Contribution à l' analyse de causalité par apprentissage automatique pour l'aide à la décision, dans un contexte de supervision des systèmes pour l'industrie 4.0

Kenza Amzil

HESAM UNIVERSITÉ

Arts et Métiers Sciences et Technologies

École doctorale Sciences et Métiers de l'Ingénieur

LISPEN - Campus d'Aix en Provence

THÈSE

présentée par : **Kenza AMZIL**

soutenue le : **03 Janvier 2022**

pour obtenir le grade de : **Docteur d'HESAM Université**

préparée à : **École Nationale Supérieure d'Arts et Métiers**

Discipline : **Section CNU 61**

Contribution à l'analyse de la causalité par apprentissage automatique pour l'aide à la décision, dans un contexte de supervision pour l'industrie 4.0

Thèse dirigée par :
Lionel ROUCOULES

et co-encadrée par :
Esma Yahia et Nathalie Klement

Jury

M. Marc ZOLGHADRI	Professeur des Universités, LISMMA, ISAE-Supméca	Rapporteur
Mme. Elise VAREILLES	Professeur des Universités, ISAE-Supaéro	Rapportrice
M. Vincent CHEUTET	Professeur des Universités, DISP-lab, INSA de Lyon	Président
M. Lionel ROUCOULES	Professeur des Universités, LISPEN, ENSAM Aix en Provence	Examinateur
Mme. Esma YAHIA	Maître de Conférences, LISPEN, ENSAM Aix en Provence	Examinatrice
Mme. Nathalie KLEMENT	Maître de Conférences, LISPEN, ENSAM Lille	Examinatrice

THÈSE

"I would rather discover a single causal connection, than win the throne of Persia."

Democritus (460 av. J.-C. — 370 av. J.-C.)

Remerciements

C'est par conviction très profonde, et non par convenance, que je souhaite, avant tout, remercier très chaleureusement l'équipe qui m'a suivie pendant ces trois années de thèse. Je témoigne alors ma plus grande gratitude et toute ma reconnaissance à mon directeur de thèse, Lionel Roucoules. Merci pour ta présence et ton soutien indéfectibles. Merci pour tes idées, remarques, et conseils toujours très avisés. Merci pour ton ouverture intellectuelle, pour ta bonne humeur contagieuse, et pour ton expérience et ton savoir que tu partages toujours dans la plus grande générosité. J'adresse également un très grand merci à Esma Yahia, la co-encadrante de cette thèse. Merci à toi pour ta grande disponibilité inconditionnelle, ton écoute, et tes remarques qui m'ont été très précieuses. Merci d'avoir toujours su me tendre la main, tant sur le plan scientifique que psychologique. Je te remercie pour ton humilité, ta patience, ta tolérance, et toutes tes qualités humaines que je ne saurai faire tenir en quelques lignes. Ma reconnaissance va également à Nathalie Klément, qui a co-encadré cette thèse depuis le campus Arts et Métiers de Lille. Malgré la distance et les aléas qui auraient pu en découler, tu t'es toujours rendue disponible, tu as toujours été présente, et je t'en remercie très sincèrement. Merci à toi pour ta bienveillance, tes conseils, et les encouragements que tu as su me témoigner dans les moments difficiles. Je te remercie également d'avoir parcouru toute la France pour être présente à mes côtés le jour de ma soutenance. Que mes remerciements, à vous trois, puissent témoigner de la dette que vous doit l'aboutissement de ma thèse : ce travail vous doit tout dans ses points forts, et pas assez dans ses faiblesses et limites. Sans vous, il n'aurait jamais pu voir le jour. Le seul *lien de causalité* que je peux donc affirmer avec certitude dans ce manuscrit, est que vous avez été *la cause commune* de l'aboutissement de cette thèse d'une part, et du goût et de la passion que j'ai développé pour la recherche, d'autre part.

Mes sincères remerciements s'adressent aussi au professeur Vincent Cheutet. D'abord pour m'avoir fait l'honneur de présider mon jury de thèse, mais également pour avoir fait partie de mon comité de suivi de thèse pendant les deux premières années, et pour m'avoir donné des conseils qui m'ont été d'une grande aide. Je tiens également à remercier très sincèrement les professeurs Marc Zolghadri et Elise Vareilles, pour avoir accepté de rapporter ce travail de thèse. Je vous remercie pour le temps que vous avez pris pour relire ce manuscrit, et pour vos remarques aussi judicieuses qu'enrichissantes.

Je remercie également tous mes collègues du LISPEN, doctorants, enseignants-chercheurs, ingénieurs, et techniciens, pour m'avoir permis de bien m'intégrer, et pour la très bonne ambiance qu'il font régner au sein du laboratoire.

Enfin, mais non des moindres, il va sans dire que je dirige toute ma gratitude à mes très chers parents Souad et Khalid, sans lesquels je ne serai jamais parvenue au bout de cette entreprise, ni même à son début. Je les remercie pour leur soutien sur tous les plans,

leur accompagnement, leur amour inconditionnel, leurs questions, leurs conseils, et leurs paroles apaisantes, consolantes, réconfortantes, et encourageantes. Je remercie également mes deux sœurs Fatima-Ezahra et Meryem, pour leur écoute, leur présence à mes côtés, leur compréhension, et leur patience. Merci également à la petite Maria, ma nièce qui a illuminé mes jours. Enfin, je dédie cette thèse aux âmes de mes grands parents que je porterai à jamais dans mon cœur, et auxquels j'ai pensé très profondément dans chaque moment fort de cette thèse.

Kenza Amzil

Table des matières

Table des figures

Liste des acronymes et abréviations

AMDEC : Analyse des modes de défaillances, de leurs effets et de leurs criticités

AIC : Akaike Information Criterion

BD : Bayesian Dirichlet

BDe : Bayesian Dirichlet Equivalent

BIC : Bayesian Information Criterion

CI : Causal Inference

CPDAG : Graphe acyclique partiellement dirigé complété

CPS : Système Cyber-Physique

DAG : Diagramme Acyclique Dirigé

D-séparation : Ensemble de règles permettant d'identifier les indépendances conditionnelles à partir d'un graphe

FCI : Fast Causal Inference

HC : Iterated Hill Climbing

IC : Inductive Causation

IHC : Iterated Hill Climbing

IoT : Internet des Objets

KPI : Indicateur clé de performance

LBFGS : Broyden-Fletcher-Goldfarb-Shanno à mémoire limitée

MDL : Minimum description length

MLP : Perceptron Multi-couches

MWST : Maximum Weighted Spanning Tree

NP : Non déterministe polynomial

PDAG : Diagramme acyclique partiellement dirigé

QQOQCP : Quoi, Qui, Où, Quand, Comment, Pourquoi

ReLU : Unité de rectification linéaire

SADT : Technique d'analyse fonctionnelle descendante et de conception structurée

SGD : Descente de gradient stochastique

SGS : Sparse Graph Search

TRS : Taux de Rendement Synthétique

V-structure : Connexion convergente de deux causes vers un effet commun

5P : Les cinq pourquoi

Introduction générale

De nos jours, l'industrie, comme de nombreux autres secteurs, est confrontée à une compétitivité croissante, en raison de l'évolution des marchés et de la mondialisation. Cela conduit à un besoin impérieux pour les entreprises d'améliorer leurs performances, que ce soit en réponse à des enjeux économiques, écologiques, ou sociétaux et éthiques. Dans le but de répondre à ces besoins, les décideurs sont contraints de déterminer les objectifs industriels souhaités et de les surveiller en permanence. Ceci leur permet, entre autres, de pouvoir fournir des décisions profitables à la résolution du problème qu'ils rencontrent ou à l'opportunité d'amélioration. Pour ce faire, les indicateurs clés de performance (KPIs) sont largement utilisés. Les KPIs servent de moyen de contrôle qui quantifie l'efficacité et l'efficience des processus et des actions engagées, ainsi que les performances des produits, la qualité ou la réalisation des objectifs commerciaux (Yin, Zhu, & Kaynak, 2015 ; Neely, Gregory, & Platts, 2005). Par conséquent, le suivi des KPIs facilite la détection des déviations et des évolutions inattendues de l'entité surveillée (Laudon & Laudon, 2019). Ainsi, ils peuvent être considérés comme étant aussi bien des déclencheurs que des pilotes des processus décisionnels. Ceci nous persuade que pour améliorer la prise de décision, une attention particulière devrait être portée sur les KPIs. En effet, lorsqu'un KPI révèle une situation anormale, la compréhension de l'origine de cette déviation est indispensable pour rechercher des solutions, et pour en sélectionner une parmi plusieurs.

Les travaux de cette thèse considèrent donc la question de l'amélioration des processus décisionnels par le biais de l'amélioration de l'analyse des KPIs. Plus précisément, nous cherchons à déceler les liens de causalité que les KPIs peuvent entretenir avec les entités du contexte dans lequel ils évoluent, de manière à améliorer la prise de décision sur deux axes : la pertinence de la décision ; et la durée associée aux processus décisionnels. Pour ce faire, nous proposons de faire une analyse de la causalité, basée sur les données, afin de réduire la subjectivité et les biais cognitifs liés à l'expérience de l'expert. Nous proposons par ailleurs de construire des modèles de prédiction afin d'anticiper les éventuelles déviations des KPIs d'intérêt, et permettre ainsi une certaine proactivité.

Ainsi, la méthode que nous proposons doit répondre à trois fonctions : l'identification des variables contextuelles liées causalement à un KPI, sous forme structure causale ; la hiérarchisation de ces variables selon leurs forces respectives d'association au KPI d'intérêt ; et la possibilité de prédiction du KPI pour des fins de proactivité.

Ce manuscrit de thèse est sectionné en cinq chapitres, selon l'organisation suivante :

— Le chapitre 1 introduit le contexte général, et décrit la réflexion qui nous a mené à définir la problématique à laquelle nous souhaitons répondre. Cette problématique est ensuite détaillée, et les objectifs fixés pour y répondre sont identifiés. Ils sont exprimés sous forme de trois fonctions principales. Les hypothèses globales de la

proposition sont également énoncées dans ce chapitre ;

— Le chapitre 2 fait un état de l'art réparti en trois parties, portant chacune sur une des trois fonctions principales. La première partie, destinée à l'analyse causale, expose la difficulté de donner une définition exacte et exploitable de la notion de causalité, et liste les différentes caractéristiques qui distinguent un lien causal entre deux entités. Cette première partie passe ensuite en revue les différentes techniques et méthodes utilisées pour l'analyse causale, et nous permet de faire un choix parmi ces techniques. Les deux autres parties concernent la construction automatique des réseaux de neurones qui permettront la prédiction des déviations des entités surveillées, et motivent les choix que nous avons fait pour répondre à la fonction de hiérarchisation des causes ;

— Le chapitre 3 a pour objectif d'introduire progressivement l'architecture de la méthode que nous proposons, brique par brique, à travers un exemple d'illustration. Il a également pour objectif de présenter les entrées et sorties des briques, et la manière dont elles interagissent les unes avec les autres ;

— Le chapitre 4 décrit le développement détaillé de chaque brique de l'architecture de la proposition. Ce chapitre est réparti en trois partie, dont chacune détaille les choix et les développements effectués pour répondre à l'une des trois fonctions ;

— Le chapitre 5 présente l'application de la méthode proposée sur deux cas d'études étalons différents. Ce chapitre discute également des performances de la méthode proposée, grâce à une comparaison entre les résultats de la méthode et les étalons, et grâce à une comparaison entre les résultats de la méthode et ceux d'autres techniques ayant les mêmes objectifs que les nôtres. Ce chapitre vérifie également la correspondance des résultats issus de la méthode, aux caractéristiques d'évaluation définies dans l'état de l'art ;

— Une conclusion générale résume les constats importants établis tout au long du manuscrit. Les perspectives identifiées au terme de cette thèse viennent enfin proposer quelques pistes d'amélioration.

Chapitre 1

Contexte, problématique, et objectifs

1.1 Contexte

1.1.1 Histoire, enjeux, et pilotes de la révolution industrielle

Au vu des liens étroits qui existent entre l'industrie, les progrès technologiques, et l'évolution des modes de consommation, le secteur industriel a connu de nombreux changements afin de répondre aux nouveaux besoins, tout en tirant profit des nouveaux développements technologiques. Les grandes transformations qui ont significativement bouleversé la façon de produire, sont communément appelées "révolutions industrielles". Depuis le milieu du $XVIII^{eme}$ siècle, quatre révolutions industrielles se sont succédées et complétées. La première révolution a émané de l'invention de la machine à vapeur et a permis la mécanisation de la production : ce fut le passage de l'artisanat à l'industrie. A la fin du XIX^{eme} siècle, l'invention de l'électricité et son utilisation massive au sein des usines ont conduit à la deuxième révolution industrielle. L'automatisation a ensuite profondément transformé les méthodes de production grâce à l'apparition de l'électronique et de l'informatique dans les années 1960, ce qui entraîna la troisième révolution industrielle. Jusqu'alors, la standardisation et la production de masse ont été à la fois les pilotes et la finalité de ces nouveaux paradigmes (Freeman & Louça, 2001).

Depuis le début des années 2010, l'urgence pour le secteur industriel de s'adapter aux changements économiques et sociétaux se fait de plus en plus menaçante. Ces changements tirent leur origine de la mondialisation de plus en plus importante, et de l'avènement des technologies dites nouvelles (Gamache, Abdulnour, & Baril, 2019). Dans un monde où nous sommes quotidiennement noyés dans le numérique, les entreprises doivent faire face à un consommateur mieux informé et plus exigeant. Outre les exigences de personnalisation de masse, le consommateur a de nombreuses autres attentes. En effet, au cours des dernières années, les modes de consommation ont considérablement évolué sur différents aspects, comme le montre une enquête effectuée sur un échantillon représentatif de 1000 personnes majeures (Pouget, Caylou, & Doignies, 2019) :

— 79% des personnes interrogées cherchent à consommer de manière plus économique, ceci se traduit par un attrait de plus en plus prononcé pour les promotions et les petits prix ;

— 66% se soucient de consommer de manière plus écologique. Ceci ne se manifeste pas seulement par une consommation plus durable et avec moins de gaspillage, mais également par la consommation de produits écologiques, notamment en fonction de

leur impact sur l'environnement. Ceci concerne aussi bien la phase d'utilisation que celle de la production ;

— 53% consomment de manière plus qualitative. Ceci implique, entre autres, un attrait vers les produits labellisés, des penchants pour les marques de renoms, ainsi qu'une consultation presque systématique et non sans conséquences, des avis laissés par les clients ayant testé le produit ou le service ;

— 63% se préoccupent de plus en plus du volet éthique et de la responsabilité sociale. En effet, les consommateurs sont de plus en plus regardants sur les conditions de travail et le respect du bien-être des travailleurs.

Face à tous ces éléments, les entreprises se voient contraintes d'accompagner cette évolution afin de pouvoir continuer à séduire leurs clients potentiels. Afin de s'aligner avec les besoins de leurs clients, les entreprises sont alors confrontées, entre autres, à la nécessité d'accroître leur efficacité opérationnelle, de manière à réduire les coûts, les délais, et l'impact environnemental, tout en innovant et en créant de la valeur. À cet égard, les entreprises industrielles réalisent de plus en plus l'intérêt d'exploiter les nouvelles technologies telles que les systèmes cyber-physiques (CPS), l'impression 3D, l'Internet des objets (IoT), ou encore le Big Data (Cagnetti, Gallo, Silvestri, & Ruggieri, 2021). La convergence de l'ensemble de ces technologies pour l'amélioration de la production se désigne aujourd'hui par le terme *"Industrie 4.0"*, en référence aux trois révolutions antécédentes. Ce terme est apparu pour la première fois en Allemagne, lors de la Foire annuelle de Hanovre de 2011, et défini comme *"l'intégration des CPS dans la fabrication et la logistique, ainsi que l'usage des IoT dans les processus industriels, dans le but d'en voir des répercussions sur la création de la valeur, l'organisation du travail, les modèles économiques, et les services en aval"* (Kagermann, Wahlster, & Helbig, 2013).

L'intérêt que portent les industriels à cette révolution se constate à partir des chiffres relatifs aux investissements entrepris, ou futurs, dans ces différentes technologies. La figure 1.1 illustre les résultats d'une enquête menée auprès d'une centaine d'industriels de différents secteurs, dont l'automobile, l'industrie pharmaceutique, l'énergétique, les industries des biens de consommation et des biens d'équipements, la chimie, l'aéronautique, et le ferroviaire (Marie & Poindextre, 2020).

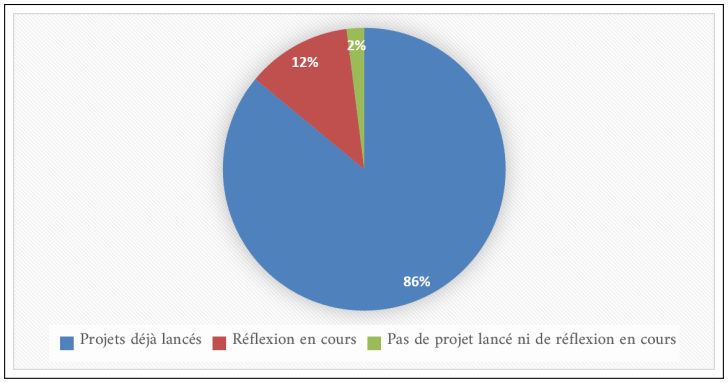


FIGURE 1.1 – Statistiques sur les projets de transformation digitale en industrie. (Adapté de (Marie & Poindextre, 2020)).

Le bilan montre qu'en 2020, 86% de ces industriels ont déjà commencé à investir dans

un ou plusieurs aspects de la transformation digitale (66% avec bénéfices constatés, et 20% n'ayant toujours pas constaté de résultats). Par ailleurs, 12% considèrent l'intérêt d'un tel investissement et sont en cours de réflexion. Cette transformation digitale se traduit, entre autres, par la mise en place d'IoT et par l'exploitation des données qu'ils génèrent.

✓ *Les industriels se dirigent de plus en plus vers l'exploitation des technologies de l'industrie 4.0 au vu des bénéfices qu'elles promettent.*

1.1.2 L'exploitation des technologies de l'industrie 4.0 pour l'amélioration de la production

Outre le besoin de faire évoluer les moyens de production pour pouvoir proposer des produits personnalisés en masse, l'engouement pour l'industrie 4.0 est surtout motivé par l'amélioration de la production, et les nombreux avantages qui en découlent. L'excellence opérationnelle, et l'amélioration des processus, représentent des objectifs prioritaires et communs à toutes les entreprises ayant investi, ou souhaitant se lancer, dans un projet de transformation digitale (Verriere-cuenot & De Noinville, 2019). Les industriels sont à la quête d'agilité, de proactivité, de qualité, et d'efficience (Masood & Sonntag, 2020). Ces attentes se traduisent par plusieurs objectifs, que nous pouvons répartir, comme le montre la figure 1.2, sur trois axes : les enjeux économiques, les enjeux écologiques, et les enjeux sociétaux et éthiques.

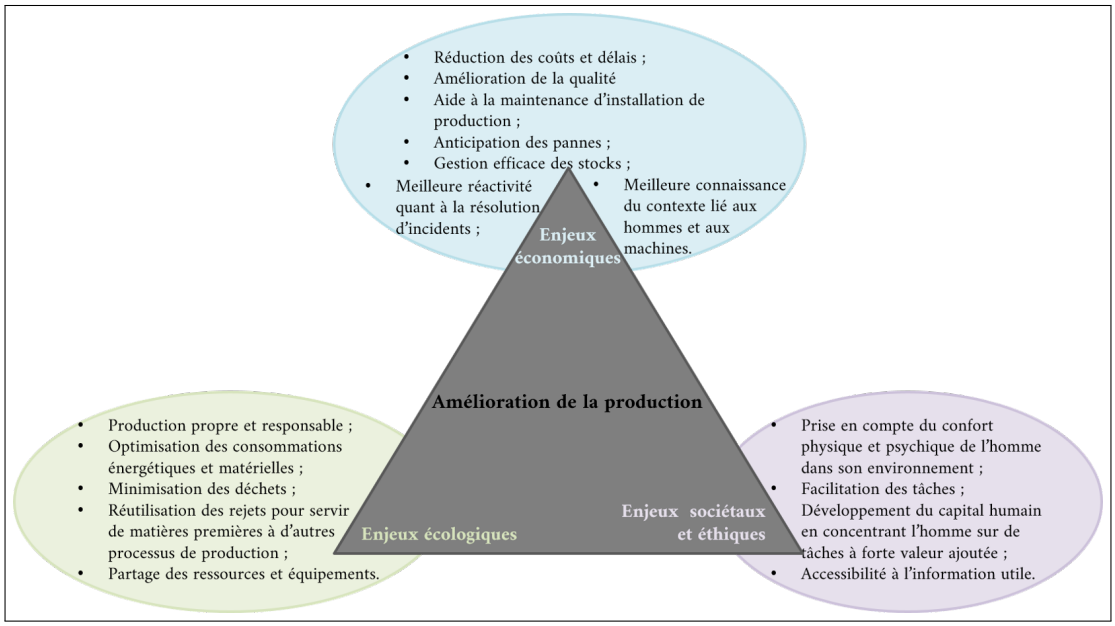


FIGURE 1.2 – Répartition des objectifs attendus de l'industrie 4.0 pour l'amélioration de la production.

✓ *L'amélioration de la production, et des conditions de production, représente un facteur déterminant incitant les industriels à adopter les nouvelles technologies.*

1.1.3 La prise de décision au cœur de l'amélioration de la production

La réponse aux ambitions liées à l'amélioration de la production passent par de nombreux moyens. En effet, les nouvelles technologies, et la modernisation des moyens de productions grâce à la robotisation, facilitent plusieurs tâches, et permettent un gain de temps et une optimisation de ressources considérables (Kiel, Müller, Arnold, & Voigt, 2017). Cependant, la prise de décision demeure la pierre angulaire qui régit tous les résultats. C'est le dénominateur commun à l'aboutissement de tous les objectifs précédemment cités (Bousdekis, Lepenioti, Apostolou, & Mentzas, 2021). La prise de décision, est retrouvée comme élément majeur, composant ou résultant, de plusieurs concepts du paradigme de l'industrie 4.0, comme les CPS, la simulation, ou l'analyse des données.

La décision est définie par le dictionnaire *L'internaute* comme "un processus cognitif complexe consistant à faire un choix d'action parmi plusieurs alternatives. Le choix final résulte de ce processus, et ce choix peut être une opinion ou une action". Dans *Le Robert*, une décision est un "jugement qui apporte une solution". En neurosciences, la décision est un processus mêlant à la fois le conscient et l'inconscient, qui consiste à envisager, face à un problème, différentes alternatives possibles, puis à faire un choix suite à une réflexion (Allain, 2013). Dans ce sens, l'inconscient peut intervenir dans deux phases : la réalisation de la nécessité de prendre une décision, et l'imagination de solutions possibles, qui est conditionnée par l'expérience du décideur et ses ressentiments ou intuitions. Nous distinguons trois types de décisions : les décisions en situation de certitude, où le décideur est certain de la conséquence qu'engendrera son choix ; les décisions en situation d'incertitude, où le décideur ne connait pas la probabilité d'arriver au résultat escompté suite à son action ; et les décisions à risque, où la conséquence n'est pas certaine, mais le décideur connaît la probabilité de chacune des conséquences possibles (Allain, 2013). En ce qui concerne les situations problématiques auxquelles les décideurs font face en entreprise, il s'agit de situations incertaines ou à risque. Dans la suite, nous entendons donc par *décision*, le processus ayant pour objectif de sélectionner une solution parmi plusieurs possibles, en situation incertaine et/ou à risque, afin de résoudre un problème ou d'améliorer une situation. À cet égard, nous considérons que la résolution de problèmes représente une situation particulière de prise de décision, compte tenu du fait que le besoin de prendre une décision peut également émaner d'une simple volonté d'améliorer une situation, ou de saisir ou non une opportunité.

L'aide à la décision est un domaine auquel les entreprises s'intéressent de plus en plus. Bien que la décision représente la finalité de plusieurs applications dans le cadre de l'industrie 4.0, il n'en reste pas moins que les applications qui impliquent une prise de décision ou une aide à la décision sont, pour la majorité, limitées à des décisions locales sur un système ou un équipement, et ne prennent pas compte de tous les éléments du processus de production à la fois (Hoffmann Souza, da Costa, de Oliveira Ramos, & da Rosa Righi, 2020).

Que ce soit d'un point de vue économique, écologique, social, ou éthique, la prise de décision s'impose et conditionne l'achèvement des objectifs à atteindre. Ceci est valable pour tous les niveaux : opérationnel, tactique, et stratégique (Pillet, 2004).

✓ *L'amélioration de la production, dans tous ses aspects, est conditionnée par l'amélioration de la prise de décision.*

1.1.4 La mesure de la performance, pilote de la décision

La relation entre les décisions et la mesure des performances peut être décrite comme interactive : d'un côté la prise de décision a une influence sur les performances, d'un autre côté, la mesure des performances est à la fois un déclencheur et un pilote de la prise de décision. Afin d'évaluer leurs performances et d'assurer un suivi de leurs activités, les entreprises ont recours à un ensemble d'indicateurs clés de performances (KPIs). Les KPIs sont des métriques d'importance stratégique pour mesurer les performances spécifiques à une activité (production, chaîne logistique, ventes, etc.), ou générales à l'ensemble de l'entreprise (Liebetruth, 2017). Ils servent à quantifier l'efficacité d'une action (Liebetruth, 2017). Relativement à un KPI, une donnée peut être perçue comme étant un petit bloc de construction, parmi d'autres, pour le KPI (Rennell, 2021). Dans la suite, nous entendons par KPI, une mesure utilisée pour suivre le déroulement d'une activité ou d'un processus, et évaluer le succès d'une activité ou l'efficacité d'une action, en référence à un objectif préalablement fixé. Ceci peut concerner aussi bien le niveau opérationnel que le niveau tactique ou stratégique.

Les KPIs font l'objet de *reporting*, afin de surveiller et analyser les déroulements des activités. Ceci implique deux éléments : **la prise de conscience de la nécessité d'agir, et l'analyse des KPIs pour rechercher et sélectionner les solutions**. Afin de lever toute ambiguïté, nous donnons ci-dessous les définissions de "surveillance" et "supervision", selon lesquelles ces termes seront utilisés dans la suite :

— La surveillance : ou monitoring, permet de collecter les données d'un processus et/ou système, et de déterminer l'état actuel du processus ou système contrôlé (Combacau, Berruet, Zamai, Charbonnaud, & Khatab, 2000). Il permet donc d'informer la personne qui surveille, afin qu'elle puisse décider en fonction des informations dont elle dispose (Grislin, 1995).

— La supervision : doit offrir des fonctions pouvant avoir des incidences sur le déroulement des processus et/ou sur les systèmes surveillés, comme le paramétrage, le calcul, la replanification, l'optimisation, la redéfinition, en fonction de l'état actuel du processus et/ou système, et ce dans les cas de fonctionnement normal ou anormal (Combacau et al., 2000). Un système de supervision doit également être capable de reconnaître les situations anormales et de les signaler.

Par rapport à ce qu'implique l'exploitation des KPIs, la prise de conscience de la nécessité d'agir est liée à la surveillance, tandis que l'analyse des KPIs pour rechercher et sélectionner les solutions est liée à la supervision. Fournir des informations sur les performances est essentiel, mais demeure insuffisant pour améliorer les résultats. La plus value réside dans l'interprétation des performances (Nudurupati, Arshad, & Turner, 2007). La pertinence des décisions dépend alors, entre autres, de la manière dont les KPIs sont exploités. Par ailleurs, une étude effectuée auprès de 21 petites et moyennes entreprises soulève que les outils d'aide à la décision basée sur les KPIs sont jugés essentiels dans les stratégies de mise en œuvre de l'industrie 4.0 (Gamache, 2019). Selon la même étude, ils

offriraient un gain potentiel de performances d'au moins 20%.

✓ *L'amélioration de la prise de décision est liée aux deux aspects de l'exploitation des KPIs : la prise de conscience de la nécessité d'agir, et l'analyse des KPIs pour rechercher et sélectionner les solutions.*

1.1.5 Question de recherche de premier niveau

Notre question de recherche de premier niveau émane du raisonnement déductif que nous avons présenté dans cette section. Notre besoin a été constaté partant du général au spécifique. Le point de départ représente un constat sur la nécessité et l'intérêt que montrent les entreprises à l'égard des technologies de l'industrie 4.0. Nous en avons ensuite déduit que cet engouement émane de la nécessité d'améliorer la production et ses conditions sur différents aspects. Nous avons établi que cette amélioration passe par l'amélioration de la prise de décision, et que cette dernière est liée à l'amélioration de la manière d'exploiter les KPIs. La figure 1.3 illustre la réflexion ayant mené à notre question de recherche de premier niveau : "Comment améliorer les deux aspects de l'exploitation des KPIs : l'anticipation de la prise de conscience de la nécessité d'agir et l'analyse des KPIs pour bien agir ?"

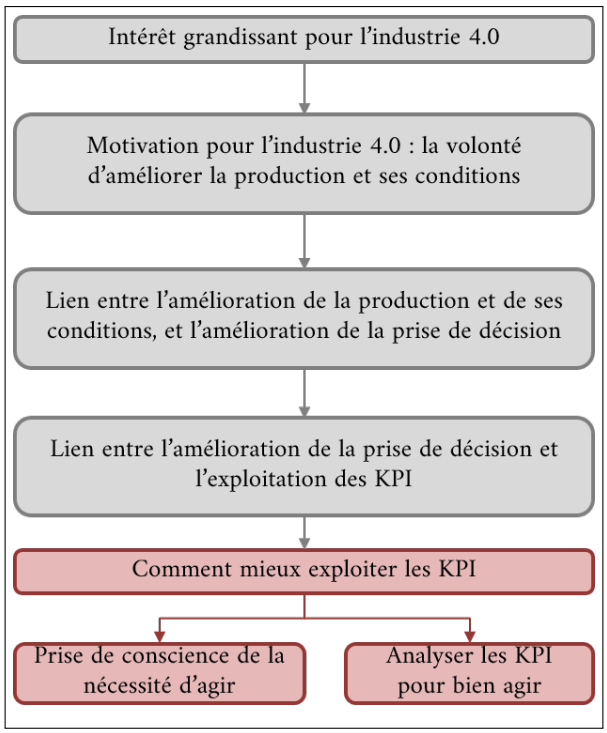


FIGURE 1.3 – Illustration de la réflexion ayant mené à la question de recherche de premier niveau.

1.2 Problématique

1.2.1 Analyse causale sur les KPIs

Comme mentionné dans la section précédente, la valeur ajoutée de l'exploitation des KPIs réside dans l'action qui découle de leurs observations. Cette action est le résultat d'une décision. Or, une décision "consiste à **établir des liens de cause à effet** entre les événements, à les évaluer, et à trouver une solution satisfaisante" (Benoît, 2011). Lorsqu'un KPI indique une déviation de l'objectif qu'il devait atteindre, la difficulté de l'interprétation de cette mesure réside dans la recherche des éléments ayant causé la déviation. Il s'agit dans ce cas là d'une situation problématique, et dans toute démarche de résolution de problèmes, la recherche de causes est impérative. Dans un cas plus général, où il faut prendre une décision sans qu'il y ait forcément de problème à résoudre, la connaissance des liens des causes à effets représente un avantage de taille qui augmente considérablement les chances d'aller dans le bon sens. En effet, l'efficacité de toute action dépend de la connaissance des causes (Drouet, 2007).

Plusieurs recherches soulignent que l'analyse des valeurs des KPIs et l'identification de leurs facteurs d'influence, sont souvent réalisées par des experts de manière empirique et descriptive, ne permettant pas d'identifier de manière exhaustive et exacte toutes les causes directes et indirectes de chaque déviation (Pérez-Alvarez, Maté, Gó mez López, & Trujillo, 2018 ; Georgakopoulos, Jayaraman, Fazia, Villari, & Ranjan, 2016 ; Moeuf, Pellerin, Lamouri, Tamayo-Giraldo, & Barbaray, 2018 ; Nudurupati et al., 2007). Les méthodes classiques de recherche de causes, comme les cinq pourquoi, le diagramme d'Ishikawa, le QQOQCP, ou l'analyse des modes de défaillance, de leurs effets et de leur criticité (AMDEC), pour ne citer qu'eux, sont des méthodes descriptives présentant deux inconvénients. Le premier est relatif aux ressources humaines et temporelles que mobilisent ces démarches, puisqu'elles nécessitent, le plus souvent, un travail d'équipe pouvant s'étendre sur une durée non négligeable. Le deuxième inconvénient est lié à la subjectivité pouvant biaiser la recherche des causes lors de l'interprétation des valeurs des KPIs, ainsi qu'à la non exhaustivité entraînant plutôt un déplacement du problème que sa résolution. En effet, l'expérience des décideurs, peut se transformer en inhibiteur lors de leurs processus décisionnels (Allain, 2013). D'abord lorsqu'ils rencontrent une situation similaire mais pas identique à une autre ayant déjà été résolue, de fausses analogies peuvent être déduites et entraver la décision. Ensuite, lors du déroulement de ces démarches, il a été établi que les participants ont tendance à commencer la recherche de solutions avant d'avoir identifié les causes racines du problème (Duret & Pillet, 2011).

Dans le contexte de l'industrie 4.0, les différentes technologies utilisées génèrent une large quantité de données. Ces données peuvent provenir des systèmes d'acquisition et concerner les produits, les systèmes, et les éléments composant l'environnement de production, comme elles peuvent provenir des systèmes d'informations. Dans la suite, on désignera l'ensemble de ces données par "*données contextuelles*". L'exploitation de ces données et leur conversion en informations utiles impactent positivement les performances des entreprises (Bordeleau, Mosconi, & Santa-Eulalia, 2018). Dans notre contexte, nous voulons exploiter les données disponibles et les KPIs afin pouvoir faire ressortir les relations causales qui les régissent. L'exploitation de ces données passe obligatoirement par une analyse de données (Bordeleau et al., 2018).

