIIC algorithm accurately identifies the parent nodes of X_2. Consequently, both the MIIC-SR and MIIC-GLM models incorporate X_0, X_1, and Z_2 in the regression formula, with both finding the correct mathematical equation.

The parents of X_3 are identified properly by both MIIC and PC algorithms. As a result, the equations provided by GLM and SR correctly use the exact parent nodes. Moreover, the SR method finds again the exact equation (Table 4d), while, for nonlinearity reasons, GLM can only get an approximation in the estimated formula.

In the final step, we evaluated the Mean Squared Error (MSE) of the SCMs for each node of the Symprod Simpson Graph (Figure 4e), to better compare SR and GLM results in the ability to predict the data. The highest MSE occurs at node X_2 when using the PC-SR method, due to the inability of the PC algorithm to accurately identify the predictors of X_2. In contrast, it can be seen that the MSE of the equations estimated by MIIC-SR is equal to 0 since MIIC identifies the predictors of all nodes precisely for SR.

To assess the stability of our method and understand the impact of sample size on regression performances, we conducted the Symprod benchmark using multiple sample sizes (50, 150, 250, 500, 2500 and 5000). The mean squared error (MSE) was used to evaluate the regression error. Figure S1 shows that while generalized linear models (GLMs) achieve a performance plateau for X_1 and X_3, indicating the inability to capture nonlinear associations accurately, symbolic regression (SR) achieves an MSE of 0 when provided with sufficient sample sizes. MIIC-SR exhibits performances comparable to PC-SR for X_1 and X_3, except in cases with the smallest sample sizes, and consistently outperforms PC-SR across all sample sizes for X_2. In particular, for node X_2, even with 5,000 samples, PC-SR fails to achieve low MSE scores. Causal discovery performances, measured by precision, recall, and F1 score, are shown in

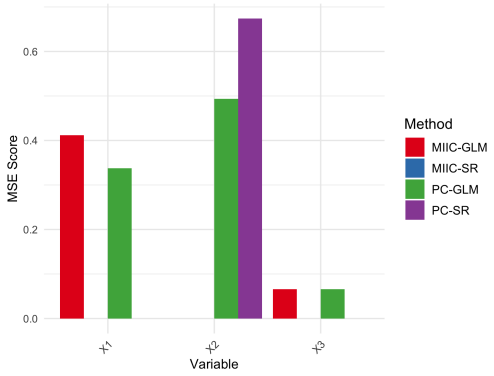
Figure S2. These results highlight the reliability of the MIIC algorithm in reconstructing causal networks, as demonstrated by both precision and recall measures.



(a) **(b)** **(c)**

True SCM	Method	Estimated SCM
$X_1 = 2\tanh(2X_0) + \frac{1}{\sqrt{10}}Z_1$	MIIC-GLM	$X_1 = 1.458X_0 + 0.323Z_1 + 0.001X_0Z_1$
	MIIC-SR	$X_1 = 2\tanh(2X_0) + \frac{1}{\sqrt{10}}Z_1$
	PC-GLM	$X_1 = 1.146X_0 + 0.453X_3 + 0.325Z_1 + 0.004X_0Z_1 - 0.005Z_1X_3 - 0.006X_0Z_1X_3 + 0.001$
	PC-SR	$X_1 = 2\tanh(2X_0) + \frac{1}{\sqrt{10}}Z_1$
$X_2 = \frac{1}{2}X_0X_1 + \frac{1}{\sqrt{2}}Z_2$	MIIC-GLM	$X_2 = \frac{1}{2}X_0X_1 + \frac{1}{\sqrt{2}}Z_2$
	MIIC-SR	$X_2 = \frac{1}{2}X_0X_1 + \frac{1}{\sqrt{2}}Z_2$
	PC-GLM	$X_2 = 0.709Z_2 + 0.729$
	PC-SR	$X_2 = Z_2 + 0.726$
$X_3 = \tanh\left(\frac{3}{2}X_0\right) + \sqrt{\frac{3}{10}}Z_3$	MIIC-GLM	$X_3 = 0.689X_0 + \sqrt{\frac{3}{10}}Z_3$
	MIIC-SR	$X_3 = \tanh\left(\frac{3}{2}X_0\right) + \sqrt{\frac{3}{10}}Z_3$
	PC-GLM	$X3 = 0.689X0 + \sqrt{\frac{3}{10}}Z3$
	PC-SR	$X3 = \tanh\left(\frac{3}{2}X_0\right) + \sqrt{\frac{3}{10}}Z_3$
$Z_1 \sim t_3, Z_2 \sim Laplace(1)$, $Z_3 \sim \mathcal{N}(0,1), X_0 \sim \mathcal{N}(0,1)$		

(d)



(e)

FIG. 4 – Comparison of the Symprod Simpson Graph (a) and the graph identified by MIIC (b) and PC (c) algorithms. Table (d) presents the structural equations corresponding to the true SCM and equations identified by each method. (e) MSE evaluation using the estimated equations. This graph is inspired by Microsoft's CSuite for causality benchmarking Geffner et al. (2022)

4 Discussion

In this paper, we demonstrated that integrating a state-of-the-art causal discovery algorithm with symbolic regression enables accurate estimation of SCMs without making prior assumptions about either data distribution or model structure. Our results emphasize the importance of causal discovery in accurately identifying parent nodes, which is essential for understanding causal mechanisms and specifying predictive models correctly. The mistaken identification of the causal structure by the PC algorithm in the Symprod Sympson graph (Figure 4c), can result in erroneous regression formulas and misguided interpretations of variable relationships. In contrast, the consistent accuracy of the MIIC algorithm highlights the importance of reliable causal discovery methods for accurately capturing relationships, leading to more robust models and reliable insights. Moreover, our flexible pipeline allows for the incorporation of domain expert knowledge, when already estimated feature equations or physical-chemical laws are known.

Structural causal equations provide a fully explainable framework that suggests the causal direction and magnitude of the effect of a variable on another. SCM estimation can be used in real use cases as a simulator or numerical twin to test interventions and evaluate how modifying some rules affects the target variables and the entire system. SCMs can also be used to generate data for better training machine learning algorithms, such as in imbalanced data or in the presence of missing data.

The complexity of the SCM estimation primarily lies within the SR phase. The MIIC algorithm has already been applied in relatively large real scenarios (100 variables, 10,000 samples) and runs efficiently in a few minutes. The complexity of SR estimation is associated with the connectivity of the resulting causal network.

From a theoretical perspective, some challenges remain. One of them arises from the class of networks found by constraint-based causal discovery methods that do not guarantee the output of a completely oriented graph (DAG), but rather a combination of directed and undirected edges representing a class of Markov equivalences Kalisch and Bühlmann (2007); Gillispie and Perlman (2013). An alternative approach, such as that proposed by Maathuis et al. (2009), could be considered for future research to address this challenge. This method uses the PC algorithm to estimate the equivalence class of DAGs consistent with the observed data. It then applies intervention calculus to each DAG in the equivalence class to compute the possible causal effects, leveraging the structure of the graph to efficiently estimate the intervention outcomes. This approach offers a promising direction for handling the ambiguity introduced by Markov equivalence classes in causal discovery.

To go further, conducting additional executions considering more complex systems and real-world use cases would be beneficial. This would help to analyze the performance of our methodology in a precise way. To estimate the ability of MIIC-SR to capture the relations between real distributions of an observational dataset where the true graph is unknown, we would need to use some distance metric applied to the multivariate distribution of the generated data against the original ones, such as the Wasserstein distance.

However, it is crucial to assess the accuracy of the estimated SCMs without relying solely on the evaluation of regression errors or distribution distances. In this context, a relevant approach would involve computing distances between estimated and true equations using tree representation-based equations Akutsu et al. (2021), incorporating customized penalties for

non-correct trees. This would provide a better understanding of the accuracy and correctness of the estimated SCMs.

