

# Modèle tabulaire adaptatif de classement des outils intelligents d'aide à la conception architecturale

Louis Roobaert<sup>1,2\*</sup>, and Damien Claeys<sup>1,2</sup>

<sup>1</sup>TSA-LAB, LAB, UCLouvain, Rue Wafelaerts 47/51, 1060 Bruxelles, Belgique

<sup>2</sup>LOCI, UCLouvain, Rue Wafelaerts 47/51, 1060 Bruxelles, Belgique

**Résumé.** L'usage d'outils intelligents d'aide à la conception redéfinit les pratiques en conception architecturale. Pour permettre aux non-experts d'appréhender les fonctions et les combinaisons potentielles de ces différentes formes d'intelligence artificielle, un modèle de tableau périodique et adaptatif des outils intelligents est proposé et discuté. Pour révéler la pertinence du modèle, différentes variétés algorithmiques sont présentées et placées dans le tableau.

**Mots-clés.** intelligence artificielle, conception architecturale, tableau périodique, apprentissage automatique, apprentissage profond

**Abstract.** The use of intelligent design assistance tools is redefining practices in architectural design. To enable non-experts to understand the functions and potential combinations of these different forms of artificial intelligence, a model of a periodic and adaptive table of intelligent tools is proposed and discussed. To demonstrate the relevance of the model, different algorithmic varieties are presented and placed in the table.

**Keywords.** artificial intelligence, architectural design, periodic table, machine learning, deep learning

## 1 Introduction

Alors que la digitalisation du réel affecte profondément tous les secteurs de l'activité humaine, l'usage croissant d'outils *intelligents* d'aide à la conception modifie les pratiques de la conception de l'environnement construit. De récentes études comparatives entre les spectres cognitifs des humains et des machines analysent les hybridations potentielles au sein

---

\* Corresponding author: [louis.roobaert@uclouvain.be](mailto:louis.roobaert@uclouvain.be)

d'écosystèmes de concepteurs [1,2]. Elles explorent des modalités collaboratives de travail dans lesquelles la cognition artificielle ne remplace pas la cognition humaine, mais elle l'étend et elle l'enrichit. En effet, les outils dotés d'*intelligence artificielle* (IA) posséderaient des aptitudes inédites dans des tâches spécifiques, telles que le traitement de *données en masse*, l'effectuation de calculs complexes ou l'établissement de prédictions depuis un *jeu de données*. L'intégration raisonnée d'outils intelligents dans les processus de projection architecturale enrichirait donc les pratiques actuelles, tout en les amenant à se réinventer.

Pourtant, la formation initiale des architectes intègre peu la problématique de l'IA. De plus, les praticiens bénéficient rarement de formations continues en la matière. L'intelligence artificielle considérée avec défiance comme une *black box*, conçue et implémentée par d'autres, dont la complexité et l'évolution constante briment l'autonomie et la créativité des concepteurs architecturaux. Plus précisément, la connaissance des formes d'IA et de leurs hybridations potentielles est difficilement accessible aux non experts. Enfin quand elle est utilisée, elle l'est de manière limitée, principalement pour l'automatisation de tâches répétitives, plutôt que comme une véritable partenaire de conception.

En partant du principe que le risque n'est pas que l'IA remplace l'être humain, mais qu'un nombre limité d'êtres humains ayant des connaissances en IA assujettissent tous les autres, des efforts répétés sont nécessaires pour garantir un accès équitable aux initiés et à ceux qui ne le sont pas. Plus précisément, l'objectif est de faciliter la connaissance des outils intelligents d'aide à la conception architecturale en proposant un modèle de structure tabulaire adaptative, permettant de les classer objectivement. Inspiré du tableau périodique des éléments chimiques, le tableau proposé ici est destiné à renforcer, auprès des non experts, une compréhension intuitive des potentialités de ces outils, ainsi que de leurs combinaisons possibles, avant d'en faire des partenaires avec lesquels collaborer lors de processus de conception architecturale.

Dans le présent texte, l'IA et ses sous-parties sont d'abord définies globalement, à partir des questions liées à la gestion des données et à l'apprentissage. Ensuite, un modèle de classification tabulaire est établi (périodes, familles, blocs, étiquettes). Sélectionnés pour leurs liens avec le domaine de la conception architecturale, des outils intelligents sont placés dans la structure du tableau. Enfin, la pertinence du modèle est discutée.

## 2 Intelligence artificielle

Sachant que le concept d'*intelligence* ne possède aucune définition définitive, l'expression *intelligence artificielle* (IA) est controversée dès son introduction [3], si bien qu'aucune définition d'elle et de ses sous-parties n'est communément partagée. Plutôt que de tenter de définir ce qu'est l'IA en tant que tel, il est préférable d'étudier la constitution de la discipline, l'objet qu'elle poursuit et les différentes approches utilisées pour l'atteindre.

En tant qu'outil de résolution de problèmes à partir de données, l'IA a pour objet d'« émuler par un programme d'ordinateur des comportements intelligents sans pour autant reproduire le fonctionnement correspondant de l'être humain » [4]. Ce vaste programme de recherche comprend un ensemble de théories et de procédures simulant le comportement humain pour reproduire les raisonnements cognitifs et les actions mécaniques. Contrairement à la méthode classique des algorithmes dédiés à la résolution de problèmes algébriques (description exhaustive de la séquence d'opérations à mener pour résoudre un problème

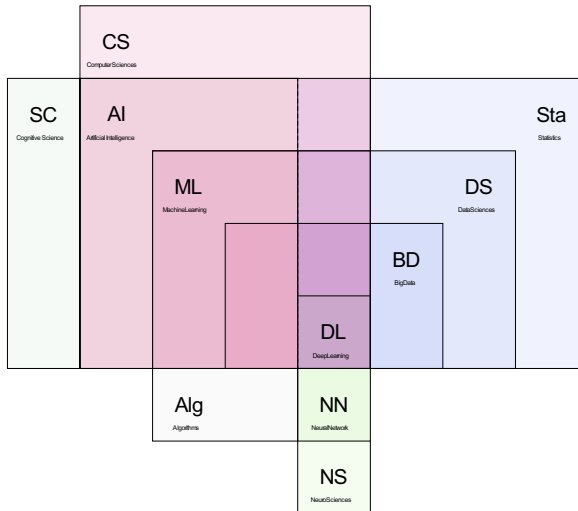
donné), l'IA utilise la méthode des heuristiques (méthode de résolution non déterministe à partir d'essais et d'erreurs, sans garantie de résultat, mais plus rapide).

Historiquement, deux approches sont régulièrement distinguées : (1) l'approche *connexionniste*, menant au connexionnisme, modélisant la cognition comme un processus émergent, calquée sur la structure physiologique du cerveau pour créer une *machine neuronale universelle*, donnant vie aux premiers réseaux de neurones artificiels (NN) [7,8] ; (2) l'approche *symbolique*, menant au cognitivisme, calquant les processus mentaux sur le fonctionnement symbolique du langage humain par la manipulation de signes à partir de règles logiques, pour établir une *machine symbolique universelle*, mettant en place les premiers agents conversationnels et systèmes experts [9–11]. Actuellement, la discipline s'est considérablement étoffée parallèlement au récent renouvellement de l'engouement pour l'approche connexionniste (apprentissage par expérience), longtemps éclipsée par les résultats de l'approche symbolique (systèmes experts). De plus, les deux approches sont régulièrement hybridées (algorithmes évolutionnaires testant des variations, inférences bayésiennes intégrant l'incertitude en phase d'apprentissage, systèmes apprenant par analogie en reconnaissant des tendances dans des jeux de données...) [2].