Par ailleurs, il n'est pas sans rappeler que l'objectif derrière l'identification des liens causaux entre le KPI étudié et les données contextuelles disponibles, est bien de pouvoir bien agir. Cette action portera sans doute sur une ou plusieurs causes identifiées. La connaissance de ces liens de causalité permettra de restreindre le nombre d'actions efficaces possibles. Néanmoins, en admettant que toutes les relations causales concernant le KPI étudié aient bien été identifiées, le besoin de les hiérarchiser devient alors fondamental afin de pouvoir faire un choix parmi les alternatives possibles.

Ceci nous ramène alors à notre question de recherche de deuxième niveau, à laquelle nous tenterons de proposer une réponse : **"Comment capturer les relations causales entre les KPIs et les données contextuelles, et les hiérarchiser selon la force de la liaison cause à effet ?"**

1.2.2 Prise de conscience du besoin d'agir

Avant de passer à l'action, le processus décisionnel devant y mener nécessite un certain temps. Cette durée a pour point de départ, le moment de la prise de conscience de la nécessité d'agir, qui se situe à l'origine de déclenchement du processus décisionnel. Le point d'arrivée est, dans notre contexte, représenté par le moment où le choix de la solution est effectué. Si l'on se place dans un cas de résolution de problème, une durée trop importante de prise de décision peut avoir des conséquences considérables sur le déroulement des activités, et entraîner de sérieuses pertes. Dans une telle situation, et outre les conséquences directes et concrètes qu'une longue durée du processus décisionnel peut entraîner, des conséquences indirectes sont aussi à prendre en compte . En effet, la pression du temps, dont le décideur est bien souvent conscient, engendre un stress pouvant avoir plusieurs effets négatifs sur la prise de décisions : sous stress temporel, les ressources mentales peuvent être réduites et des comportements néfastes à la bonne décision ont été remarqués, comme des erreurs d'évaluation, des dénis d'informations importantes, etc (Ariely & Zakan, 2001).

La figure 1.4 représente un problème qui se manifeste par une déviation par rapport à un niveau attendu. Sur la figure, nous pouvons apercevoir que la réduction du temps de retour à la situation nominale est dépendante de : la diminution du décalage entre l'apparition et la perception du problème, la réduction de la durée de la prise de décision, et la rapidité du déroulement de l'action.

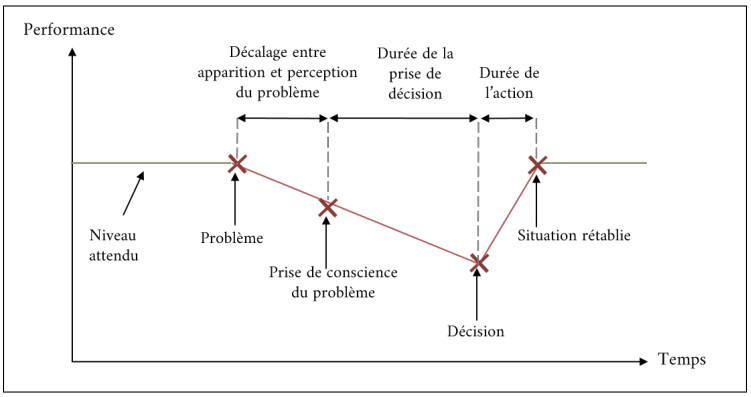


FIGURE 1.4 – Visualisation des étapes pour revenir à la situation normal suite à une déviation.

Pour ce qui est de la durée de la prise de décision, nous avons établi, dans la section précédente, que l'analyse de causalité à partir des données réduirait considérablement le temps nécessaire à la recherche des solutions possibles, et que la hiérarchisation des liens causaux diminuerait le temps de sélection d'une alternative. Ce qui entraînera systématiquement une diminution du temps de la prise de décision.

Le décalage entre l'apparition et la perception du problème est inévitable si l'on est pas proactif. En effet, il serait judicieux d'anticiper l'arrivée des déviations, afin de pouvoir agir le plus tôt possible. Ceci permettra au meilleur des cas d'éviter complètement la déviation, et au pire des cas de gagner du temps, à condition, bien sûr, que l'alerte de la déviation ne soit pas intempestive. Ceci revient donc à dire que pour réduire encore plus le temps de retour à la situation nominale, il serait préférable de prédire les valeurs des KPIs à surveiller.

Concernant la durée entre la prise de décision et le retour à la situation normale, elle dépend surtout de la nature et de la difficulté de l'action à entreprendre. Elle dépend donc de la sortie du processus décisionnel (*i.e.* la solution choisie), mais n'affecte pas le processus de prise de décision pas en soi.

1.3 Objectifs

1.3.1 Objectif et fonctions principales du travail de recherche

Cette thèse s'inscrit donc dans le contexte de l'industrie 4.0, et vise à répondre au besoin de l'amélioration de la production et de ses conditions, à travers l'amélioration de la prise de décision et la réduction de ses délais. À partir des questions de recherche, et des besoins identifiés dans les sections précédentes, nous définissons ici l'objectif principal de cette recherche, qui est de : **proposer une approche d'analyse causale visant à améliorer la prise de décision et à réduire la durée qu'elle prend, en exploitant à la fois les KPIs mesurés et les données contextuelles**. Afin que cet objectif global puisse être atteint, il a été décliné sous forme des fonctions suivantes, auxquelles la méthode proposée devrait répondre :

— F1 : Identifier, à partir des données, les relations causales entre un KPI d'intérêt, et les données contextuelles, ainsi que les autres indicateurs de performances ;
— F2 : Hiérarchiser les relations causales identifiées pour un KPI donné, selon la force de la relation, afin de prioriser les actions à mener pour améliorer l'indicateur étudié ;
— F3 : Prédire les valeurs des indicateurs clé de performance, et alerter des déviations.

1.3.2 Postulats de travail

Dans la suite, nous nous appuyons sur les postulats de travail suivants, que nous considérons comme étant remplis :

— P1 : Les systèmes d'acquisition des données sont disponibles et déjà mis en place. Ce postulat est étayé par la constatation de l'augmentation considérable de l'utilisation

des IoT, et la continuelle diminution de leurs coûts (Kamienski et al., 2019 ; Marie & Poindextre, 2020) ;

— P2 : Puisque nous allons effectuer un apprentissage à partir des données contextuelles et les anciennes mesures des KPIs, nous postulons que les données contextuelles historiques et les mesures historiques des KPIs, qui nous permettrons d'apprendre à partir des situations antérieures, sont disponibles. Nous soulignons que le quantité de données historiques nécessaire est positivement corrélée au nombre de variables que nous souhaitons inclure dans le périmètre de l'étude ;

— P3 : Nous postulons que la hiérarchisation des causes d'un effet se base uniquement sur les forces des relations causales qui lient cet effet à ses différentes causes. C'est à dire que nous prenons uniquement compte du degré d'influence entre la cause et son effet, sans considérer d'autres aspects comme le coût qu'engendrerait une action prise sur telle ou telle cause, ou sa difficulté de mise en place ;

1.3.3 Méthodologie de recherche

Cette thèse représente le résultat d'un travail de recherche scientifique fondamentale, suivant une méthodologie analytique et expérimentale. Comme le décrit la figure 1.5, cette méthodologie est composée d'une succession d'itérations entre l'émission d'hypothèses, et le test et l'observation, jusqu'à arriver à des résultats pouvant être évalués.

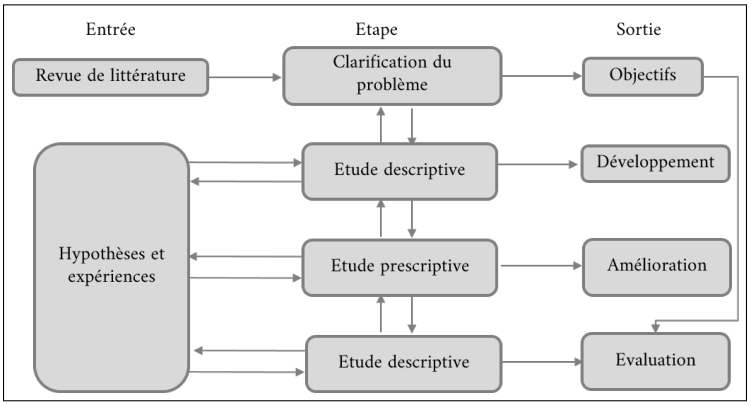


FIGURE 1.5 – Méthodologie de recherche.

L'évaluation des résultats se fera d'abord en vérifiant par étalonnage si la méthode proposée répond bien à l'objectif, et donc aux trois fonctions fixées. Ensuite, une validation se fera à travers une comparaison qualitative et quantitative avec d'autres solutions proposées pour répondre aux mêmes fonctions que les nôtres.

Chapitre 2

État de l'art

2.1 Concept de la causalité

2.1.1 Place de la causalité dans la prise de décision

Comme précédemment illustré sur la figure 1.4, le processus de prise de décision commence à partir du moment de la prise de conscience d'un problème ou d'une opportunité d'amélioration, et se conclut par la mise en place des actions décidées. Entre les deux, plusieurs étapes, représentées par la figure 2.1, sont nécessaires (Newman, 1971).

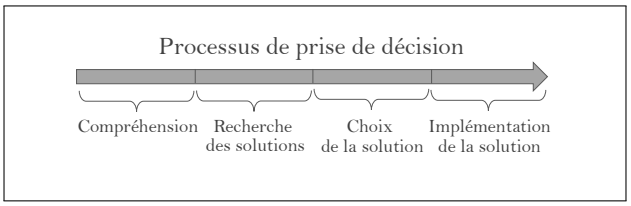


FIGURE 2.1 – Phases d'un processus de prise de décision.

La phase de compréhension est fondamentale et conditionne tout le reste du processus décisionnel (Dash, Bhattacharjee, Singh, Aftab, & Sagesh, 2017). La compréhension concerne à la fois celle du problème ou de l'opportunité d'amélioration, et celle des conséquences que pourrait avoir une décision sur le problème traité, ainsi que sur son environnement. Or, l'évaluation du rapport de cause à effet est indispensable à la compréhension et à l'évaluation des conséquences : le raisonnement causal est le fondement même de toute démarche de compréhension (Dash et al., 2017). Il en va de même pour la recherche des solutions, puisque le fait d'envisager de potentielles alternatives passe par la considération et la prédiction des effets que chaque alternative pourrait avoir sur le problème à résoudre (Güss & Robinson, 2014). Une meilleure connaissance des liens causaux jouerait donc un rôle clé dans les processus de prise de décision : d'une part, elle permettrait de réduire les délais de compréhension et de recherche des solutions, d'autre part, la découverte de nouvelles causes permettrait de s'ouvrir sur de nouvelles alternatives pour traiter le problème. S'agissant du choix de la solution, les décideurs peuvent prendre en compte un ou plusieurs critères de sélection, outre l'ampleur annoncée de chacune des conséquences potentielles des décisions envisageables.

De manière plus générale, la connaissance des liens causaux s'avère utile dans les trois situations décisionnelles suivantes (Kayser & Levy, 2004) :

— La planification : lorsque nous connaissons l'effet escompté, et que nous sommes à la recherche des moyens à mettre en œuvre pour y parvenir, la connaissance des chaînes causales conduisant à l'objectif permet d'aiguiller la recherche des actions à engager ;

— Le diagnostic : lorsqu'un évènement se produit, la connaissance des liens causaux permet d'identifier ce qui l'a provoqué ;

— La prédiction : lorsque nous observons un évènement, et que nous savons qu'il est une cause d'un autre évènement, nous pouvons alors prédire le comportement à venir.

En pratique, les décisions sont les résultats d'une accumulation d'expériences (Jensen & Nielsen, 2001), chose qui peut parfois restreindre le champs des possibles lorsqu'il s'agit d'envisager des alternatives. L'évolution des environnements industriels représente une opportunité pour tenter de développer des méthodes ou outils d'aide à la décision basés sur l'analyse de la causalité à partir des données. En effet, la découverte de nouvelles relations causales à partir des données, et l'invalidation d'autres relations causales supposées connues mais qui sont en réalité erronées, pourraient considérablement améliorer et accélérer la prise de décision. De ce fait, l'inférence causale à partir des données fait l'objet de débats depuis plusieurs années concernant sa faisabilité, et les techniques pour y parvenir le cas échéant.

Dans la suite de cette section, nous allons passer en revue les pratiques de l'analyse de la causalité, ainsi que les techniques d'inférence causale. Afin de pouvoir se positionner, et au vu des désaccords autour de ce qui définit la causalité, nous ne pouvons pas faire l'économie de faire d'abord un panorama sur les différentes définitions et caractéristiques du concept de la causalité.

2.1.2 Définition de la causalité

La causalité est une notion paradoxalement ambiguë et évidente à la fois. C'est un principe que nous utilisons quotidiennement et naturellement pour désigner une relation entre deux entités, l'une ayant *causé* l'autre. Ces entités peuvent représenter des évènements (le convoyeur *s'est arrêté* car le capteur *a détecté une surcharge*), des états (le convoyeur est *à l'arrêt* parce que le capteur est *en panne*), ou des propriétés ou variables (la *rapidité* d'un opérateur à exécuter ses tâches est due à son *expérience*). Malgré le fait qu'il nous soit très familier, le concept de la causalité a longtemps été étudié par plusieurs disciplines, et reste encore aujourd'hui sujet à plusieurs divergences de points de vues. Si la majorité adhère à l'idée que la causalité est essentielle à toute théorie de compréhension, sa définition n'est pas tout à fait arrêtée (Zheng & Pavlou, 2010). Nous entendons ici par définition, la ou les conditions que doivent remplir les entités concernées, ainsi que la relation entre elles, pour que cette dernière soit qualifiée de relation de cause à effet : qu'est ce qui nous fait dire que A a causé B, ou que A est une cause de B ?

De nombreux philosophes se sont d'abord penchés sur l'élucidation de cette question pas très évidente, du fait qu'aucune observation n'est en mesure de faire le lien entre une cause et son effet : il n'y a formellement aucune perception sensorielle qui nous permette d'identifier un tel lien (Besneux, 2017). Certains parmi les philosophes s'étant

intéressés à la question, auxquels s'ajoutent quelques statisticiens, affirment ne pas croire à l'existence même de la causalité, compte tenu du fait que les relations causales ne se manifestent pas forcément par des relations physiques réelles (Esfeld, 2010). Leur thèse suppose alors que la causalité serait une pure illusion de l'esprit. Nous écartons cette idée puisque, fondamentalement, il serait inenvisageable de vivre et de travailler, sans utiliser de raisonnement basé sur la causalité, quel que soit le nom que l'on lui attribue (Spirtes, Glymour, & Scheines, 2000). Cela conduirait inévitablement à la fatalité. L'idée selon laquelle la causalité n'existerait pas est d'autant plus absurde que ses partisans eux même agissent comme si la causalité existait : n'importe qui freinerait *pour* arrêter sa voiture. Dans ce sens, nous partons donc du principe que les liens de causes à effets existent, et tenterons dans ce qui suit d'établir ce qui les caractérise. Ceci étant dit, nous soulignons que nous ne remettons pas en question l'existence du hasard, que nous concevons, dans notre contexte, comme étant l'intersection de deux chaînes causales indépendantes dans l'espace et dans le temps. Par exemple, un opérateur de maintenance s'absente car il doit garder son enfant malade, qui s'est grièvement blessé à la main en essayant d'ouvrir une boîte de conserves (chaîne causale 1, qui explique l'absence de l'opérateur) ; le même jour, dans l'usine où travaille cet opérateur, une machine tombe subitement en panne suite à une coupure d'électricité causée par les intempéries (chaîne causale 2, qui explique la panne de la machine). Ces deux chaînes causales sont indépendantes dans le sens où aucune entité de l'une n'est intervenue dans l'avènement des évènements de l'autre, et qu'elles auraient pu avoir lieu séparément à des moments différents. Le *hasard*, inattendu, se manifeste donc par la rencontre de ces deux chaînes causales. L'arrêt de production qui découlera de cette rencontre aurait alors était imprévisible. Dans notre contexte, nous ne prétendons pas maîtriser ce genre de situations. Nous tenterons plutôt d'identifier des relations causales permettant de prévoir de potentielles dérives afin de pouvoir anticiper les actions préventives adéquates. Étant donné qu'il n'y a pas de consensus autour de la définition de la causalité, nous parcourons dans ce qui suit les différentes caractéristiques de la causalité qui ont été proposées dans différents domaines d'étude, afin d'en sélectionner celles qui nous intéressent, et qui s'avèrent compatibles avec le contexte de notre recherche.

2.1.2.1 Causalité et antécédence temporelle

La caractéristique la plus rationnelle et évidente de la causalité est l'antécédence chronologique de la cause par rapport à l'effet : la cause doit toujours précéder son effet dans le temps.

$$A \text{ cause } B \Longrightarrow A \text{ précède } B \text{ dans le temps} \tag{2.1}$$

La composante temporelle de la causalité, admise et reflétée de manière implicite dans les comportements communs, fut explicitement exprimée dans l'axiome posé par Hume, initiateur des discussions autour de l'analyse de la causalité et de l'inférence causale :

> "... Nous pouvons donc définir une cause comme un objet suivi d'un autre et tel que tous les objets semblables au premier sont suivis d'objets semblables au second." [1]

1. Hume, Enquête sur l'entendement humain, 1748

On note que l'antécédence temporelle doit toujours être vérifiée quand il s'agit de causalité entre deux évènements concrets, ce qui n'est pas toujours le cas lorsqu'on parle de lien causal entre deux propriétés, qui peuvent être cause ou conséquence selon les cas. Par exemple, si le stress de l'opérateur l'a conduit à faire des erreurs de montage, on sait que le stress est arrivé avant l'apparition des défauts de montage. En revanche, nous ne pouvons pas établir si dans l'absolu, le stress est une cause de défauts, ou si c'est l'inverse, même pour le même opérateur, dans d'autres situations, les défauts pourraient être susceptibles de stresser l'opérateur. Ce point sera abordé plus tard dans cette section.

Si l'antécédence temporelle liée à la causalité fait l'unanimité (Atallah, 2014), du moins quand il s'agit d'évènements, la question de la régularité à laquelle cet axiome fait également référence a partagé les scientifiques et philosophes et provoqué plusieurs désaccords.

2.1.2.2 Causalité et régularité

À l'instar de l'antécédence temporelle, Hume a souligné le caractère régulier de la causalité. L'interprétation de son axiome laisse entendre qu'une cause est forcément suivie de son effet, quelles que soient les circonstances (Drouet, 2007). Nous pouvons traduire cela par la proposition suivante :

$$[A \text{ est une cause de } B] \Longleftrightarrow [\mathbb{P}(A) = 1 \Rightarrow \mathbb{P}(B) = 1] \tag{2.2}$$

Cette assertion fit l'objet de plusieurs critiques. En effet, elle laisse interroger sur deux points essentiels : la suffisance, et la nécessité de la cause à l'apparition de l'effet. Par ailleurs, conclure sur l'existence d'un lien causale à partir d'observations répétitives d'une même succession de deux évènements a très vite été remis en question, au travers de contre-exemples, mettant en avant le risque de confondre causalité et corrélation. Nous aborderons plus dans la sous-section 2.1.2.5 la distinction entre la causalité et la corrélation.

2.1.2.3 Causalité, suffisance, et nécessité

La notion de régularité a fait l'objet de plusieurs objections (Blanchard, 2018). L'exemple le plus courant pour réfuter cette idée est celui de l'allumette : gratter une allumette ne provoque pas toujours une flamme. Ceci nous conduit à aborder deux points supplémentaires : la généricité ou l'universalité des liens causaux, et la pluralité des causes pour un seul et même effet.

— **Généricité des liens causaux :** on peut distinguer deux catégories de liens causaux : les causalités génériques, et les causalités singulières. La causalité générique, aussi appelée jugement causal générique, fait référence aux liaisons causales entre variables ou propriétés, tandis que la causalité singulière concerne des évènements uniques (Barberousse, Bonnay, Cozic, et al., 2011). Par exemple, dire que le stress de l'opérateur A a causé son ralentissement au travail ne veut pas dire que le stress des opérateurs cause leur ralentissement. Dans la suite, nous parlerons plutôt de causalité explicative pour désigner la catégorie générique, et de causalité circonstancielle pour la catégorie singulière. Nous nous focaliserons particulièrement sur la causalité explicative parce que, nous le rappelons, l'objectif est de pouvoir identifier

les relations causales afin d'entreprendre des actions préventives. Dans ce sens, il faut que les liens causaux identifiés soient avérés dans plusieurs situations afin de pouvoir fonder les décisions dessus.

— **Pluralité des causes :** revenons à l'exemple de l'allumette ; si dans certaines conditions le grattage d'une allumette n'est pas suivi de son embrasement, ceci ne veut pas dire que dans l'absolu, le grattage d'allumettes n'est pas une cause d'apparition de flammes. Une absence de dioxygène ou une présence trop importante de vent, entre autres, peuvent empêcher l'embrasement, mais cela n'enlève rien du caractère causal du lien entre le frottement et l'apparition de la flamme. Ici, le dioxigène, la vitesse, et la direction du vent sont communément perçus comme étant des "facteurs" parmi d'autres participant à l'embrasement. En réalité, ce sont, tout comme le frottement, des causes élémentaires, d'un ensemble causal permettant d'expliquer la conséquence (Drouet, 2007). En effet, une cause peut être nécessaire, mais pas suffisante pour occasionner son effet. Par conséquent, pour percevoir la causalité comme quelque chose de régulier, déterministe et universel, il faut que la cause soit suffisante. Autrement, il est indispensable de connaître l'ensemble des causes ; qui est composé de causes non suffisantes mais nécessaires, et dont la conjonction provoque l'effet (Lacour, 2017). Cet ensemble, lui même est suffisant, mais pas forcément nécessaire à l'apparition de l'effet (Lacour, 2017) : une allumette peut s'enflammer en la rapprochant simplement d'une flamme, l'effet apparaît alors suite à un autre ensemble suffisant mais pas nécessaire de causes insuffisantes mais nécessaire. Cette idée, initialement introduite par Mackie (Mackie, 1965), et qui nous parait plus convaincante que la régularité absolue, est illustrée sur la figure 2.2. Le diagramme logique équivalent y est également présenté. Dans l'exemple illustré sur cette figure, on suppose que tous les ensembles suffisants possibles de causes sont connus, l'implication donnée par (2.3) est alors vraie :

$$[\mathbb{P}(C_1)\wedge\mathbb{P}(C_2)\wedge\mathbb{P}(C_3)=1]\vee[\mathbb{P}(C_4)\wedge\mathbb{P}(C_5)=1]\vee[\mathbb{P}(C_6)=1]\Longrightarrow\mathbb{P}(E)=1 \quad (2.3)$$

où C_1 à C_6 désignent les causes, et E désigne l'effet. De manière plus générale, on donc écrire la proposition donnée par (2.4) :

$$\forall\{C_1,...,C_n\}\text{, ensemble suffisant de causes de } E : \mathbb{P}(\bigcap_{i=1}^{n} C_i)=1\Longrightarrow\mathbb{P}(E)=1 \quad (2.4)$$

Par ailleurs, il convient de souligner que lorsqu'un effet ne peut être occasionné que par un seul et unique ensemble causal suffisant (composé d'une ou plusieurs causes), cet ensemble devient alors nécessaire à l'apparition de l'effet.

2.1.2.4 Causalité et probabilités

Théoriquement, lorsque "toutes les causes insuffisantes mais nécessaires d'un ensemble suffisant mais pas forcément nécessaire à la production d'un effet" sont connues, nous pouvons établir que l'observation de cet ensemble de causes garantit l'observation de l'effet (Lacour, 2017 ; Mackie, 1965). Cependant, dans la pratique, il se trouve qu'il n'est pas évident de connaître toutes les causes, encore moins de les assigner à un ensemble, de manière à ce qu'il soit suffisant (Pearl, 2000). En outre, rien ne garantit l'existence d'un

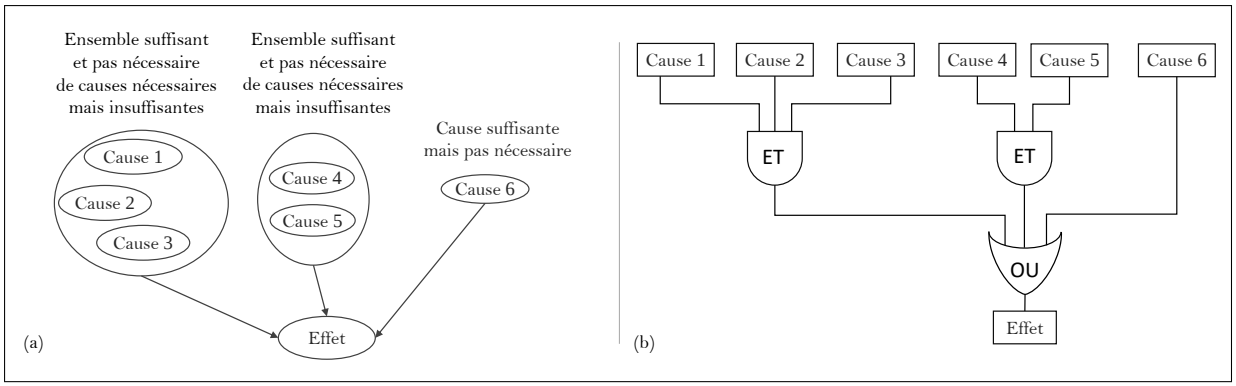


FIGURE 2.2 – Schéma explicatif des notions de suffisance et de nécessité causales (a), et diagramme logique équivalent (b).

tel ensemble (Drouet, 2007). Par ailleurs, les situations où sont énoncés des liens causaux sont souvent marquées d'indéterminisme. Nous n'énonçons souvent que les raisons "phares" qui, selon nos croyances, occasionnent l'effet (Pearl, 2000) : pour l'exemple de l'allumette, nous aurions tendance à dire qu'elle s'allume lorsqu'on la gratte, en dépit des situations où cela ne se passe pas comme prévu. La théorie probabiliste de la causalité a donc pour objectif de considérer la causalité à travers des dépendances probabilistes (Juhel, 2015). Concrètement, si une cause ne provoque pas systématiquement son effet, elle le rend tout de même plus probable (Suppes, 1970). Ainsi, Suppes a défini la causalité par la proposition (2.5) :

$$C \text{ est une cause de } E \iff \begin{cases} C \text{ est antérieur à } E & \text{et ;} \\ \mathbb{P}(C) > 0 & \text{et ;} \\ \mathbb{P}(E|C) > \mathbb{P}(E) \end{cases} \tag{2.5}$$

Cette définition pose trois conditions nécessaires que deux évènements doivent remplir pour pouvoir conclure sur l'existence d'un lien causal entre eux. Le premier axiome de cette définition suggère l'antériorité temporelle de la cause par rapport à l'effet. Le deuxième axiome exige que l'évènement cause soit possible, puisqu'il serait absurde de se prononcer sur un lien causal dont la cause est impossible à réaliser. De plus, ce deuxième axiome est implicitement exigé par le troisième : pour rappel, pour deux évènements A et B, on a : $\mathbb{P}(A|B) = \dfrac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$, il est donc indispensable que la probabilité de la cause soit strictement supérieure à 0 pour pouvoir calculer la probabilité conditionnelle de l'effet sachant la cause. Enfin, le troisième axiome établit que l'observation de l'effet est d'autant plus probable lorsque la cause est observée. Dit autrement, l'observation de la cause augmente la probabilité d'observer l'effet. Nous pouvons également dire qu'il est plus probable d'observer l'effet sachant que la cause a eu lieu, que d'observer l'effet sachant que la cause n'a pas eu lieu : $\mathbb{P}(E|C) > \mathbb{P}(E|\neg C)$. Cette expression est équivalente au dernier axiome de la définition (2.5), puisqu'on a, d'après le théorème des probabilités totales :

$$\mathbb{P}(E) = \mathbb{P}(E|C).\mathbb{P}(C) + \mathbb{P}(E|\neg C).\mathbb{P}(\neg C)$$

$$\begin{aligned}\text{alors : } \mathbb{P}(E|C) > \mathbb{P}(E|\neg C) &\iff \mathbb{P}(E|C).\mathbb{P}(\neg C) > \mathbb{P}(E|\neg C).\mathbb{P}(\neg C)\\ &\iff \mathbb{P}(E|C).\mathbb{P}(\neg C) + \mathbb{P}(E|C).\mathbb{P}(C) > \mathbb{P}(E)\\ &\iff \mathbb{P}(E|C).[\mathbb{P}(\neg C) + \mathbb{P}(C)] > \mathbb{P}(E)\\ &\iff \mathbb{P}(E|C) > \mathbb{P}(E)\end{aligned}$$

L'équivalence proposée par la définition (2.5) reste à discuter : lorsque le lien causal entre C et E est avéré, il est logique de conclure que les trois axiomes seraient vérifiés. Néanmoins, dans le sens inverse, il ne serait pas rigoureux de conclure un lien causal à partir de la vérification des trois axiomes : cela conduirait à voir des causalités là où il n'y en a pas (Besneux, 2017).

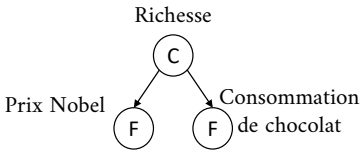
2.1.2.5 Causalité et corrélation

Inférer une causalité à partir des trois axiomes de la définition (2.5) est une erreur. Il convient donc de lever la confusion entre causalité et corrélation, et de montrer, à travers un contre-exemple, que les augmentations simultanées des probabilités de la cause et de l'effet ne représentent pas une condition suffisante pour inférer une relation causale. Nous rappelons que ces probabilités sont calculées de manière empirique, à partir des fréquences d'observation (Belis, 1995).

La première situation pouvant porter à confusion est celle où deux évènements, n'étant pas reliés causalement, partagent une cause commune. Prenons l'exemple de la consommation de chocolat, et de l'attribution des prix Nobel : une étude a publié des statistiques soulevant que les pays où la consommation de chocolat est plus élevée se voient attribuer plus de prix Nobel que les pays où le chocolat est moins consommé (Messerli, 2012). La conclusion de cette étude a rapidement été sur-interprétée et reprise par d'autres études, clamant à tort que manger du chocolat augmenterait nos chances de recevoir un prix Nobel (Prinz, 2020). L'incohérence de ce genre d'assertions est d'autant plus prononcée si l'on se demande comment le fait qu'un nominé consomme beaucoup de chocolat pourrait affecter la décision du jury. En réalité, la corrélation entre la consommation de chocolat dans un pays et le nombre de ses lauréats du prix Nobel peut être expliquée par l'existence d'une variable cachée. La richesse du pays, par exemple, peut expliquer cette corrélation : plus un pays est riche, plus ses citoyens consomment du chocolat puisqu'ils en ont les moyens ; parallèlement, plus un pays est riche, plus les investissements dans l'éducation et la recherche sont importants et nombreux, et donc plus les chances sont importantes de voir les citoyens de ce pays atteindre le niveau pour être lauréats du prix Nobel. Ici, la richesse est une cause commune, que l'on peut également appeler facteur de confusion.

De façon plus générale, si C est une cause commune de E et de F, on dit que les entités C, E, et F forment une fourche conjonctive (Reichenbach, 1956). Et on a : $\mathbb{P}(E|C) > \mathbb{P}(E)$, et $\mathbb{P}(F|C) > \mathbb{P}(F)$, et E et F sont corrélées positivement. Par conséquent, lorsque C a lieu, les deux probabilités respectives de ses conséquences E et F augmentent conjointement.

On sait que si deux variables aléatoires sont indépendantes, alors elles sont non-corrélées (Juhel, 2015). Nous en déduisons que si deux variables sont corrélées, alors elles sont dépendantes. Nous en constatons que E et F sont dépendants (nous notons



$E \not\perp\!\!\!\perp F$), puisqu'ils sont corrélés . Pour deux évènements E et F tels que $E \not\perp\!\!\!\perp F$, on a $\mathbb{P}(E \cap F) \neq \mathbb{P}(E).\mathbb{P}(F)$ (Pearl, 2000). Dans notre cas, la corrélation est positive, alors $\mathbb{P}(E \cap F) > \mathbb{P}(E).\mathbb{P}(F)$. On peut donc écrire l'équivalence 2.6 :

$$\begin{aligned} \mathbb{P}(E \cap F) > \mathbb{P}(E).\mathbb{P}(F) &\iff \frac{\mathbb{P}(E \cap F)}{\mathbb{P}(F)} > \mathbb{P}(E) \\ &\iff \mathbb{P}(E|F) > \mathbb{P}(E) \end{aligned} \tag{2.6}$$

De la même manière, nous pouvons démontrer que $\mathbb{P}(F|E) > \mathbb{P}(F)$. Nous en déduisons que si nous ignorons l'existence de C, et quel que soit l'ordre chronologique des occurrences de E et F, les trois axiomes de la définitions (2.5) seront vérifiés dans un cas ou dans l'autre pour conclure un lien causal entre E et F : par exemple, si E précède F dans le temps, nous aurons : (1) E est antérieur à F, et (2) $\mathbb{P}(F) > 0$ (puisque F est un effet avéré de C, il n'est alors pas impossible), et (3) $\mathbb{P}(F|E) > \mathbb{P}(F)$. Dans ce cas, et en se fiant à l'équivalence donnée par (2.5), on pourra conclure, à tort, que E est une cause F. Inversement, si c'est F qui précède E dans le temps, on pourra conclure, à tort également, que F est une cause de E. Manifestement, de la causalité on peut déduire la corrélation, mais la réciproque n'est pas toujours vraie. Un évènement peut donc augmenter la probabilité d'un autre sans nécessairement en être la cause ou la conséquence. Nous avons parlé ici de corrélation, qui concerne les variables quantitatives, mais ceci est également valable plus généralement pour l'association, qui peut faire référence aux variables qualitatives également.

Par ailleurs, le premier axiome de la définition (2.5) suppose que pour conclure une relation causale entre C et E, il faut connaître la chronologie d'occurrence de C et de E. Or, comme nous l'avons mentionné plus haut, dans notre contexte, nous nous intéressons à la causalité générique, et non pas à la causalité singulière. C'est à dire que ce sont les liens causaux entre les propriétés, de manière générale, que nous voulons identifier, et non pas les liens causaux entre les instanciations de ces propriétés. Il est donc difficile d'émettre un jugement sur l'antériorité de deux propriétés dans l'absolu (Drouet, 2007). Ainsi, la deuxième situation où la définition (2.5) peut induire en erreur lorsque l'ordre chronologique absolu des propriétés n'est pas connu ou bien impossible à définir, puisqu'on a :

$$\begin{aligned} \mathbb{P}(E|C) > \mathbb{P}(E) &\iff \frac{\mathbb{P}(E \cap C)}{\mathbb{P}(C)} > \mathbb{P}(E) \\ &\iff \mathbb{P}(E \cap C) > \mathbb{P}(E).\mathbb{P}(C) \\ &\iff \frac{\mathbb{P}(E \cap C)}{\mathbb{P}(E)} > \mathbb{P}(C) \\ &\iff \mathbb{P}(C|E) > \mathbb{P}(C) \end{aligned} \tag{2.7}$$

Il est donc analytiquement prouvable que, de la même manière qu'une cause augmente la probabilité de son effet, l'effet augmente également la probabilité de sa cause.