References

Akutsu, T., T. Mori, N. Nakamura, S. Kozawa, Y. Ueno, and T. N. Sato (2021). Tree edit distance with variables. measuring the similarity between mathematical formulas.

Bezanson, J., S. Karpinski, V. B. Shah, and A. Edelman (2012). Julia: A fast dynamic language for technical computing.

Cabeli, V., L. Verny, N. Sella, G. Uguzzoni, M. Verny, and H. Isambert (2020). Learning clinical networks from medical records based on information estimates in mixed-type data. *PLoS computational biology 16*(5), e1007866.

Cranmer, M. (2023). Interpretable machine learning for science with pysr and symbolicregression.jl.

de Franca, F. O., M. Virgolin, M. Kommenda, M. S. Majumder, M. Cranmer, G. Espada, L. Ingelse, A. Fonseca, M. Landajuela, B. Petersen, R. Glatt, N. Mundhenk, C. S. Lee, J. D. Hochhalter, D. L. Randall, P. Kamienny, H. Zhang, G. Dick, A. Simon, B. Burlacu, J. Kasak, M. Machado, C. Wilstrup, and W. G. L. Cava (2023). Interpretable symbolic regression for data science: Analysis of the 2022 competition. *arXiv preprint 2304.01117*.

Geffner, T., J. Antoran, A. Foster, W. Gong, C. Ma, E. Kiciman, A. Sharma, A. Lamb, M. Kukla, N. Pawlowski, M. Allamanis, and C. Zhang (2022). Deep end-to-end causal inference.

Gillispie, S. B. and M. D. Perlman (2013). Enumerating markov equivalence classes of acyclic digraph models.

Kalisch, M. and P. Bühlmann (2007). Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *Journal of Machine Learning Research 8*, 613–636.

Koza, J. R. (1990). *Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems*, Volume 34. Stanford University, Department of Computer Science Stanford, CA.

Koza, J. R. (1994). Genetic programming as a means for programming computers by natural selection. *Statistics and computing 4*, 87–112.

Koza, J. R. et al. (1989). Hierarchical genetic algorithms operating on populations of computer programs. In *IJCAI*, Volume 89, pp. 768–774.

Kronberger, G., B. Burlacu, M. Kommenda, S. M. Winkler, and M. Affenzeller (2024). *Symbolic Regression*. CRC Press.

Maathuis, M. H., M. Kalisch, and P. Bühlmann (2009). Estimating high-dimensional intervention effects from observational data. *The Annals of Statistics 37*(6A), 3133–3164.

Makke, N. and S. Chawla (2024). Interpretable scientific discovery with symbolic regression: a review. *Artificial Intelligence Review 57*(1), 2.

Ng, I., A. Ghassami, and K. Zhang (2020). On the role of sparsity and dag constraints for learning linear dags. *Advances in Neural Information Processing Systems 33*, 17943–17954.

Sella, N., L. Verny, G. Uguzzoni, S. Affeldt, and H. Isambert (2018). Miic online: a web server to reconstruct causal or non-causal networks from non-perturbative data. *Bioinformatics 34*(13), 2311–2313.

Shimizu, S., P. O. Hoyer, A. Hyvärinen, and A. Kerminen (2006). A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research 7*, 2003–2030.

Shimizu, S., T. Inazumi, Y. Sogawa, A. Hyvarinen, Y. Kawahara, T. Washio, P. O. Hoyer, K. Bollen, and P. Hoyer (2011). Directlingam: A direct method for learning a linear non-gaussian structural equation model. *Journal of Machine Learning Research-JMLR 12*(Apr), 1225–1248.

Spirtes, P. and C. Glymour (1991). An algorithm for fast recovery of sparse causal graphs. *Social science computer review 9*(1), 62–72.

Spirtes, P., C. Glymour, and R. Scheines (2001). *Causation, prediction, and search*. MIT press.

Verny, L., N. Sella, S. Affeldt, P. P. Singh, and H. Isambert (2017). Learning causal networks with latent variables from multivariate information in genomic data. *PLoS computational biology 13*(10), e1005662.

Zheng, X., C. Dan, B. Aragam, P. Ravikumar, and E. P. Xing (2020). Learning sparse nonparametric dags.

5 Supplementary

5.1 Causal Discovery Algorithms Performance

To assess the performance of the causal discovery algorithms, we evaluate their accuracy using key metrics: Skeleton Precision (or Positive Predictive Value) $Prec = TP/(TP + FP)$, Recall (or Sensitivity) $Rec = TP/(TP + FN)$, and $F\text{-}score = (2\times Prec \times Rec)/(Prec + Rec)$. True Positives (TP) refer to correctly identified causal relationships, False Positives (FP) are incorrectly identified relationships, and False Negatives (FN) are missed relationships.

TAB. S1 – Causal Discovery: Performances on lingauss graph

Method	Precision	Recall	Fscore
DirectLiNGAM	0	0	-
ICALiNGAM	0.333	0.5	0.4
PC	0.333	1	0.5
NotearsNonlinear	0.333	0.5	0.4
GOLEM	0.333	0.5	0.4
MIIC	1	1	1

5.2 PySR Parameters

In the Symbolic Regression experiments, we employed the PySR package with the parameters reported in Table S3.

TAB. S2 – Causal Discovery: Performances on Symprod Sympson graph

Method	Precision	Recall	Fscore
DirectLiNGAM	0.3333	0.4286	0.375
ICALiNGAM	0.2222	0.2857	0.25
PC	0.7143	0.7143	0.7143
NotearsNonlinear	0.1	0.1429	0.1176
GOLEM	0.1	0.1429	0.1176
MIIC	1	1	1

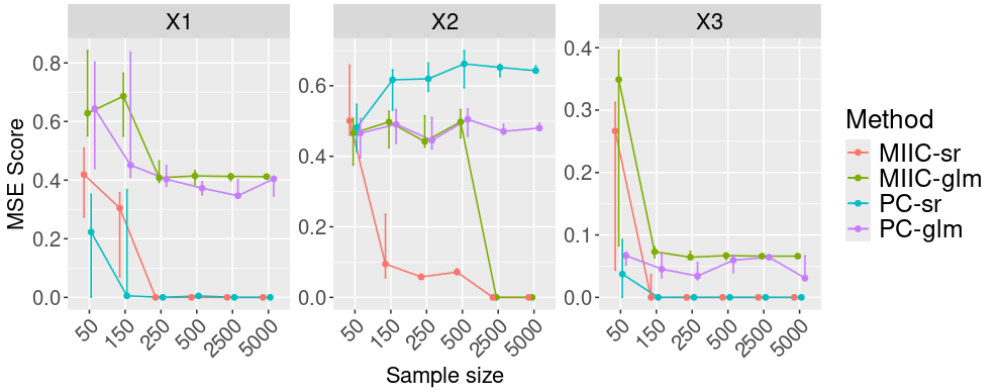


FIG. S1 – Evaluation of MSE in predicting X_1, X_2, and X_3 for the different methods and multiple sample sizes (50, 150, 250, 500, 2500, 5000) in the Symprod Simpson Graph. Median values with first and third quartiles as error bars are reported.