En tant que discipline, l'IA s'est progressivement constituée par la convergence de plusieurs disciplines [12] (figure 1) : (1) l'*algorithmique* (Alg) ; (2) l'*informatique* (CS) fournit les données structurées, les langages de programmation et les modèles computationnels [13] ; (3) les *statistiques* (Sta) abordent les questions liées aux probabilités et à la structuration géométrique des données [14], elles recouvrent les *sciences des données* (DS), elles-mêmes donnant des outils pour gérer les *données de masse* (BD) ; (4) les *neurosciences* (NS) de l'approche connexionniste (apprentissage, plasticité synaptique, traitement de l'information) fournissent différents types de réseaux de neurones artificiels (NN) [15] ; (5) les *sciences cognitives* (SC) de l'approche symbolique combinent notamment la psychologie, la linguistique et la philosophie, pour se concentrer sur l'étude des processus mentaux [16].

Au cœur de cette convergence transdisciplinaire, l'IA est ici divisée en deux catégories principales : (1) l'« apprentissage machine » (ML), issu de l'approche symbolique, est un système permettant de prendre des décisions à partir d'un ensemble de données avec un facteur d'activation unidimensionnel [17,18]; (2) l'« apprentissage profond » (DL), issu de l'approche connexionniste, en tant que système utilisant des réseaux de neurones artificiels à plusieurs couches pour modéliser des relations complexes entre des entrées et des sorties, induisant l'apprentissage avec un grand ensemble de données, de s'affiner avec des boucles de rétroaction et une lecture multidimensionnelle .

Bien que régulièrement considérés comme une catégorie intermédiaire entre le ML et le DL, les NN interviennent plutôt dans le DL et ils sont placés ici à la rencontre des NS et de l'IA [22–24]. De nombreux autres sous-domaines sont clairement distingués, mais ils ne sont pas repris ici.



**Figure 1.** La discipline de l'intelligence artificielle et ses grandes sous-parties

La croissante, plus qu'exponentielle, des Systèmes d'Intelligence Artificielle (SIA) ou *Artificial Intelligence System* [AIS] [29] a conduit certains groupes de chercheurs à cartographier leur paysage selon des critères bien spécifiques. Parmi d'autres, la *Foundation Model Transparency Index* (FMTI) a été créée par le *Center for Research on Foundation Models* (CRFM) à l'Université de Stanford en collaboration avec des chercheurs du MIT Media Lab et du Center for Information Technology Policy de l'Université de Princeton [30]. Le FMTI vise à évaluer la transparence des SIA par leur évaluation au travers d'une grille critériée en demandant aux entreprises d'intelligence artificielle de soumettre des rapports détaillant divers aspects de leurs modèles [31]. Une autre forme de proposition d'évaluation des modèles de SIA est émise par François Chollet un chercheur en intelligence artificielle chez Google. Dans le programme Open Ended Electronic Intelligent Robot Operating System (ONEIROS) il a développé une librairie *open-source*, Keras, pour l'interface Python qui mesure le niveau de performance de SIA par des benchmarks, standardisant les tests et permettant de faire ressortir des mesures comparables de la performance des différents modèles de SAI. Les modèles Keras permettent de tester les performances sur des ensembles de données standardisés (comme ImageNet), et de mesurer des métriques telles que la précision, la vitesse d'inférence, et l'efficacité des calculs [32].

Ainsi, en complément de ces études qui analysent et comparent les SIA qui peuplent et hybrident de plus en plus nos environnements, cet article propose une autre approche du sujet. Plutôt que de se concentrer sur les SIA eux-mêmes, l'objectif est l'établissement d'une cartographie des différentes techniques propres au niveau de lecture de l'intelligence artificielle – lorsqu'elle n'est pas transposée dans un système. L'idée est l'établissement d'une table périodique des différents algorithmes propres à l'apprentissage artificiel en conception architecturale combinables pour générer des SIA.

### 3 La structure du tableau périodique

Une structure est proposée pour constituer un tableau permettant le classement des différents outils intelligents d'aide à la conception architecturale.

Attribué au chimiste russe Dmitri Ivanovitch Mendeleïev [33], le tableau périodique des éléments a été adapté de nombreuses fois avant d'apparaître sous sa forme actuelle. Les éléments chimiques connus y sont classés, en fonction de leurs propriétés physiques et chimiques, tout en permettant l'ajout futur d'éléments encore inconnus. La structure du tableau actuel comprend : sept *périodes* horizontales (alignement horizontal d'éléments ayant un même niveau d'énergie électronique), dix-huit *familles* verticales (alignement vertical d'éléments ayant le même nombre d'électrons de valence) et quatre *blocs* (groupes d'éléments dont les électrons de valence appartiennent aux mêmes sous-couches électroniques).

Par analogie, le tableau périodique de l'apprentissage artificiel adopte une configuration similaire : (1) huit *périodes* horizontales fournies par une échelle verticale décrivant le niveau de complexité algorithmique des outils disponibles ; (2) six *familles* verticales distinguées par une stratification de base, correspondant aux différentes méthodes d'apprentissage mobilisées par les outils (supervisé, non supervisé, renforcement) et à l'intensité de cet apprentissage (non profond, profond) ; (3) quatre *blocs* répètent horizontalement cette stratification en reproduisant la succession des quatre phases habituelles de traitement des données lors d'un processus d'IA (analyse, modélisation, décision, inférence) ; (4) une *étiquette* [*label*] identifie individuellement chaque élément en apportant des informations supplémentaires.

Les quatre sections suivantes explicitent les différents éléments structurels du tableau : les périodes, les familles, les blocs et les étiquettes.

### 3.1 Niveaux de complexité algorithmique (périodes)

À première vue, les outils intelligents peuvent être distingués par l'observation directe de leurs limites *matérielles* (par exemple, les types de mémoires vives ou de processeurs utilisés) ou *logicielles* (par exemple, les types de langages informatiques utilisés). Mais leur efficacité est distinguée ici *formellement*, d'après la valeur générale de leur niveau de complexité algorithmique, calculée indépendamment de leur environnement d'exécution [34].

En informatique théorique, le *degré de complexité* d'un problème correspond au niveau de difficulté pour le résoudre. La complexité est à la fois *spatiale* (l'espace mémoire, ou la taille occupée par les données impliquées dans le processus de résolution du problème) et *temporelle* (le temps de calcul, ou le nombre d'opérations élémentaires à effectuer pour clôturer ce processus). De là, une *classe de complexité* est un ensemble de problèmes algorithmiques dont la résolution nécessite une même quantité de ressources. En fonction du degré de complexité auquel répond un algorithme, ce dernier peut être réparti dans une classe d'*efficacité*, chacune d'elles étant descriptible de manière rigoureuse par des ordres de grandeurs asymptotiques (bornage supérieur de fonctions mathématiques).