Par conséquent, dans les cas où il est impossible d'établir un ordre chronologique entre deux propriétés dans l'absolu, et même en ayant la certitude de l'existence d'un lien causal entre les deux propriétés, l'augmentation de la probabilité ne nous permet pas d'inférer le sens de la causalité.

Cette analyse, où ont été relevés les deux points pouvant induire en erreur quant à l'inférence causale à partir de l'équivalence (2.5), nous conduit donc à considérer seulement une implication à sens unique. Nous retenons alors la proposition donnée par (2.8) :

$$C \text{ est une cause de } E \Longrightarrow \begin{cases} C \text{ est antérieur à } E \text{ en instanciation} & \text{et ;} \\ \mathbb{P}(C) > 0 & \text{et ;} \\ \mathbb{P}(E|C) > \mathbb{P}(E) \end{cases} \tag{2.8}$$

2.1.2.6 Causalité et causes sous-jacentes

Suppes (Suppes, 1968) a essayé de pallier au problème de confusion entre la corrélation et la causalité en ajoutant un axiome supplémentaire, qui exige cette fois l'absence de toute cause pouvant faire écran ente deux effets, et donc induire en erreur. Cela se traduit par :

$$C \text{ est une cause de } E \Longrightarrow \begin{cases} C \text{ est antérieur à } E \text{ en instanciation} & \text{et ;} \\ \mathbb{P}(C) > 0 & \text{et ;} \\ \mathbb{P}(E|C) > \mathbb{P}(E) & \text{et ;} \\ \nexists\, C' \text{ antérieur à } E \text{ tel que : } \mathbb{P}(E|C, C') = \mathbb{P}(E|C') \end{cases} \tag{2.9}$$

Ce dernier axiome suggère de vérifier qu'il n'y a aucune autre variable C' qui augmente la probabilité de l'effet E, de telle manière que lorsque C' est observée, l'observation de C n'apporte aucune information supplémentaire. Pour les mêmes raisons discutées plus haut, et pour d'autres raisons que nous citerons ici, nous nous contentons d'une implication à sens unique.

Théoriquement, ce dernier axiome aurait pu permettre de résoudre le problème de causalité erronée due à la présence d'une ou plusieurs causes communes sous-jacentes de deux effets. Cependant, il demeure rationnellement impossible de démontrer la vérification de cet axiome, et de prouver que toutes les variables pouvant être observées l'ont effectivement été (Kenny, 1979). L'impossibilité de garantir l'observation de toutes les variables susceptibles d'être causes communes de deux effets, représente l'un des plus grands freins à l'inférence automatique de la causalité (Kenny, 1979). Raison pour laquelle les extrapolations de causalités fallacieuses à partir de simples corrélations ou associations sont très courantes. Inversement, il est impossible de prouver que toutes les variables devant être observées pour éviter ce problème ne l'ont pas été.

Bien que ce dernier axiome ne permette pas, dans la pratique, d'identifier avec certitude les liens de causalité, il porte tout de même une idée de fond qui puisse permettre d'approcher la causalité. En effet, l'idée de cet axiome nous incite naturellement à dire que plus le nombre de variables de contexte observées est important, moins élevé serait le risque de lier causalement et à tort deux effets d'une même cause sous-jacente. Faisons l'analogie avec un lancer de dé non pipé, qui traditionnellement, est l'un des exemples les plus symboliques du hasard. Le résultat obtenu lors d'un lancer de dé est alors disculpé

de toute relation causale ayant conduit à obtenir ce résultat et pas un autre. De cette perspective, le pronostic le plus ambitieux que nous pourrions faire quant au résultat serait de dire qu'il serait compris entre 1 et 6. Cette attribution de résultat au hasard n'est en réalité pas tout à fait rigoureuse, puisque si nous connaissons quelques variables supplémentaires, nous pourrions faire un pronostic plus précis. Le fait de connaître entre autres la position initiale exacte du dé, sa vitesse initiale, sa structure, son poids, et l'ensemble des forces qui vont être exercées sur le dé durant sa trajectoire, nous permettrait de prédire de façon plus précise la face sur laquelle il va tomber. Mais comme l'ensemble des valeurs de ces variables est souvent difficile à obtenir, nous attribuons le résultat au phénomène du hasard, que nous utilisons quelque part pour justifier notre ignorance. La connaissance des variables de contexte diminue donc l'étendue de notre ignorance, et permet par conséquent d'éviter de faire des erreurs d'interprétation sur les phénomènes qu'on essaie d'expliquer, comme elle permet de limiter les phénomènes que nous attribuons, par manque de connaissances, au hasard. À l'avenant de cet exemple, l'explication de tout phénomène devient d'autant plus recevable et plausible lorsqu'elle prend compte de plus de variables de contexte. Si nous revenons à l'exemple de corrélation entre la consommation de chocolat et le nombre de lauréats de prix Nobel par pays, l'observation de variables de contexte supplémentaires aurait pu éviter l'interprétation abusive des résultats de l'étude et l'incitation à la consommation de chocolat pour augmenter les chances de recevoir un prix Nobel.

Pour clore ce point, nous dirions que bien que l'aspect probabiliste puisse nous persuader qu'il y a une relation entre deux entités, il demeure insuffisant pour affirmer l'existence d'un lien causal. Cependant, le nombre de variables de contexte observées et le risque de confusion entre causalité et corrélation sont inversement corrélés. Par conséquent, une corrélation qui demeure pertinente entre deux entités malgré le fait d'observer un nombre important de variables de contexte nous permet d'approcher la présomption de relation causale.

Par ailleurs, il est vrai que si aucune corrélation n'est soulevée entre deux entités, il serait inutile de tenter de trouver un lien causal entre elles. Dans ce cas, l'exploration des variables de contexte reste malgré tout profitable pour découvrir de nouvelles corrélations, et ensuite potentiellement les causalités liant chacune de ces entités avec les variables explorées du contexte. En outre, l'espace de recherche des causes est important non seulement pour identifier les corrélations trompeuses et découvrir de nouvelles corrélations, mais également pour identifier les indépendances trompeuses. Lorsque deux variables A et B sont observées comme étant indépendantes, il se peut que l'indépendance observée soit due à une troisième variable C non observée qui est inversement corrélée à l'une des deux variables A ou B, et que ce soit cette corrélation inversée qui annule la dépendance entre A et B : considérons par exemple trois variables A) Fumer, et B) Problèmes cardiaques, et C) Activité physique, et supposons que dans la population étudiée, les fumeurs exerçant une activité physique ne développent pas de problèmes cardiaques. Si on observe uniquement les variables A et B sans avoir d'information sur la variables C on conclura à tort que le fait d'avoir des problèmes cardiaques n'est pas lié au fait d'être fumeur. Mais en observant la variable C, on conclu que c'est sa corrélation inverse avec A qui annule la dépendance entre A et B, et que B est dépendante de A et C à la fois. Là encore, le constat selon lequel l'observation d'un grand nombre de variables favorisera la présomption correcte de la présence ou l'absence d'un lien causal.

2.1.2.7 Causalité et transitivité

L'ambition d'explorer les causes dans l'objectif d'éviter des événements redoutés serait intenable si l'on se cantonne aux causes directes de l'évènement en question. En effet, lors de la recherche des causes racines d'un problème, le raisonnement transitif est intuitivement adopté (Bonnefond, Castelain, Cheylus, & Van der Henst, 2014). C'est notamment le principe de la démarche des cinq pourquoi. Le caractère transitif de la causalité, s'il est avéré, permettrait d'inférer des relations causales indirectes non observées à partir de relations connues (Von Sydow, Hagmayer, & Meder, 2016). Par exemple, si nous savons que $A \rightarrow B$, et que $B \rightarrow C$, et comme le montre la figure 2.3, le caractère transitif de la causalité nous permettrait de déduire que $A \rightarrow C$. L'emploi du conditionnel ici est éloquent, au vu des désaccords autour de cette caractéristique de la causalité, c'est d'ailleurs pourquoi nous avons choisi de mettre en pointillés l'arc ajouté sur la figure 2.3 pour effectuer la fermeture transitive. Ainsi, Lewis, défendeur de la transitivité de la causalité, pose l'assertion (2.10) (Lewis, 1986) :

$$A \text{ est une cause de } B \iff \text{Il existe une chaîne causale entre } A \text{ et } B \tag{2.10}$$

La transitivité est souvent indissociable de la causalité, il n'est cependant pas toujours valable de faire des raccourcis entre causes et effets (Halpern, 2016). L'équivalence (2.10) a fait l'objet de plusieurs critiques appuyées par des contre-exemples (Johnson & Keil, 2017). En effet, la dépendance probabiliste est non transitive (Falk & Bar-Hillel, 1983 ; Bourgeois-Gironde, 2002). La non transitivité de la dépendance probabiliste peut également être démontrée grâce à un contre-exemple : admettons que l'on dispose d'un dé non pipé, considérons les trois évènements A) le résultat est impair, B) le résultat est un multiple de 5, et C) le résultat est supérieur à 4, on a alors $\mathbb{P}((A \cap B) = \frac{1}{6}$ et $\mathbb{P}(A).\mathbb{P}(B) = \frac{3}{6}.\frac{1}{6}$, donc A et B sont dépendants puisque $P(A \cap B) \neq \mathbb{P}(A).\mathbb{P}(B)$. On a aussi $\mathbb{P}(B \cap C) = \frac{1}{6}$, et $\mathbb{P}(B).\mathbb{P}(C) = \frac{1}{6}.\frac{2}{6}$, donc B et C sont dépendants puisque $\mathbb{P}(B \cap C) \neq \mathbb{P}(B).\mathbb{P}(C)$. Enfin, on a $\mathbb{P}(A \cap C) = \frac{1}{6}$, et $\mathbb{P}(A).\mathbb{P}(C) = \frac{3}{6}.\frac{2}{6} = \frac{1}{6}$, donc A est indépendant de C puisque $\mathbb{P}(A \cap C) = \mathbb{P}(A).P(C)$. On a alors A et B sont dépendants, et B et C sont dépendant, mais A et C sont indépendants. Par conséquent la dépendance n'est pas une propriété transitive. Par ailleurs, comme nous avons vu précédemment, une relation causale implique une dépendance. Si on considère trois évènements C, E, et F tels que C cause E et E cause F, alors C et E sont dépendants, et E et F sont dépendants, mais comme nous l'avons montré, cela n'implique pas forcément que E et F sont dépendants. La dépendance entre E est F n'est pas garantie, par conséquent la causalité ne l'est pas non plus, puisque la dépendance est une condition nécessaire à la causalité.

La causalité, selon l'approche contre-factuelle, n'est pas transitive (Kistler, 2011). En effet, l'approche contre-factuelle de la causalité assert que [A cause B] ⇒[si B n'a pas lieu, alors A n'a pas lieu]. Et selon l'approche contre-factuelle probabiliste : [A cause B] ⇒[si $\mathbb{P}(\neg B)$ augmente alors $\mathbb{P}(\neg A)$ augmente]. En supposant que la causalité est transitive, on a alors [A cause B, et B cause C] ⇒ A cause C. Selon l'approche contre-factuelle, pour dire que A cause C il faut que l'absence de C provoque l'absence de A, ou bien que l'absence de C augmente la probabilité de l'absence de A (*i.e.* $\mathbb{P}(\neg A|\neg C) > \mathbb{P}(\neg A)$). Or, si l'on considère les 4 évènements A, B, C, D comme suit : A) le guide de montage d'un produit comporte des erreurs, B) l'opérateur fait des erreurs de montage, C)

le produit comporte des défauts, et D) le client retourne le produit. Ici, l'absence du retour du produit ne provoque pas l'absence des erreurs dans le guide de montage, on peut également affirmer de manière intuitive qu'elle n'augmente même pas la probabilité d'absence des erreurs dans le guide de montage. Dans ce cas, et si l'on résonne selon l'approche contre-factuelle de la causalité, on déduit que A ne cause pas D, ce qui veut dire que la causalité n'est pas transitive selon la conception contre-factuelle de la causalité.

Paradoxalement, la causalité selon l'approche de production est conçue comme un concept transitif (Hall, 2004). Cette approche suggère de voir une cause comme quelque chose qui aide son effet à se produire, d'une façon ou d'une autre.

Nous ne nous attarderons pas sur les débats confrontant ces deux points de vue, puisque la causalité peut être vue à la fois comme une relation de dépendance contre-factuelle et de production. Selon nous, l'utilisation de la causalité pour des fins de prise de décision suppose qu'il est préférable d'avoir le plus de granularité possible dans les liens de cause à effet. Autrement dit, il est préférable de savoir que A cause B, et que B cause C, plutôt que de savoir seulement que A cause C même dans le cas où c'est vrai. D'abord pour éviter toute interprétation abusive, mais surtout pour avoir un large panel de choix d'actions à entreprendre. Par exemple, si nous savons uniquement que A cause C, et que nous voulons provoquer un changement de C en agissant sur sa ou ses causes(s), nous n'aurions pas d'autres choix que d'agir sur A pour provoquer la modification de C. Cependant, si nous savons qu'il existe une cause B telle que A cause B et B cause C, nous pourrions choisir, pour des raisons de coûts, de facilité, ou autre, d'agir sur B pour provoquer un changement sur C. Néanmoins, dans le cas où nous voudrions quand même agir sur la cause *racine* A, il faut s'assurer que la transitivité est valable pour éviter de prendre des actions qui auraient peu ou pas d'incidence sur C. Nous reviendrons sur les conditions de fermeture transitive d'un graphe causal dans la section suivante. Pour l'instant, nous affirmons que même si la fermeture transitive causale d'un graphe causal est possible, il est préférable d'avoir les deux versions du graphe pour pouvoir choisir d'agir sur la cause directe de la propriété qui nous intéresse, ou sur sa cause indirecte.

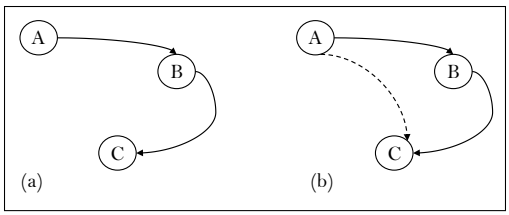


FIGURE 2.3 – Fermeture transitive C(G) (b) du graphe G (a).

2.1.2.8 Causalité et critères de Hill

Hill (Hill, 1965) s'est intéressé à la notion de la causalité pour la place importante qu'elle prend dans le domaine de l'épidémiologie. Il a ainsi défini plusieurs critères supplémentaires qu'une relation causale doit satisfaire. L'objectif initial derrière ses travaux était de pouvoir capturer les relations causales et de ne pas systématiquement voir la causalité dans toute association, comme nous en avons discuté dans la sous-section 2.1.2.5. Il a donc établi une liste de critères qui peuvent être utilisés pour valider que les liens identifiés comme causaux le sont effectivement. En effet, nous avons vu que les probabilités ne

suffisent pas à rendre compte de la causalité, et qu'il faut observer le plus de variables de contexte possibles pour éviter la sur-interprétation des corrélations, et des associations en général. Bien que cela soit vrai, et permette de repérer des corrélations non causales, nous avons vu qu'il est impossible de garantir que toutes les variables de contextes pouvant faire objet de cause commune de deux effets différents ont été observées. Ainsi, Hill a mis en place une liste de critères afin d'examiner le caractère causal d'une association (Hill, 1965). Bien que Hill ait défini ces critères dans un domaine bien précis et différent du nôtre, nous nous y intéressons tout de même afin d'en sélectionner ceux qui pourraient être pertinents dans notre contexte. En effet, la causalité est avant tout un concept de raisonnement, dont certains principes restent valables quel que soit le domaine d'études auquel on s'intéresse. Certains parmi les critères définis par Hill ont déjà été discutés dans les sous-sections précédentes, nous ne nous y attarderons donc pas.

— Force de l'association : plus la cause est présente, plus l'effet l'est aussi. Ceci est équivalent au troisième axiome de l'implication donnée par (2.9) ;

— Relation temporelle : la cause doit être observée avant l'effet, comme discuté dans la sous-section 2.1.2.1 ;

— Relation dose-effet : une plus large *dose* de cause provoque une plus large *dose* d'effet. C'est à dire que si A est une cause de B qui lui est positivement corrélée, alors, à l'observation une augmentation de la valeur de A, on observe une augmentation de la valeur de B relative à l'augmentation de B. Cette relation étant valable uniquement pour les variables quantitatives, n'est pas toujours vérifiable dans notre contexte. De plus, ceci suppose que la causalité est une relation monotone (Bard et al., 2005). Or, une cause peut provoquer son effet uniquement au delà d'un certain seuil, ou encore provoquer son effet positivement jusqu'à un certain seuil, et le provoqué négativement quand le seuil est dépassé ;

— Stabilité : selon ce critère, il est d'autant plus probable que la relation identifiée soit causale lorsqu'elle est observée chez toutes les populations, et qu'elle est stable dans l'espace et dans le temps. Ce critère n'est pas valide dans notre contexte, puisque contrairement à l'humain dont l'anatomie est presque semblable chez tous, un contexte industriel se distingue par ses propres caractéristiques et configurations (par exemple le nombre de postes, leurs configurations et ordonnancements, la planification, les moyens, etc) ;

— Cohérence : c'est à dire que les études concluant sur des causalités ne doivent pas se contredire quand il s'agit des mêmes propriétés étudiées (Lavigne & Forrest, 2020). Pour les mêmes raisons d'invalidité de la stabilité de la causalité, ce critère n'est pas retenu dans notre contexte.

— Plausibilité : un relation, prétendue causale, doit être plausible pour les décideurs et experts du domaine, c'est à dire qu'elle doit être cohérente avec leurs connaissances (Bard et al., 2005). La notion de plausibilité peut aider à définir le sens de la causalité lorsqu'il est inconnu, ou lorsqu'on ne peut pas observer directement les instanciations du lien causal. Ce critère étaye le fait de qualifier une relation comme

causale. Néanmoins, il n'est pas nécessaire, puisque nos connaissances peuvent être limitées (Bard et al., 2005). L'objectif, étant justement d'identifier des relations causales insoupçonnées, il n'est pas impossible que les causalités identifiées paraissent non plausibles. Nous soulignons néanmoins que la plausibilité reste un critère important pour ne pas confondre séquence et conséquence ;

— Spécificité : une cause doit provoquer un seul effet. Ce critère n'est pas applicable dans notre contexte, puisqu'à l'inverse des causalités en biologie, une entité peut être liée causalement à plusieurs effets, c'est le principe même de la cause commune. Même en biologie, ce critère semble être très critiqué et non accepté par la communauté : une exposition peut être causalement liée à plusieurs maladies (Bard et al., 2005). Ce critère est donc considéré comme invalide dans notre contexte ;

— Vérifiabilité : une relation, prétendue causale, doit pouvoir être vérifiée, par exemple en agissant intentionnellement sur les causes afin de vérifier l'hypothèse selon laquelle elles provoquent leurs effets. C'est le même principe que celui de *l'intervention* proposé plus tard par Pearl (Pearl, 1988). Ce critère semble être intéressant pour valider l'existence d'un lien causal. Cependant il n'est pas toujours facile de vérifier s'il est satisfait ou pas, notamment des points de vue éthique et économique ;

— Analogie : ce critère n'est pas discutable dans notre contexte puisqu'il est propre au domaine biologique (par exemple lorsque la causalité est argumentée grâce à une analogie sur les animaux).

Parmi ces neufs critères, aucun n'est suffisant pour établir avec certitude l'existence d'une causalité. Hill d'ailleurs, ne prétend pas que sa liste soit exhaustive ou suffisante dans son ensemble, il estime cependant que plus le nombre de critères satisfaits par une association est important, plus il est probable que cette association soit causale. **Nous récapitulons donc que dans notre contexte, nous retenons quatre parmi ces neuf critères : celui de la force d'association (puisque nous retenons l'aspect probabiliste de la causalité. Dans la suite, nous préférons continuer à utiliser la dénomination *aspect probabiliste*) ; celui de la plausibilité, que nous jugeons cependant comme facultatif, mais qui reste tout de même utile pour valider les relations causales, dit autrement, la plausibilité argumente la validation d'un lien causal, mais l'absence de plausibilité n'invalide pas l'existence d'un lien causal ; celui de la vérifiabilité, qui, en cas de sa faisabilité, doit obligatoirement être satisfait ; et enfin celui de la relation temporelle, nécessaire uniquement pour les instanciations de causalité.** On a donc :

Causalité $\Rightarrow$ Association/corrélation (aspect probabiliste) ;
Causalité $\Rightarrow$ Vérifiabilité (quand la vérification est faisable) ;
Causalité $\Rightarrow$ Relation temporelle (obligatoire uniquement en instanciation) ;
Association/corrélation $\rightsquigarrow$ Causalité ;
Plausibilité $\rightsquigarrow$ Causalité ;
Vérifiabilité $\Rightarrow$ Causalité (quand la vérification est faisable).

Le symbole $\rightsquigarrow$ utilisé ici veut dire que le terme de gauche peut servir d'argument

pour "défendre" l'existence d'un lien causal. Nous utilisons le terme défendre car il s'agit d'arguments insuffisants pour affirmer la causalité avec certitude.

2.1.2.9 Causalité et acyclicité

La composante temporelle sous-entend l'absence de cycle causal. Cependant, si l'antécédence temporelle est toujours vérifiable dans les instanciations des causalités (*i.e.* pour la causalité singulière), il se peut qu'elle ne le soit pas pour la causalité générique. En effet, lorsqu'on sait qu'il existe un lien causal entre deux propriétés, il se peut qu'il soit difficile, voire impossible de définir un sens unique pour ce lien (Drouet, 2007). Ceci peut être le cas par exemple entre le stress chez des opérateurs de fabrication et la présence de défauts sur les produits : l'apparition des défauts peut provoquer du stress, inversement, le stress peut également provoquer des erreurs de fabrication. Dans de tels cas, les deux sens de la causalité sont plausibles. Plusieurs situations où deux entités se causent mutuellement peuvent être citées pour rejeter l'acyclicité systématique de la causalité générique, comme par exemple le célèbre cercle vicieux entre la maladie et la faiblesse immunitaire (Williamson et al., 2005 ; Davis, 1988). Ceci n'exclue pas le fait que lors de l'observation de la manifestation du lien causal, nous observons un seul sens de la causalité à la fois.

2.1.2.10 Causalité et représentation graphique

La causalité est une relation qui se prête à la représentation graphique. La visualisation des liens de causalité sous forme de graphe est souvent préférée pour sa facilité d'interprétation (Wright, 1934). Les relations causales, caractérisées par leur aspect probabiliste, peuvent être représentées par des modèles graphiques probabilistes (Kechaou, 2020), où les noeuds représentent les propriétés étudiées , et les arcs représentent les liens de causalité entre les propriétés. Ce type de modèle fournit un schéma de représentation permettant de prendre en compte les dépendances et les indépendances conditionnelles entre les propriétés étudiées (Mechri, Simon, & Morel, 2017).

2.1.3 Conclusion

Du fait qu'aucune définition ne permet de cerner la notion de la causalité sans qu'elle ne soit contredite ou incomplète, les énoncés causaux sont souvent plus faciles à formuler qu'à évaluer. L'objectif de notre recherche est de fournir une aide à la décision basée sur la connaissance des liens de causalité. Afin qu'une décision soit bien fondée, il ne faut pas que les liens de causalité sur lesquels elle repose soient faux. Nous avons alors passé en revue ce qui semble caractériser les relations causales pour deux raisons :

— Identifier les liens de causalité à partir des observations et ;
— Vérifier la pertinence des liens de causalité identifiés ou déjà connus.

Dans cette section, l'exploration et la discussion des caractéristiques de causalité suggérées par les différents domaines a débouché sur la sélection de celles qui nous paraissent convenables et applicables dans notre contexte. Ces caractéristiques sont à prendre en compte lors de la recherche de liens causaux, ainsi que lors de l'évaluation de la pertinences des liens causaux identifiés ou déjà connus. Nous avons également expliqué les moyens qui permettent d'examiner si les relations entre les variables satisfont bien ces

caractéristiques ou non. Nous soulignons que lorsque la vérifiabilité n'est pas faisable, l'ensemble des autres caractéristiques n'est pas suffisant pour affirmer de façon certaine qu'un lien est effectivement causal. C'est justement parce que la notion de causalité résiste à toute définition exacte, et par manque de caractéristiques tranchantes de la causalité que nous adoptons **une approche plutôt pluraliste**. Nous avons donc essayé de lister une série de caractéristiques et d'arguments, qui, ajoutés les uns aux autres, nous permettrons **d'approcher la présomption de relation causale, et non pas de l'établir avec certitude**. La figure 2.4 résume ces caractéristiques et arguments, ainsi que les prendre en compte lors de la vérification et/ou identification des liens causaux, afin d'arriver à nos objectifs relatifs à l'analyse causale. Les connaissances humaines n'étant pas toujours disponibles, nous utilisons des pointillés pour tracer leur participation à l'atteinte des objectifs de construction de structure causale.

Cette section nous a permis d'identifier ce qui caractérise un lien de causalité. Pour remplir nos objectifs, nous avons également besoin de faire ressortir ce que nous attendons d'une analyse causale dans son ensemble. La prochaine section concernera les pratiques et méthodes de résolution de problèmes impliquant un raisonnement causal, usuellement employées dans le monde industriel et en ingénierie.

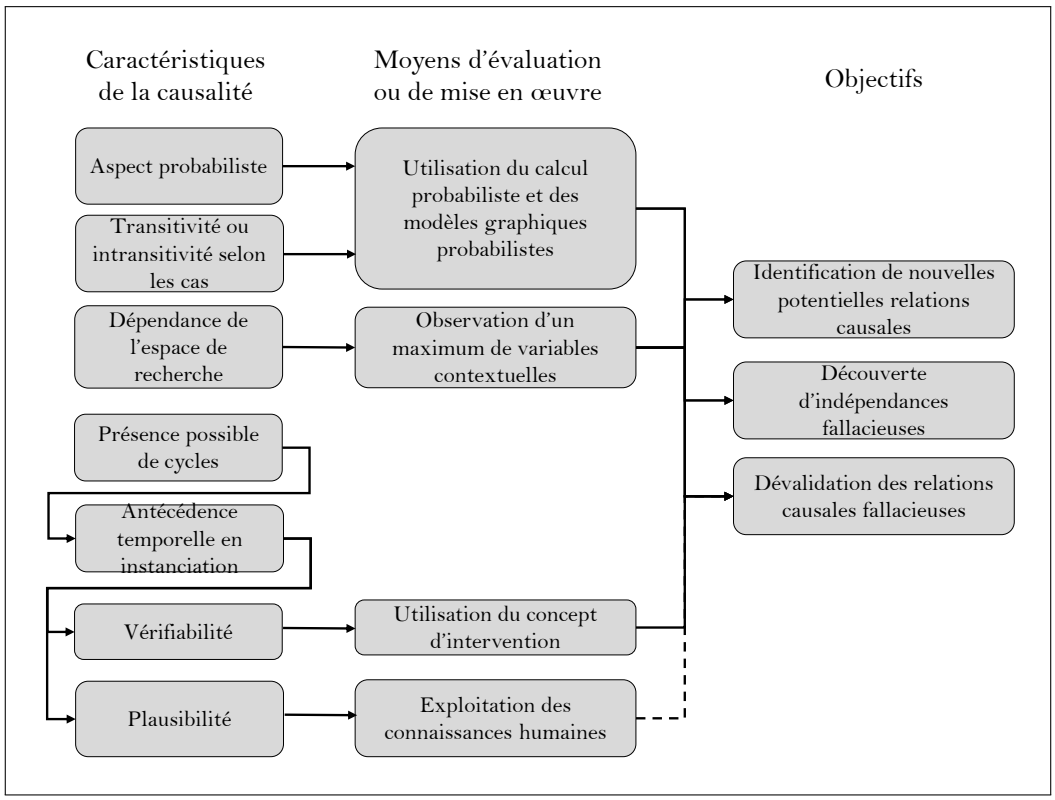


FIGURE 2.4 – Schéma explicatif des caractéristiques de la causalité et les moyens de les utiliser pour vérifier ou identifier des liens causaux, ainsi que les objectifs relatifs à l'analyse causale.

2.2 Analyse causale

Dans la section précédente, nous avons exploré les caractéristiques des liens causaux afin de pouvoir évaluer le type de relation qu'il y a entre deux entités, de manière à

approcher la présomption de causalité, ou au contraire d'écarter l'hypothèse selon laquelle la relation étudiée serait causale. L'objectif de cette section est d'introduire les méthodes de résolution de problèmes basées sur un raisonnement causal, afin de faire ressortir ce que nous attendons d'une analyse causale dans son ensemble, et de faire valoir l'intérêt de nos contributions que nous présenterons dans le prochain chapitre. Nous introduirons ensuite l'inférence causale à partir des données.

2.2.1 Les indicateurs clés de performance

Un indicateur clé de performance (KPIs), est un attribut quantifiable d'une entité ou d'une activité qui sert à décrire sa performance (Stark, 2016). Les entreprises utilisent souvent un ensemble de KPIs représentant un ensemble équilibré d'aspects pour mesurer les performances spécifiques pour chaque activité (comme la poduction, la chaine logistique, les ventes, etc.), ou générales à l'ensemble de l'entreprise (Liebetruth, 2017). La mesure des performances permet également de quantifier l'efficacité des actions entretenues (Neely, Gregory, & Platts, 1995), et représente ainsi un support pour orienter la décision, comme le montre la figure 2.5. Selon une enquête menée par (Moeuf, 2018), les experts considèrent qu'une sélection correcte de l'ensemble des paramètres à surveiller est une clé de réussite. La difficulté la plus importante réside dans le fait d'établir une définition des KPIs, de manière à ce qu'elle soit utile aux choix des actions à entretenir pour améliorer les performances. L'objectif principal derrière toute mesure de performance est de pouvoir réagir face aux aléas, ou de simplement entreprendre des actions pour améliorer une situation. Le résultat de l'action entreprise se reflète ensuite sur le même KPI. Ceci sous-entend que l'interprétation et l'analyse du KPI conditionne le choix de l'action à entreprendre, et par conséquent la mesure de performance à posteriori. En effet, plusieurs recherches ont confirmé que l'analyse des valeurs des KPIs et l'identification de leurs facteurs d'influence, souvent réalisées par des experts de manière empirique et descriptive, ne permettent pas une identification objective, exhaustive et exacte de toutes les causes directes et indirectes ayant une influence sur les KPIs (Pérez-Alvarez et al., 2018 ; Georgakopoulos et al., 2016 ; Nudurupati et al., 2007 ; Moeuf, 2018). Les KPIs représentent à la fois le déclencheur et la cible finale de plusieurs méthodes de résolution de problèmes, pendant que l'analyse causale en représente le fil conducteur.

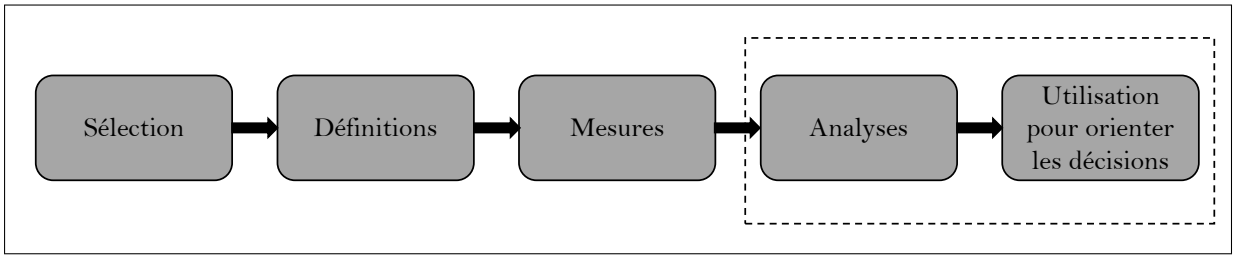


FIGURE 2.5 – Étapes et objectif de la mesure des performances par le biais des KPIs.

Dans notre contexte, nous nous intéressons à l'analyse des KPIs, et la façon dont elle oriente les décisions. Dans la suite de cette section, nous expliquerons alors en quoi l'analyse des KPIs n'est pas une tâche évidente, avant de présenter les méthodes usuelles de résolution de problèmes pouvant être régies déclenchées ou régies par les KPIs.

2.2.2 Interprétation des valeurs des KPIs

Afin de mesurer les performances, il faut d'abord poser la définition de la mesure, en sélectionnant les paramètres qui interviendront dans le calcul. Selon les objectifs et les enjeux de chaque activité, un ensemble de KPIs est sélectionné pour être suivi. La méthode de calcul de chaque KPI sélectionné est ensuite définie en fonction d'un ensemble de mesures quantifiables. Bien que ces définitions permettent de mesurer les performances de manière à ce qu'elles reflètent fidèlement l'état de l'entité étudiée, elles ne permettent cependant pas de diriger les actions à entreprendre. En effet, lorsque le KPI indique une déviation, la définition grâce à laquelle il a été mesuré ne fournit pas assez d'éléments profitables à la prise de décision. Par exemple, si l'indicateur surveillé est le taux de rendement synthétique (TRS), la formule pour le calculer est donnée par la formule (2.11) :

$$TRS = \frac{Production\ réelle}{Production\ maximum\ théorique} \tag{2.11}$$

Si une déviation par rapport à un seuil prédéfini est détectée, la définition de base donnée par (2.11) n'est pas suffisante pour identifier les éléments sur lesquels une action directe peut être entreprise pour rétablir la situation de manière efficace. En effet, comme il n'est pas possible d'agir sur le temps de production, puisque ce n'est pas un paramètre manipulable directement, cette formule ne permet pas, à elle seule, d'engager une action concrète sans avoir à chercher les causes sous-jacentes de la déviation. Comme illustré sur la figure 2.6, la phase d'analyse comprend la recherche de liens entre le KPI en question et les variables contextuelles manipulables (*i.e.* sur lesquelles une action peut être entreprise directement). La complexité de cette analyse relève du fait que les dépendances entre toutes les données disponibles et les KPIs sont inconnues (Rabenoro, 2015).