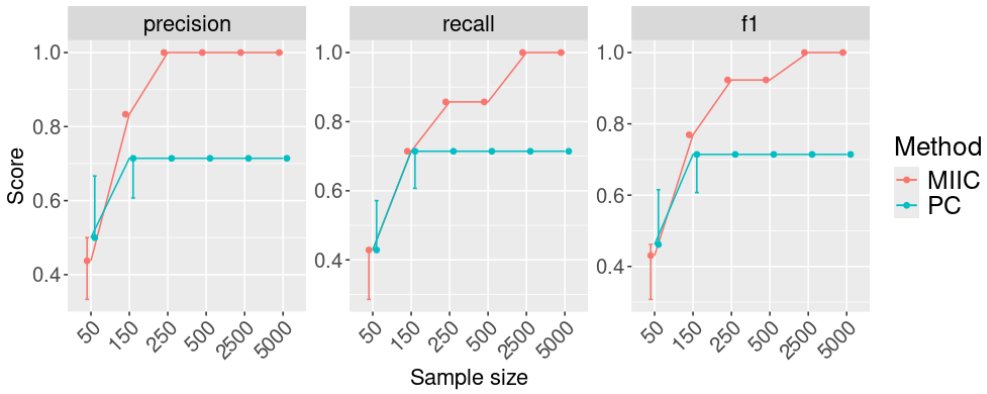


FIG. S2 – Evaluation of Precision, Recall, and F-score for MIIC and PC algorithms in multiple sample sizes (50, 150, 250, 500, 2500, 5000) in the Symprod Simpson Graph. Median values with first and third quartiles as error bars are reported.

Parameter	Description
`random_state`	42
`niterations`	100
`populations`	10
`population_size`	100
`maxsize`	10
`binary_operators`	`["+", "-", "*", "/", "^"]`
`unary_operators`	`cos`, `sin`, `exp`, `inv(x) = 1/x`, `log`, `sqrt`, `square`, `cube`, `abs`, `tan`, `tanh`, `sinh`, `cosh`.
`constraints`	`{"^": (-1, 1)}`
`elementwise_loss`	`loss(prediction, target) = (prediction - target)^2`
`early_stop_condition`	`stop_if(loss, complexity) = loss< 1e-10 && complexity < 10`

TAB. S3 – PySR Parameters

X-Train: eXplanations for Training
Exploitation des explications dans l'entraînement multimodal des Transformers

Meghna P Ayyar*, Jenny Benois-Pineau*, Akka Zemmari*

*LaBRI, CNRS, Univ. Bordeaux, UMR 5800, F-33400, Talence, France
{meghna-parameswaran.ayyar, jenny.benois-pineau, akka.zemmari}@u-bordeaux.fr,

Résumé. L'intelligence artificielle explicable (XAI) vise à améliorer la transparence et la fiabilité des décisions prises par les réseaux neuronaux. Bien que les transformers soient devenus l'état de l'art dans diverses tâches, notamment la vidéo, le traitement automatique du langage naturel et l'analyse de signaux, l'impact des méthodes XAI sur l'apprentissage des modèles reste peu exploré. Cet article présente une méthode appelée *X-Train : eXplanations for Training*, qui exploite une méthode XAI pour identifier les parties saillantes des entrées et oriente l'entraînement du modèle à se concentrer uniquement sur ces éléments pertinents. Nous appliquons X-Train dans un cadre d'analyse multimodale, utilisant un transformer pour traiter conjointement des vidéos et des signaux. Les résultats expérimentaux démontrent que X-Train surpasse systématiquement les approches d'entraînement classiques ainsi que la méthode basée sur XAI IFI, dans des contextes d'apprentissage monomodal et multimodal.

1 Introduction

Afin d'amélioration des performances des modèles d'apprentissage automatique, les chercheurs se sont efforcés de repérer et d'exploiter les caractéristiques saillantes des données d'entrée, un objectif qui a conduit à des avancées significatives depuis les premiers réseaux neuronaux convolutifs (CNN) jusqu'à aujourd'hui. Par exemple, des cartes d'attention basées sur des principes psychovisuels ont été utilisées pour créer des fenêtres d'intérêt dans des tâches de détection d'objets (de San Roman et al., 2017). Plus récemment, les architectures Transformer se sont imposées comme l'état de l'art pour diverses applications en vision et en traitement du langage, grâce à leur mécanisme d'auto-attention capable de capturer des dépendances à longu

Au-delà de la simple identification des caractéristiques, les méthodes d'intelligence artificielle explicable (XAI) offrent un potentiel considérable pour améliorer l'entraînement des modèles en mettant l'accent de manière sélective sur les régions d'entrée pertinentes, contribuant ainsi à une meilleure généralisation. À l'image de l'apprentissage humain, où l'attention se concentre sur des éléments essentiels tout en ignorant les détails superflus, nous proposons *X-Train* [1] qui utilise des explications pour guider l'entraînement des modèles. Plus précisé-

1. Code X-Train

ment, nous étendons la méthode d'explication des caractéristiques (FEM) (Fuad et al., 2020) [2] afin de tirer parti des composants saillants des images vidéo et des signaux de capteurs, en optimisant ces informations pour l'entraînement de transformers appliqués à la vidéo et aux signaux, et, en fin de compte, aux architectures multimodales. Cette approche est particulièrement pertinente pour les tâches complexes du monde réel, telles que la détection des risques dans des situations quotidiennes (Mallick et al., 2024).

Les principales contributions de notre travail sont les suivantes :

1. Nous proposons X-train, une méthode qui utilise uniquement les parties saillantes des images d'entrée pour l'entraînement d'un transformer vidéo.
2. Nous étendons X-Train en intégrant des poids d'importance dérivés des mécanismes d'attention d'un transformer appliqué aux signaux, afin de mettre en évidence les segments pertinents du signal d'entrée durant l'entraînement.
3. En outre, nous intégrons X-Train dans une architecture multimodale utilisant une fusion tardive, permettant d'entraîner conjointement les branches vidéo et signaux pour la reconnaissance des événements à risque.

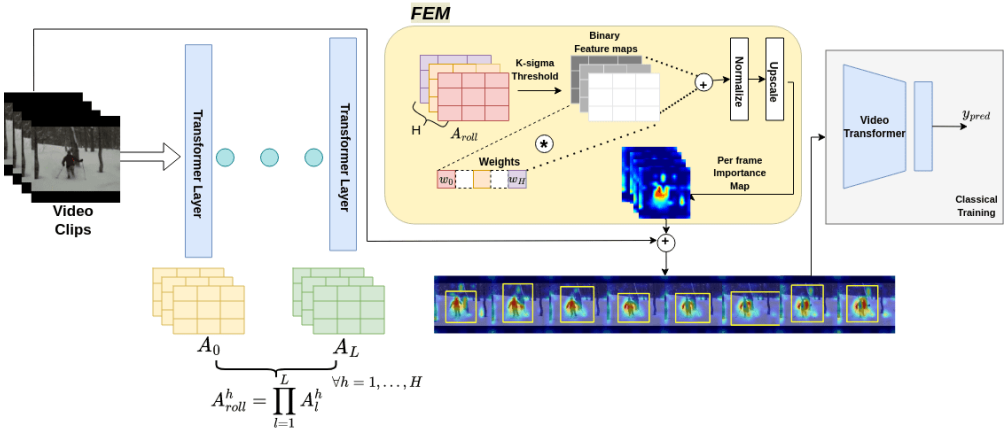


FIG. 1 – *X-Train pour la vidéo avec les cartes d'explication FEM*

2 Méthodologie

Entraînement avec la modalité vidéo :

Le rollout de l'attention (Abnar et Zuidema) combine les matrices d'attention à travers les couches du Transformer jusqu'à la couche l , comme le montre l'équation (1), où I représente la matrice identité pour les connexions résiduelles, h est l'indice de la tête d'attention, et H

2. https://github.com/labribkb/fem

désigne le nombre total de têtes d'attention.

$$A'^{l} = I + \sum_{h=1}^{H} A_h^l, \qquad A_{roll} = \prod_{l=1}^{L} A'^{l} \tag{1}$$

La méthode de rollout calcule une moyenne des attentions spécifiques à chaque tête, analogue à la moyenne des canaux dans les cartes de caractéristiques des CNN. Nous proposons d'utiliser la méthode FEM pour cette agrégation. Au lieu d'une moyenne sur h , les cartes d'attention de chaque couche sont multipliées récursivement pour obtenir $A_{h,\text{roll}}$, comme indiqué dans l'équation (2).

$$A_h'^{l} = I + A_h^l, \ \forall h = 1, ...H, \qquad A_{h,roll} = \prod_{l=1}^{L} A_h'^{l} \tag{2}$$

Chaque carte est seuillée à l'aide de la règle K-sigma pour ne conserver que les régions d'attention «forte» (voir l'équation 3).

$$b_h(A_{h,roll}) = \begin{cases} 1 & \text{if } a_{i,h} \geq \mu_h + K * \sigma_h \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

où μ_h et σ_h désignent respectivement la moyenne et l'écartype de la carte d'attention. Cela produit des cartes binaires b_h et, en utilisant la moyenne des cartes d'attention μ_h comme poids, nous calculons la combinaison linéaire $E^{'}$. Cette carte est normalisée puis redimensionnée aux dimensions de l'entrée pour obtenir la carte d'explication finale E de même dimension que l'entrée.