Introduite à la fin du XIX<sup>e</sup> siècle par la théorie des nombres [35] avant d'être largement adoptée en informatique pour analyser la complexité algorithmique [36,37], la notation  $O$  désigne l'ordre de complexité de l'algorithme, tandis que la complexité relative aux données est exprimée par la variable  $n$ . Elle fournit un cadre standardisé pour évaluer l'efficacité des algorithmes en fonction de leur complexité temporelle ou spatiale, sans avoir à les implémenter et à mesurer leur performance empiriquement.

Dans le tableau, les outils intelligents sont ici classés en huit périodes horizontales correspondant chacune à un niveau de complexité, l'échelle verticale allant du moins

complexe au plus complexe : (1) constant  $O(1)$  ; (2) logarithmique  $O(\log n)$  ; (3) linéaire  $O(n)$  ; (4) linéarithmique  $O(n \log n)$  ; (5) quadratique  $O(n^2)$  ; (6) cubique  $O(n^3)$  ; (7) exponentiel  $O(2^n)$  ; (8) factoriel  $O(n!)$ . Ainsi, le niveau de complexité d'un algorithme noté  $O(1)$  ne varie pas en fonction de la taille des données, tandis que celui d'un algorithme noté  $O(n^2)$  aura une croissance quadratique en fonction de la taille des données d'entrée (si la taille des données est doublée, le temps d'exécution est quadruplé). En plaçant la taille des données  $n$  en abscisses et le temps  $t$  de leur traitement en ordonnées, les algorithmes peuvent être comparés graphiquement : plus la pente de la courbe est raide, plus la complexité algorithmique est élevée [38].

### 3.2 Méthodes d'apprentissage (familles)

Dans le tableau, les outils intelligents sont classés en fonction du type de méthode d'apprentissage qu'ils mobilisent, par colonne, selon six *familles* de base. Deux critères de classification sont utilisés pour déterminer ces six familles : (1) le mode d'apprentissage de l'outil ((I) supervisé, (II) non supervisé, (III) renforcement), entre intervention humaine et autonomie ; (2) l'intensité de cet apprentissage ((a) non profond, (b) profond), sachant que les trois dernières impliquant du DL sont des versions plus complexes des trois premières impliquant du ML, le tableau est donc subdivisé en six types de colonnes :

(1) l'apprentissage supervisé et non profond (I.a –  $N$ ) regroupe les algorithmes entraînés à partir de jeux de données *étiquetées* (classées avec des attributs connus et associées à des réponses connues) pour établir des correspondances entre des entrées et des sorties. La méthode est très précise, mais nécessite un traitement humain des données et elle possède une capacité de généralisation limitée. Les techniques couramment utilisées incluent la classification et la régression, la régression linéaire ou les réseaux de neurones supervisés [23] ;

(2) l'apprentissage non supervisé et non profond (II.a –  $S$ ) reprend les algorithmes capables d'identifier des modèles et des structures dans des jeux de données sans directives explicites. À partir d'un jeu de donnée, l'algorithme s'entraîne à identifier les modèles, les structures et les relations implicites entre les entrées et les sorties. Cette approche est nettement plus complexe car elle demande à l'algorithme de développer sa propre interprétation des données sans directives explicites. Les techniques couramment utilisées incluent le *clustering* (regroupement par similitude), l'association ou la réduction de dimensionnalité [39] ;

(3) l'apprentissage par renforcement non profond (III.a –  $R$ ) reprend les algorithmes autonomes devant apprendre par eux-mêmes les actions qu'ils doivent réaliser. Ce type d'apprentissage consiste à laisser des êtres artificiels apprendre de leurs propres expériences, sans leur fournir de données, grâce à un système cumulatif de récompense ou de pénalité [40].

(4) l'apprentissage supervisé et profond (I.b –  $NP$ ) décrit des réseaux de neurones entraînés à partir d'ensemble de données étiquetées apprenant à prédire avec plus de précision les sorties à partir des entrées fournies, comme pour la reconnaissance vocale ou le traitement automatique du langage [41] ;

(5) l'apprentissage non supervisé et profond (II.b –  $SP$ ) comprenant des structures comme des GAN ou des auto-encodeurs sont utilisés pour découvrir des structures dans des données sans étiquetage permettant d'apprendre plusieurs niveaux de caractéristiques ou de représentation de la donnée pour par exemple effectuer de la génération d'image ou de texte [19] ;

(6) l'apprentissage par renforcement profond (III.b – *RP*), combinant les techniques d'apprentissage profond avec des algorithmes d'apprentissages par renforcement, permettant à un agent autonome de prendre des décisions stratégiques et d'optimiser son comportement en vue de maximiser les récompenses obtenues [42].

### 3.3 Phases des processus intelligents (blocs)

Les processus de résolution de problème à l'aide d'outils intelligents passent par quatre phases fondamentales liées à la gestion de données et à l'apprentissage. De là, reprenant les six familles d'apprentissage, quatre blocs sont définis :

(1) analyse de données (traitement et collecte des entrées) : le concepteur collecte et analyse les données dont il dispose, issues de ses expériences passées, ou spécifiquement récoltées pour le processus qu'il entame. En mobilisant ses compétences cognitives, il traite et structure ces données de façon à les visualiser ou à les comparer. Cette phase de traitement est polymorphe allant du regroupement, au filtrage ou encore à la classification permettant l'organisation des données de manière structurée facilitant la reconnaissance de *patterns* ou la déduction d'*insights* ;

(2) développement de modèles (développement et ajustement) : comme le processus de mise en forme des données pour créer différentes propositions de projets admissibles, les données analysées et les *insights* sont utilisés pour développer des modèles d'apprentissage capables d'effectuer des prédictions. Les modèles prédictifs fonctionnent en utilisant les caractéristiques des données passées pour générer des prévisions ;

(3) prise de décision (fondées sur les résultats des modèles) : à l'instar de l'évaluation des différentes propositions de projets pour en sélectionner une, une prise de décision équilibre le lien entre l'exploration (présentation des risques et des coûts) de nouvelles alternatives et l'exploitation de stratégies éprouvées, (procédures réussies du passé) [40] ;

(4) inférence causale (vérification et interprétation) : comme lors de la vérification et l'évaluation de la décision prise, où le concepteur vérifie par des simulations que la proposition de projet choisie produit les résultats attendus et répond aux critères prédéfinis, les modèles développés avec l'IA sont rigoureusement évalués et testés pour confirmer leur précision, leur fonction de rappel permet de confirmer leur caractère généralisable et leur robustesse. Enfin, l'inférence causale consiste à identifier les causes d'un phénomène et à les relier aux effets observés, permettant ainsi une rétroaction sur la validité des décisions prises. À chaque étape, l'apprentissage profond peut implémenter une variété d'architectures neuronales utilisant des fonctions d'activation, comme pour réguler l'activation des neurones. Les méthodes d'optimisation et de régularisation sont appliquées pour affiner les modèles et prévenir le surajustement [43].