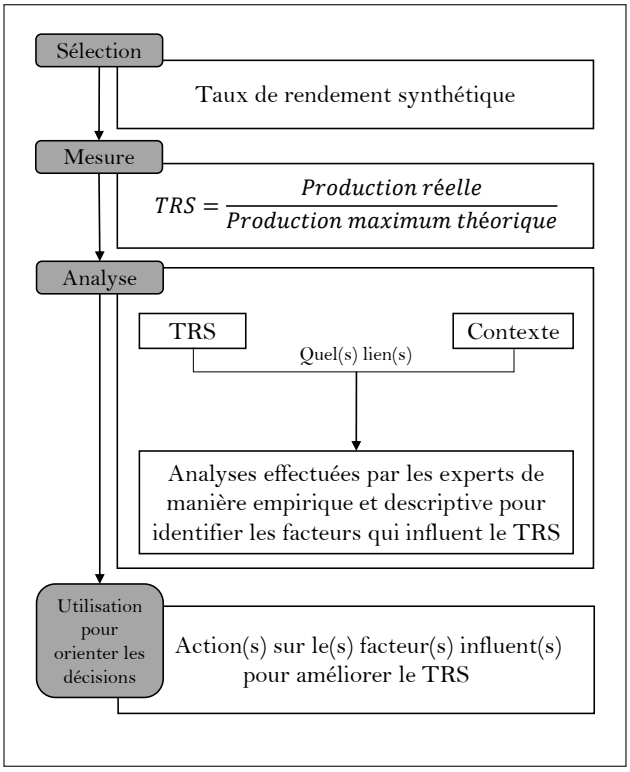


FIGURE 2.6 – Illustration des phases de l'exploitation d'un KPI en utilisant une analyse descriptive.

2.2.3 Analyses descriptives des KPIs

— **Les cinq pourquoi**

Comme son nom l'indique, cette méthode a pour objectif de remonter aux causes racines d'un problème en reposant à plusieurs reprises la question pourquoi. Cette méthode soutient l'idée selon laquelle **l'analyse causale doit être exhaustive** afin de pouvoir éradiquer le problème depuis sa source. Bien que l'idée ambitieuse d'exhaustivité soit pertinente, cette méthode descriptive souffre d'inconvénients sur plusieurs aspects. D'abord, elle contient une part importante de subjectivité, et ses conclusions peuvent changer en fonction de la personne qui la conduit. Ensuite, pour contrer les problèmes de subjectivité, elle mobilise beaucoup de ressources, puisqu'elle se déroule souvent en groupe. En outre, pour limiter les biais, il est préconisé de la déployer en l'absence des personnes directement impliquées dans le problème traité. Par ailleurs, le temps consacré au déploiement de cette méthode est également conséquent.

— **Diagramme d'Ishikawa**

Également appelé diagramme de causes à effets, ou diagramme des 5M, ce diagramme est très utilisé, compte tenu de sa généricité et applicabilité à plusieurs types de problèmes (Liliana, 2016). Il a pour principe de classifier les causes selon leurs catégories (notamment des causes liées aux méthodes, aux matières, à la main d'œuvre, au milieu, ou aux machines). Cependant, avoir des résultats pertinents en utilisant le diagramme d'Ishikawa nécessite un travail d'équipe. Ainsi, à l'instar de la méthode des 5P, cette méthode est sujette à des problèmes de mobilisation de ressources importantes et de subjectivité. Elle inspire tout de même un aspect notable pour l'analyse de causalité : tenter d'atteindre l'exhaustivité en cherchant au delà du périmètre du problème, à travers **l'exploration du contexte dans sa globalité**.

— **AMDEC**

L'Analyse des Modes de Défaillances, de leurs Effets et de leur Criticité est une méthode d'analyse de risques faisant appel aux connaissances des équipes en ingénierie et de fiabilité pour optimiser la qualité et la sûreté d'un système, une conception, un processus, ou un service (Stamatis, 2003). Elle aide ainsi à la sélection des alternatives avec une fiabilité et un meilleur potentiel de sûreté (Stamatis, 2003). Nous la citons parmi les méthodes de résolution de problèmes, puisqu'elle traite des risques et repose sur un raisonnement causal. Pareillement aux deux méthodes préalablement citées, l'AMDEC requiert du temps et un travail d'équipe, et reste subjective. Par ailleurs, elle nécessite une identification préalable des évènements redoutables (Faucher, 2009), ce qui a pour conséquence un manque de réactivité face à l'apparition d'évènements non traités *à priori*. Dans cette méthode, la quête d'exhaustivité passe par **la décomposition du système traité** et l'étude de ses composants.

— **QQOQCP**

Il s'agit de la méthode "Quoi, Qui, Où, Quand, Comment, Pourquoi", qui est souvent labellisée comme méthode d'analyse causale et de résolution de problèmes basée sur un questionnement méthodique. Son objectif est avant tout d'**identifier et caractériser les circonstances** dans lesquelles le problème a émergé. Cette méthode souffre des mêmes inconvénients que celles précédemment présentées.

— **Diagramme de Paretto**

Il ne s'agit pas ici d'une méthode d'analyse causale à proprement parler, puisque l'élaboration du diagramme de Paretto suppose une connaissance préalable des causes, ou du moins les macro-causes. Nous le citons tout de même ici parce qu'il soulève un aspect intéressant pour l'analyse causale qu'est **le classement des causes par ordre d'importance ou du moins de récurrence**.

— **Arbre de causes**

Semblablement au diagramme d'Ishikawa, cette méthode adopte une représentation graphique, à la différence qu'elle ne classe pas les causes par catégorie. Cette méthode fait appel à un raisonnement causal arborescent. Elle représente la forme la plus simplifiée **des modèles graphiques causaux** lorsqu'elle ne fait pas intervenir de calcul probabiliste, et possède une déclinaison : l'arbre de défaillances, qui est basé sur le calcul de la probabilité de l'évènement indésirable à partir des probabilités d'occurrences des causes racines (Mortureux, 2002). L'utilisation d'un arbre de défaillances suppose une identification préalable de l'évènement indésirable, tandis que les arbres de causes classiques sont souvent utilisés à posteriori de l'évènement indésirable (Mortureux, 2002).

— **Graphes causaux**

Les graphes causaux représentent une généralisation des arbres causaux. Ils s'en distinguent par le fait que les liens causaux ne sont pas organisés sous forme d'arbres. Les structures de réseaux Bayésien sont un exemple des arbres causaux. Ces graphes sont catégorisés parmi les méthodes descriptives lorsqu'ils sont spécifiés par les experts, et qu'ils possèdent des probabilités associées estimées par les experts ou ne possèdent pas de probabilités associées du tout. Dans un tel cas, ils souffrent des mêmes inconvénients que les arbres de causes, mais permettent une meilleure représentation lorsque les liens causaux ne sont pas arborescents.

— **Standards et méthodes internes**

Afin de résoudre un problème, les entreprises possèdent souvent une feuille de route bien définie permettant, entre autres, de remonter aux causes racines du problème. Cette feuille de route peut aussi bien être composée de méthodes propres à l'entreprise et d'une ou plusieurs méthodes précédemment citées, qu'être exclusivement interne. Les standards internes, qu'ils soient analytiques ou descriptifs, restent en tout état de cause propres à l'entreprise pour laquelle ils ont été conçus, et manquent

ainsi de **généricité**. Ceci peut représenter un frein au sein même de l'entreprise génératrice du standard en cas de changements importants, ce qui a pour conséquence un manque d'agilité.

Comme nous venons de le soulever, l'ensemble de ces analyses descriptives partagent des inconvénients communs : subjectivité, implication importante de ressources humaines, et aspect chronophage. La négation de ces trois principaux inconvénients constituent alors pour nous des caractéristiques auxquelles nous aspirons pour notre analyse causale. De plus, les analyses descriptives présentées ci-dessus nous ont permis d'identifier des attributs intéressants, qui feraient également partie des caractéristiques que notre analyse causale devrait idéalement satisfaire. Par conséquent, pour qu'une analyse causale soit profitable à la prise de décision de manière efficace, il faudrait, selon nous, qu'elle satisfasse les caractéristiques suivantes :

— Objectivité ;
— Implication minimale des ressources humaines ;
— Rapidité ;
— Exhaustivité ;
— Prise en compte des variables de contexte ;
— Généricité ;
— Prise en compte de l'importance des causes ;
— Respect des caractéristiques de causalité (identifiées dans la section précédente).

L'ensemble de ces caractéristiques sera utilisé lors de la validation de la méthode proposée, au chapitre 5.

Les analyses descriptives, ne fournissant pas assez d'éléments pour satisfaire l'ensemble de ces caractéristiques à la fois, sont alors inexploitables pour atteindre notre objectif. C'est pourquoi nous parlerons dans la sous-section suivante de l'inférence causale à partir des données.

2.2.4 Inférence causale à partir des données

Le constat selon lequel les données deviennent de plus en plus accessibles grâce à l'accroissement de l'utilisation des nouvelles technologies en industrie nous amène à poser nos postulats. En effet, nous considérons que les réponses que nous souhaitons amener à nos fonctions principales définies dans le chapitre 1, reposent sur le fait que les environnements de production peuvent être équipés de systèmes d'acquisition permettant de récupérer toutes les données dont on pourrait avoir besoin. Ceci représente le premier postulat P1 introduit dans le premier chapitre, et est fondé sur la constatation d'une utilisation croissante des systèmes d'acquisition de données (Kamienski et al., 2019), ainsi que sur la chute continuelle des coûts d'acquisition des IoT dans le milieu industriel, comme le montre la figure 2.7 (Microsoft, 2019). Nous tenons alors à préciser que les problématiques liées aux contraintes d'acquisition des données sont hors de notre périmètre.

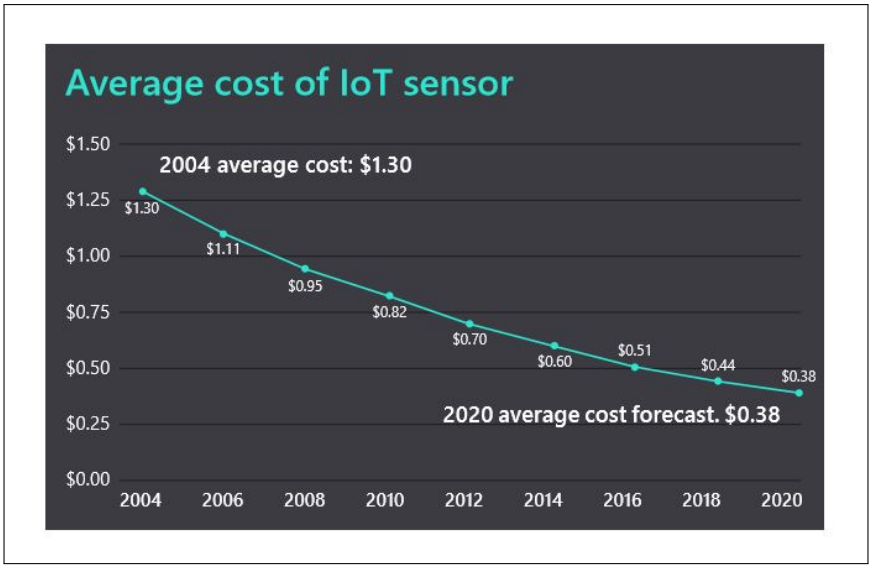


FIGURE 2.7 – Courbe décrivant la chute du coût moyen des IoT.

Le deuxième postulat (P2) que nous posons, concerne la possibilité de récupérer les données historiques issues de ces systèmes d'acquisition, ainsi que des valeurs historiques mesurées des KPIs que l'on souhaiterait analyser. Ces deux postulats sont essentiels, puisque pour inférer des liens de causalité à partir de données, il nous faut des données à partir desquelles l'inférence sera faite (P2), et pour collecter ces données, il nous faut des systèmes d'acquisition (P1).

Ainsi, afin d'améliorer la prise de décision par le biais de l'analyse causale, nous proposons d'abord de transformer la figure 2.6 décrivant les pratiques usuelles d'interprétation de KPIs via une analyse causale descriptive, par la figure 2.8, qui décrit l'idéal auquel nous aspirons. En l'occurrence, l'analyse descriptive des valeurs des KPIs est alors remplacée par une inférence causale.

2.2.4.1 Analyse causale à partir des données

Plusieurs techniques de statistiques traditionnelles permettent de décrire les relations entre les variables. Ces techniques, notamment celles de régression, d'estimation, et de test d'hypothèses, permettent surtout de déduire des associations entre des variables, ou d'estimer les probabilités d'évènements passés ou à venir (Pearl, 2009a). Dans le contexte de détection de relations entre variables, nous pouvons citer le test de Pearson, le test de Spearman, les différentes déclinaisons de l'analyse factorielle, celles de l'analyse de la variance. Si la fouille de données exploratoire est très performante pour repérer des corrélations, en particulier des corrélations subtiles qu'une analyse d'ensembles de données plus petits pourrait manquer, les analyses de la Big data ne révèlent pas nécessairement la causalité, et n'informent pas sur la pertinence et la signification des corrélations détectées [2].

Dans un autre registre, les plans d'expériences permettent de fournir des éléments per-

2. https ://www.nytimes.com/2014/04/07/opinion/eight-no-nine-problems-with-big-data.html

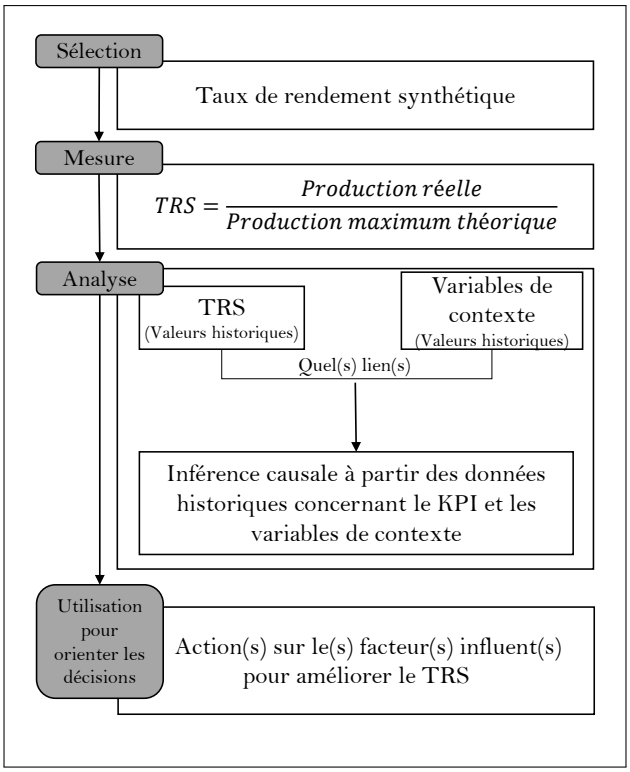


FIGURE 2.8 – Illustration des phases d'exploitation d'un KPI en utilisant l'inférence causale.

tinents pour l'identification des liens de causalité. Généralement utilisés dans un souci de qualité, les plans d'expériences permettent d'évaluer les effets d'un ensemble de variables sur la variable d'intérêt. Bien qu'ils aient pour objectif de fournir un maximum d'information tout en minimisant le nombre d'expériences à effectuer (Goupy, 2000), ils demeurent coûteux en termes de temps de mise en œuvre, de temps d'analyse des résultats, et de ressources humaines.

Les techniques que nous venons de citer ici se basent sur les données et permettent de limiter la subjectivité humaine dans le processus de prise de décision (Tufféry, 2011). Néanmoins, aucune de ces techniques ne satisfait les caractéristiques que nous nous sommes fixées pour établir l'analyse causale.

Comme nous l'avons cité dans la section 2.1, les modèles graphiques probabilistes se prêtent à la description et l'interprétation des liens de la causalité. Ceci est d'autant plus avéré dans notre contexte, puisque, comme mentionné dans le chapitre 1, nous nous intéressons aux situations de prise de décision incertaines ou à risques. Les modèles graphiques probabilistes présentent l'avantage de pouvoir prendre en considération les incertitudes, et sont donc en effet adaptés à ce genre de situations (Koehler, 1996 ; Kechaou, 2020). En particulier, les réseaux Bayésiens causaux offrent une structure de graphe causal ordinaire, avec en plus des probabilités (Faghraoui, 2013), permettant ainsi de construire un modèle de raisonnement causal probabiliste. Dans la suite, nous allons présenter les réseaux Bayésiens, avant d'introduire les réseaux Bayésiens causaux. Par la suite, et comme l'un de nos objectifs est de limiter les problèmes d'interprétation liés à la subjectivité humaine, nous allons explorer les techniques d'apprentissage des réseaux Bayésiens à partir des données.

2.2.4.2 Les réseaux Bayésiens

Un réseau Bayésien est un modèle graphique caractérisé par une structure composée de nœuds et d'arcs, ainsi que par des probabilités conditionnelles ou marginales affectées aux nœuds (Taroni, Aitken, Garbolino, & Biedermann, 2006). Le graphe G associé à un réseau Bayésien est un graphe acyclique dirigé (DAG), tel que $G = \langle V, E \rangle$, où $V = \{V_1, V_2, ..., V_n\}$ est un ensemble fini de variables discrètes représentées par les nœuds du graphe et pouvant prendre un nombre fini de valeurs ; et E un ensemble fini d'arcs reliant les variables entre elles. Un réseau Bayésien R est alors défini par le couple (G, P), où P représente la distribution de probabilités conditionnelles et marginales associée au graphe G. Ainsi, nous distinguons deux composantes principales d'un réseau Bayésien : une composante graphique ou structurelle qu'est le DAG associé, et une composante numérique qu'est la distribution de probabilités associée (Ben Amor, 1988). Un réseau Bayésien peut être utilisé dans l'objectif de 1) représenter schématiquement et de manière compacte les hypothèses sur les indépendances, ou de 2) représenter les influences causales dans un langage graphique (Pearl, 1995). Dans notre cas, c'est plutôt ce deuxième objectif qui nous intéresse : nous présentons donc ici les réseaux Bayésiens, pour parler ensuite des réseaux Bayésiens causaux, puisque tous les réseaux Bayésiens ne sont pas nécessairement causaux (Pearl, 2009b). En effet, l'interprétation d'un réseau Bayésien est valide à partir du moment où ce dernier traduit correctement les dépendances et indépendances probabilistes entre les variables. Or, nous avons vu dans la sous-section 2.1.2.5 que la causalité implique la dépendance, mais que la réciproque est fausse, dans le sens où la dépendance implique corrélation et non pas causalité. Dans un réseau Bayésien causal, les arcs reliant les variables représentent alors les liens de causalité probabilisés, où la cause est représentée par l'origine de l'arc, et l'effet par son extrémité.

— **Parents Markoviens**

Pour chaque nœud V_i nous désignerons l'ensemble des parents immédiats de V_i par la notation $Pa(V_i)$. Si le nœud A est un parent du nœud B, ceci sera représenté sur le graphe par un arc allant de A vers B, on dira également de B est l'enfant (ou le descendant) de A. Nous écrirons alors $A \rightarrow B$. À chaque noeud V_i est associée une distribution de probabilité $\mathbb{P}(V_i|Pa(V_i))$, la condition porte uniquement sur les parents direct du nœud V_i (Boreux, Parent, & Bernier, 2009), ce qui a pour avantage de réduire considérablement le nombre de probabilités conditionnelles à stocker ou à estimer (François, 2006 ; Bellot, 2002). De plus, la représentation des indépendances conditionnelles doit satisfaire la condition de Markov, qui établit que chaque variable est indépendante de ses non descendants conditionnellement à ses parents directs. Ce qui veut dire que si $\{V_1, V_2, ..., V_{i-1}\}$ est l'ensemble des non-descendants de V_i, on a l'égalité donnée par (2.1) :

$$\mathbb{P}(V_i|V_{i-1}, ..., V_2, V_1) = \mathbb{P}(V_i|Pa(V_i)) \tag{2.12}$$

En utilisant la règle de chaînage, nous pouvons écrire la probabilité jointe de l'ensemble

$\{V_1, V_2, ..., V_n\}$ sous la forme suivante :

$$\begin{aligned}\mathbb{P}(V_1, V_2, ..., V_n) &= \mathbb{P}(V_n|V_{n-1}, ..., V_2, V_1).\mathbb{P}(V_{n-1}, ..., V_2, V_1) \\ &= \mathbb{P}(V_n|V_{n-1}, ..., V_2, V_1).\mathbb{P}(V_{n-1}|V_{n-2}, ..., V_2, V1).\mathbb{P}(V_{n-2}, V_{n-3}, ..., V_2, V_1) \\ &= \mathbb{P}(V_n|V_{n-1}, ..., V_2, V_1).\mathbb{P}(V_{n-1}|V_{n-2}, ..., V_2, V_1). \ ... \ .\mathbb{P}(V_2|V_1).\mathbb{P}(V_1) \\ &= \prod_i^n \mathbb{P}(V_i|V_{i-1}, ..., V_2, V_1) \end{aligned} \tag{2.13}$$

Pour un réseau Bayésien, l'égalité (2.12) étant satisfaite, il peut alors être déduit de l'égalité (2.13), la formule donnée par (2.14) :

$$\mathbb{P}(V_1, V_2, ..., V_n) = \prod_i \mathbb{P}(V_i|Pa(V_i)) \tag{2.14}$$

Ainsi, les parents $Pa(V_i)$ de V_i sont également appelés *Parents Markoviens*, en référence à la condition de Markov. Ils sont alors définis comme étant "un ensemble minimal de prédécesseurs de V_i, rendant V_i indépendant de tous ses autres prédécesseurs (ou non descendants)" (Bellot, 2002). Dit autrement, les parents Markoviens $Pa(V_i)$ forment un sous-ensemble de V_i suffisant pour calculer la probabilité de V_i. Si un nœud V_i n'a pas de parents, sa distribution de probabilités est alors "inconditionnelle" (*i.e.* marginale), ou *a priori* ; à l'inverse, s'il en a, sa distribution de probabilités est alors conditionnelle.

— **Couverture de Markov**

Pour un nœud V_i, ses parents directs, ses enfants directs, ainsi que ses époux, représentent sa couverture de Markov. Un nœud V_j est l'époux du nœud V_i si V_i et V_j ont un enfant en commun. La figure 2.9 illustre la notion de couverture de Markov dans un réseau Bayésien.

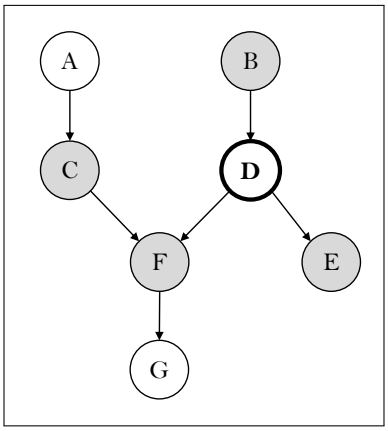
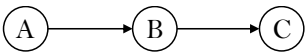
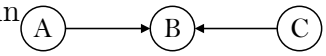


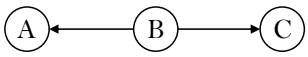
FIGURE 2.9 – Mise en évidence de la couverture de Markov du nœud D.

— **Types de connexion**

Dans un réseau Bayésien, on peut distinguer trois types de connexions (Naïm, Wuillemin, Leray, Pourret, & Becker, 1999) :

- Connexion en série : A est une cause de B, et B cause de C. 

- Connexion convergente (ou V-structure) : B est un effet commun de A et C. 

- Connexion divergente : B est une cause commune de A et C. 

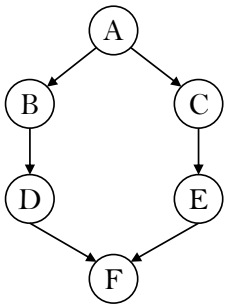
— ***D-séparation***

La règle de la *d-séparation* est composée d'un ensemble de critères permettant d'identifier les indépendances conditionnelles (Pearl, 1995). En d'autres termes, elle permet d'identifier si un nœud A est indépendant d'un autre nœud B, sachant un ensemble non vide de nœuds C : c'est à dire lorsque C est observé. Deux nœuds A et B indépendants sachant C si **tous** les chemins entre A et B sont bloqués par un ou plusieurs nœuds appartenant à C. C'est justement cette règle de *d-séparation* qui permet de dire, si oui ou non, un chemin est entre deux nœuds est bloqué.

Un chemin entre A et B est bloqué par un ensemble de nœuds C si :

- s'il contient un connexion en série $A \to D \to B$, ou $B \to D \to A$, où $D \in C$: le chemin est alors bloqué conditionnellement à D. Ou bien ;
- s'il contient une connexion divergente $A \leftarrow D \to B$, où $D \in C$: le chemin est alors bloqué conditionnellement à D. Ou bien ;
- s'il contient un connexion convergente $A \to E \leftarrow B$, où $E \notin C$, et tel qu'aucun descendant de D n'appartient à E : le chemin est alors bloqué conditionnellement à E.

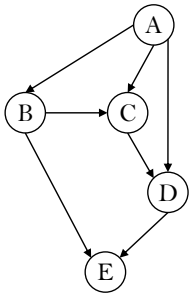
Nous concevons que cette notion de *d-séparation* n'est pas très intuitive de prime abord. Nous l'explicitons donc avec un exemple. Considérons le réseau Bayésien suivant :



Nous voulons évaluer l'indépendance entre les nœuds C et D conditionnellement à A, les seules variables observées sont alors A, C, et D. Afin d'évaluer cette indépendance conditionnelle, il faut vérifier que tous les chemins liant C et D sont bloqués par le nœud A. Il y a un premier chemin $D - B - A - C$, ce chemin contient la connexion divergente $B \leftarrow A \to C$, et cette connexion est bloquée par A, donc le chemin entre D et C est bloqué par A. Le deuxième chemin entre C et D est $C - E - F - D$, ce chemin contient une connexion convergente, avec un effet commun F non observé ($F \notin \{A\}$). Cette connexion est alors bloquée par A. Le chemin est alors bloqué par A. Les deux seuls chemins reliant

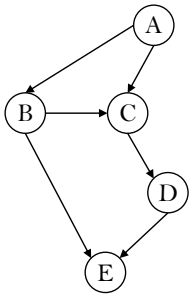
C et E sont bloqués par A. On en déduit que C est indépendant de D conditionnellement à A ($C \perp\!\!\!\perp D|A$). On a alors aussi : $\mathbb{P}(C|D,A) = \mathbb{P}(C|A)$, c'est à dire que lorsque A est observée, l'observation de D n'apportera aucune information supplémentaire sur C. Nous avons donc également $\mathbb{P}(D|C,A) = \mathbb{P}(D|A)$, c'est à dire que lorsque A est observée, l'observation de C n'apporte aucune information supplémentaire sur D.

Prenons maintenant le graphe suivant, et évaluons l'indépendance entre B et D conditionnellement à C :



Un premier chemin $B - A - D$ liant B et D est une connexion divergente $B \leftarrow A \rightarrow D$ où A n'est pas observée ($A \notin \{C\}$), ce chemin n'est alors pas bloqué. Il est inutile de vérifier le blocage des autres chemins, puisque pour que B et D soient indépendants sachant C, il faut que tout les chemins soient bloqués par C. Nous concluons donc que $B \not\perp\!\!\!\perp D \mid C$. L'observation de D apporte alors une information supplémentaire sur B lorsque C est observée, et réciproquement.

Modifions maintenant ce même graphe, en enlevant l'arc entre A et D. Nous obtenons alors le graphe suivant :



Si les deux graphes précédents décrivent des liens de causalité, une première fausse intuition que l'on pourrait avoir est que l'information circule de la même manière dans les deux graphes. Ceci est faux, puisque si nous examinons cette fois l'indépendance entre les mêmes nœuds B et D conditionnellement au nœud C, nous obtiendrons des résultats différents. En effet, un premier chemin reliant B et D est la connexion en série $B - C - D$, qui est donc bloquée par C. Le deuxième chemin est une connexion convergente $B - E - D$, qui est également bloquée par C, puisque la connexion est convergente en E, et on sait que $E \notin \{C\}$, et E n'a pas de descendants. Le troisième chemin $B - A - C - D$ contient une connexion en série $A - C - D$ qui est bloquée par C. Les trois chemins sont donc bloqués par C. Nous en déduisons alors que $B \perp\!\!\!\perp D|C$, par conséquent, si C est observée,

l'observation de B n'apportera pas d'information supplémentaire sur D, et inversement.

Nous tenons à préciser que la *d-séparation* est une notion purement graphique qui permet de déduire les indépendances conditionnelles à partir d'un graphe supposé correct (Naïm et al., 1999). La comparaison des deux derniers graphes ci-dessus nous amène à revenir sur la notion de transitivité évoquée dans la section 2.1. Nous précisons que ces deux graphes ne représentent pas la même chose, et que dans le premier graphe, la fermeture transitive de la chaîne $A \rightarrow C \rightarrow D$ reflète les dépendances et indépendances probabilistes entre les variables. Pour le deuxième graphe, les dépendances et indépendances probabilistes entre les variables dans ce cas là ne permettent pas la fermeture transitive de la chaîne $A \rightarrow C \rightarrow D$. La transitivité de la causalité n'est alors pas systématique, et est conditionnée par les dépendances et indépendances entre les variables étudiées.

2.2.4.3 Probabilités *a priori*

Une table de probabilités conditionnelles doit être associée à chacun des nœuds ayant un ou plusieurs parents. Concernant les nœuds n'ayant pas de parents, leurs probabilités marginales doivent alors être spécifiées. Compte tenu du fait que la valeur d'une variable représentée par un nœud dépend seulement des valeurs que prennent les parents de ce nœud, il suffit de spécifier les probabilités du nœud en question conditionnellement à ses parents, au lieu de spécifier les tables de probabilités jointes complètes. Ceci a pour conséquence de restreindre drastiquement la complexité de l'estimation des probabilités conditionnelles. En effet, si un réseau Bayésien contient n variables, le nombre de probabilités à spécifier devient linéaire en n : la complexité est de $O(nm)$ au lieu de $O(2^n)$ si l'on veut spécifier toutes les probabilités conditionnelles ; avec m le nombre maximum de parents (Stuart J. Russell, 2016). Par exemple, pour le graphe illustré sur la figure 2.10, le nombre de probabilités à spécifier sera de $2 \times 5 = 10$ au lieu de $2^5 = 35$.

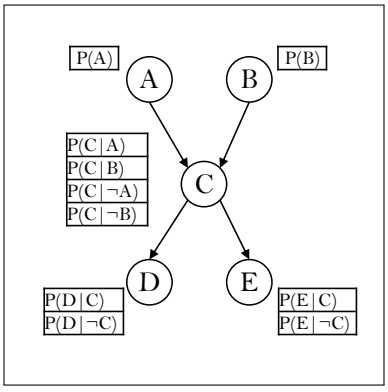


FIGURE 2.10 – Probabilités conditionnelles et marginales à estimer pour un réseau Bayésien.

L'estimation des paramètres d'un réseaux Bayésiens (*i.e.* les probabilités conditionnelles ou marginales des nœuds du réseau), peut se faire par le biais d'une estimation "manuelle" en interrogeant plusieurs experts du domaine d'étude sur leurs croyances (Kechaou, 2020). En général, cette estimation est précédée par une définition de la structure du réseau Bayésien en adoptant la même démarche. Cette méthode a pour avantage que la structure définie par les experts est souvent garantie d'être construite dans un raisonnement causale. Néanmoins, elle demeure subjective, et les experts peuvent omettre certaines relations causales sous-jacentes, ou qualifier une relation de causale alors qu'elle

ne l'est pas en réalité. Dans ce cas, les probabilités définies en se basant sur une telle structure seraient simplement fausses. Et quand bien même la structure soit parfaitement exacte, une définition de probabilités basée sur une estimation manuelle ne peut être que fortement approximative. La hiérarchisation des causes qui pourra en découler aura donc de fortes chances de ne pas être fiable.

Lorsque la structure est fixée et supposée correcte, l'approche fréquentiste peut être utilisée pour estimer les probabilités marginales et conditionnelles des nœuds du réseaux (Naïm et al., 1999). Par exemple, pour estimer la probabilité marginale qu'une variable multinomiale V_i prenne la valeur v_i par $\mathbb{P}(V_i = v_i) \approx \frac{N_{V_i=v_i}}{N}$. La probabilité est alors assimilée à la fréquence d'observation, il s'agit donc d'une probabilité fréquentielle ou expérimentale, qui peut être obtenue *a posteriori* d'une suite d'observations, en faisant le ratio entre le nombre de cas favorables et le nombre de cas possibles. Cette approximation de probabilité doit cependant obéir à la loi des grands nombres, selon laquelle la probabilité peut être interprétée comme la fréquence de réalisation avec plus de certitude lorsque la taille de l'échantillon tend vers l'infini. Or, dans notre contexte, nous avons supposé, par le biais du postulat P2, la disponibilité d'une quantité massive de données compte tenu de l'utilisation de plus en plus répandue des systèmes d'acquisition des données. Ce postulat suppose également la disponibilité et la complétude des données historiques, et donc des observations à partir desquelles des fréquences de réalisation peuvent être calculées. Nous en déduisons que l'estimation des probabilités à partir des fréquences de réalisation peut être réalisable dans notre contexte, à partir du moment où la structure du réseau est fixée, et que de cette manière, le caractère subjectif de l'estimation des probabilités serait minimisé, et que la présomption à l'exactitude des probabilités estimées serait plus approchée.

2.2.5 Réseaux Bayésiens causaux

Comme précédemment mentionné, tous les réseaux Bayésiens ne sont pas forcément des réseaux Bayésiens causaux. En effet, pour les mêmes probabilités conditionnelles et marginales, plusieurs graphes peuvent être équivalents, mais un seul parmi ces graphes pourrait capturer les liaisons causales (Leray, 2006). Une classe d'équivalence regroupe tous les réseaux équivalents. La figure 2.11 illustre trois réseaux Bayésiens équivalents.