Les cartes d'explication E peuvent alors être utilisées pour masquer les régions sans importance de l'entrée et ne conserver que les régions à forte saillance. Le recadrage à taille fixe, tel qu'utilisé par (Saha et Roy, 2023), peut être efficace pour les images statiques avec des objets uniques et petits, mais échoue souvent dans les contextes vidéo où les objets peuvent se déplacer, risquant ainsi de ne pas prendre en compte le contexte de l'objet. Pour y remédier, nous proposons une méthode de recadrage dynamique basée sur la plus grande zone saillante de la carte d'explication E pour chaque image d'entrée. Ici, nous effectuons le seuillage encore avec la règle K-sigma sur la carte d'explication E. La plus grande composante connexe de la carte E binarisée est retenue et sa boîte englobante (BB) est calculée. Ces BB varient par image, en fonction du mouvement de l'objet, pour recadrer de manière adaptative les images d'entrée, tandis que les pixels situés en dehors de la BB sont mis à zéro. La figure 1 illustre le schéma d'apprentissage de notre transformer vidéo ainsi que le recadrage des BB pour le clip vidéo. On peut voir que la taille des boîtes varie pour chaque image et suit le mouvement de l'objet dans le clip.

Entraînement avec la modalité du signal : Comme pour la stratégie d'apprentissage de la modalité vidéo, les cartes d'explication E du transformer sont extraites à l'aide de la combinaison de Rollout et de FEM. Contrairement aux vidéos, où le recadrage spatial est intuitif, les données de capteurs exigent que tous les capteurs aient la même fréquence d'échantillonnage et l'identification de l'événement (classe) à la même position temporelle. Par conséquent, le système de recadrage des vidéos pour les données de signaux provenant des capteurs n'est pas adaptée. Au lieu de cela, les scores d'importance peuvent être utilisés comme des poids pour

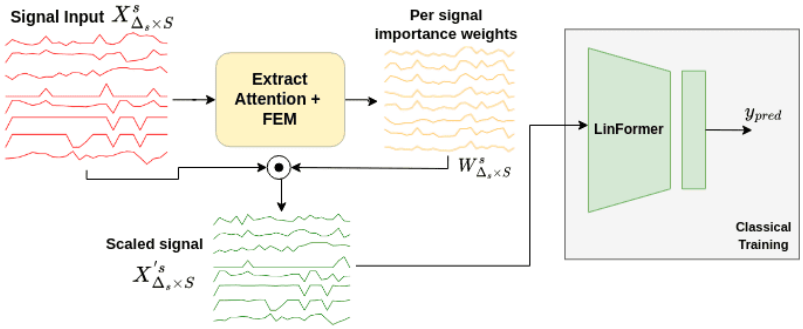


FIG. 2 – *X-Train pour la modalité du signal*

indiquer la pertinence de chaque signal. Les scores sont ensuite normalisés en fonction de la dimension du signal S pour obtenir les poids W^S et l'entrée signal X^S est pondérée par une multiplication élément par élément avec les poids, comme l'illustre la Figure 2.

Architecture Multimodale : Le transformer multimodal est une architecture à deux branches, i) la branche vidéo utilisant le Pooling Transformer (Mallick et al., 2022) ii) la branche signal utilisant le modèle LinFormer (Wang et al., 2020). Ces deux branches sont formées conjointement à l'aide de la stratégie de fusion tardive avec la fonction de perte combinée donnée par l'équation 4 où $\mathcal{L}^v$ et $\mathcal{L}^s$ sont les fonctions de perte (entropie croisée) pour les branches vidéo et capteur, et $\lambda \geq 0$ est l'hyperparamètre pour pondérer les fonctions de perte.

$$\mathcal{L} = \lambda \mathcal{L}^v + (1 - \lambda)\mathcal{L}^s \tag{4}$$

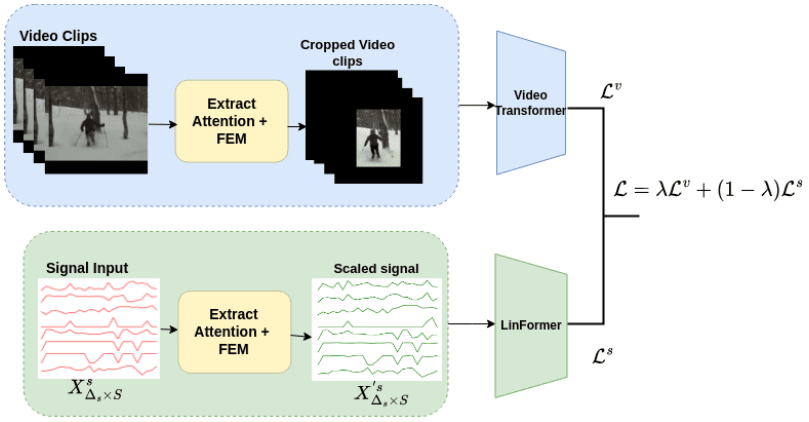


FIG. 3 – *X-Train pour l'entraînement multimodale avec la perte combinée des deux modalités*

3 Expérimentation et résultats

Afin de valider notre approche, nous l'avons appliquée au corpus BIRDS (Bio-Immersive Risk Detection System) (Mallick et al., 2022). Ce corpus représente une sorte de journal multimodal de la vie quotidienne, composé de vidéos égocentriques et de données de signaux provenant de 16 capteurs portables, pour détecter les situations à risque chez les personnes fragiles vivant seules à domicile. Il y a cinq classes de risque et une classe sans risque.

Les résultats de l'entraînement sur la modalité vidéo uniquement du corpus sont indiqués dans le tableau 1. l'entraînement avec X-Train a la meilleure performance globale et montre une amélioration de $\sim 9\%$ par rapport à l'entraînement de base. Il montre également une amélioration de $\sim 8\%$ par rapport à la méthode IFI de l'état de l'art, laquelle utilise le gradient de l'attention pour fournir une supervision supplémentaire pendant l'entraînement.

Model	Top-1 Acc
TimeSFormer (Bertasius et al.)	74.11%
Swin Transformer (Swin-T) (Liu et al., 2022)	73.39%
Pooling Transformer (Pool-T) (Mallick et al., 2022)	75.19%
Video Swin-T-In (IFI) (Mallick et al., 2024)	76.37%
Pool-T + X-Train (Ours)	**84.45**%

TAB. 1 – *Top-1 test accuracy on the BIRDS dataset for videos*

Aucun entraînement préalable n'est effectué pour le transformer de signaux en raison de l'indisponibilité d'ensembles de données similaires. Le tableau 2 montre la comparaison de l'entraînement sur les données de signal. L'utilisation de scores d'importance pour pondérer les parties pertinentes du signal à l'aide de X-Train montre une amélioration globale par rapport aux autres modèles. Notre méthode présente une amélioration de $\sim 13.5\%$ par rapport à l'entraînement de base et $\sim 8.83\%$ par rapport à l'entraînement IFI de l'état de l'art (Mallick et al., 2024).