En pratique ces quatre étapes constituent la base du travail d'un *data scientist* cherchant à établir un système algorithmique pour résoudre un problème donné. Par analogie, un parallélisme partiel peut être identifié entre celles-ci et des tâches apparaissant dans les phases de conception engagées par les architectes. En effet, la projection architecturale peut être considérée comme la tentative de résolution d'une gamme particulière de « problèmes mal définis » [44] nécessitant des méthodes de résolution spécifiques. À défaut d'être linéaire, ce type processus est plutôt « circulaire », « itératif » et « réflexif », passant alternativement par des phases de « convergence » et de « divergence » avant d'aboutir à une « solution sous-optimale » [44,45]. Dans ce contexte, le concepteur opère une combinaison dynamique d'outils intelligents d'aide à la conception, entre lesquels il navigue constamment pour affiner le projet qu'il conçoit.

### 3.4 Propriétés des outils intelligents (étiquettes)

Pour améliorer la compréhension des effets potentiels de leurs usages en conception architecturale, outre le niveau de complexité algorithmique, le mode d'apprentissage et la phase des processus, chaque algorithme associé à un outil intelligent est précisément identifié à l'aide d'une étiquette [*label*] comprenant cinq informations supplémentaires :

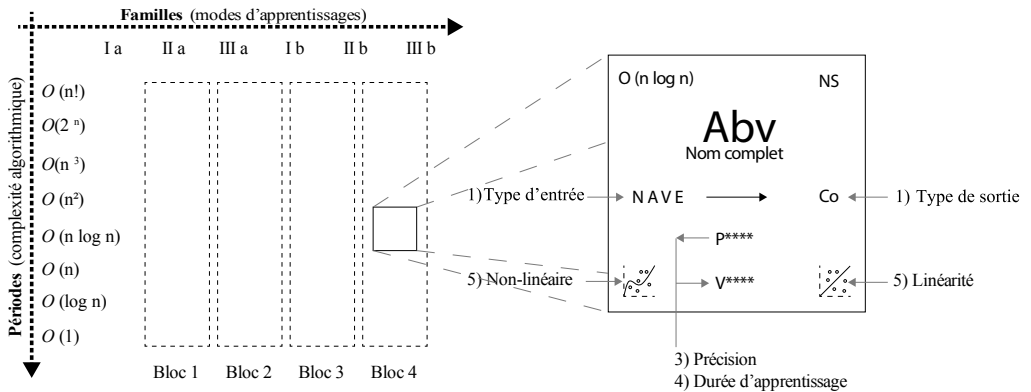
(1) le type d'entrées [*inputs*] : décrit la nature des entrées traitées par l'algorithme, pouvant être *numériques* (N), *audios* (A), *vidéos* (V) ou *écrites* (E), sachant que des algorithmes utilisent plusieurs types de donnée [46] ;

(2) le type de sorties [*outputs*] : décrit la nature des sorties traitées par l'algorithme, pouvant être *continues* (production d'une valeur appartenant à la même classe que l'entrée), *catégorielles* (classification des d'entrée), *séquentielles* (séquence de valeur ou de labels), *structurées* (organisation des entrées créant une nouvelle étiquette), ou encore *de classement* (classement des entrées selon des métriques spécifiques), ou *probabilité* (mesure de l'incertitude) [47,48] ;

(3) la précision (ou « valeur prédictive positive ») : indique la proportion de prédiction correcte (vrais positifs) par rapport au nombre total de prédiction positives (la somme des vrais positifs et des faux positifs) (valeur échelonnée de 1\* à 4\*, de la précision minimale à la précision maximale) ;

(4) la durée d'apprentissage : indique la vitesse d'apprentissage de l'algorithme à partir de données d'entraînement avant d'atteindre un seuil de maturation de performance ou de précision suffisant (valeur échelonnée de 1\* à 4\*, de lent à très rapide) [49] ;

(5) la linéarité : les algorithmes modélisent des relations *linéaires* ou *non linéaires*, selon que toute modification des entrées est directement ou non proportionnelle à celles des sorties, bien que certains soient capables de modéliser les deux types de relations à la fois [49].



**Figure 2.** (a) Structure du tableau à l'aide des périodes, des familles et des blocs. (b) Structure d'une étiquette. (c) Calcul de la précision d'un algorithme

## 4 Placement de variétés algorithmiques dans le tableau

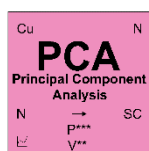
Après avoir établi la structure générale du modèle tabulaire des outils intelligents (périodes, familles, blocs, étiquettes), la pertinence de celui-ci est testée par un positionnement non exhaustif de variétés algorithmiques. Le caractère *adaptatif* du modèle tabulaire intègre la possibilité d'adapter et d'ajouter d'autres algorithmes, reflétant ainsi le caractère dynamique



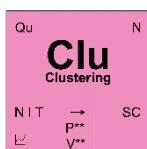
et cyclique des processus de conception et d'implémentation constante de nouveaux outils intelligents. Les algorithmes présentés ci-dessous sont décrits, bloc par bloc, en parcourant successivement les quatre phases de traitement de données constituant des processus intelligents de résolution de problèmes : (1) analyse de données ; (2) développement de modèles ; (3) prise de décision ; (4) inférence causale.

### 4.1 Analyse de données (bloc 1)

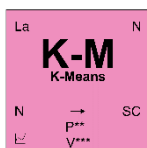
Contient l'ensemble des algorithmes permettant le traitement des données en facilitant leur visualisation. Trois principales catégories d'algorithmes existent dont chacune est composée d'une diversité de sous-algorithmes [50] : (1) la *réduction de la dimensionnalité* permet le traitement de données de haute dimension où le nombre de variables, d'étiquettes et d'attributs empêche leur corrélation pour la cognition humaine [51] ; (2) le *clustering* regroupe des données en sous-ensembles homogènes, révélant des structures et des patterns internes ; (3) la *classification* attribue des étiquettes aux données non étiquetées basées sur des modèles appris [52], la sélection d'une classe d'algorithmes dépend de la structuration des données dont dispose le concepteur. Tandis que le type de données (texte, nombres, vidéos, images) détermine l'outil le plus approprié pour l'analyse [23].



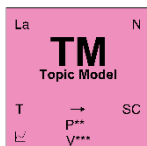
Principal Component Analysis (PCA) : l'*analyse en composantes principales* réduit les dimensionnalités d'un ensemble de données (permet la simplification de données complexe en un nombre plus petit de variables non corrélées).



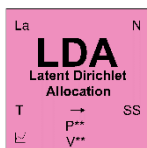
Clustering (Clu) : le *regroupement* explore un ensemble de données et les regroupe en fonction de leurs similitudes, en faisant ressortir les caractéristiques similaires et les motifs récurrents, tout en simplifiant la visualisation spatiale des entrées.



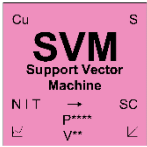
K-Means (KM) : le *K-Moyen* est une sous-technique du regroupement, identifiant les *clusters* naturels, en minimisant la variance *intracluster*, par l'évaluation de la distance entre chaque donnée par rapport à la moyenne la plus proche [53].