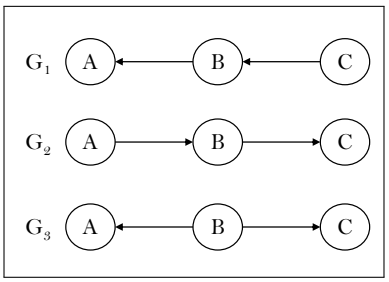


FIGURE 2.11 – Réseaux Bayésiens équivalents.

Considérons les trois réseaux Bayésiens de la figure 2.11, tels que A, B, et C représentent les mêmes variables dans les trois graphes. Nous allons à présent montrer que les graphes G_1, G_2, et G_3 codent les mêmes informations en termes de probabilités, bien que

l'interprétation en termes de causalité ne soit pas la même visuellement. En effet, pour le graphe G_1, en utilisant la règle de chaînage et la condition de Markov (voir l'égalité donnée par (2.14)), on a :

$$\mathbb{P}_{G_1}(A, B, C) = \mathbb{P}(A|B) \times \mathbb{P}(B|C) \times \mathbb{P}(C) \tag{2.15}$$

Pour le graphe G_2, en utilisant encore une fois la règle de chaînage et la condition de Markov, ainsi que la règle de Bayes selon laquelle $\mathbb{P}(X|Y) = \dfrac{\mathbb{P}(Y|X) \times \mathbb{P}(X)}{\mathbb{P}(Y)}$, on a :

$$\begin{aligned}
\mathbb{P}_{G_2}(A, B, C) &= \mathbb{P}(A) \times \mathbb{P}(B|A) \times \mathbb{P}(C|B) \\
&= \mathbb{P}(A) \times \frac{\mathbb{P}(A|B) \times \mathbb{P}(B)}{\mathbb{P}(A)} \times \frac{\mathbb{P}(B|C) \times \mathbb{P}(C)}{\mathbb{P}(B)} \\
&= \mathbb{P}(A|B) \times \mathbb{P}(B|C) \times \mathbb{P}(C)
\end{aligned} \tag{2.16}$$

Pour le graphe G_3, en utilisant, comme pour le graphe G_2 la règle de chaînage, la condition de Markov, ainsi que la règle de Bayes, on a :

$$\begin{aligned}
\mathbb{P}_{G_3}(A, B, C) &= \mathbb{P}(A|B) \times \mathbb{P}(C|B) \times \mathbb{P}(B) \\
&= \mathbb{P}(A|B) \times \frac{\mathbb{P}(B|C) \times \mathbb{P}(C)}{\mathbb{P}(B)} \times \mathbb{P}(B) \\
&= \mathbb{P}(A|B) \times \mathbb{P}(B|C) \times \mathbb{P}(C)
\end{aligned} \tag{2.17}$$

A partir des égalités (2.15), (2.16), et (2.17), nous déduisons que $\mathbb{P}_{G_1}(A, B, C) = \mathbb{P}_{G_2}(A, B, C) = \mathbb{P}_{G_3}(A, B, C)$. Ces trois graphes sont donc des représentations différentes des mêmes indépendances conditionnelles, et des mêmes probabilités jointes. Pour les trois graphes, on a : $A \not\perp\!\!\!\perp B$, $B \not\perp\!\!\!\perp C$, et $A \perp\!\!\!\perp C|B$. Ceci peut être directement déduit des graphes en utilisant la notion de la *d-séparation*. Dans ces conditions, si des liens directs de causalité existent effectivement entre A et B, et entre B et C, un seul parmi ces trois graphes les modélise correctement. Ceci est cohérent avec la distinction entre la causalité et la corrélation que nous avons abordée dans la section 2.1.

La question qui se pose désormais est comment construire un réseau Bayésien capturant les liens de causalité de manière objective (*i.e.* en se basant uniquement sur les données), et dans le cas où cela ne serait pas possible, comment construire un réseau Bayésien causal en maximisant l'exploitation des données observées, et en minimisant la subjectivité humaine.

2.2.6 Apprentissage des réseaux Bayésiens

Afin de limiter la subjectivité liée à la construction des réseaux bayésiens, l'apprentissage constitue un atout fort, puisqu'il permet d'exploiter au maximum les données. Comme le montre la figure 2.12, il existe deux grandes familles dans le domaine de l'apprentissage des réseaux Bayésiens : l'apprentissage de la structure, et l'apprentissage des paramètres. L'apprentissage des paramètres est conditionné par le fait que la structure soit connue. Nous nous intéressons donc à l'apprentissage de la structure d'un réseau Bayésien. Une fois la structure connue, les paramètres peuvent être estimés d'une manière fréquentielle. Afin d'approcher la présomption d'exactitude d'une structure de réseau Bayésien causal, il est nécessaire que la suffisance causale soit approchée ; c'est à dire que pour une variable

d'intérêt, il ne faut pas qu'il y ait de variable qui soit cause de la variable d'intérêt en question, et qui ne soit pas observée. Ceci traduit la caractéristique de causalité relative à la dépendance de l'espace de recherche que nous avons précédemment identifiée, ainsi que la nécessité de la prise en compte et donc l'observation d'un maximum de variables de contexte, que nous avons soulevée dans les sections 2.1 et 2.2 comme étant une condition *sine qua non* à la conduite d'une analyse causale.

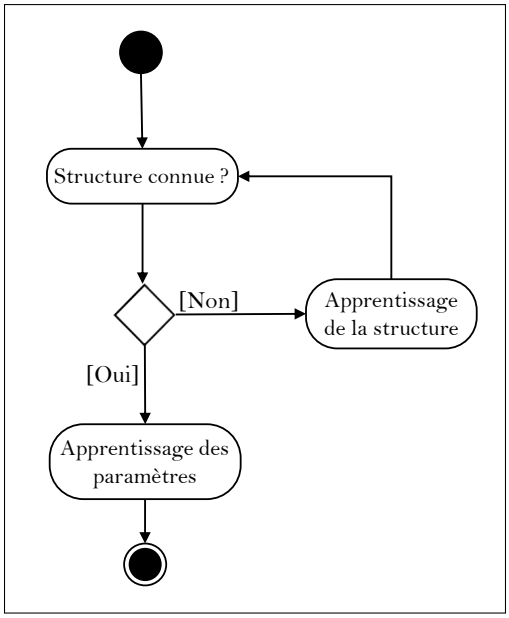


FIGURE 2.12 – Diagramme d'activités correspondant au processus de construction de réseaux Bayésiens par apprentissage.

2.2.6.1 Un problème NP-Complet

La définition de la structure d'un réseau Bayésien causal s'avère être une tâche difficile compte tenu de la complexité du problème. En effet, si nous supposons que nous cherchons à modéliser dans un graphe les liens causaux entre n variables ; le nombre $N(n)$ de combinaisons possibles constituant l'espace de tous les graphes possibles, donné par 2.18, est exponentiel par rapport au nombre n de variables observées (Robinson, 1977) :

$$N(n) = \begin{cases} 1 & \text{si n=0 ou n=1 ;} \\ \sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} 2^{k(n-k)} N(n-k) & \text{si n>1} \end{cases} \tag{2.18}$$

Par exemple, si nous voulons chercher un graphe modélisant les liens causaux entre trois variables A, B, et C, il y aura 25 graphes possibles, comme illustré sur la figure 2.13. Dans un contexte plus complexe, dans lequel la suffisance causale doit être approchée le plus possible, le nombre de variables du réseau devient important. Ceci conforte l'idée selon laquelle la construction manuelle, bien qu'elle soit la plus simple, n'est pas forcément la plus fiable et peut souvent induire en erreur. Par ailleurs, nous venons de mettre en avant le fait que l'apprentissage de la structure correcte d'un réseau bayésien est un problème NP-complet.

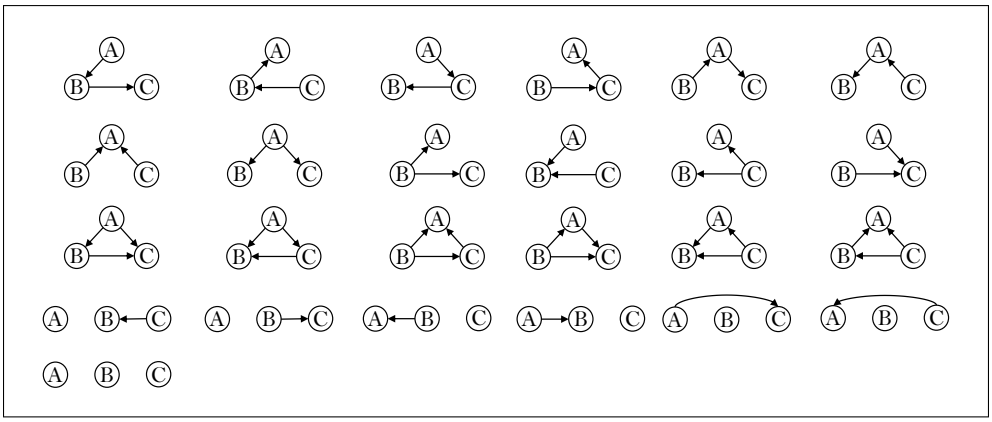


FIGURE 2.13 – Réseaux Bayésiens possibles pour trois variables.

2.2.6.2 Approches d'apprentissage de la structure

Les approches d'apprentissage de la structure d'un réseau Bayésiens peuvent être répertoriées dans deux catégories : les approches basées sur la recherche d'indépendances conditionnelles, et les approches basées sur le calcul d'un score (Vandel, 2012).

2.2.6.2.1 Approches basées sur la recherche d'indépendances conditionnelles

Ces approches ont pour objectif de construire la structure d'un réseau bayésien à partir des contraintes d'indépendances, qui se traduisent par l'ajout ou non d'arcs entre les variables. Comme précédemment discuté, les opérateurs et relations testables statistiquement, comme l'indépendance et la corrélation entre deux variables, sont symétriques (Hourbracq, Wuillemin, Gonzales, & Baumard, 2018). Leur utilisation individuelle sur un couple de variables ne permet alors pas l'orientation du graphe. L'idée principale derrière les approches basées sur les indépendances est alors de d'abord chercher les corrélations ou indépendances entre les variables par paires pour construire un graphe non-orienté, puis détecter les V-structures permettant d'orienter le squelette pour obtenir un graphe dirigé, en utilisant la notion de la *d-séparation*. Le point de départ de ces approches peut être un graphe vide (recherche de corrélations et ajout d'arêtes pour former le squelette du réseau), ou un graphe complet (recherche d'indépendances et suppression d'arêtes pour former le squelette du réseau). L'étape de recherche des V-structures reste inchangée quel que soit le point de départ choisi, et s'effectue en se basant sur le squelette obtenu à l'issu de la première étape, et sur les indépendances conditionnelles. En effet, si deux variables sont indépendantes conditionnellement à un ensemble de variables $\mathcal{X}$, alors elles sont d-séparées par $\mathcal{X}$, et cette d-séparation permet ensuite de détecter les v-structures. À titre d'illustration, si le squelette contient une chaîne $A - B - C$, et qu'il n'y a aucune autre chaîne reliant A et C, et qu'une indépendance conditionnelle $A \perp\!\!\!\perp C|\mathcal{X}$ a été détectée (A et C détectés par les tests comme étant indépendants conditionnellement à un ensemble de variable $\mathcal{X}$), et si $B \notin \mathcal{X}$ alors la seule orientation qui sera possible est $A \rightarrow B \leftarrow C$, puisque dans le cas où $B \notin \mathcal{X}$, le seul blocage possible du chemin par $\mathcal{X}$ est obtenue par la troisième composante de la règle de la *d-séparation*. À l'issue de cette opération, l'orientation de certaines arêtes tant que cela est possible par déduction directe du graphe obtenu à l'étape précédente. Il est important de souligner que les algorithmes fondés sur cette approche ne permettent d'orienter un graphe que de manière partielle. Le graphe obtenu en utilisant une approche basée sur les tests d'indépendance est alors un graphe

acyclique partiellement dirigé (PDAG), qu'il faudra ensuite compléter en ayant recours aux connaissances des experts pour construire le réseau bayésien causal final, qui sera donc un graphe acyclique partiellement dirigé complété (CPDAG) (Andersson, Madigan, & Perlman, 1997).

Les tests utilisés pour détecter les indépendances conditionnelles et/ou les corrélations peuvent différer d'un algorithme à l'autre. Le test χ^2 et le test χ^2 conditionnel, de Pearson, sont souvent utilisés pour détecter respectivement les indépendances et les indépendances conditionnelles. D'autres tests existent également comme la corrélation ρ de Spearman, le coefficient de corrélation de Pearson, le coefficient V de Cramér, le test G^2, ou encore le tau de Kendal. Ces tests ne sont pas tous applicables dans les mêmes situations, puisqu'il faut respecter les types de variables admis pour chaque test. Par ailleurs, certains parmi ces tests, notamment le coefficient de corrélation de Pearson, sont paramétriques : ils sont valables sous l'hypothèse selon laquelle les données suivent une distribution normale, et il est préconisé d'effectuer un test de normalité avant d'utiliser les tests paramétriques concernés par cette hypothèse. Néanmoins, l'absence de normalité ne représente pas toujours un obstacle, dans le sens où elle ne rend pas forcément ces test invalides : il suffirait que les effectifs soient importants pour s'affranchir de cette hypothèse (Tufféry, 2011).

2.2.6.2.2 Approches basées sur le calcul d'un score

La deuxième catégorie d'approches d'apprentissage de structure d'un réseau Bayésien est celle basée sur le calcul et la maximisation d'un score. Ce score doit refléter le degré de correspondance d'un graphe final obtenu par apprentissage, au problème traité (Naïm et al., 1999). En d'autres termes, il doit refléter la vraisemblance que les données puissent être générées à partir du graphe final obtenu. Plus le score est élevé, plus on s'approche du graphe correct, en admettant qu'il existe un graphe correct. L'apprentissage de la structure, dans ces approches, s'apparente donc à un problème d'optimisation. Ces approches se décomposent à leur tour en (1) approches de recherche du réseau Bayésien dans tout l'espace des DAG possibles (espace $\mathbb{B}$), et en (2) approches effectuant la recherche dans un espace restreint composé de DAGs représentant chacun une parmi les classes d'équivalence possibles (espace $\mathbb{E}$) (Ben Amor, 1988) ; chaque classe d'équivalence étant composée d'un ensemble complet de réseaux Bayésiens équivalents. Les approches (2), suggérant la réduction de l'espace de recherche, justifient cela par la complexité du problème qui rend l'exhaustivité de la recherche inatteignable dès que le nombre de variables est important, comme discuté dans la sous-section 2.2.6.1. Ce choix est également motivé par le fait que tous les réseaux Bayésiens équivalents obtiennent le même score, puisqu'ils encodent tous les mêmes indépendances et indépendances conditionnelles. Par conséquent, au lieu de parcourir tous les graphes équivalents afin de calculer leurs scores un par un, il suffirait de calculer le score d'un graphe parmi plusieurs graphes équivalents. Ce graphe sera alors le "représentant" de sa classe d'équivalence. Concernant les algorithmes basés sur les approches (1), certains sont gloutons, tandis que d'autres tentent de contourner le problème de complexité en effectuant la recherche dans un espace réduit de $\mathbb{B}$ en ordonnant les nœuds, ou en se limitant aux DAGs ayant des structures arborescentes. Étant donné la complexité de parcours de tous les réseaux différents et celle du parcours des représentants des classes d'équivalence, le score à calculer doit être décomposable localement afin d'éviter de le recalculer complètement à nouveau pour chaque graphe candidat, et ce quel que soit l'espace de recherche. C'est à dire que le score doit pouvoir être calculé à partir des scores locaux de chaque nœud. Les expressions des formules fréquemment utilisées

pour calculer le score $S_{G,D}$ de l'adéquation d'un graphe candidat G à la base de données D, son déclinées de la forme (2.19) :

$$S_{G,D} = c_G + \sum_{i=1}^{n} s_{(X_i|Pa(X_i)),D} \tag{2.19}$$

où c_G représente une constante attribuée au graphe candidat pour le pénaliser sur sa complexité. Cette pénalité est fonction du nombre de tables de probabilités conditionnelles nécessaires pour paramétrer le réseau ; et $s_{(X_i|Pa(X_i)),D}$ le score local de vraisemblance que $Pa(X_i)$ selon le graphe candidat soient les parents de X_i, calculé pour chaque variable X_i.

À l'instar des méthodes basées sur les indépendances, les méthodes basées sur le calcul d'un score ne permettent pas non plus d'identifier un réseau Bayésien causal avec certitude, puisque les scores sont égaux pour les graphes équivalents. Les connaissances des experts sont alors sollicitées afin d'identifier le graphe équivalent capturant les relations causales. L'ordonnancement des nœuds dont certaines approches sont concernées, peut également être considéré comme une implication de connaissances humaines.

Le problème de l'inférence causale en général, et de la découverte de réseaux Bayésiens causaux à partir des données en particulier, n'a jusqu'à présent pas de solution. Les avis en la matière convergent largement sur le fait qu'il est impossible d'inférer la causalité en se basant uniquement sur les données (Juhel, 2015 ; Boreux et al., 2009 ; John-Mathews, 2017). En effet, la causalité est une notion dont la portée dépasse les données, elle n'admet jusqu'à présent aucune représentation mathématique, et les connaissances humaines demeurent indispensables pour compléter l'apprentissage.

2.2.7 Conclusion

Après avoir identifié les caractéristiques des liens de causalité dans la section 2.1, nous avons abordé, dans cette section, l'analyse causale dans sa globalité. Nous avons d'abord décrit l'utilité de l'exploitation des KPIs pour la décision, puis mis en avant le rôle crucial que joue l'interprétation des valeurs des KPIs dans la prise de décision et l'amélioration des performances. Cette interprétation des KPIs, qui oriente les décisions, est basée sur l'analyse des liens existants entre le KPI d'intérêt et les éléments y exerçant des influences. Les analyses descriptives, pouvant se décliner en plusieurs méthodes ou standards, présentent l'inconvénient d'être très subjectives, et coûteuses en termes de temps et de ressources humaines. La subjectivité de ce type d'analyse peut induire en erreur, et par conséquent donner lieu à de mauvaises décisions. Le caractère chronophage peut également représenter un obstacle à la prise de décision, puisque dans certaines situations, le problème traité peut s'aggraver, voire engendrer d'autres problèmes supplémentaires. En outre, le caractère chronophage des analyses descriptives est également du à leur caractère singulier, puisque le processus doit être redéroulé pour chaque situation problématique. Par ailleurs, l'utilisation des ressources humaines n'est pas optimisée, dans le sens où l'humain ne se voit pas assigner autant de tâches à valeur ajoutée comme il l'aurait pu. Le parcours de certaines méthodes d'analyse causale descriptive ne nous a pas seulement permis de soulever les défis à relever, mais également de repérer les forces et les idées intéressantes sous-jacentes à chacune des méthodes parcourues. Ceci nous a permis d'identifier les caractéristiques que, selon nous, devrait satisfaire une analyse causale, et

dont nous nous servirons lors de la validation de la proposition. À partir de là, et en se référant aux caractéristiques des liens causaux, et en particulier au caractère incertain de ces liens, nous nous sommes intéressés au réseaux Bayésiens, pour ce qu'ils apportent en termes de modélisation de l'incertitude, et également pour la représentation graphique intuitive qu'ils offrent. Aussi, compte tenu des caractéristiques que nous avons identifiées pour l'analyse causale en référence aux défis et forces des méthodes descriptives, nous écartons la construction manuelle des réseaux Bayésiens, qui est finalement descriptive aussi. La construction des réseaux Bayésiens causaux par apprentissage, bien qu'elle soit basée sur les données, ne peut pas être complètement affranchie de l'immobilisation des connaissances humaines. Malgré cela, la construction par apprentissage comprend tout de même moins de subjectivité que la construction manuelle, puisqu'elle permet d'inférer un DAG ou un PDAG à partir des données, qui doit ensuite être complété. Ceci nécessite donc moins de connaissances et de temps qu'une construction complètement manuelle, et comporte moins de risques quant à la déduction à tort de relations causales fallacieuses. La construction par apprentissage des réseaux Bayésiens causaux, qui reste donc en tout état de cause une meilleure alternative que la construction manuelle, est tout de même soumise à une contrainte forte de suffisance causale. Dans la pratique, il est simplement impossible de garantir la réalisation, ou du moins la preuve de réalisation, de cette hypothèse. Cependant, grâce à l'abondance des données observée de nos jours, et à l'utilisation de plus en plus répandue des systèmes d'acquisition des données, on peut tendre vers la suffisance causale, sans pour autant la garantir. L'exploitation des systèmes d'acquisition des données ne permet pas seulement d'approcher la suffisance causale qui permet d'invalider les relations causales fallacieuses en découvrant des causes communes sous-jacentes, mais elle permet aussi de découvrir de nouvelles relations insoupçonnées auparavant. Elle permet également la prise en compte du facteur humain qui peut être équipé de systèmes d'acquisition, sans pour autant enfreindre ce qui relève de l'éthique. Pour toutes ces raisons, nous optons dans notre proposition pour l'apprentissage des réseaux Bayésiens causaux. L'objectif serait alors de réduire les connaissances nécessaires pour obtenir des DAG en se basant sur l'une des approches présentées dans la sous-section précédente (basée sur le calcul d'un score ou sur la recherche d'indépendances conditionnelles). Le choix de l'approche sera présenté et motivé dans le chapitre 4 lors de la présentation de l'algorithme que nous proposons. **Dans notre proposition, nous allons donc procéder à la sélection d'une approche puis d'un algorithme pour l'apprentissage des réseaux Bayésiens, en motivant notre choix, afin d'y apporter des améliorations de façon à minimiser l'apport nécessaire de connaissances.**

2.3 Prédiction des KPIs

La prédiction représente un atout majeur pour la prise de décision. En effet, comme nous l'avons précédemment expliqué, et comme illustré sur la figure 1.4, la notion du temps est cruciale pour tout processus de décision. Lorsqu'un KPI donné est surveillé et qu'un écart par rapport à l'objectif est découvert, une action doit être entreprise pour rétablir la situation. Afin d'être pro-actif, et de réduire le temps de résolution des problèmes en prenant des décisions en temps opportun, il est important de prévoir l'évolution future de l'indicateur surveillé (Zeng, Lingenfelder, Lei, & Chang, 2008).

La prédiction est un vaste domaine abordé depuis plusieurs années par différentes disciplines. Nous parlons ici de la prédiction basée sur des données mesurées, collectées,

simulées, ou issues d'expérimentations. Dit autrement, il s'agit ici de la prédiction par apprentissage, qui s'avère profitable lorsqu'on travaille sur des systèmes ou processus complexes, pour lesquels les connaissances sont souvent trop approximatives pour prétendre à des prédictions précises (Dreyfus et al., 2008). Dans le domaine de l'apprentissage, la prédiction (ou régression) désigne le fait de prédire des valeurs numériques quantitatives. Par abus de langage, nous employons dans la suite le terme prédiction pour désigner 1) le classement et la classification automatique, qui permettent de faire une analyse discriminante pour prédire la catégorie d'un objet, et 2) la régression.

2.3.1 Apprentissage supervisé

Dans notre contexte, nous avons supposé, par le biais du postulat P2, que l'ensemble des données contextuelles historiques est disponible, ainsi que les mesures historiques des KPIs d'intérêt. Nous faisons également un postulat sur la structuration de ces données, dans le sens où nous admettons que la correspondance entre les KPIs mesurés et les observations des données contextuelles est déjà effectuée. L'apprentissage se fera alors sur une table de données s'apparentant à la table 2.1, où les colonnes A, B, C, D, et E correspondent aux variables contextuelles, et la colonne KPI correspond au KPI d'intérêt.

TABLE 2.1 – Structuration des données pour un apprentissage supervisé.

#	A	B	C	D	E	KPI
1	a_1	b_1	c_1	d_1	e_1	kpi_1
2	a_2	b_2	c_2	d_2	$e_1 2$	kpi_2
...	...	...	...	...	...	...
n	a_n	b_n	c_n	d_n	e_n	kpi_n

La disponibilité des données historiques rend possible l'apprentissage supervisé, auquel nous nous intéresserons donc. L'objectif est donc de construire un modèle de prédiction qui soit capable de prédire le KPI d'intérêt à partir d'une observation des variables contextuelles. Par ailleurs, si plusieurs indicateurs clés de performance doivent être prédits, et si plusieurs systèmes d'acquisition sont installés pour surveiller les entités concernées, le modèle de prédiction doit être capable de traiter des types de données qualitatives et quantitatives. Ceci s'avère être le cas dans notre contexte.

Une revue de la littérature nous a permis d'identifier les différentes méthodes d'apprentissage automatique existantes. Notre besoin de traiter les KPIs qualitatifs et quantitatifs nous a contraint à écarter les méthodes de classification automatique non hiérarchiques, telles que la méthode des centres mobiles et les nuées dynamiques, et les méthodes hiérarchiques tels que les arbres de décision ; puisqu'elles elles ne permettent pas la prédiction de variables continues. De plus, la disponibilité des données historiques rend insignifiant l'utilisation d'un apprentissage non-supervisé, et il serait regrettable de ne pas tirer profit des données dont on dispose. Les méthodes de classification supervisée telles que les k plus proches voisins sont également écartées, pour la même raison évoquée pour la classification

automatique. La régression et la régression multiple sont largement utilisées pour prédire les variables quantitatives, cependant leur pouvoir prédictif est limité uniquement aux cas où la variable à expliquer et les variables explicatives sont quantitatives (Tu, 1996). Les réseaux Bayésiens peuvent également servir à faire de la prédiction, cependant, les résultats restent fortement dépendants de la validité de la structure du réseau, qui peut ne pas être exacte. Elle dépend également des tables de probabilités conditionnelles qui doivent donc être spécifiées, et pour cela, une discrétisation des variables continues est nécessaire, ce qui a pour conséquence une perte d'information et donc un risque élevé de prédiction inexacte quand ceci est cumulé avec une structure approximative. Par ailleurs, il existe également la régression logistique, permettant de prédire une variable dichotomique, donc qualitative, mais s'avère très sensible à la multicolinéarité entre les prédicateurs. Enfin, les réseaux de neurones représentent une technique très répandue pour répondre au besoin de prédiction. Ils bénéficient de nombreux avantages, dont l'admissibilité de variables quantitatives et qualitatives en entrée et en sortie, même si quelques transformations peuvent être nécessaires pour les admettre ensemble en entrée d'un réseau (Tu, 1996). En outre, les réseaux de neurones présentent une capacité élevée de détection des relations non-linéaires complexes entre des variables dépendantes et indépendantes, grâce aux couches intermédiaires qu'ils peuvent mettre en œuvre. Ils existent en plusieurs types et peuvent être appliqués différemment en fonction du problème traité. Aussi, l'utilisation massive des réseaux de neurones sur différentes applications a eu pour conséquence la disponibilité d'un nombre important de librairies facilitant leur développement, et ce pour leurs différentes variantes.

Pour toutes ces raisons, nous nous intéressons aux réseaux de neurones pour répondre à notre objectif de prédiction F3.

2.3.2 Réseaux de Neurones

Les réseaux neuronaux constituent un outil d'apprentissage automatique très efficace qui a été largement utilisé dans de nombreuses applications industrielles telles que le contrôle des processus de fabrication, le diagnostic des processus et des machines, l'analyse de la maintenance des machines, et la planification (Martin, Howard, Mark, & Orlando, 2014). Comme leur nom le laisse penser, les réseaux de neurones s'inspirent de la biologie, et leur développement a émané de la tentation d'imiter le traitement des information dans le cerveau humain, qui a ensuite suscité un intérêt pour le neurone biologique. Cette unité de base, qui représente le composant principal du traitement de l'information, est constituée 1) d'un corps cellulaire contenant le noyau qui traite l'information et dont l'influx nerveux reflète l'état d'activité du neurone (Boularas & Djalab, 2020), 2) de dendrites, qui captent les informations provenant des autres neurones, représentent les entrées principales du neurone, 3) de synapses, qui véhiculent l'information vers le neurone, en moyennant des poids synaptiques, et 4) l'axone, qui représente la sortie du neurone. L'analogie avec ce mode de fonctionnement a donc donné naissance au réseaux de neurones, dont le composant unitaire est le neurone artificiel, illustré sur la figure 2.14.

Dans notre contexte, la sortie du réseau représentera soit la valeur quantitative du KPI, ou bien le dépassement ou non d'un seuil de tolérance préalablement fixé pour le KPI étudié (*i.e.* une classification sur la déviation ou la non déviation du KPI). Nous travaillerons alors sur les réseaux de neurones à sortie unique.

Le traitement d'un vecteur de données d'entrée $\langle X_1, X_2, ..., X_n \rangle$ s'effectue en deux

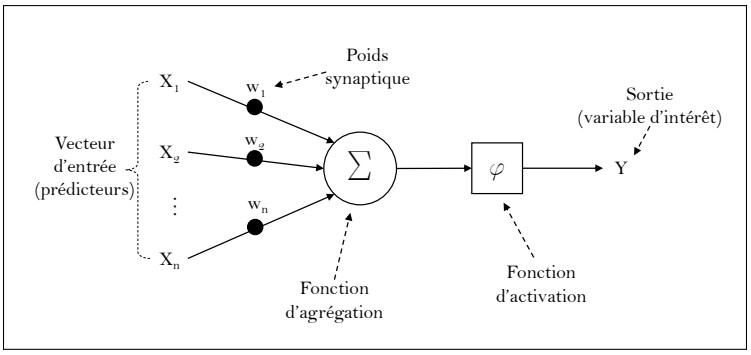


FIGURE 2.14 – Neurone artificiel formel.

phases : la fonction d'agrégation calcule la somme des entrées liées au neurones, pondérées par les poids synaptiques des liaisons, à laquelle s'ajoute un biais b. Un biais se présente comme un neurone supplémentaire dont la valeur est toujours égale à 1, et qui est lié aux autres neurones par un poids. Autrement dit, il agit comme une somme de poids supplémentaires exerçant une transformation affine sur la somme pondérée des entrées. La fonction d'agrégation s'écrit donc sous la forme $\sum_{i=1}^{n} x_i.w_i + b$, où w_i représente le poids synaptique associé à l'entrée i ; le résultat de cette somme pondérée est ensuite passé en paramètre d'une fonction d'activation φ associée au nœud. La sortie du neurone correspond alors à la valeur de $\varphi(\sum_{i=1}^{n} x_i.w_i + b)$. Lorsqu'une entrée n'est pas connectée au neurone, le poids synaptique les liant prend la valeur de 0.

La fonction d'activation φ, également appelée fonction de transfert et souvent non linéaire, reflète le potentiel d'activation. À l'image du fonctionnement biologique, elle tient pour rôle de donner la permission de passage d'information ou non, selon si le seuil de stimulation est atteint ou non. L'intérêt de son utilisation est de pouvoir capturer et représenter les dépendances non linéaires. En effet, se cantonner à la fonction d'agrégation, qui n'est autre qu'une combinaison linéaire, aura pour conséquence que l'efficacité du réseau ne se remarquerait que dans les cas de dépendances linéaires. De nombreuses fonctions d'activations usuelles existent et les choix de leurs utilisations dépendent du problème traité. Elles sont souvent bornées (souvent entre 0 et 1, ou entre -1 et 1) afin d'éviter les calculs mobilisant de grands nombres et une consommation de mémoire importante. Toutefois, toutes les fonctions d'activation ne sont pas bornées, certaines ayant un caractère illimité peuvent s'avérer utiles, entre autres, lors des entrainements lents pour empêcher une potentielle saturation provoquée par l'approximation progressive de 0. Aussi, toutes les fonctions d'activation ne respectent pas la non-linéarité. Le choix d'une fonction d'activation dépend des cas d'utilisation et des particularités qu'ils rendent souhaitables pour ces fonctions.

La figure 2.15 illustre un exemple de fonction d'activation : la fonction *sigmoïde*, qui est souvent utilisée, pour les avantages qu'elle présente, tels que la facilité de se dérivation permettant d'accélérer le calcul de la dérivée et par conséquent de réduire le temps de l'entraînement, ainsi que ses bornes 0 et 1 induisant une interprétation probabiliste de l'activation de la du neurone. Cette fonction s'écrit sous la forme suivante : $\varphi(z) = \dfrac{1}{1+e^{-z}}$,

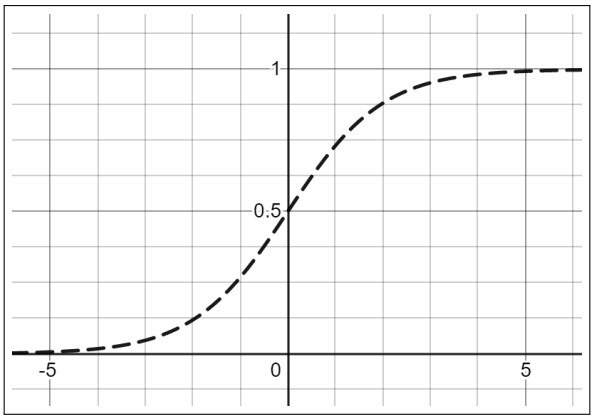


FIGURE 2.15 – Fonction *sigmoïde*.

où z est le résultat de la fonction d'agrégation. Pour un neurone avec n entrées connectées chacune au neurone avec un poids w_i, l'activation de la sortie du neurone prendra alors la valeur de $\varphi(\sum_{i=1}^{n} x_i.w_i + b) = \frac{1}{1+e^{-\sum_{i=1}^{n} x_i.w_i+b}}$. Nous pouvons percevoir, à travers cette formule, que la normalisation des données d'entrée est nécessaire, afin que les variables d'entrée ayant de très grandes valeurs n'écrasent pas celles prenant de petites valeurs. Avec des variables variant sur la même échelle, et dans le cas d'une classification utilisant la fonction Sigmoïd, l'apprentissage ajuste les poids w_i et le biais b de manière à ce que $\varphi(\sum_{i=1}^{n} x_i.w_i+b)$ prenne une grande valeur (positive) pour les observations $\langle X_1, X_2, ..., X_n \rangle$ dont la sortie attendue est 1, et une petite valeur (négative) pour les observations dont la sortie attendue est 0.