Model	Top-1 Acc
LinFormer (Wang et al., 2020)	35.55%
LinFormer-In (IFI) (Mallick et al., 2024)	40.26%
LinFormer-X-Train (Ours)	**49.09**%

TAB. 2 – *Top-1 test accuracy on the BIRDS dataset for signal modality*

L'entraînement multimodal basé sur la fusion tardive sans aucune supervision basée sur l'attention atteint une précision de $76,51\%$ et $78,26\%$ avec la méthode IFI. Avec notre méthode X-Train pour les branches vidéo et signal, nous obtenons une précision finale de $85,12\%$, soit une augmentation de $\sim$ **8,6%** et $\sim$ **7%** par rapport à ces deux méthodes. Cela montre que le transformer multimodal peut utiliser les informations supplémentaires des signaux pour améliorer les performances, et notre méthode X-Train peut encore améliorer les performances pendant l'entraînement comme l'illustre la Figure 4. Ici nous comparons le cadre X-Train (XT) avec les modèles de la littérature analysés dans (Mallick et al., 2022).

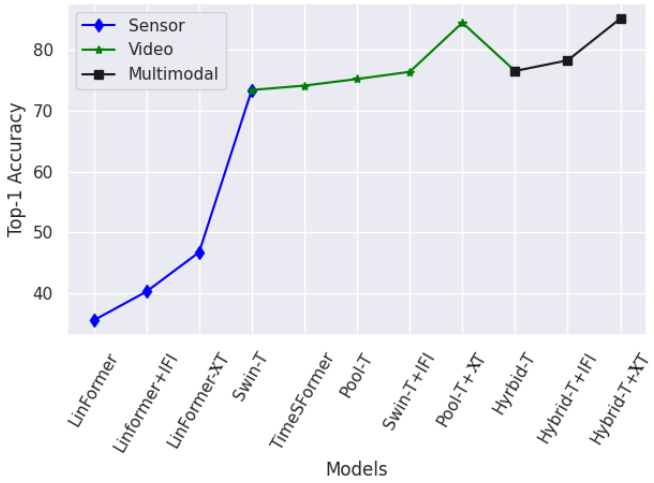


FIG. 4 – *Top-1 de la précision des modèles de transformers sur l'ensemble de données BIRDS, XT : X-Train*

4 Conclusion et travaux futurs

Dans ce travail, nous avons proposé le cadre d'entraînement des Transformers, appelé X-Train, qui intègre des méthodes issues de l'intelligence artificielle explicable (XAI). Notre approche démontre que les méthodes d'explication peuvent être utiles pour guider l'entraînement de modèles basés sur l'auto-attention. X-Train optimise les entrées en exploitant les valeurs d'attention générées par le modèle pendant l'apprentissage. Nous avons validé cette méthode sur un ensemble de données multimodales comprenant des vidéos égocentriques et des signaux de capteurs. Une comparaison avec IFI, une autre technique utilisant les poids d'auto-attention pour l'entraînement, montre que X-Train surpasse systématiquement l'apprentissage classique ainsi qu'IFI pour les deux modalités. En entraînement multimodal à deux flux, X-Train offre une amélioration de $\sim 8,6\%$ par rapport à l'approche de base et de $\sim 7\%$ par rapport à la méthode IFI sur des ensembles de données complexes.

Un axe de recherche futur intéressant serait d'analyser l'évolution des régions saillantes des entrées pour mieux comprendre le processus de formation des décisions du modèle au fil de l'entraînement.

Références

Abnar, S. et W. H. Zuidema. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of ACL, 2020*.

Bertasius, G., H. Wang, et L. Torresani. Is space-time attention all you need for video understanding ? In *ICML, 2021*.

de San Roman, P. P., J. Benois-Pineau, J.-P. Domenger, F. Paclet, D. Cataert, et A. De Rugy (2017). Saliency driven object recognition in egocentric videos with deep cnn : toward application in assistance to neuroprostheses. *Computer Vision and Image Understanding*.

Fuad, K. A. A., P. Martin, R. Giot, R. Bourqui, J. Benois-Pineau, et A. Zemmari (2020). Features understanding in 3d cnns for actions recognition in video. In *Tenth International Conference on Image Processing Theory, Tools and Applications, IPTA 2020, Paris, France, November 9-12, 2020*, pp. 1–6. IEEE.

Liu, Z., J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, et H. Hu (2022). Video swin transformer. In *Proceedings of the IEEE/CVF CVPR*, pp. 3202–3211.

Mallick, R., J. Benois-Pineau, et A. Zemmari (2024). Ifi : Interpreting for improving : A multimodal transformer with an interpretability technique for recognition of risk events. In *MultiMedia Modeling*, pp. 117–131. Springer Nature Switzerland.

Mallick, R., J. Benois-Pineau, A. Zemmari, T. Yebda, M. Pech, H. Amieva, et L. Middleton (2022). Pooling transformer for detection of risk events in in-the-wild video ego data. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pp. 2778–2784. IEEE.

Saha, G. et K. Roy (2023). Saliency guided experience packing for replay in continual learning. In *IEEE/CVF WACV*.

Wang, S., B. Z. Li, M. Khabsa, H. Fang, et H. Ma (2020). Linformer : Self-attention with linear complexity. *arXiv preprint arXiv :2006.04768*.

Summary

Our X-Train framework uses self-attention-based explanations to guide transformer models to focus on salient input regions and ignore the less relevant parts during training. X-Train is able to improve model performance by retaining only the important input regions while training.

De l'explicabilité à l'explication des IA : perspective située incarnée par et pour les métiers

Ranya Bennani*, Myriam Frejus**
Marc-Eric Bobillier-Chaumon***

*EDF lab, 7 Bd Gaspard Monge, 91120 Palaiseau
Ranyabennani2@gmail.com
**EDF lab, 7 Bd Gaspard Monge, 91120 Palaiseau
Myriam.frejus@edf.fr
***CNAM, 41 Rue Gay-Lussac, 75005 Paris
marc-eric.bobillier-chaumon@lecnam.net

Résumé. L'intelligence artificielle (IA) a réalisé des progrès importants, notamment grâce aux algorithmes d'apprentissage automatique, mais l'augmentation de la complexité des modèles pose un problème de transparence. L'intelligence artificielle explicable (XAI) a été proposée pour rendre l'IA plus transparente et favoriser son appropriation. Bien que de nombreuses études aient identifié des défis et des pistes de recherche, celles-ci restent très technocentrées. Cet article s'inscrit dans le cadre d'une thèse sur l'explicabilité et l'appropriation de l'intelligence artificielle (IA) dans les environnements professionnels et socio-domestiques, en se concentrant d'abord sur le milieu professionnel. Il vise à clarifier la notion d'explicabilité et à apporter une perspective complémentaire et anthropocentrée aux perspectives existantes, en s'appuyant sur une étude exploratoire en contexte professionnel.

1 Introduction

Cet article s'inscrit dans le cadre d'une thèse en ergonomie portant sur l'explicabilité et l'appropriation de l'intelligence artificielle (IA) dans les environnements professionnels et socio-domestiques, en se concentrant d'abord sur le milieu professionnel.

Pourquoi l'explicabilité ?

L'intelligence artificielle (IA) a réalisé des progrès importants, notamment grâce aux algorithmes d'apprentissage automatique (ML), mais l'augmentation de la complexité des modèles, souvent opaques (« boîtes noires »), pose un problème de transparence. Cela est particulièrement critique dans des domaines sensibles comme la santé et la sécurité, où la précision ne suffit pas. L'intelligence artificielle explicable (XAI) a été proposée pour rendre l'IA plus transparente et favoriser son appropriation. L'XAI cherche à développer des techniques permettant aux utilisateurs de mieux comprendre et gérer les systèmes d'IA. Bien que de nombreuses études aient identifié des défis et des pistes de recherche, celles-ci restent très technocentrées (Saeed et Omlin, 2021).