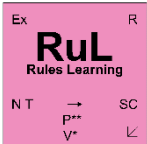
Topic Model (TM) : le *modèle de sujet* identifie les thèmes récurrents dans les entrées textuelles. Il facilite la recherche de sujet, aidant à classer et structurer l'information en procédant à une analyse thématique par la détection des sujets prédominants et des mots-clés associés.



Latent Dirichlet Allocation (LDA) : l'*allocation de dirichlet latente* est similaire en concept au K-Means, mais spécifique au traitement de texte. Elle aide à identifier les tendances dans un document textuel.



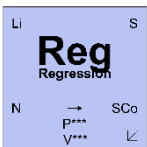
Support Vector Machine (SVM) : la *machine à vecteurs de support* observe la façon dont les données de grande dimensionnalité sont distribuées sur un hyperplan cartésien en les séparant en classes, en fonction de la distance aux points de données les plus proches de toutes les classes appelée la marge maximale [54].



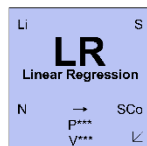
Rules Learning (RuL) : l'*apprentissage par règle* analyse un ensemble de données et sépare celles qui respectent ou non les règles établies (tel qu'un règlement urbanistique, des normes de sécurité...)

## 4.2 Développement de modèles (bloc 2)

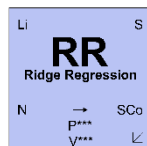
Contient l'ensemble des réseaux et des modèles permettant le développement et l'ajustement des modèles prédictifs. Le but de ces éléments est d'effectuer des décisions à partir des données traitées. Cette section contient habituellement trois familles : (1) la *régression* comprend les méthodes visant à estimer les relations entre variables et à prédire des valeurs continues [55] ; (2) la *classification* regroupe les techniques assignant des catégories discrètes aux instances de données ; (3) les *réseaux de neurones* sont les systèmes modélisant les interactions complexes non linéaires entre les entrées et les sorties, simulant le comportement du cerveau humain pour reconnaître les *patterns* complexes [56].



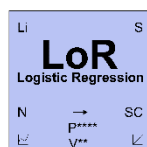
Regression (Reg) : la *régression* identifie des relations de causalité au sein d'une série de données historiques et effectue une modélisation afin de prévoir des prédictions. Elle aide à comprendre l'impact de variables sur des données quantitatives (Freedman, 2009).



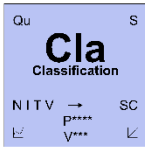
Linear Regression (LR) : la *régression linéaire*, considérée comme le modèle de régression le plus élémentaire établit des relations linéaires entre les variables indépendantes et la variable dépendante en utilisant des pondérations pour évaluer la contribution de chaque attribut à une variable cible.



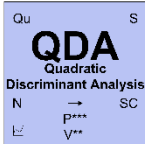
Ridge Regression (RR) : la *régression de crête*, ajoutant un terme de pénalité à la LR réduisant ainsi la complexité du modèle, sans accroître significativement le biais, tout en aidant à prévenir le surajustement améliorant la précision des prédictions.



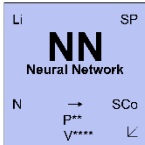
Logistic Regression (LoR) : la *régression logistique* est une adaptation de la RL pour les problèmes de classification, en particulier pour les réponses binaires (vrai/faux). Elle est fréquemment utilisée pour modéliser la probabilité d'un événement en fonction de plusieurs prédicteurs et à déterminer les facteurs critiques de réussite.



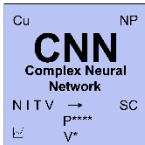
Classification (Cla) : la *classification* organise les données en catégories distinctes. Elle vise à identifier les motifs et les relations dans un ensemble de donnée permettant de catégoriser les nouveaux éléments sur base des catégories préétablies [57].



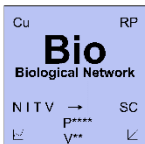
Quadratic Discriminant Analysis (QDA) : l'*analyse discriminante quadratique* est une méthode plus sophistiquée de Cla, car elle utilise une fonction quadratique qui fournit des courbures permettant d'obtenir des modèles prédictifs plus perfectionnés et précis que la version linéaire qui aide à prévoir la catégorie d'un nouvel échantillon en fonction de ses caractéristiques.



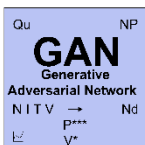
Simple Neural Network (NN) : les *réseaux de neurones simples*, inspirés des travaux initiaux de McCulloch et Pitts, disposent d'une unique couche connectant les entrées aux sorties. Ils comportent des entrées (variables indépendantes) qui, après pondération et ajout d'une constante, génèrent des sorties (variables dépendantes), au travers d'une fonction d'activation [7].



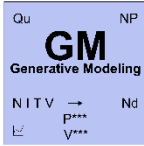
Complex Neural Network (CNN) : les *réseaux de neurones complexes* ont une structure plus développée que les NN, car ils ont des couches intermédiaires entre les deux extrémités (entrée et sortie), appelée *couche cachée*, dont le rôle est d'effectuer des calculs intermédiaires qui se déplacent d'une couche à une autre jusqu'à l'obtention d'un résultat final. L'architecture d'un réseau de neurones complexe dépend du nombre de couches cachées, du nombre de neurones par couche cachée et de la typologie des connexions entre les neurones.



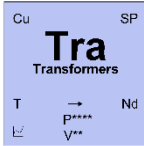
Biological Networks (Bio) : les *réseaux biologiques* sont des CNN dont l'architecture est inspirée de la structure et du fonctionnement des systèmes neuronaux biologiques. Ils tentent de modéliser les mécanismes complexes du cerveau pour créer des algorithmes capables d'apprentissage et de traitement de l'information de manière similaire aux organismes vivants.



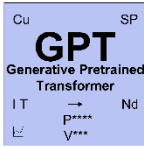
Generative Adversarial Network (GAN) : le *réseau adverse génératif* est un modèle d'apprentissage non supervisé ou double réseaux neuronaux, généralement un générateur et un discriminateur qui sont entraînés simultanément dans un jeu compétitif. Le générateur crée des données qui ressemblent aux données d'entrées, tandis que le discriminateur essaie de distinguer les données générées des vraies données [58].



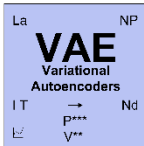
Generative Modeling (GM) : le *modelage génératif* apprend à générer de nouvelles données similaires à celles d'entraînement en modélisant la distribution statistique. Ces modèles sont utilisés dans diverses applications, allant de la synthèse de texte à la création d'images.



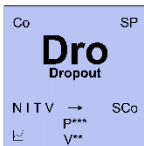
Transformer (Tra) : le *transformateur* possède une architecture qui est spécifique au traitement du langage naturel. Cette méthode utilise des mécanismes d'attention pour améliorer la qualité de l'élément de sortie. Contrairement au CNN le Tra crée des liens de dépendance entre les sorties.