Comme précédemment mentionné, l'atout majeur des réseaux de neurones réside dans leur capacité de traitement de relations complexes, ce qui est le cas dans notre contexte. Cette capacité est atteinte grâce à la possibilité de circulation de l'information entre plusieurs neurones interconnectés et disposés en une ou plusieurs couches intermédiaires précédant la sortie finale. En effet, bien que les fonctions d'activation fréquemment utilisées sont assez simples et monotones, leur combinaison fait qu'elles peuvent approcher n'importe quelle fonction (Tufféry, 2011). Ceci est notamment le cas dans les réseaux de neurones de type "perceptron multi-couches" (MLP), où l'information se propage dans un sens direct : des entrées vers la sortie en traversant la ou les couches cachées. La figure 2.16 illustre la topologie d'un réseau de neurones du type perceptron multi-couches complètement connecté, à deux couches intermédiaires, et la figure 2.17 illustre la propagation de l'information dans ce même réseau, où la première couche intermédiaire comprend n neurones, et la deuxième en contient m.

L'exactitude de la prédiction dépend alors des fonctions utilisées pour calculer la sortie, qui elles mêmes dépendent des poids qu'il faut fixer correctement. Par conséquent, la force de la prédiction dépend, entre autres, du choix de la fonction d'activation et de l'estimation des poids. L'estimation de ces poids est conditionnée par la structure du réseau et par la fonction d'activation utilisée, puisque pour un seul et même problème et avec les mêmes données d'entrée, les poids seraient différents d'un réseau à un autre si les deux réseaux ne possèdent pas le même nombre de couches et de neurones traversés, et si les couches et les neurones cachés ne sont pas traversés de la même manière (*i.e.* avec

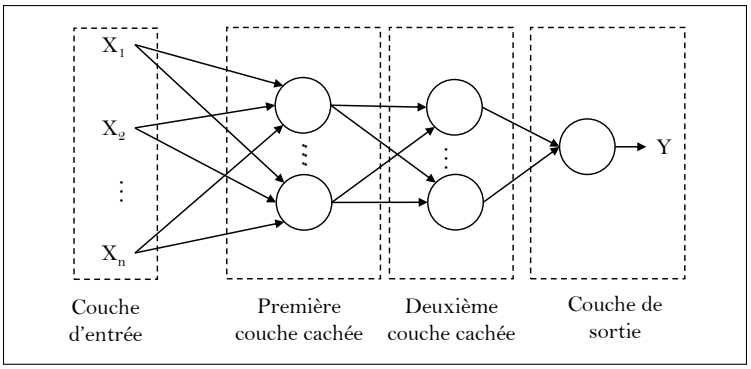


FIGURE 2.16 – Représentation d'un réseau de neurones du type perceptron multi-couches à deux couches intermédiaires.

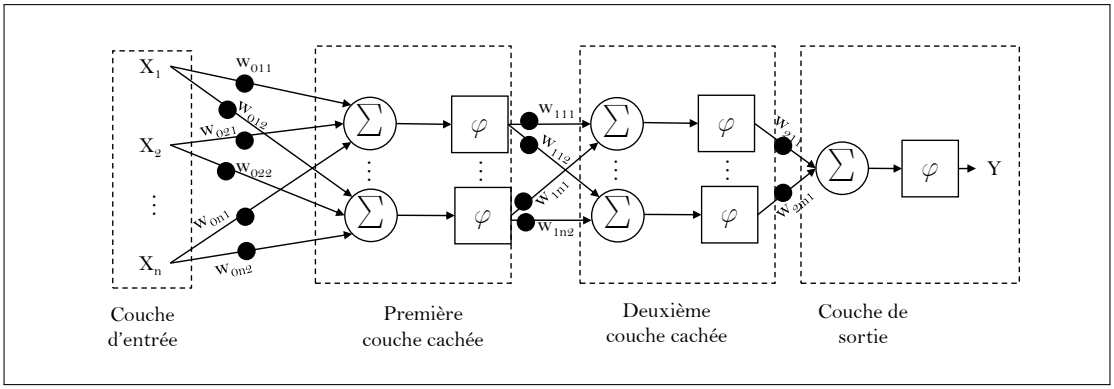


FIGURE 2.17 – Représentation de la propagation de l'information dans un réseau de neurones du type perceptron multi-couches à deux couches intermédiaires.

des fonctions d'activation différentes). L'apprentissage qu'effectue un réseau de neurone, dont la structure est fixée, se fait alors à travers l'optimisation des poids. En général, dans les réseaux type MLP, l'apprentissage peut commencer avec des poids aléatoires, qui évoluent à chaque itération de l'apprentissage, en fonction de l'erreur en sortie, jusqu'à ce que l'erreur en sortie soit minimisée (*i.e.* l'écart observé entre la sortie prédite Y^* et la sortie attendue Y). Il existe plusieurs algorithmes pour optimiser les poids dans le cadre d'un réseau MLP, comme par exemple la rétro-propagation du gradient.

L'utilisation d'un réseau de neurones pour la prédiction nécessite alors la mise en œuvre de plusieurs étapes (Tufféry, 2011) :

— La préparation des données d'entrée : qui correspond à la normalisation pour éviter l'écrasement et les problèmes d'approximation rapide des limites lorsque la fonction d'activation est bornée ;

— La construction du réseau de neurones : qui comprend la définition de la structure du réseau de neurones, et ses paramètres, notamment les poids synaptiques, le taux d'apprentissage, le choix de la ou les fonctions d'activation à utiliser, le choix de l'algorithme d'optimisation des poids, et d'autres paramètres auxquels nous reviendrons plus en détail dans le prochain chapitre ;

— L'apprentissage du réseau : qui se traduit par l'optimisation des poids de liaisons entre les neurones jusqu'à obtenir des poids finaux minimisant l'erreur ;

— Le test du réseau ;
— L'application du réseau ;
— La dénormalisation des données en sortie si nécessaire.

2.3.3 Conclusion

Les réseaux de neurones peuvent servir dans notre contexte à la prédiction des valeurs des KPIs, ou de leurs états afin d'alerter sur de potentielles futurs déviations, sur la base d'un ensemble de variables contextuelles. Par ailleurs, dans le cadre de nos objectifs, nous allons voir, dans la prochaine section, que les réseaux de neurones peuvent éventuellement avoir une autre utilité que celle de faire de la prédiction, ce qui rend le choix de l'utilisation de cette technique encore plus privilégié. Cependant, la construction des réseaux de neurones, étape clé conditionnant leur efficacité, n'est pas très évidente. En effet, elle est le résultat d'un processus empirique itératif de définition de la structure et des paramètres, puis apprentissage et test des performances du réseau. Ceci entrave la rapidité de déploiement et l'aspect générique que nous voulons donner à notre proposition, puisqu'il faut reconduire ce processus itératif pouvant être long et fastidieux pour chaque KPI étudié. **Par conséquent, la construction des réseaux de neurones représente désormais pour nous un sous-objectif, dans le sens où nous allons tenter d'automatiser ce processus.**

2.4 Hiérarchisation des causes

La hiérarchisation des causes représente un atout majeur pouvant nettement améliorer l'efficacité du processus de prise de décision. En effet, si nous pouvons faire de la prédiction sur un KPI qui nous intéresse, et que nous connaissons les entités qui lui sont liées causalement, la connaissance du degré avec lequel chacune de ces entités influe sur le KPI pourrait nous aider à faire la sélection d'une alternative d'action parmi plusieurs. Le rôle que peut jouer la hiérarchisation dans un processus de prise de décision se ressent donc particulièrement dans la phase "choix de la solution", illustrée précédemment sur la figure 2.1.

Les tables de probabilités conditionnelles associées à un réseau Bayésien peuvent servir à l'interprétation des influences des différentes causes sur le KPI d'intérêt. Or, dans les contextes industriels, les systèmes sont souvent complexes, et un KPI auquel on s'intéresse fait partie d'un grand nombre de variables pouvant expliquer la déviation. Il en résulte un réseau bayésien complexe décrivant les relations entre le KPI étudié et les variables contextuelles, dans lequel chaque nœud peut prendre différents états et peut avoir des parents, qui peuvent à leurs tours prendre plusieurs états également. Cela conduit à devoir inférer et interpréter, pour chaque variable, les combinaisons des différents états, issues de nombreuses grandes tables de probabilités conditionnelles qui peuvent être très difficiles à interpréter (Castellani, 2013).

À cela s'ajoute le fait que la plupart des personnes ne peuvent pas interpréter les informations au-delà de quatre dimensions, ce qui peut être très vite atteint lorsque les

nombres de variables et de leurs états augmentent (Pollino & Henderson, 2010). Les niveaux d'influence peuvent toujours être saisis en effectuant une analyse de sensibilité sur les différentes combinaisons d'états des nœuds impliqués, mais cela prend souvent du temps (Kjærulff & van der Gaag, 2013), et n'est donc pas pratique dans un contexte où des décisions doivent être prises rapidement.

Comme précédemment expliqué, l'accumulation des fonctions d'activation sur plusieurs couches, permet d'approcher n'importe quelle fonction. En s'intéressant de plus près à cette façon de prédire la sortie, on se rend compte qu'en plus de pouvoir prédire les événements futurs, une fois que le modèle fournissant de bonnes performances a été construit, leurs poids finaux peuvent être récupérés et analysés dans le but d'identifier les facteurs ayant les plus importants impacts sur la sortie. Les poids synaptiques qui évoluent pendant l'apprentissage représentent les seuls inconnus à déterminer, pour que la fonction globale de prédiction puisse approcher la fonction réelle inconnue liant les entrées à la sortie. En effet, les poids des connexions peuvent être considérés, par analogie, comme étant équivalents aux coefficients β dans les modèles de régression, et contiennent la "connaissance" acquise par un réseau de neurones après son entraînement. Ces poids représentent alors en quelque sorte la force des connexions entre les unités d'un réseau de neurones, et mettent en évidence les degrés d'importance des valeurs des entrées (Han, Pool, Tran, & Dally, 2015). Ceci n'est sans aucun doute valable que lorsque le réseau est performant.

Par conséquent, afin de remplir notre objectif de hiérarchisation des causes, nous nous intéresserons à l'exploitation des poids finaux des réseaux de neurones pour l'interprétation des influences des variables d'entrée sur la sortie. Ceci n'étant possible que si nous disposons d'un réseau de neurones ayant un bon pouvoir de prédiction, notre sous-objectif de construction automatique des réseaux de neurones prend encore plus de sens, puisqu'il sera utile à notre démarche de hiérarchisation des causes.

2.5 Conclusion générale

Dans ce chapitre, nous avons abordé les points indispensables à l'introduction de notre proposition. Notre objectif étant de fournir une méthode d'analyse causale basée sur les données, nous avons commencé par explorer la notion complexe de la causalité. Cette notion, n'ayant à ce jour pas de définition arrêtée, et n'étant pas mathématiquement représentable, nous avons listé l'ensemble des caractéristiques communes des liens causaux génériques. Nous en avons sélectionné un ensemble qui nous a paru adapté à notre contexte. Les caractéristique de causalité identifiées vont servir à la fois de fondement pour développer notre méthode d'analyse causale, et de support pour vérifier la validité des liens causaux une fois ces derniers détectés par la méthode proposée. Par la suite, nous avons fait un panorama sur les pratiques usuelles de résolution de problèmes basées sur une analyse des causes. Ceci nous a permis de souligner les manques dont souffrent chacune des analyses descriptives, et les forces et idées intéressantes à partir desquelles nous pouvons nous inspirer pour notre analyse causale. Par conséquent, les forces identifiées, et la négation des inconvénients identifiés, représentent les caractéristiques relatives à notre analyse causale dans sa globalité. Ces caractéristiques nous ont aidé à faire le choix sur l'approche que nous adopterons pour construire notre analyse causale, et seront

utilisées lors de la validation de la proposition, afin de vérifier si l'analyse causale que nous développons répond bien à nos attentes. Nous nous sommes par la suite arrêtés sur les réseaux Bayésiens, pour les atouts qu'ils offrent quant à l'analyse causale à partir des données, et à la représentation graphique aisément abordable. Nous avons déduit, pour plusieurs raisons, que la construction des réseaux Bayésiens par apprentissage représente une meilleure alternative que la construction manuelle. Nous avons tout de même souligné que l'apprentissage de la structure causale des réseaux Bayésiens n'est pas tout à fait garantie dans son ensemble, et que les connaissances humaines restent indispensables, quelle que soit la méthode d'apprentissage adoptée. L'enjeu est alors de tenter de réduire au maximum l'apport en connaissances.

Par ailleurs, le besoin de proactivité nous a conduit à aborder les techniques pour prédire les évènements futurs. Nous nous sommes ensuite attardés sur les réseaux de neurones, en raison de leur forte capacité de modéliser les non-linéarités, de la nature des données que nous voulons manipuler dans notre contexte, de la grande disponibilité de la littérature à ce sujet, des nombreuses applications déployées, et de l'accessibilité de librairies facilitant leur développement selon le besoin. Par la suite, nous avons mis en évidence la complexité de construction d'un réseau de neurones, qui représente alors un obstacle à notre ambition de généricité et de rapidité. Nous nous sommes donc fixés le sous-objectif d'automatiser le processus empirique et itératif de la construction des réseau de neurones. En outre, nous avons expliqué que l'utilisation des réseaux de neurones peut servir non seulement à la prédiction du KPI, mais également à la hiérarchisation des causes du KPI prédit, puisqu'ils apprennent des poids qui lient mathématiquement les entrées à la sortie. Nous nous sommes donc fixé pour objectif l'exploitation des poids finaux d'un réseau de neurones servant à la prédiction, pour répondre à l'objectif de la hiérarchisation des causes d'un KPI donné. Nous avons tout de même soulevé le fait que la hiérarchisation des causes ne peut être conduite de cette manière que si l'on dispose de réseaux de neurones performants, puisqu'un réseau de neurones prédisant mal la sortie encode forcément des poids qui ne reflètent pas correctement les forces de liaisons entre les entrées et la sortie. Ceci rend d'autant plus utile pour notre sous-objectif de construction automatique des réseaux de neurones.

Étant donné l'ensemble de ces éléments, nous allons construire, pas à pas, dans le prochain chapitre, une architecture à notre proposition, conformément aux besoins identifiés dans ce chapitre, et aux choix que nous y avons effectués. Cette architecture sera donc composée de trois briques, correspondant chacune à un besoin relevé dans le présent chapitre : une brique pour faire de l'analyse causale par apprentissage de réseaux Bayésiens, une brique pour automatiser la construction des réseaux de neurones, et une brique pour la hiérarchisation des causes en utilisant les poids issus du réseau de neurones issu de la brique précédente. À ces trois briques s'ajoute une quatrième brique servant à combiner les résultats et fournir une aide à la décision.

Chapitre 3

Architecture de la proposition

3.1 Introduction

Dans le premier chapitre, nous avons d'abord introduit les motivations, le contexte, et les objectifs de nos travaux, pour ensuite postuler un ensemble d'hypothèses dont nous avons tenu compte lors de notre analyse de l'état de l'art dans le deuxième chapitre. Cette analyse nous a avant tout permis d'élucider la notion complexe de la causalité. L'absence de définition satisfaisante et complète pour cette notion nous a conduit à établir une liste de caractéristiques qui permettent d'approcher la présomption d'existence d'un lien causal entre deux entités. Les méthodes d'analyse causale ont ensuite été abordées et discutées, et il en a résulté que les réseaux Bayésiens semblent représenter une méthode intéressante pour répondre à notre objectif d'analyse causale. La prédiction, faisant également partie de nos objectifs, a ensuite été abordée, et le choix d'utiliser les réseaux de neurones à cette fin a été justifié. Nous avons également soulevé que notre objectif de classer par ordre d'importance les entités liées causalement à un KPI d'intérêt pourrait être rempli en utilisant les réseaux de neurones.

Dans ce chapitre, nous allons introduire l'architecture de notre proposition, pour ensuite mettre en avant, dans le prochain chapitre, l'ensemble des contributions que nous suggérons pour pallier aux limites de chacune des techniques choisies, et pour répondre à l'objectif de la thèse dans son ensemble, tout en respectant l'architecture que nous proposons dans le présent chapitre.

3.2 Construction progressive de l'architecture de la proposition

L'objectif de nos travaux s'inscrit dans le cadre de la supervision des systèmes pour l'aide à la décision. La supervision peut aussi bien concerner des projets d'ingénierie, des systèmes de production, des systèmes d'information, ou encore l'état de santé d'un individu, pour ne pas se restreindre au domaine de l'ingénierie. Selon (Mirdamadi, 2009), la supervision est fondée sur l'acquisition des données en temps réel, et tient un rôle décisionnel d'optimisation dans le sens où elle doit inclure une analyse des données collectées afin de les rendre utiles pour le processus de prise de décision. Elle doit donc permettre une accélération de la prise de décision, et une amélioration de la pertinence des décisions afin de mieux garantir l'atteinte des objectifs fixés.

Il convient également de souligner que la supervision ne doit toutefois pas être confondue avec le pilotage. En effet, ces deux notions sont complémentaires et agissent dans deux sens opposés (Mirdamadi, 2009) : le pilotage se manifeste par la mise en œuvre des actions, en fonction des résultats issus de la supervision. Il s'agit, en d'autres termes, de l'exploitation des données rendues utiles par la supervision. Le pilotage dépend donc des objectifs à atteindre et de la supervision, et vient en aval de cette dernière. La figure 3.1, adaptée de (Pujo & Kieffer, 2002), illustre le déroulement des opérations de supervision et de pilotage dans un contexte de production.

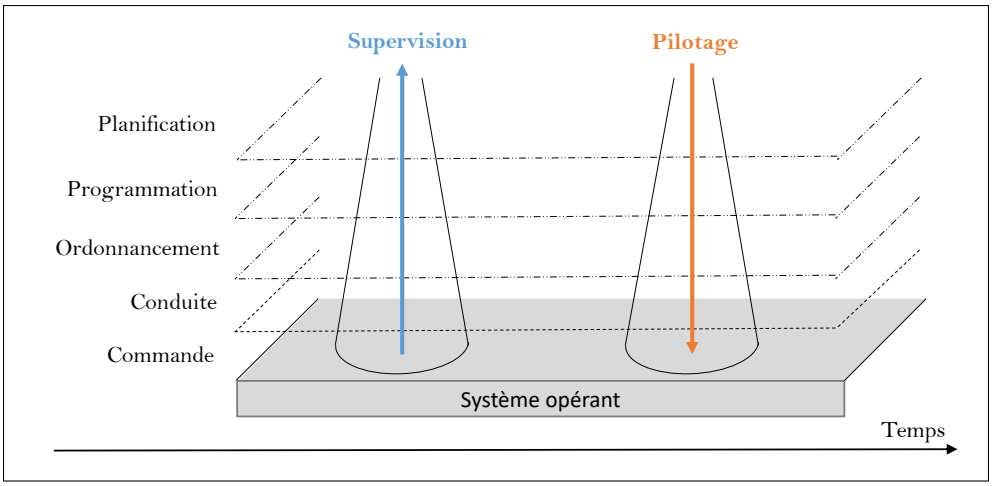


FIGURE 3.1 – Déroulement des opérations de supervision et de pilotage dans un contexte de production.

Notre proposition porte principalement sur la phase d'analyse des données requise par la supervision, et ses résultats peuvent être exploités pour réaliser le pilotage. La figure 3.2 met en évidence le positionnement de notre proposition dans les opérations de supervision et de pilotage. Elle permet alors de rendre compte que notre proposition ne traite pas de la supervision dans sa globalité, mais plus spécialement de l'analyse des données qui en fait partie. L'exploitation des résultats issus de cette analyse de données, ainsi que la prédiction, représenteront ensuite un fondement pour la prise de décision, et par conséquent pour le pilotage. Comme illustré sur la figure 3.2, l'analyse des données suppose la collecte et la mise en disponibilité des données, ainsi que leur agrégation et leur pré-traitement. Dans notre proposition, l'analyse des données est "orientée causalité", par conséquent, le pilotage qui s'en suit l'est également. L'objectif est donc d'améliorer les performances par le biais de la découverte des liens causaux qu'elles ont avec les entités manipulables, sur lesquelles des actions pourraient être engagées pour améliorer les performances de manière indirecte.

La prédiction des KPIs, qui n'est pas à proprement parler incluse dans l'analyse des données, et qui n'est pas représentée sur la figure 3.2, a pour objectif de permettre la proactivité face aux potentielles futures situations critiques. Elle se base sur les valeurs historiques des données contextuelles et du KPI traité pour permettre l'apprentissage, et suppose la disponibilité en temps réel des données contextuelles afin d'appliquer le modèle appris et donc de prédire le KPI. Le pilotage est donc déclenché dès lors que la prédiction dévoile une déviation, et est orienté par les résultats des analyses préalablement effectuées. Ceci suppose alors de connaître un seuil de tolérance au delà ou en deça duquel la performance ne correspondrait pas aux attentes. La prédiction peut concerner

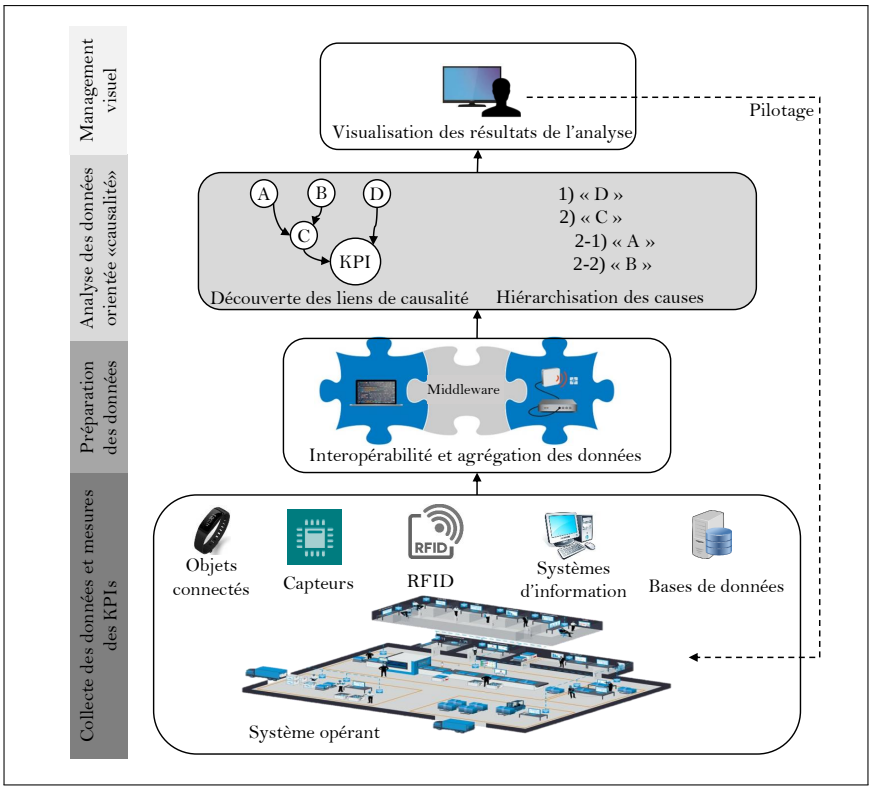


FIGURE 3.2 – Positionnement de notre proposition dans le cadre des opérations de supervision et de pilotage.

directement 1) la valeur du KPI : dans ce cas, le seuil de tolérance ou la valeur attendue doit être spécifié(e) afin d'effectuer une comparaison selon laquelle des actions devraient être engagées ou non ; ou 2) inclure implicitement cette comparaison : dans ce cas, la prédiction se fait sur l'acceptabilité du KPI, il s'agirait alors plutôt d'une classification permettant de statuer si la performance serait adéquate ou non, auquel cas des actions devraient alors être engagées sur les entités manipulables liées causalement au KPI d'intérêt. La figure 3.3 illustre les diagrammes SADT correspondant au déclenchement du pilotage dans ces deux situations. Nous soulignons qu'ici, le terme "déclencher" n'a pas de connotation automatique. En effet, dans le périmètre de notre proposition, les actions ne sont pas déclenchées de manière automatique comme c'est le cas notamment des systèmes cyberphysiques grâce à des commandes envoyées à des actionneurs, mais par les décideurs ou les personnes concernées, qui, en fonction de la prédiction et des informations sur les liens causaux qu'ils ont à disposition, prennent une décision et la mettent en œuvre.

Afin de construire progressivement l'architecture globale de notre proposition, nous suggérons ici de procéder en utilisant un exemple en guise de fil conducteur. Nous reprenons alors l'indicateur de performance calculant le taux de rendement synthétique pour le montage d'un produit donné $P1$, pour représenter le KPI d'intérêt dans notre exemple. Nous rappelons que le TRS est calculé moyennant la définition donnée par la formule (3.1) :

$$TRS = \frac{Production\ réelle}{Production\ maximum\ théorique} \tag{3.1}$$

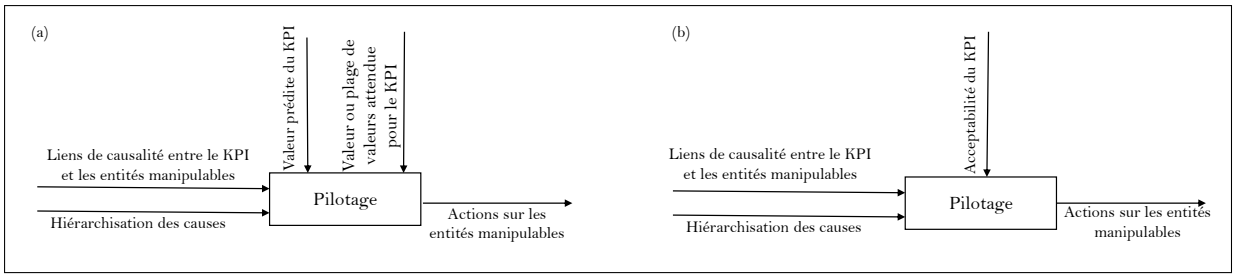


FIGURE 3.3 – Déclenchement du pilotage, prenant en entrée les résultats de l'analyse causale, et conditionné par (a) la comparaison des valeurs prédite et attendue pour le KPI étudié, ou par (b) la prédiction sur l'acceptabilité du KPI.

3.2.1 Analyse causale

Le premier objectif, correspondant à notre fonction F1, est d'identifier les variables contextuelles, et éventuellement d'autres KPIs, liés causalement à ce KPI. Pour ce faire, nous proposons une analyse causale basée sur les données afin de construire un graphe décrivant les liens de causalité. Cette analyse sera basée sur l'apprentissage de la structure d'un réseau Bayésien causal, et fera également intervenir des connaissances humaines, puisque la détection de la causalité uniquement à partir des données est impossible. Selon notre démarche, nous évaluerons les liens de causalité au delà des variables que l'on soupçonne habituellement de manière empirique, afin de découvrir de nouveaux liens jusqu'alors ignorés. Autrement dit, l'analyse causale sera déroulée sur toutes les variables disponibles. Pour notre exemple, ceci requiert la disponibilité des valeurs historiques du TRS, ainsi que celles des données contextuelles collectées. Nous supposons que le montage de $P1$ nécessite 3 pièces et un outil avec un maximum de 3 opérateurs, qu'il n'est pas toujours monté selon le même ordonnancement, que le TRS ne doit pas être en deça de 70%, qu'il est recalculé toutes les 10 minutes, et que son état est spécifié (*i.e* s'il rempli l'objectif ou non). L'apprentissage se fera alors à partir des variables disponibles. Pour cet exemple, supposons que les variables contextuelles observées sont les suivantes :

— Nombre de produits défectueux (Df) ;
— Nombre de postes (NP) ;
— Choix de l'ordonnancement (O) ;
— Inventaire pièce 1 (I_1) ;
— Inventaire pièce 2 (I_2) ;
— Inventaire pièce 3 (I_3) ;
— Nombre de pièces 1 défectueuses (Df_1) ;
— Nombre de pièces 2 défectueuses (Df_2) ;
— Nombre de pièces 3 défectueuses (Df_3) ;
— Disponibilité de l'outil (DiOu) ;
— Durée de panne de l'outil (POu) ;
— Niveau de stress (S) ;
— Niveau de formation de l'opérateur au poste 1 (F_1) ;
— Niveau de formation de l'opérateur au 2 (F_2) ;
— Niveau de formation de l'opérateur au 3 (F_3) ;

— Niveau de détail et de clarté du guide de montage (G) ;
— Absence d'opérateur (Ab) ;
— Attentes (At) ;
— Température ambiante (Temp) ;
— Bruit ambiant (B) ;
— Niveau de luminosité (L) ;
— KPI : Acceptabilité du TRS (TRS).

Les variables contextuelles entrant en compte dans l'analyse causale peuvent éventuellement contenir d'autres KPIs, pour lesquelles une autre analyse causale peut être effectuée de la même manière. Dans cet exemple, ceci est représenté, entre autres, par le nombre de produits défectueux (qui est équivalent au taux de rebut), les attentes, ou encore le stress observé chez les opérateurs qui peut également être considéré comme un indicateur de performance dont l'entreprise peut se soucier pour améliorer le confort de ses opérateurs. Comme illustré sur la figure 3.4, un graphe global contenant toutes les causes directes et indirectes du KPI_1 d'intérêt, calculables ou mesurables, peut être construit. Afin de faciliter la prise de décision, surtout lorsque le système étudié est complexe, une visualisation des causes directes uniquement du KPI étudié est prévue dans notre proposition, comme le montre la figure 3.4.a. Le décideur peut ensuite s'attarder sur une cause en particulier, qui peut être un autre KPI (dans la figure 3.4, il s'agit du KPI_2), ou simplement une variable mesurée ou calculée, afin de voir les éléments lui étant liés causalement, et par conséquent de pouvoir remonter aux causes racines si on le souhaite.

Un KPI, ou plus généralement une quelconque variable peut être commune à deux ou plusieurs tables de données donnant lieu à deux ou plusieurs graphes causaux. Quand c'est le cas, les graphes résultants peuvent être associés pour donner lieu à un seul graphe global, comme illustré sur la figure 3.4.b. Ce graphe global peut à son tour être décomposé pour ne visualiser que les causes directes d'un KPI ou une variable d'intérêt.

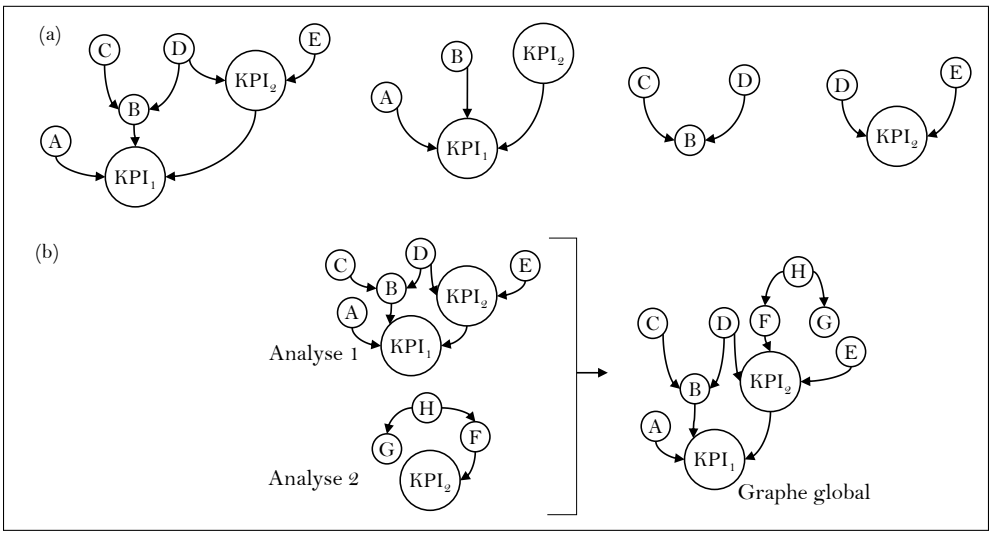


FIGURE 3.4 – (a) Décomposition d'un graphe causal pour visualisation des causes directes, (b) association de deux graphes issus d'analyses différentes.

Dans cet exemple, une observation est enregistrée toutes les 10 minutes. Une observation correspond à une ligne de tableau, dont les colonnes correspondent aux différentes variables citées plus haut. Pour chaque observation, chacune des variables est collectée et/ou calculée. Pour les variables changeant plusieurs fois dans un intervalle de 10 minutes, comme cela peut être le cas pour la température ou le bruit, leurs valeurs enregistrées pour une observation peuvent être des agrégations représentatives des différentes valeurs mesurées (comme la moyenne). Par ailleurs, dans certaines situations, certaines variables, comme c'est le cas dans notre exemple pour les niveaux de formation des opérateurs affectés aux postes 2 et 3, peuvent être non applicables, c'est notamment le cas lorsque le produit $P1$ est monté sur un seul poste par un seul opérateur. Dans ce cas, les valeurs manquantes doivent être traitées de manière adéquate, c'est à dire que l'on se trouve dans un cas où les valeurs sont manquantes "volontairement" car elles sont facultatives, ou plus précisément impossibles à renseigner lorsque le produit est monté sur un seul poste par un seul opérateur. On parle alors de valeurs dont l'absence dépend d'une ou plusieurs autres variables observées (Minini & Chavance, 2004) ; dans notre exemple, il s'agit du nombre de postes et par conséquent de l'ordonnancement choisi. Pour rappel, nous avons émis comme hypothèse que toutes les données sont complètes et qu'il n'y a pas de valeurs manquantes pour des variables par omission ou par erreur. Les cas semblables à celui que nous venons de citer sont alors les seuls cas où des valeurs peuvent être manquantes, toujours selon notre hypothèse. Il n'en reste pas moins que le constat est que certaines valeurs "doivent" être manquantes dans certains cas comme celui que nous venons de voir, et que leur traitement demeure indispensable, puisque la majorité des méthodes statistiques ne peut pas les manipuler en l'état (Tufféry, 2011).