Le courant de recherche « XAI » se concentre sur l'explicabilité du point de vue des sciences des données (approche technocentrée). Cependant, il existe peu de recherches sur cette notion dans le domaine des sciences humaines et sociales (approche anthropocentrée) (cf. Vuarin et Steyer (2023)). L'objectif de cet article est de clarifier la notion d'explicabilité et d'apporter une perspective complémentaire et anthropocentrée aux perspectives existantes.

Nous présenterons les différentes définitions et visions de l'explicabilité. Nous illustrerons également nos propos avec les résultats d'une étude exploratoire réalisée par Frejus et Turpin (2023).

2 Cadre théorique

2.1 L'explicabilité : un processus souvent abordé selon un angle technocentré

L'explicabilité est une notion sans terminologie fixe, définie différemment par les chercheurs en sciences des données. Les définitions varient, allant d'une approche très technocentrée, focalisée sur les systèmes et modèles mathématiques, à des approches moins technocentrées, prenant en compte l'individu dans ce processus. Cependant, nous verrons que cette prise en compte de l'individu est loin d'être complète.

Nous notons également, que dans cet article, nous faisons la distinction entre l'explicabilité et l'interprétabilité. En effet, nous soutenons l'idée que l'interprétabilité fait partie du processus de l'explicabilité, comme soutenu par Markus et al. (2020) qui considèrent « l'interprétabilité comme une propriété liée à une explication et l'explicabilité comme un concept plus large se référant à toutes les actions visant à expliquer ». Pour Markus et al. (2020), l'explicabilité englobe toutes les actions nécessaires pour clarifier le fonctionnement des systèmes d'IA, bien que la nature exacte de ces actions reste floue. Ces chercheurs soulignent une méconnaissance concernant les explications requises pour rendre les systèmes pleinement compréhensibles. De plus, ils ne considèrent pas l'individu comme acteur ou cible de l'explicabilité.

D'autres définitions complètent cette première approche en s'éloignant de la vision technocentrée. Beaudouin et al. (2020) définissent l'explicabilité comme « la capacité, l'inclination ou l'aptitude à rendre clair ou compréhensible, ou à expliquer le sens d'un algorithme ». Gornet et Maxwell (2023) ajoutent un aspect relatif aux décisions humaines, définissant l'explicabilité comme « la capacité d'expliquer à la fois les processus techniques d'un système d'IA et les décisions humaines qui s'y rapportent ». Cependant, ces définitions sont encore trop centrées sur les systèmes et les algorithmes.

Selon Ali et al. (2023), une définition récente et largement acceptée de l'IA explicable est celle de Barredo Arrieta et al. (2020), qui mettent l'accent sur le récepteur de l'explication : pour un public donné, une IA explicable produit des détails ou des raisons rendant son fonctionnement clair ou facile à comprendre. Ali et al. (2023) ajoutent que l'explicabilité exprime ce qui se passe dans le modèle en fournissant une explication compréhensible par les humains sur la décision du modèle. Jouis (2023) rejoint cette définition et propose une distinction entre deux dimensions de l'explicabilité : (1) la transparence ; il s'agit du degré de compréhension permis par le modèle pour des utili-

sateurs spécifiques et (2) la pertinence, qui fait référence au niveau d'adéquation entre les explications fournies par le modèle et le besoin de compréhension dans un contexte donné (Saeed et Omlin, 2021). Nous précisons que le contexte que nous étudions dans cet article est le contexte d'activité en milieu professionnel. Parler de l'activité au travail, consiste « à parler des acteurs au travail et à les considérer comme des individus ayant des pratiques qui leur sont propres mais également comme des individus engagés dans un collectif organisé avec des règles plus ou moins explicites, des manières de faire individuelles et partagées. » (Paganelli, 2016). Pour revenir à la définition de l'explicabilité, celle-ci soulève d'abord des questions sur qui sont les destinataires de l'explicabilité ? S'agit-il des chargés de conception ? ou plutôt des utilisateurs finaux du système ? Elle soulève aussi la question du degré de compréhension nécessaire en fonction des variétés individuelles et sur les destinataires de l'explicabilité (concepteurs ou utilisateurs finaux). Elle évoque également l'importance du contexte dans le processus d'explicabilité et le potentiel surcroît de travail pour l'individu, qui doit comprendre le fonctionnement de l'algorithme avant d'évaluer ses résultats. Paradoxalement, l'objectif des dispositifs à base d'IA est de réduire la charge de travail, comme dans le domaine médical où l'IA vise à libérer du temps pour les médecins (Vuarin et Steyer, 2023). Ces deux notions, semblent ainsi évoquer la nécessité de se pencher sur les utilisateurs et sur le statut de l'explicabilité au sein d'un processus de conception, mais également sur le contexte externe à celle-ci, celui de l'activité organisationnelle, individuelle et collective, avec toutes les règles et contraintes propres à cette activité. Plusieurs études viennent confirmer ces perspectives. Plusieurs études ont exploré la question de la place de l'individu et de la diversité des profils dans le processus d'explication des décisions issues de systèmes automatisés. Kirsch (2017) a démontré le besoin de correspondre l'explication aux besoins du public visé. Barredo Arrieta et al. (2020) ont montré que l'intelligibilité des explications dépend du public cible (par exemple, on n'explique pas une décision de la même manière à un enfant qu'à un adulte). Ras et al. (2018) ont distingué experts et non- experts, soulignant que le niveau d'expertise influence la compréhension des explications (Wang et Ying, 2021). Ainsi plusieurs auteurs (Beaudouin et al., 2020) soutiennent la nécessité d'adapter l'explication aux variétés intraindividuelles (e.g., expertise) du public cible. Gornet et Maxwell (2023) ont confirmé ces propos dans leur étude sur une IA dédiée à la détection des ceintures et téléphones au volant. Ils ont montré que les contrôleurs avaient besoin d'explications simples et visuels (image), alors que les concepteurs, régulateurs et organismes de certification exprimaient des besoins d'explications plus détaillées et que les recommandations liées à l'explicabilité et la transparence dans l'AI act semblaient les concerner davantage. Anichini et Geffroy (2021) ont étudié ces notions chez les radiologues : « Quand il s'agit de la phase de détection, l'opacité qui caractérise le choix automatisé s'avère problématique en raison du sentiment d'expropriation de la décision qu'elle produit, ce qui témoigne d'une mise à mal de l'engagement de la responsabilité médicale dont chaque interprétation est porteuse. ». Ici, l'inexplicabilité entraine un sentiment d'expropriation de la décision donnée par la machine, alors que l'opérateur engage toujours sa responsabilité. Or, l'objectif est de créer des dispositifs appropriables et de confiance. Ainsi, l'explication va venir renforcer la confiance des utilisateurs dans le système et permettre l'appropriation de ce dernier (Biran et Cotton, 2017). Bouzekri et Rivière

(2022) complètent ces observations par les lacunes relevées de leur étude ; il semble qu'il y a un manque de méthodes et outils pour garantir que les systèmes autonomes destinés aux utilisateurs non-experts soient à la fois compréhensibles et fiables. Amershi et al. (2019) ont également constaté que les informations fournies par ces systèmes sont souvent inadaptées au contexte spécifique, négligeant des aspects comme l'environnement, les interactions et les variabilités intra-individuelles. Ces manques font référence à la notion d'explication (informations, compréhension).

2.2 De l'explicabilité à l'explication

« Une explication n'est pas une construction mathématique, une explication est bonne si les gens la trouvent utile dans un contexte spécifique ». Telle est définie l'explication selon Kirsch (2017). D'abord, nous allons préciser qui sont ces personnes (destinataires de l'explication), puis nous aborderons les questions suivantes : qu'est-ce qu'une bonne explication ? Quels sont les critères qui cadrent une explication et qui la rendent utile ?