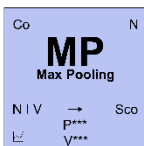
Generative Pretrained Transformer (GPT) : le *transformateur pré-entraîné génératif* est un modèle de traitement de langage naturel basé sur l'architecture du Tra. Il est préentraîné sur un large corpus de texte et peut être affiné pour accomplir une grande variété de tâches linguistiques. Il est largement utilisé pour la reconnaissance vocale, résumer des textes et répondre à des questions.



Variational Autoencoders (VAE) : l'*autoencodeur variationnel* produit de nouvelles données en convertissant les données d'entrées. Pour ce faire celles-ci sont représentées et compressées dans un espace latent et ceux-ci sont reconstruit sous une autre forme [58].



Dropout (Dro) : la *régularisation* simplifie les calculs dans les CNN. Elle ne considère qu'une sélection aléatoire de neurones lors du calcul du gradient, réduisant ainsi la complexité et le risque de surajustement [59].

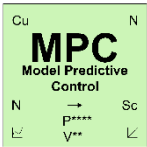


Max Pooling (MP) : le *pooling maximum* est utilisé principalement dans les réseaux de neurones convolutionnels. Il simplifie l'information, en sélectionnant la plus grande valeur de chaque sous-région de la couche précédente, réduisant ainsi la dimensionnalité et permettant au réseau de rester insensible aux petites variations et à l'échelle des données [60].

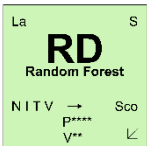
### 4.3 Prise de décision (bloc 3)

La prise de décision dépend largement de la nature et de la quantité des données disponibles. Après une phase d'apprentissage autonome, les modèles d'IA sont capables de générer des prédictions (phase 2). Sur base de ces prédictions, une prise de décision peut être effectuée, en fonction de la disponibilité des données (de faible à élevée) et de l'état de l'environnement

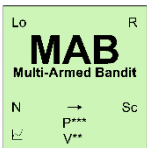
(fixe ou dynamique). L'environnement est fixe lorsqu'il ne change pas ou alors de façon très lente (par exemple, l'espace bâti existant) et *dynamique* lorsqu'il change à chaque action, interaction ou unité de temps (par exemple, lors de la conception d'un projet d'architecture). En croisant la disponibilité des données et l'état de l'environnement, quatre scénarios distincts sont définis pour optimiser la prise de décision et ils influencent le choix des outils intelligents.



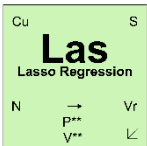
Model Predictive Control (MPC) : la *commande prédictive* est un modèle algorithmique visant à améliorer les décisions futures. Elle utilise des données historiques pour *apprendre* l'état d'un environnement, puis prédire et optimiser la trajectoire de sortie [61].



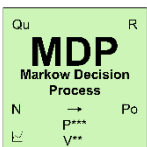
Random Forest (RD) : la *forêt aléatoire* est un ensemble d'arbres de décision, où chaque arbre est construit à partir d'un échantillon de données et utilise un sous-ensemble aléatoire de variables à chaque division, permettant d'améliorer la décision et de réduire le surajustement. Chaque arbre vote pour une classe et la majorité est choisie pour la prédiction finale.



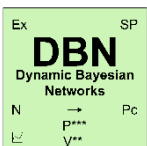
Multi-Armed Bandit (MAB) : le *bandit multi-bras* permet à la fois l'exploration et l'exploitation d'un ensemble d'options alternatives. L'algorithme évalue toutes les options et choisit celle qui offre la plus grande récompense. Le nom de l'algorithme provient du terme désignant une personne qui joue sur plusieurs machines à sous en même temps.



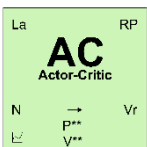
Lasso Regression (Las) : la *régression lasso* est une technique pour améliorer la prédictibilité et d'interprétabilité de modèles. Elle introduit une pénalité de régularisation sur les coefficients favorisant la sélection automatique de variables. Las est largement utilisée pour les ensembles de données avec de nombreuses variables [62].



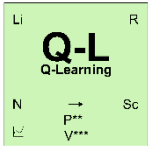
Markov Decision Process (MDP) : le *processus de décision markovien* est un modèle simulant l'interaction constante entre un agent et un environnement. L'agent choisit certaines actions, et l'environnement réagit à ces actions tout en présentant de nouvelles situations à l'agent. Lorsque plusieurs décisions sont en jeu, le but ultime pour l'agent est d'obtenir la plus grande récompense possible à la fin du processus.



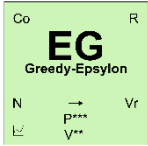
Dynamic Bayesian Networks (DBN) : les *réseaux bayésiens dynamiques* modélisent des séquences de données ou des processus qui évoluent dans le temps pour reconnaître des séquences, grâce à une série d'interdépendances entre des variables.



Actor-Critique (AC) : l'*acteur-critique* utilise deux composants principaux : l'acteur qui choisit les actions à entreprendre et le critique qui évalue les actions prises en calculant une fonction de valeur. Ce double processus réduit la variance des estimations de la récompense [40].



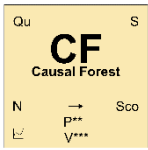
Q-Learning (QL) : l'*apprentissage-Q* permet à un agent, sans modélisation de l'environnement, d'apprendre une fonction de valeur qui estime la valeur optimale des actions possibles à partir d'un état donné. Au fur et à mesure des décisions prises, QL met à jour ses valeurs en utilisant les récompenses obtenues et en maximisant les récompenses futures attendues.



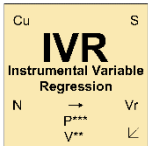
Epsilon-Greedy (EG) : l'*epsilon-Avide* est une stratégie qui équilibre l'exploration et l'exploitation. À chaque étape, l'agent choisit une action au hasard (exploration), puis il choisit l'action qui, par probabilités, a la plus haute estimation de valeur (exploitation) [63,64].

#### 4.4 Inférence causale (bloc 4)

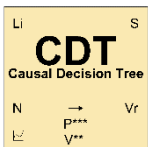
L'inférence causale se concentre sur l'identification et l'attribution des causes aux phénomènes observés. Elle vise à confirmer l'existence d'une relation de cause à effet entre divers événements, permettant ainsi d'évaluer l'efficacité des décisions prises.