Il existe plusieurs manières pour le traitement des valeurs manquantes (Giorgi, 2013 ; Tufféry, 2011) :

- L'analyse de données complètes qui revient à ignorer les observations (lignes) contenant des valeurs manquantes pour certaines variables : lorsque les données sont manquantes d'une manière complètement aléatoire, ceci peut s'avérer être une bonne alternative, bien que ce ne soit pas toujours le cas, puisqu'en supprimant les observations incomplètes, on peut se retrouver avec des données très réduites en nombre bien que pour chaque variable, la probabilité que sa valeur soit manquante soit faible. Cependant, lorsque les valeurs sont manquantes pour une certaine catégorie des observations, ce qui est le cas dans cet exemple, ceci peut introduire des biais importants dans l'analyse, puisque toute(s) la/les catégorie(s) concernée(s) sera/seront ignorée(s). Dans notre exemple, ceci reviendra à ignorer tous les cas où le produit est monté sur un ou deux postes, et seuls les cas où le produit est monté sur trois postes seront pris en compte, ce qui biaisera l'analyse de la causalité que nous souhaitons conduire par la suite ;
- L'analyse de données complètes qui revient à ignorer les variables (colonnes) contenant des valeurs manquantes : ceci requiert un jugement préalable sur la pertinence de la contribution de la/les variables concernée(s) dans l'analyse à venir. Le fait d'établir qu'une variable est essentielle à notre analyse ou non va à l'encontre de nos objectifs, puisque l'objectif est de découvrir de nouveaux liens de causalité entre le KPI étudié et les variables contextuelles ;
- L'imputation des données : qui revient à substituer les valeurs manquantes par des valeurs déterminées statistiquement ou grâce à des connaissances. En général, si les connaissances sont indisponibles, les méthodes les plus simples utilisées sont le

remplacement par la valeur la plus fréquente pour l'imputation des variables qualitatives, et par la moyenne pour l'imputation des variables quantitatives. D'autres méthodes plus élaborées, permettant de moins altérer la distribution des données, existent également, telles que la régression et la classification (Nakache & Confais, 2004). Néanmoins, dans les cas similaires au nôtre, cette méthode n'est pas raisonnable, puisque les variables concernées ne peuvent simplement prendre aucune valeur dans des situations bien précises ;

— La substitution par indicateur de données manquantes : qui revient à traiter les valeurs manquantes par l'introduction de nouvelles valeurs contenant de l'information. Une nouvelle catégorie est alors créée pour représenter ces valeurs manquantes. Néanmoins, cette méthode n'est pas directement applicable aux variables quantitatives car n'importe quelle valeur choisie pour représenter les valeurs manquantes aurait une autre signification. Ce problème peut être contourné en discrétisant la variable puis en ajoutant une nouvelle classe représentant les valeurs manquantes. Nous faisons alors le choix d'opter pour cette méthode puisqu'elle nous semble être la plus appropriée à notre cas de figure, où les valeurs manquantes ont un sens particulier et contiennent de l'information.

Dans le chapitre précédent, nous avons expliqué que nous procéderons à l'analyse causale par apprentissage de structure de réseaux Bayésiens, couplé à l'injection de connaissances humaines pour apport d'information et/ou pour vérification. Nous représentons alors cette première brique composant notre proposition, conformément à notre exemple, comme illustré sur la figure 3.5. Ce diagramme SADT illustre le cas de figure le plus générique, où une structure causale est obtenue grâce à un algorithme faisant l'apprentissage à partir d'un seul ensemble de données disponibles, et conditionné par des connaissances humaines qui permettent 1) de restreindre le champs de recherche en contraignant la structure par des relations déjà connues, et/ou 2) de valider ou modifier la structure obtenue par apprentissage. Nous verrons dans les sections à venir comment cet apprentissage est effectué et couplé aux connaissances humaines.

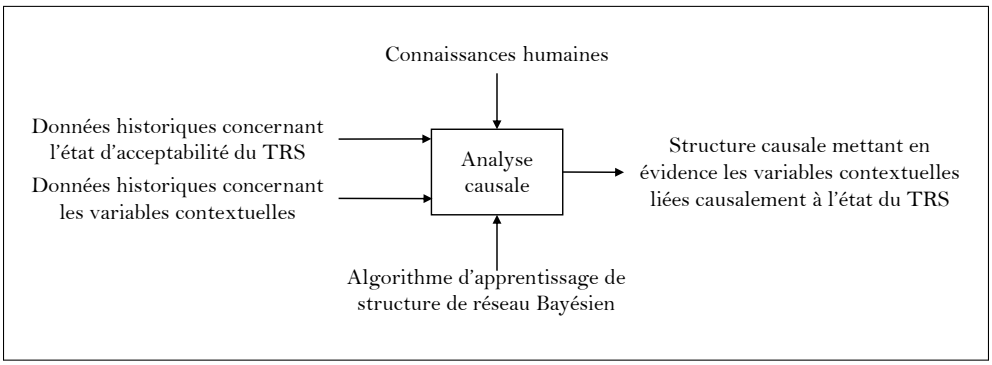


FIGURE 3.5 – L'analyse causale, première brique de l'approche proposée, répondant à la fonction F1.

Lorsque plusieurs jeux de données différents ayant des éléments communs sont disponibles, comme nous l'avons expliqué plus haut, cette brique devrait être dupliquée autant de fois que le nombre des jeux de données en question. Cette brique devient alors une macro-brique contenant plusieurs sous-briques d'analyse causale, et donnant lieu à une structure causale qui est obtenue par la mise en commun des structures causales obtenues pour chaque jeu de données.

3.2.2 Hiérarchisation des causes

La deuxième étape de la proposition, qui répond à la fonction F2, est de classer les causes identifiées par ordre d'importance, afin de fournir une information utile au décideur en lui indiquant l'entité sur laquelle l'action serrait la plus efficace. Nous nous concentrons alors dans cette étape uniquement sur les variables identifiées comme étant liées causalement au KPI d'intérêt. Comme nous l'avons expliqué à la section 2.4, cette étape est mise en œuvre grâce à la construction d'un modèle de réseau de neurones fournissant un bon pouvoir de prédiction, dont on exploitera les poids finaux. Comme précédemment expliqué, l'interprétation des tables de probabilités conditionnelles associées à un réseau Bayésien n'est pas toujours aisée lorsque le système étudié est complexe. En effet, pour un réseau Bayésien donné, le nombre de probabilités conditionnelles npc_x devant être spécifiées et interprétées pour chaque nœud x ayant des parents peut être calculé à partir de la formule (3.2) :

$$npc_x = (ne_x - 1) \times \prod_{i=1}^{np_x} ne_i \tag{3.2}$$

où ne_x représente le nombre d'état que prend le nœud x, np_x le nombre des parents du nœud x, et ne_i le nombre d'état que prend un parent i du nœud x. Pour un nœud y n'ayant pas de parents, le nombre de probabilités marginales à spécifier peut être calculé à partir de la formule (3.3) :

$$npm_y = ne_y - 1 \tag{3.3}$$

Pour l'ensemble d'un réseau, le nombre np de probabilités conditionnelles et marginales à spécifier et interpréter est donné par la formule (3.4), où n représente le nombre de nœuds avec parents, et m le nombre de nœuds sans parents :

$$\begin{aligned} np &= \sum_{k=1}^{n} npc_k + \sum_{kj1}^{m} npm_j \\ &= \sum_{k=1}^{n} [(ne_k - 1) \times \prod_{i=1}^{np_k} ne_i] + \sum_{j=1}^{m} (ne_j - 1) \end{aligned} \tag{3.4}$$

Si nous admettons que le graphe illustré sur la figure 3.6 représente le réseau Bayésien associé à notre exemple, et obtenu suite à l'étape de l'analyse causale, nous pouvons alors conclure que pour l'ensemble du graphe présumé, il faudra interpréter 18 tables de probabilités conditionnelles et marginales, avec un total de $\sum_{k=1}^{8}[(ne_k - 1) \times \prod_{i=1}^{np_k} ne_i] + \sum_{j=1}^{11}(ne_j - 1)$ probabilités . Si nous admettons, en plus, que chaque nœud ne peut prendre que deux états différents, nous aurions alors 361 probabilités réparties sur les 18 tables (11 probabilités marginales réparties sur 11 tables associées aux 11 nœuds n'ayant pas de parents, et 350 probabilités conditionnelles réparties sur 8 tables associées aux 8 nœuds ayant des parents). Dans le cas où les nœuds possèdent un nombre d'états encore plus important, la complexité des tables augmente et les rends plus difficiles à interpréter (Talvitie, Eggeling, & Koivisto, 2019). Par conséquent, l'obtention d'un classement des causes qui soit exploitable pour la prise de décision, à partir de ces tables, est une tâche laborieuse et coûteuse en temps et en ressources humaines.

L'approche que nous proposons pour le classement des causes s'appuie sur les poids finaux d'un réseaux de neurones, et se base par conséquent uniquement sur les données.

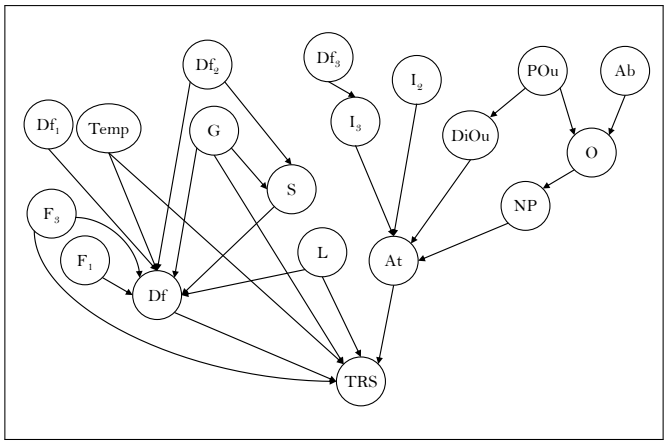


FIGURE 3.6 – Graphe causal issu de l'analyse causale associée à l'exemple explicatif.

Nous pouvons assimiler l'objectif de cette approche à celui d'une analyse de la variance permettant d'identifier les influences qu'ont les prédicteurs sur le KPI d'intérêt, pour ensuite les hiérarchiser. Par conséquent, l'approche proposée ne permet pas de hiérarchiser les causes à partir d'un jeu de données où les causes sont inconnues, mais permet plutôt de hiérarchiser tous les prédicteurs que le réseau de neurones prend en entrée. Pour cette raison, cette approche doit obligatoirement se dérouler après l'analyse causale d'un point de vue chronologique. En effet, afin que la hiérarchisation obtenue ait un sens causal, seules les causes identifiées devront faire partie des données d'entrée du réseau de neurones. Lorsque qu'il est avéré que les entrées du réseaux de neurones sont des causes de sa sortie, les importances prédictives des variables peuvent être assimilées avec leurs importances causales (Hippert & Taylor, 2010).

Par ailleurs, l'utilisation de cette technique après avoir obtenu un graphe causal permet également de réduire le nombre de prédicteurs en entrée du réseau de neurones, puisque la qualité d'un modèle dépend du nombre de variables discriminantes retenues pour effectuer l'apprentissage. Dans l'exemple que nous décrivons ici, la construction de la structure causale a permis de passer de 21 à 18 variables explicatives. En effet, l'utilisation d'un grand nombre de variables en entrée peut causer un sur-entrainement du modèle et par conséquent une capacité de généralisation médiocre, tandis que l'utilisation d'un nombre trop faible de variables en entrée aurait pour conséquence une qualité de prédiction médiocre. Nous soulignons tout de même qu'il n'existe pas de nombre ni d'intervalle idéal pour le nombre de variables d'entrée nécessaires pour construire un réseau de neurones, puisqu'il est à moduler selon la taille du jeu de données devant être représentatif des différentes situations rencontrables : plus le nombre d'observations disponibles pour entrainer le réseau est grand, plus il est possible d'augmenter le nombre de variables d'entrée si besoin (Tufféry, 2011).

Dans le cas idéal, toutes les causes identifiées peuvent être utilisées en entrée du réseau de neurones, comme le montre la figure 3.7. Si toutefois le nombre de variables discriminantes s'avère malgré tout très grand, et qu'il s'avère impossible de construire un réseau de neurones avec un bon pouvoir de généralisation, le graphe causal obtenu suite à l'étape précédente peut être segmenté pour se concentrer dans un premier temps uniquement sur les causes directes du KPI d'intérêt et construire un réseau de neurones qui prend en entrée uniquement les causes directes ; puis se concentrer dans un deuxième temps sur la

ou les causes directes la ou les plus importantes, et construire le(s) réseau(x) de neurones correspondant(s) pour classer leur(s) causes. Dans notre exemple, ceci revient à dire de se concentrer sur les causes directes de la variable TRS mises en évidence sur le sous-graphe causal illustré sur la figure 3.8. Un réseau de neurones sera ensuite construit pour la prédiction de la variable TRS, et ses poids finaux seront ensuite exploités, comme nous le verrons plus tard dans ce chapitre, pour hiérarchiser ses causes directes. Si l'on admet que la hiérarchisation dévoile que la variable *At* est la plus importante, la même opération sera ensuite effectuée pour hiérarchiser les causes de la variable *At*, comme le décrit la figure 3.8. D'autres méthodes existent également pour réduire le nombre de variables en entrée d'un réseau de neurones, la plus utilisée étant d'effectuer une analyse factorielle au préalable, dont les composantes principales représenteront les entrées du réseau de neurones (Tufféry, 2011). Néanmoins, l'utilisation d'une telle technique entraînera une perte d'information, dans le sens où l'information contenue dans l'analyse causale ne sera pas exploitée. Cette technique peut tout de même être utilisée en amont de l'analyse causale, où les composantes principales représenteront les nœuds du graphe causal. De cette manière, nous traiterons alors des "groupes de causes", et non des causes à part entière, ce qui entraînera donc une perte de granularité dans l'analyse et par conséquent une perte d'information, mais qui resterait tout de même moins importante que de conduire une analyse factorielle directement en entrée du réseau de neurones et après l'obtention du graphe causal.

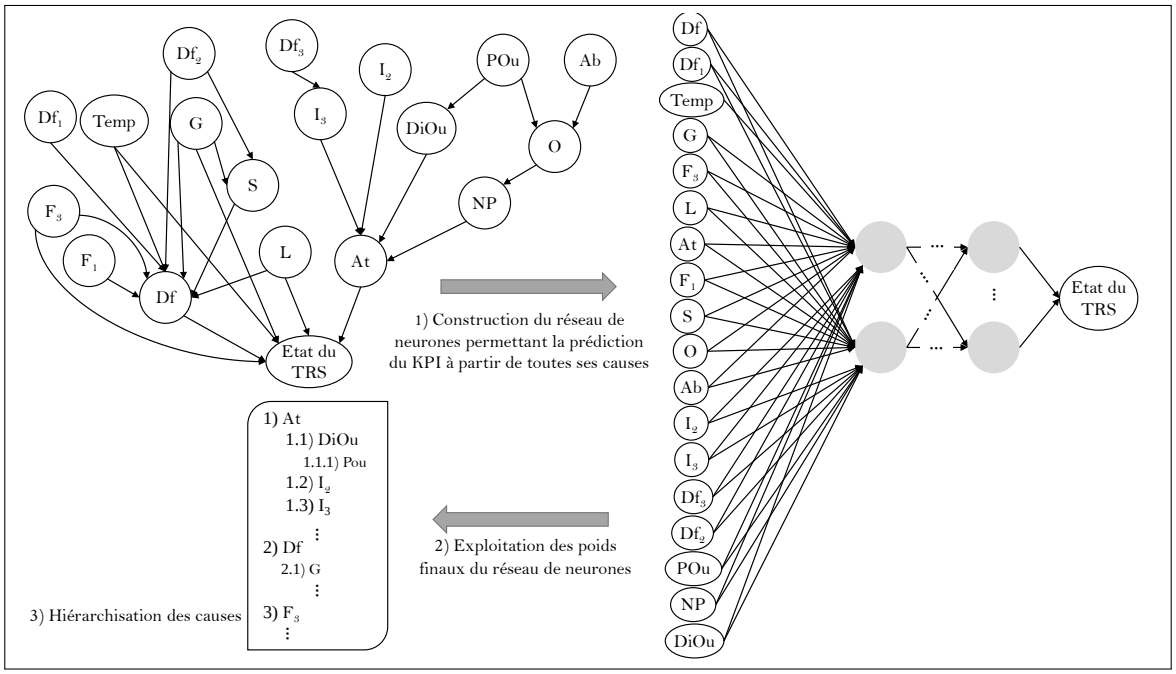


FIGURE 3.7 – Processus de hiérarchisation à partir d'un réseau de neurones prenant l'ensemble des causes en entrée.

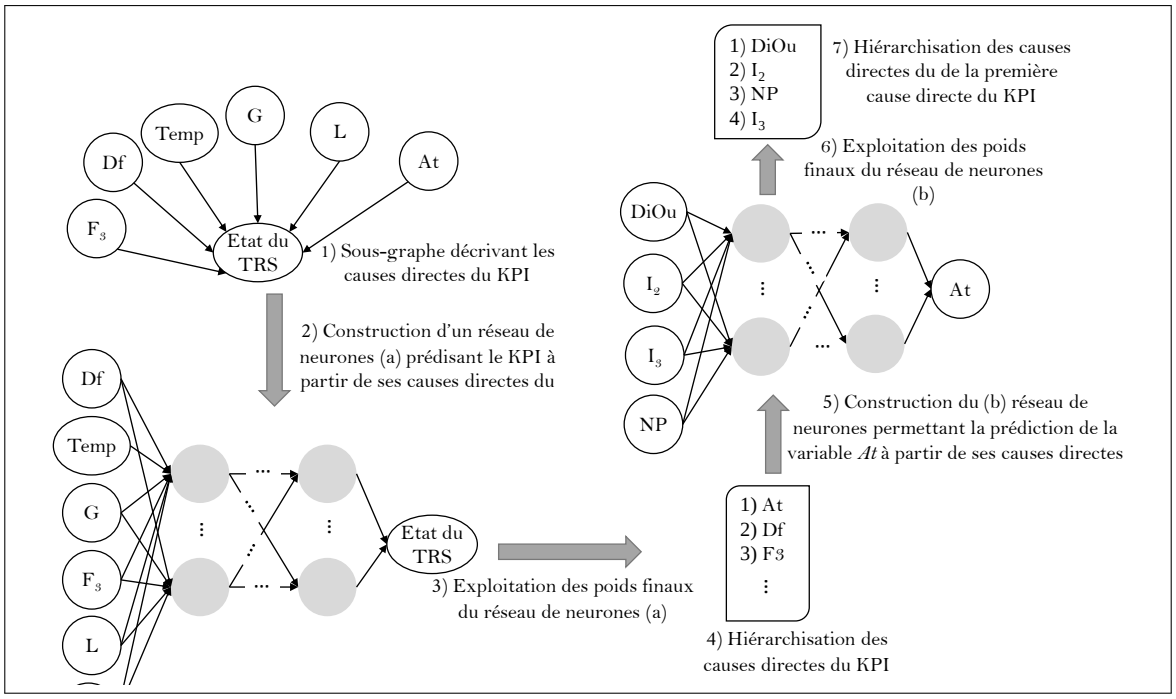


FIGURE 3.8 – Processus progressif de hiérarchisation des causes à partir d'un réseau de neurones prenant en entrée les causes directes.

Dans les deux cas de figure que nous venons d'énoncer, l'essence même de la méthode reste la même pour cette seconde composante de notre proposition. Les entrées nécessaires pour cette brique sont sensiblement les mêmes que pour la première : il s'agit des données historiques du KPI traité, ainsi que celles des données contextuelles, à la différence qu'ici, seules les données historiques des variables contextuelles ayant été identifiées comme causes sont nécessaires. Sur le diagramme SADT présenté sur la figure 3.9, ceci se traduit par la présence d'une contrainte de contrôle représentant la structure causale préalablement identifiée. En effet, c'est cette structure causale qui permettra d'identifier les variables d'entrée du réseau de neurones dont les poids finaux seront exploités, ou, si nécessaire ou préférable, de segmenter la structure causale pour identifier les causes directes qui seront attribuées en entrées du réseau de neurones.

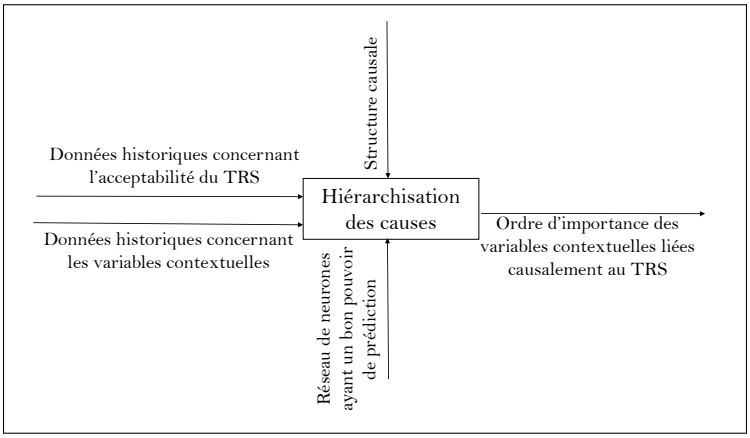


FIGURE 3.9 – La hiérarchisation des causes, deuxième brique de l'approche proposée.

La conjugaison des deux briques qui ont été décrites jusqu'ici donne alors le diagramme

SADT illustré sur la figure 3.10, qui représente les réponses aux deux fonctions principales F1 et F2 de notre proposition.

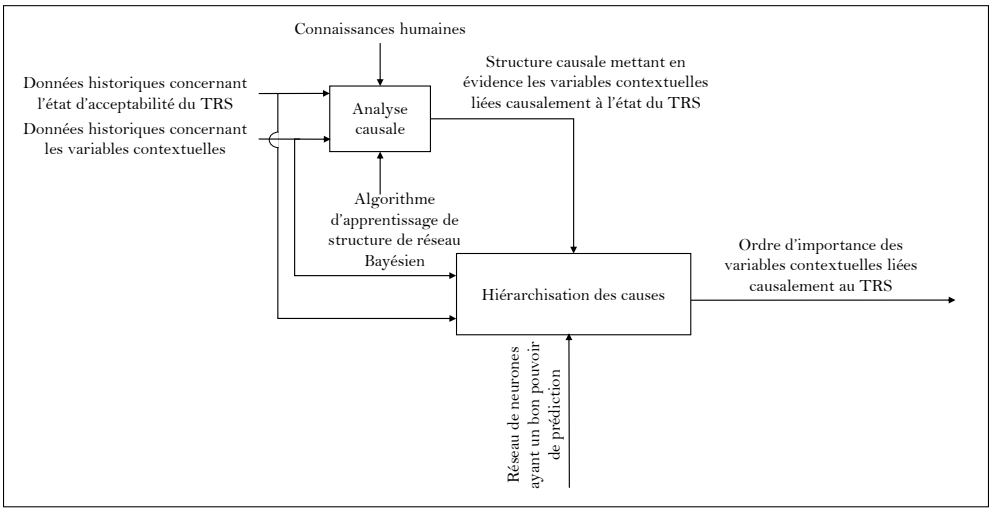


FIGURE 3.10 – Diagramme SADT représentant l'enchainement des deux fonctions F1 et F2 de la proposition.

3.2.3 La neuro-évolution

Afin que la fonction de hiérarchisation des causes soit accomplie avec rigueur, et que ses résultats soient les plus fiables possible, il faut qu'elle soit basée sur des poids issus d'un réseau de neurones ayant un bon pouvoir de prédiction. En effet, un réseau de neurones ayant de mauvaises performances en termes de prédiction signifie que l'apprentissage effectué n'a pas bien capturé les relations entre les prédicteurs et la variable à prédire, et que les poids finaux issus de cet apprentissage ne reflètent donc pas la réalité. Par conséquent, toute opération réalisée sur la base des paramètres d'un tel réseau donnera forcément lieu à des résultats erronés, puisque les paramètres utilisés comme base, en l'occurrence les poids, sont fallacieux. Il est donc primordial de pouvoir disposer, pour chaque ensemble données d'entrée/variable à expliquer, un réseau de neurones performant (*i.e.* ayant appris des paramètres qui lui auront permis de prédire correctement). Ceci nous permettra par la même occasion de disposer d'un modèle de prédiction qui permettra de répondre à la fontion F3.

La performance d'un réseau de neurones est conditionnée par sa structure, également appelée topologie, ainsi que les paramètres associés aux réseau, dont la définition est conditionnée par la structure. En effet, la première étape lors d'une démarche classique de construction d'un réseau de neurones à Perceptron multicouche est de définir la structure du réseau, c'est à dire les entrées, le nombre de couches cachées, le nombre de neurones par couche cachées, et enfin la sortie attendue. Concernant la définition des entrées, nous l'avons déjà abordée plus haut, et elle ne constitue généralement pas un obstacle insurmontable lors de la construction des réseaux de neurones. Le problème se pose plutôt lors de la définition du nombre de couches cachées et de neurones par couche cachée. Lorsque ces éléments sont fixés, les poids évoluent afin de minimiser l'erreur en sortie au fur et à mesure de l'apprentissage. La performance du réseau dépend donc de la manière dont les poids évoluent, mais cette évolution est restreinte par la structure du réseau, et la performance du réseau est alors limitée par le choix de sa structure (Mouret, 2015). Afin

d'illustrer ce problème, admettons que nous cherchons à construire le réseau de neurones permettant la prédiction de l'acceptabilité (oui ou non) du TRS à partir de ses causes directes, que nous exploiterons ensuite pour hiérarchiser les causes directes, comme précédemment illustré sur la figure 3.8. Admettons aussi que nous définissons une structure au préalable pour conduire l'apprentissage, constituée d'une couche cachée contenant trois neurones, comme le décrit la figure (3.11).

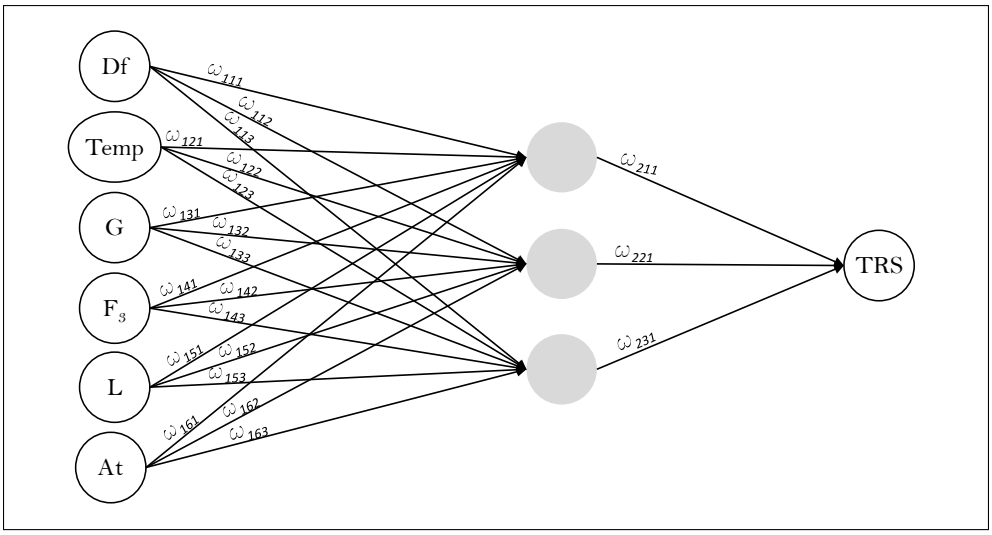


FIGURE 3.11 – Exemple de réseau de neurones à une couche cachée de trois neurones pour la prédiction du TRS à partir de ses causes directes.

L'apprentissage, qui consiste en l'optimisation pour estimer les poids et biais du réseau, est limité à cette structure fixe. Pour simplifier l'explication, nous faisons ici abstraction des biais associés aux neurones du réseau. L'objectif de l'apprentissage est de minimiser la fonction de perte, qui reflète en quelques sortes la qualité du réseau en terme d'apprentissage. Nous nous situons dans un cas de classification binaire, nous admettons alors que la fonction de perte choisie est la perte d'entropie croisée, qui est définie, dans le cas binaire, par l'égalité 3.5.

$$\begin{aligned} e(y_i^*, y_i) &= -y_i.log(y_i^*) - (1-y_i).log(1-y_i^*) \\ &= -y_i.log(f(x_i, W)) - (1-y_i).log(1-f(x_i, W)) \end{aligned} \tag{3.5}$$

où y_i représente la sortie attendue pour la $i^{ème}$ observation $[x_{i_1}, x_{i_2}, ..., x_{i_n}]$, et y_i^* représente la sortie estimée par le réseau pour la $i^{ème}$ observation. Cette estimation est obtenue en calculant $f(x_i, W)$, qui opère, au fil des couches, les transformations effectuées par les fonctions d'activation sur les combinaisons linéaires déterminées par les valeurs d'entrée des neurones et leurs poids associés. La sortie estimée est alors fonction du vecteur d'entrées x_i du réseau, et de W, la matrice des poids associés à chaque liaison entre les neurones. L'erreur $e(y_i^*, y_i)$ est calculée pour chaque observation x_i du jeu d'entraînement, ce qui permet ensuite de calculer l'erreur moyenne, donnée par (3.6), pour l'ensemble du jeu d'entrainement.

$$E = \frac{1}{N}. \sum_{i=1}^{N} e((x_i, W), y_i) \tag{3.6}$$

où N représente la taille de l'échantillon d'entraînement. Il existe deux grandes familles de méthodes pouvant être utilisées pour optimiser l'erreur en sortie du réseau de neurones : les méthodes dites *en-ligne*, et les méthodes dites *hors-ligne*. Les méthodes *en-ligne* rétro-propagent l'erreur commise pour chaque observation, et mettent à jour les poids du réseau après chaque observation ou "exemple" qui lui est présenté. Quant aux méthodes *hors-ligne*, également appelées méthodes *batch*, ne font évoluer les poids qu'à la fin du passage de toutes les observations du jeu d'entraînement, elles sont alors basées uniquement sur la rétro-propagation de l'erreur moyenne sur l'ensemble du jeu de données. Dans cette section, nous ne rentrons pas dans les détails des techniques utilisées pour rétro-propager l'erreur et faire évoluer les poids, auxquelles nous reviendrons dans le prochain chapitre. L'objectif ici est plutôt d'introduire ce en quoi la neuro-évolution nous serait utile dans notre proposition. En effet, quelle que soit la technique utilisée pour effectuer l'apprentissage, le constat reste le même : la matrice W des poids intervient en tout état de causes, et conditionne alors le champs des possibles lors de la minimisation de l'erreur en sortie. Si nous admettons, en plus, que la fonction d'activation choisie pour les neurones de la couche cachée et celui de la couche de sortie de notre exemple est la fonction sigmoïde, l'erreur sur la sortie d'une observation sera alors calculée conformément à la formule donnée par (3.7) :

$$\begin{aligned} e(y_i^*, y_i) &= -y_i.log(f(x_i, W)) - (1 - y_i).log(1 - f(x_i, W)) \\ &= -y_i.log(\frac{1}{1 + e^{-\sum_{j=1}^{3} w_{2j1}.\frac{1}{1+e^{-\sum_{k=1}^{6} x_{i_k}.w_{1kj}}}}}) \\ &- (1 - y_i).log(1 - (\frac{1}{1 + e^{-\sum_{j=1}^{3} w_{2j1}.\frac{1}{1+e^{-\sum_{k=1}^{6} x_{i_k}.w_{1kj}}}}})) \end{aligned} \tag{3.7}$$

Si la fonction d'activation choisie pour le neurone de sortie est la fonction sigmoïde, et que celle choisie pour les neurones de la couche cachée est la fonction de la tangente hyperbolique, l'erreur sur la sortie d'une observation sera alors calculée conformément à la formule donnée par (3.8) :

$$\begin{aligned} e(y_i^*, y_i) &= -y_i.log(f(x_i, W)) - (1 - y_i).log(1 - f(x_i, W)) \\ &= -y_i.log(\frac{1}{1 + e^{-\sum_{j=1}^{3} w_{2j1}.tanh(\sum_{k=1}^{6} x_{i_k}.w_{1kj})}}) \\ &- (1 - y_i).log(1 - (\frac{1}{1 + e^{-\sum_{j=1}^{3} w_{2j1}.tanh(\sum_{k=1}^{6} x_{i_k}.w_{1kj})}})) \end{aligned} \tag{3.8}$$

Nous pouvons constater qu'en plus de la topologie qui définit la taille de la matrice des poids associée au réseau, le choix de la ou les fonction(s) d'activation conditionne également la qualité de l'apprentissage. Le choix d'utiliser une fonction d'activation plutôt qu'une autre dépend de plusieurs caractéristiques du problème traité, dont le nombre de neurones de sortie du réseau, le type de la ou les sortie(s) attendue(s), le(s) type(s) des données d'entrée et leurs intervalles de variation. En outre, l'algorithme utilisé pour l'optimisation des poids conditionne également la qualité de l'apprentissage. A cela s'ajoute aussi d'autres hyper-paramètres pouvant être associés à l'entraînement du réseau, tels que le taux d'apprentissage et son adaptabilité, le nombre maximal des itérations (epochs) de l'entraînement, le taux de régularisation, l'arrêt prématuré, etc. Nous reviendrons plus en

détails sur l'ensemble de ces éléments dans le chapitre suivant.

La méthode classique pour définir un réseau de neurones pour un problème donné est d'enchaîner les entraînements et tests pour différentes combinaisons de topologies, algorithmes d'optimisation, et hyper-paramètres, pour ensuite sélectionner celle qui fournit les meilleurs résultats d'apprentissage et de généralisation. Cette démarche expérimentale itérative est à effectuer de façon individuelle pour chaque problème à traiter (*i.e.* dans notre cas, pour chaque KPI d'intérêt). Si nous considérons par exemple le processus illustré sur la figure 3.8, il faudra alors dérouler ces expérimentations d'une part pour construire le réseau de neurones prédisant la variable TRS, et d'autre part pour le réseau de neurones prédisant la variable *At*. Au vu du nombre important de KPIs d'intérêt dans un contexte industriel, et du nombre de combinaisons à explorer, la démarche itérative d'entrainement et de test des différentes combinaisons devient rapidement très coûteuse en temps et en ressources humaines, chose qui va à l'encontre de nos objectifs.