L'explication est définie comme une « justification ou motif invoqué pour éclairer l'origine ou la raison d'être d'une situation complexe ou contestable. » (Larousse, 2024). Cette définition suppose que l'explication est nécessaire uniquement lorsque les situations sont complexes ou contestables. Cependant, des études viennent contredire cette définition. Ebel (1981) définit trois conditions pour qu'un discours soit considéré comme une explication :

— Le fait à expliquer doit être hors de contestation
— L'interlocuteur doit se poser une question
— L'expliquant doit être neutre et compétent en la matière

Adadi et Berrada (2018) complètent ces conditions en précisant qu'il existe quatre principales raisons de demander une explication : justifier, contrôler, améliorer et découvrir.

Miller (2019) considère que les explications actuelles sont statiques. Selon cet auteur, l'explication idéale est celle où l'expliquant et l'expliqué interagissent. L'auteur précise que les explications sont sociales et doivent être interactives à travers la communication avec les utilisateurs. Il rejoint les propos de Frejus (1999) qui précise que l'explication dépend de son interlocuteur. « En effet, une explication est considérée comme telle du moment où l'interlocuteur considère la séquence comme une explication (Grize, 1998) ».

Ainsi, considérer l'explication comme telle dépend du sujet et du contexte dans lequel il se trouve. Si nous transposons ces propos sur la conception d'un dispositif, cela signifie que l'explication qui émane du dispositif devrait dépendre de son utilisateur (variabilités individuelles) et du contexte dans lequel il procède à son utilisation (professionnel ou sociodomestique).

C'est pourquoi, l'explication doit s'incarner dans l'activité et doit toujours être située et contingente au contexte. Elle contribuera ainsi au système organisationnel et aux ressources de l'individu.

2.3 Une approche située de l'activité

Pour comprendre les conditions d'adoption des systèmes d'IA explicables dans les activités professionnelles et socio-domestiques, nous nous appuyons sur les approches de l'acceptabilité et plus précisément l'acceptabilité située (Dubois et Bobillier-Chaumon, 2009).

Il existe trois approches principales de l'acceptabilité des technologies.

1. Acceptabilité pratique : Se concentre sur la compatibilité entre l'utilisateur, ses tâches et l'ergonomie de la technologie (utilité, utilisabilité, accessibilité) (Brangier et al., 2009). Elle inclut désormais des aspects liés à l'expérience utilisateur, mais néglige les dimensions sociales de l'acceptabilité.
2. Acceptabilité sociale : Évalue principalement la perception et la satisfaction des utilisateurs. Bien que répandue, elle traite souvent les technologies de manière statique et ne prend pas en compte l'évolution des besoins des utilisateurs (Bobillier Chaumon, 2016).
3. Acceptabilité située : Dubois et Bobillier-Chaumon (2009) proposent une analyse contextuelle des bénéfices et contraintes de la technologie dans son usage réel. Elle privilégie l'étude des pratiques facilitées ou entravées par la technologie, en mobilisant les théories de l'appropriation technologique (Rabardel, 2005) et de l'activité (Engeström, 2001). L'acceptation résulte ici d'un processus d'appropriation où les utilisateurs personnalisent les outils pour les intégrer entièrement à leurs activités quotidiennes.

3 Méthodologie

La méthodologie présentée ci-dessous concerne une étude exploratoire menée au sein d'EDF RetD par Turpin et Frejus (2023).

3.1 Analyse des besoins : demande initiale

Dans le cadre d'un projet de RetD, les chercheurs (datascience) ont étudié la faisabilité d'un modèle d'apprentissage d'un phénomène rare qui se produit en centrale nucléaire afin d'anticiper sa survenue. Les centrales qui sont situées au bord d'un fleuve utilisent l'eau du fleuve pour refroidir leurs circuits, essentiels à la production d'électricité. Mais peut alors se produire un phénomène de colmatage (ou phénomène Sar), qui, bien que rare, pose des enjeux critiques de production et de fonctionnement. Ce phénomène se caractérise par l'accumulation de débris (feuilles, morceaux de bois, etc.) susceptibles de bloquer les canalisations d'eau refroidissant les réacteurs. La survenue d'un colmatage est très rapide et contraint à l'arrêt du réacteur. Il est donc important de pouvoir l'anticiper.

L'ingénieur « source froide » est responsable de la supervision et de l'anticipation de ce phénomène. Il est donc considéré par les datascientists comme le futur utilisateur du système d'aide. Conscients des problèmes d'explicabilité qui pourraient se poser et souhaitant traiter des questions d'interaction avec l'outil (en cours de conception/recueil

des données nécessaires à la création du modèle d'apprentissage), ils ont souhaité l'intervention d'ergonomes (Frejus, 1999). Ces professionnels ont pour objectif d'apporter un regard complémentaire en permettant une meilleure prise en compte de l'humain dans les projets de conception. Cet éclairage passe par un travail de compréhension du contexte de l'activité dans lequel s'inscrit ce projet. Pour se faire, un premier travail de compréhension du travail en centrale relatif à la source froide a été réalisé grâce à des entretiens semi-directifs et des observations ouvertes. Ces données nous permettent de répondre à certains questionnements abordés dans le premier chapitre.

4 Discussion

4.1 Les interlocuteurs métiers ne sont pas les seules cibles des explications

Par expert métier, nous désignons les travailleurs experts de leur domaine, ayant plusieurs années d'expérience dans le même poste. L'étude nous permet de définir deux éléments importants relatifs aux cibles exactes des explication du SIA : d'une part, celui de la nature des destinataires de l'explication. D'autre part, de la différence d'explication à apporter à chacun.

Le premier résultat a permis l'identification de plusieurs métiers concernés par la gestion de la source froide, et non seulement l'ingénieur source froide. Les acteurs identifiés ont des profils différents. Premièrement, ils font partie de services différents (service ingénierie, équipes de conduite, métiers de la maintenance). Ensuite, ils regroupent des niveaux d'expertise différents (ingénieur expert, ingénieur novice, ingénieurs non experts de la source froide, etc.). Enfin, les acteurs impliqués dans les missions liées à la source froide consacrent des pourcentages variables de leur temps de travail en fonction de leur rôle et de leur expertise. Certains y allouent tous leurs temps, tandis que d'autres y consacrent beaucoup moins. Ceci signifie que leurs besoins en termes de conception et d'explication vont être différents. Le besoin des développeurs a orienté cette recherche vers l'ingénieur source froide, qui était considéré comme le destinateur principale de l'explication de ce dispositif. Cependant, le fait d'adopter un regard systémique nous a permis d'identifier un ensemble d'acteurs concernés et impliqués dans la gestion du risque. De plus, les acteurs identifiés ne sont pas experts du métier ni impliqués dans le suivi quotidien de la gestion de la source froide. Cette observation rejoint les propos de Barredo Arrieta et al. (2020), Kirsch (2017), Ras et al. (2018) et de Wang et Ying (2021) qui soutiennent l'idée que l'explication doit être adaptée aux besoins spécifiques de chaque public et que la compréhension dépendra de l'expertise des sujets.

Nous pouvons également supposer que le degré d'explication souhaitée serait différent entre les experts métiers et les non experts. En effet, comme souligné par Beaudouin et al. (2020) et Gornet et Maxwell (2023), un expert nécessitera une explication détaillée alors qu'un non expert nécessitera une explication plus concise. Un des besoins fonctionnels relevé dans cette étude, nous laisse supposer un lien entre cette affirmation et notre cas d'étude. En effet, l'ingénieur SF expert a souligné l'importance primordiale de pouvoir actualiser le système et également le besoin d'avoir accès aux logiciels qui

prélèvent les différents indicateurs utilisés pour la surveillance malgré la possibilité de génération d'une anticipation de la situation. Cet accès lui permettra de les confronter aux informations et résultats fournies par le logiciel.