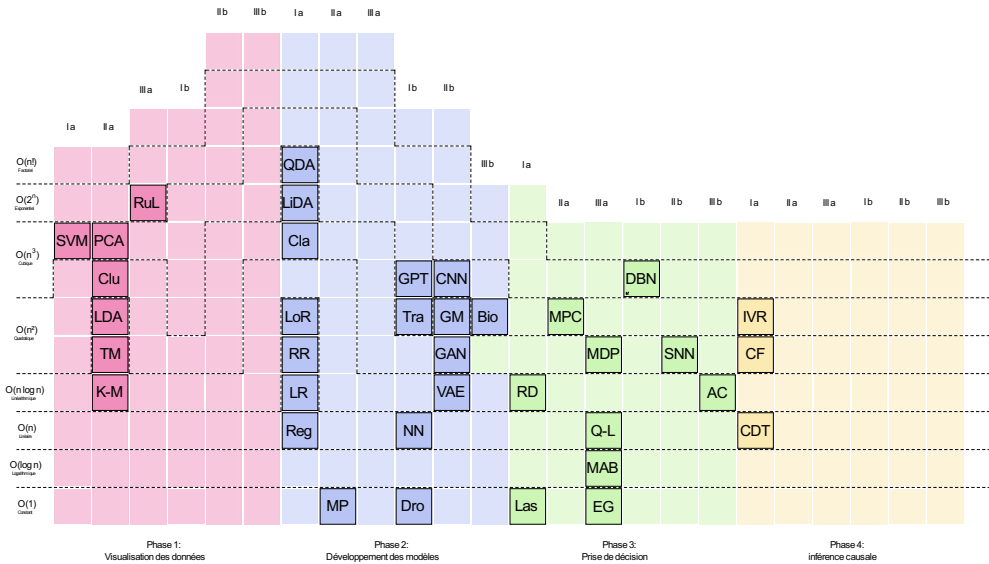
Causal Forest (CF) : la *forêt aléatoire causale* estime l'effet causal sur une variable du résultat en modélisant la relation entre les variables de traitement, les covariables, et les résultats. En générant plusieurs arbres de décision sur des sous-ensembles aléatoires des données, elle déduit comment la modification d'une variable influence une autre.



Instrumental Variable Regression (IVR) : la *régression avec variable instrumentale* estime des relations causales, lorsque le modèle de régression est sujet à une endogénéité, c'est-à-dire lorsque les variables explicatives sont corrélées avec le terme d'erreur. Cela permet d'obtenir des estimations non biaisées des effets causaux.



Causal Decision Tree (CDT) : l'*arbre de décision causal* identifie et quantifie les effets causaux d'une intervention sur un résultat, en subdivisant les données en sous-groupes homogènes en fonction des covariables. Chaque nœud de l'arbre représente une décision basée sur une variable, et les feuilles de l'arbre donnent une estimation de l'effet causal de l'intervention pour les individus dans le sous-groupe correspondant [65].



**Figure 3.** Le tableau périodique et évolutif des outils intelligents d’aide à la conception

## 5 Discussion et conclusions

À l’instar du tableau périodique de Mendeleïev classant les éléments chimiques et permettant de prévoir les combinaisons formant diverses molécules, ou encore, de logiciels de *visual scripting* permettant des combinaisons de composants intelligents pour construire des scripts capables de résoudre des problèmes spécifiques (Grasshopper ou Dynamo), le tableau périodique et évolutif des outils d’aide à la conception présenté ici a pour objectif d’aider les concepteurs non experts à situer et à envisager l’interopérabilité des différents outils disponibles, en comparant visuellement les algorithmes qu’ils mobilisent.

Cette approche combinatoire permet au moins de : (1) démystifier partiellement les processus d’IA mobilisables en conception architecturale ; (2) comprendre les fonctions individuelles de chaque algorithme, ainsi que leurs entrées et sorties ; (3) identifier les synergies possibles entre les algorithmes classés pour résoudre des problèmes complexes ; (4) imaginer des processus associant plusieurs outils avant leur mise en œuvre et identifier les solutions les plus adaptées à la puissance de calcul et aux données disponibles ; (5) anticiper l’interopérabilité entre toutes les phases d’un processus de conception utilisant l’IA.

Le positionnement d’outils intelligents dans le tableau proposé montre la pertinence de sa structure. L’objectif n’était pas de placer définitivement *tous* les algorithmes existants dans le tableau (seuls des algorithmes ayant un intérêt direct pour la conception architecturale ont été privilégiés), mais de fournir un modèle de référence situant chaque algorithme par rapport aux autres au sein d’une structure cohérente.

Enfin, la réflexion sur un modèle tabulaire de classification des outils intelligents participe à l’approfondissement de la compréhension du spectre cognitif au sein des réseaux hybrides en conception architecturale – englobant humains, machines et cyborgs –, cette recherche introduit une structure pour appréhender les interrelations entre l’IA et les outils des architectes. Cet instrument est à la fois un outil pédagogique qui déchiffre et rend

accessibles les concepts d'apprentissage automatique et profond, mais permet également leur intégration tant dans la pratique professionnelle que dans l'enseignement de l'architecture.

## Références

- [1] D. Claeys (2021), « Entre cogitation et computation : Comment déjouer l'obsolescence programmée de l'architecte ? » in *Anticrise architecturale : Analyse d'une discipline immergée dans un monde numérique*, PUL (Louvain-La-Neuve), pp. 157–176
- [2] L. Roobaert, D. Claeys, and S. Cleven (2024), « Espaces d'hybridation en conception architecturale : modalités collaboratives entre cognitions naturelles et artificielles in SHS Web of Conferences (Tunisie)
- [3] J. McCarthy, M. L. Minsky, N. Rochester, and C. E. Shannon (1955), A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955, *Artificial Intelligence*, 12
- [4] J.-P. Haton and M.-C. Haton (1989), *L'Intelligence artificielle*, Presses universitaires de France (Paris)
- [5] J. McCarthy (2007), *What is artificial intelligence?*, University Stanford Press
- [6] S. Legg and M. Hutter (2007), A Collection of Definitions of Intelligence,
- [7] W. McCulloch and W. Pitts (1943), A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biology* **65**, 99
- [8] F. Rosenblatt (1958), The perceptron: A probabilistic model for information storage and organization in the brain., *Psychological Review* **65**, 386
- [9] A. Turing (1950), *Computing Machinery and Intelligence*, *Mind* **LIX** (433)
- [10] J. Von Neumann (1945), First Draft of a Report on the EDVAC, Moore School of Electrical Engineering, Univeristy of Pennsylvania
- [11] J. Weizenbaum (1966), ELIZA– a computer program for the study of natural language communication between man and machine, *Communications of the ACM*, **9** (36)
- [12] S. J. Russell and P. Norvig (2016), *Artificial Intelligence: A Modern Approach*, 3rd ed (Pearson, Upper Saddle River)
- [13] R. A. Brooks, Intelligence without representation (1991), *Artificial Intelligence*, **47** (139)
- [14] B. Efron and T. Hastie (2016), *Computer Age Statistical Inference: Algorithms, Evidence, and Data Science* (Cambridge University Press, New York)
- [15] E. R. Kandel, J.-H. Schwartz, T. M. Jessel, S. Siegelbaum, and A. J. Hudspeth (2013), *Principles of Neural Science*, 5th ed (McGraw-Hill, New York)
- [16] M. S. Gazzaniga (2018), *The Cognitive Neurosciences*, 4th ed (MIT Press, Cambridge, Mass)
- [17] A. L. Samuel (1959), Some Studies in Machine Learning Using the Game of Checkers, *IBM Journal of Research and Development*, **3** (210)
- [18] C. M. Bishop (1995), Training with Noise is Equivalent to Tikhonov Regularization, *Neural Computation*, **7**, 108
- [19] Z. Han, M. Hong, and D. Wang (2017), *Signal Processing and Networking for Big Data Applications*, 1st ed. (Cambridge University Press)
- [20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014), *Generative Adversarial Nets in Advances in Neural Information Processing Systems*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (Curran Associates, Inc.)
- [21] G. Hinton (2007), Boltzmann machine, *Scholarpedia*, **2** (1668)
- [22] J. J. Hopfield (1982), Neural networks and physical systems with emergent collective computational abilities., *Proceedings of the National Academy of Sciences*, **79**, 2554
- [23] Y. LeCun, Y. Bengio and G. Hinton (2015), Deep learning, *Nature*, **521** (436)



- [24] J. Schmidhuber (2015), Deep learning in neural networks: An overview, *Neural Networks*, **61**, 85
- [25] M. I. Jordan and T. M. Mitchell (2015), Machine learning: Trends, perspectives, and prospects, *Science*, **349**, 255
- [26] M. Dascal (1989), Artificial intelligence and philosophy: The knowledge of representation, *Systems Research*, **6**, 39
- [27] M. A. Boden, editor (1990), *The Philosophy of Artificial Intelligence* (Oxford University Press, Oxford [England] ; New York)
- [28] D. Andler (2023), *Intelligence artificielle, intelligence humaine: la double énigme* (Gallimard, Paris)
- [29] C. S. R. Silva and J. M. Fonseca (2019), Artificial Intelligence and Algorithms in Intelligent Systems: Advanced Analytics: Moving Forward Artificial Intelligence (AI), Algorithm Intelligent Systems (AIS) and General Impressions from the Field in Artificial Intelligence and Algorithms in Intelligent Systems, edited by R. Silhavy (Springer International Publishing, Cham), pp. 308–317
- [30] W. Bauer and C. Vocke (2020), Work in the Age of Artificial Intelligence – Challenges and Potentials for the Design of New Forms of Human-Machine Interaction in Advances in Human Factors, *Business Management and Leadership*, edited by J. I. Kantola and S. Nazir (Springer International Publishing, Cham), pp. 493–501
- [31] R. Bommasani, K. Klyman, S. Longpre, S. Kapoor, N. Maslej, B. Xiong, D. Zhang, and P. Liang (2023), The Foundation Model Transparency Index
- [32] F. Chollet, Deep Learning with Python (2021), 2nd ed (Manning Publications, Shelter Island)
- [33] D. Mendeleiev (1869), *La dépendance entre les propriétés des masses atomiques des éléments*
- [34] J. Gustafson, D. Rover, S. Elbert and M. Carter (1991), The design of a scalable, fixed-time computer benchmark, *Journal of Parallel and Distributed Computing*, **12** (388)
- [35]. P. Bachmann (1894), *Die Analytische Zahlentheorie, Zahlentheorie, Band 2* (B. G. Teubner, Leipzig)
- [36] Z. Bimurat, Y. Kim, R. Ismailova and B. Sagindykov (2023), Methods Of Navigating Algorithmic Complexity: Big-Oh And Small-Oh Notations, *Scientific Journal of Astana IT University*, 160
- [37] F. A. Mala and R. Ali (2022), The Big-O of Mathematics and Computer Science, *Journal of Applied Mathematics and Computation*, **6** (1)
- [38] S. Bae (2019), Big-O Notation in JavaScript Data Structures and Algorithms (Apress, Berkeley, CA), pp. 1–11
- [39] T. Hastie, R. Tibshirani, and J. H. Friedman (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. (Springer, New York)
- [40] R. S. Sutton and A. G. Barto (2018), *Reinforcement Learning: An Introduction*, Second edition (The MIT Press, Cambridge, Massachusetts)
- [41] M. R. B. Emdad (2017), Deep Learning is the Core Method of Machine Learning, *International Journal of Engineering Research and Technology*, **6**
- [42] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare and J. Pineau (2018), An Introduction to Deep Reinforcement Learning, *Foundations and Trends® in Machine Learning* **11** (219)
- [43] D. P. Kingma and M. Welling (2013), *Auto-Encoding Variational Bayes*
- [44] W. Reitman (1964), *Heuristic decision procedures, open constraints, and the structure of ill-defined problems in Human Judgments and Optimality* (John Wiley & Sons, New-York), pp. 282–315

- [45] D. Claeys (2013), Architecture & complexité : Un modèle systémique du processus de (co)conception qui vise l'architecture, PhD Thesis, université Catholique de Louvain
- [46] T. Oladipupo (2010), *Types of Machine Learning Algorithms in New Advances in Machine Learning*, edited by Y. Zhang (InTech)
- [47] C. M. Bishop (2006), *Pattern Recognition and Machine Learning* (Springer, New York)
- [48] K. P. Murphy (2022), *Probabilistic Machine Learning: An Introduction* (The MIT Press, Cambridge, Massachusetts)
- [49] F. Osisanwo, O. Awodele, J. Hinmikaiye, O. Olakanmi and A. J. Akinjobi (2017), Supervised Machine Learning Algorithms: Classification and Comparison, *International Journal of Computer Trends and Technology*, **48** (128)
- [50] G. E. Hinton and R. R. Salakhutdinov (2006), Reducing the Dimensionality of Data with Neural Networks, *Science*, **313** (504)
- [51] L. van der Maaten and G. Hinton (2008), Visualizing Data using t-SNE, *Journal of Machine Learning Research*, **9** (2579)
- [52] C. Cortes and V. Vapnik (1995), Support-vector networks, *Machine Learning*, **20** (273)
- [53] J. MacQueen (1967), Some methods for classification and analysis of multivariate observations, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* **1** (281)
- [54] W. Feller (1976), *An Introduction to Probability Theory and Its Applications*. **1**, 3rd ed., rev. print (Wiley, New York, NY)
- [55] D. C. Montgomery, E. A. Peck and G. G. Vining (2012), *Introduction to Linear Regression Analysis* (John Wiley & Sons)
- [56] I. Goodfellow, Y. Bengio and A. Courville (2016), *Deep Learning* (The MIT Press, Cambridge, Massachusetts)
- [57]. G. James, D. Witten, T. Hastie and R. Tibshirani (2013), editors, *An Introduction to Statistical Learning: With Applications in R* (Springer, New York)
- [58] S. Chaillou (2019), AI + Architecture | Towards a New Approach (Harvard University)
- [59] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov (2014), Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *Journal of Machine Learning Research* **15** (1929)
- [60]. B. Zhou, A. Khosla, A. Lapedriza, A. Oliva and A. Torralba (2016), Learning Deep Features for Discriminative Localization, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2921–2929
- [61] E. F. Camacho and C. Bordons (2007), *Simple Implementation of GPC for Industrial Processes in Model Predictive Control* (Springer, London), pp. 81–126
- [62] T. Chai and R. R. Draxler (2014), Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature, *Geoscientific Model Development*, **7** (1247)
- [63] S. V. Stehman (1997), Selecting and interpreting measures of thematic classification accuracy, *Remote Sensing of Environment*, **62** (77)
- [64] J. Louëdec, M. Chevalier, A. Garivier and J. Mothe (2015), Algorithmes de bandits pour la recommandation à tirages multiples, *Document numérique*, **18** (59)
- [65] P. R. Rosenbaum and D. B. Rubin (1983), The central role of the propensity score in observational studies for causal effects, *Biometrika*, **70** (41)