Ainsi la confirmation du diagnostic fourni par le SIA, s'appuierait davantage sur des éléments externes (indicateurs), qui sont propres au phénomène étudié, plutôt que sur une explication des raisons pour lesquelles le système produit son résultat.

4.2 Quelle est la place du système organisationnel dans la définition de l'explication ?

Dans notre cas d'étude, le système organisationnel joue un rôle important dans la gestion du risque Sar. D'une part, les acteurs concernés, s'étendent sur un large champ, allant de l'ingénieur SF spécialiste du phénomène aux agents de maintenance. En effet, la gestion du risque semble concernée une pluralité d'acteurs aux profils différents, s'inscrivant chacun dans un premier sous-système organisationnel (une équipe ou un service), puis dans un second sous système plus large (Centrale nucléaire). Le tout s'inscrivant dans un système organisationnel davantage plus large (e.g., niveau national). Nous rejoignons l'étude d'Amershi et al. (2019), qui postule l'importance de l'adaptation de l'explication au contexte, aux interactions et aux variations intra-individuelles.

5 Conclusion

Cette recherche met en lumière des aspects importants à prendre en compte dans un processus de conception et également lors de l'étude de l'explicabilité et l'explication.

D'une part, elle contribue à élargir notre compréhension des destinataires de l'explication dans ce domaine. Contrairement à une approche initialement focalisée sur les experts métiers, nos résultats révèlent une pluralité d'acteurs impliqués. Ces derniers font partie de services divers et présentent des niveaux d'expertise hétérogènes, allant d'experts à des intervenants non spécialistes. Ces différences influencent non seulement leurs besoins en termes d'explication, mais aussi le degré de détail et de transparence attendu. En ce sens, notre travail confirme les perspectives insistant sur l'importance d'adapter les explications à des profils variés. D'autre part, notre étude montre que l'explication est profondément ancrée dans un système organisationnel complexe, ne se limitant pas uniquement aux experts métiers. En effet, dans le contexte des centrales, ce type de phénomène est géré par divers acteurs allant des métiers de maintenance aux directeurs. Également, d'autres instances peuvent être impliqués au niveau national. Cela met en évidence l'importance d'adopter une approche systémique et située pour concevoir des explications adaptées aux différents contextes organisationnels et aux multiples acteurs impliqués.

Il est important de noter que cette recherche est exploratoire, elle nécessite d'être approfondie par des études complémentaires pour mieux cerner les besoins spécifiques des non-experts dans la gestion du risque et évaluer comment des outils explicables peuvent être conçus pour répondre à des audiences hétérogènes. Une analyse plus fine des interactions entre systèmes organisationnels, contextes environnementaux et technologies explicables pourrait également enrichir le champ d'application de cette recherche. Enfin,

l'intégration d'approches interdisciplinaires, mêlant psychologie cognitive, psychologie du travail et datascience, pourrait offrir des solutions innovantes pour concevoir des dispositifs explicables et de confiance.

Nous sommes actuellement en cours de développement d'une méthodologie de simulation projective permettant d'avancer dans la conception et à cibler et répondre aux besoins d'explication diverses et variées.

Références

Adadi, A. et M. Berrada (2018). Peeking inside the black-box : A survey on explainable artificial intelligence (xai). *IEEE Access 6*, 52138–52160.

Ali, H. et al. (2023). Explainable ai in healthcare. *Nature Medicine 29*, 123–145.

Barredo Arrieta, A., N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, et F. Herrera (2020). Explainable artificial intelligence (xai) : Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion 58*, 82–115.

Beaudouin, V., I. Bloch, D. Bounie, S. Clémençon, F. D'Alché-Buc, J. Eagan, W. Maxwell, P. Mozharovskyi, et J. Parekh (2020). Flexible and context-specific ai explainability : A multidisciplinary approach. *SSRN Electronic Journal*.

Bobillier Chaumon, M. (2016). L'acceptation située des technologies dans et par l'activité : premiers étayages pour une clinique de l'usage. *Psychologie du Travail et des Organisations 22*(1), 4–21.

Brangier, ., S. Hammes-Adelé, et J. Bastien (2009). Analyse critique des approches de l'acceptation des technologies : de l'utilisabilité à la symbiose humain-technologie-organisation. *European Review Of Applied Psychology 60*(2), 129–146.

Dubois, M. et M.-E. Bobillier-Chaumon (2009). L'acceptabilité des technologies : bilans et nouvelles perspectives. *Le Travail Humain 72*(4), 305–310.

Ebel, M. (1981). *L'explication : acte de langage et légitimité du discours*. Lyon : Presses Universitaires de Lyon.

Engeström, Y. (2001). Expansive learning at work : toward an activity theoretical reconceptualisation. *Journal of Education and Work 14*(1), 133–156.

Frejus, M. (1999). *Analyser l'activité d'explication pour concevoir en terme d'aide : application à la formation et à la négociation commerciale*. Ph. D. thesis, Université Paris 5.

Gornet, J. et W. Maxwell (2023). Understanding ai decisions : A human-centered approach. *AI Ethics 3*, 45–67.

Grize, J. (1998). Logique naturelle, activité de schématisation et concept de représentation. *Cahiers de Praxématique 31*, 115–125.

Jouis, G. (2023). *Explicabilité des modèles profonds et méthodologie pour son évaluation : application aux données textuelles de Pôle emploi*. Thèse de doctorat, Université de Nantes.

Kirsch, A. (2017). Explain to whom ? putting the user in the center of explainable ai. In *Proceedings of the First International Workshop on Comprehensibility and Explanation in AI and ML*. CEUR Workshop Proceedings.

Larousse (2024). *Dictionnaire de la langue française*. Paris : Éditions Larousse.

Markus, A. F., J. A. Kors, et P. R. Rijnbeek (2020). The role of explainability in creating trustworthy artificial intelligence for health care : A comprehensive survey of the terminology, design choices, and evaluation strategies. *Journal Of Biomedical Informatics 113*, 103655.

Miller, T. (2019). Explanation in artificial intelligence : Insights from the social sciences. *Artificial Intelligence 267*, 1–38.

Paganelli, C. (2016). Réflexions sur la pertinence de la notion de contexte dans les études relatives aux activités informationnelles. *Études de communication 46*.

Rabardel, P. (2005). Instrument, activité et développement du pouvoir d'agir. In *Entre connaissance et organisation : l'activité collective*, pp. 251–265.

Ras, G., M. van Gerven, et P. Haselager (2018). Explanation methods in deep learning : Users, values, concerns and challenges. *Neural Computing and Applications 30*, 4233–4252.

Saeed, W. et C. Omlin (2021). Explainable ai (xai) : A systematic meta-survey of current challenges and future opportunities. *arXiv preprint arXiv :2111.06420*.

Turpin, M. et M. Frejus (2023). Comment faciliter l'appropriation de systèmes opaques : Etude sur l'explanable artificial intelligence et préconisations ergonomiques sur l'explicabilité d'un système de machine learning. Rapport de stage, EDF.

Vuarin, L. et V. Steyer (2023). Le principe d'explicabilité de l'ia et son application dans les organisations. *Réseaux 240*(4), 179–210.

Wang, D. et Q. Ying (2021). User-centered explainable ai : Challenges and future directions. *Frontiers in Computer Science 3*, 738800.

Summary

Artificial Intelligence (AI) has made significant progress, particularly through machine learning algorithms. However, the increasing complexity of models raises concerns about transparency. Explainable AI (XAI) has been proposed to make AI more transparent and encourage its adoption. While numerous studies have identified challenges and research avenues, these remain largely technocentric. This article is part of a thesis on the explainability and adoption of AI in professional and socio-domestic environments, with an initial focus on the professional context. It aims to clarify the concept of explainability and offer a complementary, human-centered perspective to existing approaches, drawing on an exploratory study conducted in a professional setting.