À ce jour, il n'existe pas de modèle de réseau de neurones capable de résoudre n'importe quel problème, ni même tous les problèmes appartenant à une même catégorie. Dit autrement, il n'existe pas de modèle de réseau de neurones offrant, à titre d'exemple, une bonne capacité d'apprentissage et de généralisation pour n'importe quel problème à sortie binaire unique, et à n entrées numériques. Bien que les questions autour de la définition de la topologie et des paramètres ont émané depuis plus d'une vingtaine d'années (Stathakis, 2009), il n'existe pas non plus de processus ou démarche à suivre rigoureusement pour obtenir la structure la plus appropriée pour un problème donné (Karsoliya, 2012). Par conséquent, nous enrichissons l'architecture de notre proposition en y ajoutant une brique supplémentaire dédiée à cela, que nous appelons "construction de réseau de neurones". L'objectif de cette brique est donc de fournir le "meilleur" réseau de neurones pour un problème donné. À l'image de la brique précédente, elle prend en entrée l'ensemble des valeurs historiques des variables contextuelles ayant été identifiées comme causes, ainsi que celle du KPI ou de la variable d'intérêt, la contrainte de contrôle est alors la structure causale obtenue suite à l'analyse causale. La raison pour laquelle nous avons mis des guillemets autour du terme "meilleur", réside dans le fait que le moyen qui sera mobilisé pour construire le réseau de neurones est une métaheuristique, qui ne permet pas de prouver la convergence vers une solution optimale globalement. Comme le nom de cette section l'indique, nous allons faire usage d'un algorithme évolutionnaire, que nous allons détailler dans le chapitre suivant. Bien que l'évolution naturelle du cerveau humain ou animal au fil des générations ne puisse pas être égalée par les systèmes artificiels, l'inspiration de cette théorie a été d'une grande utilité pour la construction des réseaux de neurones (Stanley, Clune, Lehman, & Miikkulainen, 2019). La neuro-évolution consiste donc à utiliser des algorithmes évolutionnaires afin de faire évoluer la topologie est les hyper-paramètres d'un réseau de neurones, dans le but d'en améliorer la performance. En effet, les algorithmes évolutionnaires, tels que les algorithmes génétiques, se sont révélés efficaces pour obtenir un réseau de neurones avec de bonnes performances, ou du moins plus efficaces que les approches qui ne font évoluer que les poids et qui sont donc basées sur une structure et des hyper-paramètres fixes (Doncieux, 2010). De plus, il serait inenvisageable d'essayer de construire les réseaux de neurones par apprentissage, puisqu'il faudrait dans ce cas avoir une base d'exemples représentative, de cardinal très important, dont il est presque impossible de disposer. Il faudrait alors que l'algorithme utilisé pour construire les réseaux de neurones puisse explorer de façon autonome l'espace des solutions, et exploiter celles

qui semblent être les plus performantes.

La figure 3.12 illustre cette troisième brique de la proposition. Nous optons alors, comme décrit sur cette figure, pour l'utilisation d'un algorithme génétique, afin de faire évoluer la topologie et les hyper-paramètres des réseaux de neurones.

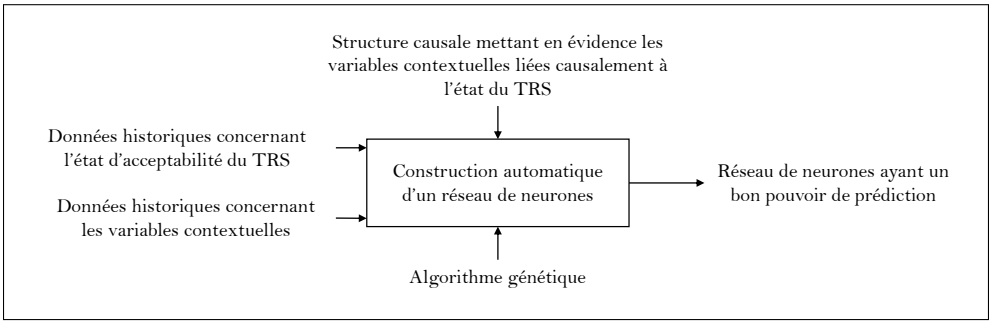


FIGURE 3.12 – Construction de réseaux de neurones par neuro-évolution, troisième brique de l'approche proposée.

Plusieurs travaux se sont intéressés à l'utilisation des algorithmes génétiques pour la construction des réseaux de neurones (Ahmadizar, Soltanian, AkhlaghianTab, & Tsoulos, 2015 ; Stanley et al., 2019 ; Stathakis, 2009 ; Mantzaris, Anastassopoulos, & Adamopoulos, 2011a ; Yang & Chen, 2012 ; Castellani, 2013 ; De Campos, Roisenberg, & de Oliveira, 2011). Nous nous en sommes donc inspirés pour produire notre propre algorithme, que nous présentons dans le chapitre suivant, et qui propose un nouvel encodage permettant de prendre en compte certains aspects n'ayant pas été traités, ou ayant été traités séparément, par les travaux que nous avons explorés. La figure 3.13 présente le diagramme SADT mis à jour de notre proposition. Comme le montre ce diagramme, nous faisons le choix de garder intactes les contrôles et entrées de la deuxième brique "Hiérarchisation des causes". Nous justifions ce choix par le fait que la brique de "Construction automatique de réseau de neurones" n'est pas indispensable pour la réalisation des objectifs liés à nos fonctions principales F1, F2, et F3, dans le sens où la construction de réseau de neurones peut très bien se faire de manière manuelle. L'ajout de cette brique permet plutôt d'assurer les caractéristiques de non-chronophagie et d'utilisation restreinte des ressources humaines, attendues pour notre proposition.

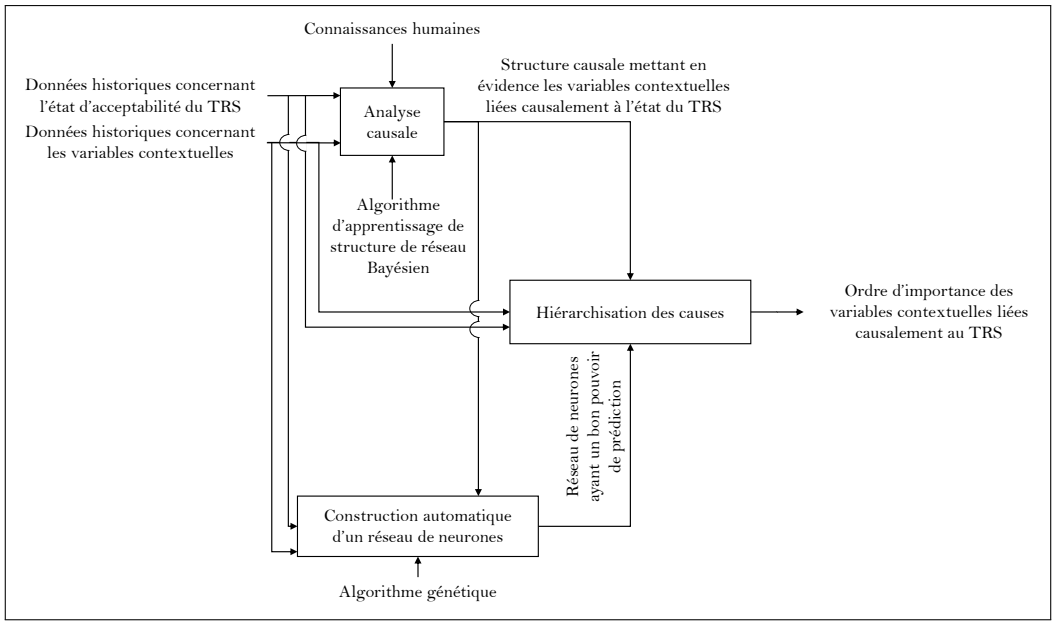


FIGURE 3.13 – Diagramme SADT illustrant le positionnement de la brique "Construction des réseaux de neurones" par rapport aux deux briques des fonctions F1 et F2.

3.2.4 Prédiction et exploitation des résultats pour la prise de décision

La dernière brique qui compose l'approche proposée représente d'abord une compilation des résultats issus des briques précédentes. Son objectif principal est de pouvoir fournir une aide à la décision sur les actions à entreprendre afin de 1) rétablir une situation normale suite à une dérive, ou 2) d'améliorer les performances sans qu'il n'y ait eu d'aléa. Pour ce qui est du premier cas de figure, et afin de maintenir une certaine cohérence avec les caractéristiques attendues pour notre proposition, nous introduisons la prédiction des situations anormales, qui se traduit par la prédiction de l'acceptabilité du KPI d'intérêt, ou la prédiction du KPI suivie d'une comparaison avec un seuil ou un intervalle de tolérance. Les bénéfices liés à l'anticipation de la prise de conscience du besoin d'agir, décrits dans la section 1.2.2, et mis en évidence sur la figure 1.4, nous amènent à considérer le besoin de faire de la prédiction, d'autant plus que le moyen nécessaire pour le faire serait, à ce stade là, déjà disponible. En effet, le réseau de neurones qui permettra de faire la prédiction sera celui qui aura résulté de la troisième brique de notre proposition, et dont les poids seront utilisés pour la hiérarchisation des causes. Nous pourrons alors tirer profit des bénéfices qu'offre la prédiction, sans devoir faire de développement supplémentaire. Comme précédemment expliqué, lorsque la prédiction indiquera une dérive, la structure causale issue de la première brique (de l'analyse causale), ainsi que la hiérarchisation issue de la deuxième brique (de construction des réseaux de neurones), pourront alors être consultés afin de prévenir la dérive, et ce en agissant sur la cause la plus influente si cela s'avère intéressant pour le décideur. Le décideur aura toujours le choix de ne pas agir sur la première cause influente s'il estime que l'alternative n'est pas intéressante financièrement ou techniquement. Nous rappelons ici que l'ordre des causes fourni par le système proposé ne prend en compte aucun autre aspect que celui de la force de la relation entre une variable contextuelle et un KPI. Dans le deuxième cas de figure qu'est celui de l'amélioration des performances sans qu'il n'y ait eu de dérive, la

prédiction n'est pas indispensable. Le décideur aura simplement à consulter les causes du KPI qu'il souhaite améliorer, et leur ordre d'importance, puis prendre la décision d'agir en fonction des préconisations du système. La figure 3.14 illustre le diagramme SADT complet décrivant l'ensemble de notre proposition.

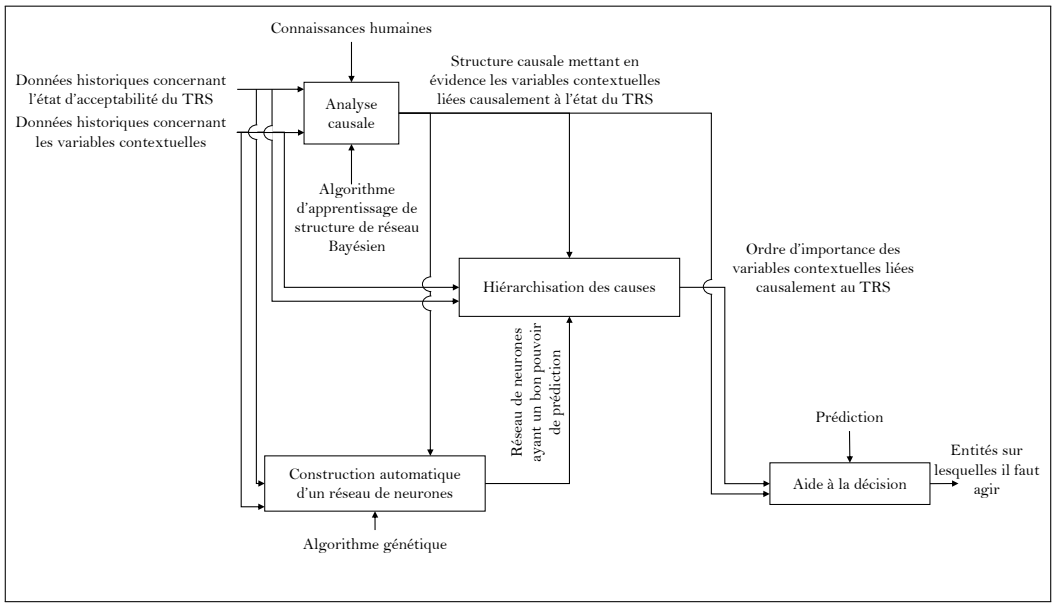


FIGURE 3.14 – Diagramme SADT décrivant l'ensemble de la proposition.

3.3 Conclusion

Dans ce chapitre, nous avons construit progressivement et à partir d'un exemple, l'architecture de notre proposition. Nous avons expliqué l'intérêt de chaque brique, et les moyens que nous envisageons de mettre en œuvre pour répondre à nos fonctions principales. Chacune de ces briques nous permettra d'apporter une réponse à l'une des trois fonctions que nous nous sommes fixé au chapitre 1.

L'objectif de ce chapitre était de permettre une meilleure compréhension des positionnements respectifs de chaque technique que nous présenterons dans le chapitre qui suit, et d'éclairer de quelle manière chacune de ces techniques contribuera à notre proposition. La figure 3.15 illustre une vue d'ensemble de notre proposition, qui peut être segmentée en deux phases : 1) une phase de développement, qui comprend les étapes à effectuer au départ pour chaque KPI d'intérêt afin d'obtenir la structure causale et la hiérarchisation des causes, cette phase peut être reconduite de façon ponctuelle si de nouvelles variables contextuelles sont disponibles ; et 2) une phase d'utilisation, qui consiste à exploiter les résultats des analyses effectuées lors de la phase de développement, afin de fournir une aide à la décision. Les résultats issus des actions prises suite à la phase de développement peuvent être perçus comme une boîte à outils, qu'il faut développer une fois pour chaque KPI d'intérêt, et qui peut être consultée dès lors que le KPI d'intérêt est impliqué dans une prise de décision. Les éléments de cette boîte à outils peuvent être reconfigurés ponctuellement en cas de disponibilité de nouvelles données.

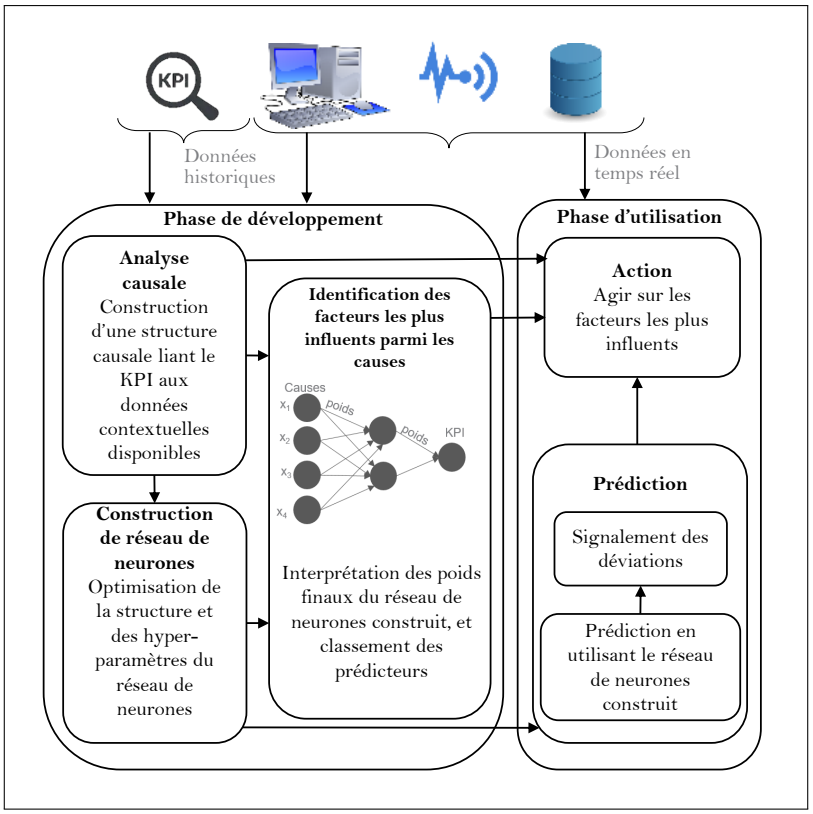


FIGURE 3.15 – Diagramme SADT décrivant l'ensemble de la proposition.

Chapitre 4

Mise en œuvre de la proposition

4.1 Introduction

Dans le chapitre précédent, nous avons introduit la conception de notre proposition, en construisant pas à pas le diagramme SADT qui décrit son architecture globale. Dans ce chapitre, nous allons présenter les techniques qui permettent de mettre en œuvre chacune des briques qui composent l'architecture proposée. Ce chapitre sera segmenté en trois sections dont chacune traitera du développement proposé pour mettre en œuvre 1) la construction de la structure causale, 2) la construction des réseaux de neurones, et 3) la hiérarchisation des causes. Le choix de présenter la technique de construction des réseaux de neurones avant celle de la hiérarchisation des causes, est d'abord de pouvoir disposer facilement d'un réseau de neurones offrant un bon pouvoir de prédiction et de généralisation pour pouvoir ensuite en exploiter les poids finaux. La deuxième raison est de pouvoir valider l'ordre fourni par la hiérarchisation, et pour cela, il faudra disposer de plusieurs réseaux de neurones, comme nous le verrons dans ce chapitre.

4.2 Construction de la structure causale

4.2.1 Choix du type de l'algorithme d'apprentissage

Comme discuté dans la section 2.2, et afin de répondre à notre première fonction F1, nous avons opté pour la construction de structure causale en se basant sur l'apprentissage des structure des réseaux Bayésien causal et sur les connaissances humaines. Comme discuté auparavant dans la sous-section 2.2.6, il existe deux familles d'approches pour l'apprentissage des réseaux Bayésiens : les approches basées sur la recherche d'indépendances conditionnelles, et celles basées sur l'optimisation d'un score.

Parmi les algorithmes basés sur la recherche d'indépendances conditionnelles les plus connus, aussi appelée recherche sous contraintes, nous pouvons citer les algorithmes IC (Inductive Causation) et son dérivé IC*, CI (Causal Inference), FCI (Fast Causal Inference), PC, et SGS (Sparse Graph Search) (Nguyen, 2012). Ces algorithmes partagent le même principe de déroulement : les recherches d'indépendances conditionnelles sont effectuées grâce à un test pour obtenir un graphe non orienté ; l'orientation du graphe s'effectue ensuite en recherchant les V-structures par le biais de la troisième composante de la règle de la *d-séparation* (Cf. section 2.2.4.2) ; l'orientation est ensuite propagée sur

les autres arêtes, d'abord en utilisant la règle de la *d-séparation* ; puis lorsqu'aucune propagation n'est possible de cette manière et que le graphe obtenu n'est pas encore dirigé complètement, les connaissances des experts viennent compléter la propagation de l'orientation. Le point de départ pour les algorithmes SGS, PC, CI et FCI est un graphe non dirigé complètement connecté, tandis que celui pour les algorithmes IC et IC* est un graphe vide composé de nœuds uniquement. Le déroulement détaillé de l'algorithme PC est donné par le pseudo-code suivant :

Algorithme 1 : Algorithme PC

Construction d'un graphe $\mathcal{G}$ complet non orienté ;
• Recherche d'indépendances conditionnelles :
$i \leftarrow 0$;
pour chaque $\{N_A, N_B\} \in \mathcal{N}^2$ *tel que* $N_A - N_B$ **faire**
  tant que $Card(Adj(\mathcal{G}, N_A) \backslash N_B) > i$ **faire**
    pour chaque $N \in Adj(\mathcal{G}, N_A) \backslash N_B$ *tel que* $Card(N) = i$ **faire**
      si $N_A \perp\!\!\!\perp N_B \mid N$ **alors**
        Supprimer de $\mathcal{G}$ l'arête $N_A - N_B$;
        $SepSet(N_A, N_B) \leftarrow SepSet(N_A, N_B) \cup N$;
        $SepSet(N_B, N_A) \leftarrow SepSet(N_B, N_A) \cup N$;
      fin
    fin
    $i \leftarrow i + 1$
  fin
fin
• Recherche des V-structures :
pour chaque $\{N_A, N_B, N_C\} \in \mathcal{N}^3$ *tel que* $N_A - N_C - N_B$ *et* $\neg\overline{N_A N_B}$ **faire**
  si $N_C \notin SepSet(N_A, N_B)$ **alors**
    Créer une V-structure $N_A \rightarrow N_C \leftarrow N_B$
  fin
fin
• Orientation du reste des arcs [tant que c'est possible] :
pour chaque $\{N_A, N_B\} \in \mathcal{N}^2$ **faire**
  si $N_A - N_B$ *et* $N_A \dashrightarrow N_B$ **alors**
    Ajouter un arc $N_A \rightarrow N_B$
  fin
  si $\neg\overline{N_A N_B}$ **alors**
    pour chaque N_C *tel que* $N_C - N_B$ *et* $N_A \rightarrow N_C$ **faire**
      Ajouter un arc $N_C \rightarrow N_B$
    fin
  fin
fin
Retourner $\mathcal{G}$;

Notations :
• $\mathcal{N}$: l'ensemble des nœuds du graphe $\mathcal{G}$;
• $Adj(\mathcal{G}, N_A)$: l'ensemble des nœuds adjacents au nœud N_A dans le graphe $\mathcal{G}$;
• $Adj(\mathcal{G}, N_A) \backslash N_B$: l'ensemble des nœuds adjacents au nœud N_A privé du nœud N_B, dans le graphe $\mathcal{G}$;

• $N_A \perp\!\!\!\perp N_B \mid N$: N_A est indépendant de N_B conditionnellement à un ensemble N de nœuds. N peut être vide (indépendance), ou bien composé d'un ou plusieurs nœuds (indépendance conditionnelle) ;
• $SepSet(N_A, N_B)$: Ensemble de séparation des nœuds N_A et N_B, qui permettra de détecter, dans la deuxième étape, les *V-structures*. Sont stockés dans cet ensemble tous les nœuds et les ensembles de nœuds rendant N_A et N_B indépendants. Dans la deuxième étape, s'il existe une chaine $N_A - N_C - N_B$, telle que N_C n'appartient pas aux nœuds qui rendent N_A et N_B indépendants ($SepSet(N_A, N_B)$), et que N_A et N_B ne sont pas adjacents, alors la seule conclusion possible est que N_C est un effet commun de N_A et N_B, d'où la création d'une v-structure convergente en N_C dans ce cas là ;
• $N_A - N_B$: N_A et N_B sont reliés par une arrête ;
• $N_A \rightarrow N_B$: N_A et N_B sont reliés par un arc allant de N_A vers N_B ;
• $N_A \dashrightarrow N_B$: Il existe un chemin allant de N_A vers N_B (arc ou ensemble d'arcs reliant N_A et N_B dans la direction de N_B ;
• $\overline{N_A N_B}$: il existe un lien direct entre N_A et N_B ($N_A - N_B$, ou $N_A \rightarrow N_B$, ou $N_A \leftarrow N_B$), c'est à dire que N_A et N_B sont adjacents.

Nous allons à présent discuter de cet algorithme, puisqu'il représente adéquatement la famille des algorithmes de construction de structure de réseaux Bayésiens sous contraintes. Nous faisons également le choix de discuter de cet algorithme parce qu'il s'agit d'une amélioration de l'algorithme SGS tout en en conservant les points forts. En effet, l'algorithme SGS requiert un nombre de tests conditionnels qui augmente de manière exponentielle avec le nombre de nœuds contenus dans le réseau (Spirtes et al., 2000). Ceci est causé par le fait que dans l'algorithme SGS, tous les tests d'indépendances marginales et conditionnelles sont effectués sur l'ensemble des variables, et conditionnellement à l'ensemble des variables, avant de mettre à jour le graphe. Contrairement à l'algorithme SGS, l'algorithme PC effectue les tests d'indépendance d'ordre 0 (marginales), puis met à jour le graphe en supprimant les arêtes entre les nœuds correspondant aux variables indépendantes, il effectue ensuite les tests d'ordre 1 (indépendance de deux variables conditionnellement à une seule variable), où sont testées uniquement les indépendances conditionnelles entre des variables dont les nœuds correspondant sont adjacents et les arêtes entre les nœuds correspondant aux variables indépendantes sont supprimées, puis les tests d'ordre 2 sont effectués de la même façon, suivis par la suppression des arêtes, ainsi de suite. Comme la suppression des arêtes reliant les variables indépendantes est effectuée après chaque test, le nombre de nœud adjacents pour une variable se voit diminuer, ce qui réduit considérablement le nombre de tests à effectuer au fur et à mesure du déroulement de la deuxième étape de l'algorithme PC.

Le premier point que nous abordons, est celui du test d'indépendance, puisqu'il conditionne toute la suite du déroulement de l'algorithme, en particulier la suppression ou non des arêtes, et par conséquent, le graphe non orienté résultant, et donc le graphe orienté final également. Différents tests d'indépendance peuvent être utilisés, selon le type de données traitées. Parmi les tests d'indépendance couramment utilisés pour les réseaux représentant des variables discrètes multinomiales, ou des variables continues ayant été discrétisées, nous pouvons citer le test du χ^2. Le test d'indépendance entre deux variables A et B est donné par (4.1) :

$$\chi^2(A,B) = \sum_{a \in A} \sum_{b \in B} \frac{(n_{ab} - \frac{n_{a.}.n_{.b}}{N})^2}{\frac{n_{a.}.n_{.b}}{N}} \tag{4.1}$$

où a et b sont des modalités que les variables A et B peuvent prendre respectivement, et n_{ab} représente le cardinal des observations où la variable A prend la valeur a, et la variable B prend la valeur b, de manière simultanée. $n_{a.}$ représente le cardinal des observations où la variable A prend la valeur a, et la variable B prend n'importe quelle valeur. De façon similaire, pour $n_{.b}$ le point en indice remplace toutes les valeurs possibles pour la variable A.

Dans le cas de test d'indépendance conditionnelle entre deux variables A et B conditionnellement à une variable C, le test du χ^2 est donné par 4.2 :

$$\chi^2(A,B|C) = \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} \frac{(n_{abc} - \frac{n_{a.c}.n_{.bc}}{n_{..c}})^2}{\frac{n_{a.c}.n_{.bc}}{n_{..c}}} \tag{4.2}$$

où a, b, et c sont des modalités que les variables A, B, et C peuvent prendre respectivement, et n_{abc} représente le cardinal des observations où la variable A prend la valeur a, la variable B prend la valeur b, et la variable C prend la valeur c, de manière simultanée. $n_{a.c}$ représente le cardinal des observations où la variable A prend la valeur a, la variable C prend la valeur c, et la variable B prend n'importe quelle valeur. De façon similaire, pour $n_{.bc}$ et $n_{..c}$, le point en indice remplace toutes les valeurs possibles pour la variable correspondante.

Dans un cas de test d'indépendance marginale par exemple, ce test permet d'accepter ou rejeter l'hypothèse nulle H_0 selon laquelle A et B seraient indépendantes, étant donné un risque α préalablement fixé. α représente le seuil de risque d'erreur de décision (risque d'erreur de type 1 de rejeter à tort l'hypothèse H_0). Dans ce cas, le nombre de degrés de liberté du test serait de $k = (m_A\text{-}1).(m_B\text{-}1)$, où m_A et m_B sont respectivement les nombres de modalités des variables A et B. La suppression ou non d'une arête dépend alors de l'acceptation ou le rejet de l'hypothèse H_0, et cette décision suit la loi suivante (Houde, 2014 ; Vandel, 2012) :

— Si $\chi^2 < \chi^2_{(k;\alpha)}$, alors H_0 est rejetée ;

— Si $\chi^2 \geq \chi^2_{(k;\alpha)}$, alors H_0 est acceptée.

Où $\chi^2_{(k;\alpha)}$ représente le χ^2 critique, ou théorique, de la loi du χ^2 à k degrés de liberté, pour un seuil de significativité α préalablement fixé pour le test. Cette valeur critique est retrouvée grâce à la table de la loi du χ^2 représentée sur la table 4.1 : il s'agit du quantile $1 - \alpha$ de la loi du χ^2 à k degrés de liberté.

Dans le cas du test d'indépendance marginale entre deux variables A et B à trois modalités chacune, le degré de liberté sera de 4. Si, nous choisissons de fixer le seuil α à 0.05, la valeur du χ^2 critique qui conditionnera le test prendra alors la valeur 9.49. Si la valeur du χ^2 observée (calculée à partir de l'égalité (4.1)) est inférieure à 9.49, alors l'hypothèse H_0 est rejetée, et les variables A et B seraient alors dépendantes, selon le test. Nous notons que la table 4.1 est un extrait de la table de la loi du χ^2, et ne représente donc pas les valeurs du χ^2 critiques de manière exhaustive pour tous les degrés de libertés, ni pour tous les seuils de significativité.

TABLE 4.1 – Table de la loi du χ^2

k	**Valeurs critiques χ^2**										
1	0.004	0.02	0.06	0.15	0.46	1.07	1.64	2.71	3.84	6.63	10.83
2	0.10	0.21	0.45	0.71	1.39	2.41	3.22	4.61	5.99	9.21	13.82
3	0.35	0.58	1.01	1.42	2.37	3.66	4.64	6.25	7.81	11.34	16.26
4	0.71	1.06	1.65	2.20	3.36	4.88	5.99	7.78	9.49	13.28	18.47
5	1.14	1.61	2.34	3.00	4.35	6.06	7.29	9.24	11.07	15.09	20.52
6	1.63	2.20	3.07	3.83	5.35	7.23	8.56	10.64	12.59	16.81	22.46
1-α	0.05	0 .1	0.2	0.3	0.5	0.7	0.8	0.9	0.95	0.99	0.999

La raison pour laquelle nous avons détaillé le déroulement de ce test, est d'introduire un premier inconvénient que présenterait un algorithme PC basé sur ce test ou, de manière plus générale, n'importe quel algorithme de construction de structure de réseau Bayésien sous contraintes. En effet, le choix du seuil α est crucial pour ce type d'algorithmes, puisque ce seuil conditionne l'acceptabilité ou non de l'hypothèse d'indépendance, et par conséquent la suppression d'un arc, selon l'algorithme PC (et analogiquement, l'ajout d'une arête pour les algorithmes IC et IC*). Si nous avions fixé le seuil à 0.01, la valeur critique du χ^2 aurait été de 13.28. Si nous admettons, en plus, que la valeur observée du χ^2 est de 10, alors H_0 serait acceptée dans le premier cas (seuil à 0.05), et rejetée dans le deuxième cas (seuil à 0.01). Ceci conduira à la suppression de l'arête reliant les deux variables dans le premier cas, et à sa préservation dans le deuxième cas. La sensibilité des algorithmes de recherche sous contraintes au seuil de significativité est d'autant plus prononcée compte tenu de la répercussion en cascade des erreurs tout au long du déroulement de l'algorithme. En effet, si l'hypothèse d'indépendance (analogiquement, de dépendance) est acceptée à tort, l'erreur ne se limitera pas à la suppression (analogiquement, la préservation) d'une arête à tort, et donc à l'obtention d'un graphe causal avec une arête en moins (analogiquement, en plus) : d'autres arêtes se verront également être supprimées (ou gardées) à tort, puisque les algorithmes de ce type sont également sensibles à l'ordre initial de sélection des variables pour le déroulement des tests d'indépendances conditionnelles et marginales, pouvant être décisif (Colombo & Maathuis, 2014 ; Abellán, Gómez-Olmedo, & Moral, 2006). En d'autres termes, pour les mêmes indépendances marginales et conditionnelles trouvées grâce à des tests, le graphe résultant peut être différent selon l'ordre dans lequel les variables ont été présentées à l'algorithme, ceci est d'autant plus remarqué lorsque le nombre de variable augmente. Cette sensibilité s'explique par le fait que 1) les arêtes sont supprimées directement après le test dans le cas de l'algorithme PC, et 2) les tests d'indépendance conditionnelles sont effectués en conditionnant uniquement sur les arêtes adjacentes du nœud testé, or l'ensemble des arêtes adjacentes dépend de la suppression ou non d'arêtes suite au tests précédents.

Par ailleurs, les erreurs commises sur les tests d'indépendance se répercuteront également sur le reste du réseau, et plus précisément au cours de l'étape d'orientation des arêtes pour obtenir les arcs finaux du graphe. D'abord, l'absence d'une arête à tort pourrait potentiellement empêcher la détection d'une ou plusieurs *V-structures* impliquant

l'arête en question, ce qui par conséquent empêchera l'orientation de la deuxième arête qui devait composer la fourche conjonctive de la *V-structure* lors de la deuxième étape de l'algorithme, ou à l'inverse, détecter une *V-structure* à tort. De façon analogique, le maintien à tort d'une arête inhiberait potentiellement aussi la détection de *V-structure(s)*, ou conduirait à une détection de *V-structure(s)* illusoire(s). L'erreur s'étendra également sur la propagation de l'orientation déroulée lors de la troisième étape de l'algorithme, en empêchant probablement certaines orientations dans le cas de suppression d'arêtes à tort, ou en effectuant des orientations à tort dans le cas de préservation d'arêtes à tort. La répercussion sur cette troisième étape est d'autant plus accentuée puisque les erreurs lors de la deuxième étape (étape de détection des *V-structures*) se répercuteront sur la troisième étape : si une V-structure est détectée à tort, cela conduira forcément à l'orientation à tort des arêtes selon la règle de la deuxième étape, et un arc erroné pourrait éventuellement compléter les conditions de la troisième étape et conduire à une orientation supplémentaire erronée. Par ailleurs, lorsque le graphe comprend un nombre conséquent de variables, il est souvent suggérer de limiter l'incrimination des ensembles de conditionnement des tests, ce qui rajoute une contrainte supplémentaire et une subjectivité lors de la spécification de la taille maximale de l'ensemble de conditionnement des tests d'indépendance.

Le dernier point que nous abordons pour ce type d'algorithmes concerne la manière dont sont exploitées les connaissances humaines, qui demeurent nécessaires en tout état de cause, comme nous avons pu le voir auparavant dans la sous-section 2.2.6.2 du chapitre 2. En effet, les connaissances humaines sont exploitées dans le but de compléter la propagation des orientations uniquement à la fin du déroulement de l'algorithme. Ceci présente d'abord l'inconvénient de restreindre le champs des possibles pour l'expert en charge de compléter le graphe, puisqu'il serait contraint par le PDAG (graphe acyclique partiellement orienté) trouvé par l'algorithme, surtout lorsque celui ci comporte des erreurs. En outre, les seules contributions possibles que l'expert pourrait apporter concerneront l'orientation des arêtes n'ayant pas pu être orientées lors du déroulement de l'algorithme : l'expert ne peut ni ajouter des arcs, ni en supprimer, ni en inverser, faute de quoi les étapes d'orientation des arcs de l'algorithme devraient être redéroulées de façon à ne pas violer les règles de la d-séparation et d'acyclicité. Par ailleurs, et en dépit des points que nous venons de relever quant à l'intégration des connaissances humaines, il est selon nous défavorable de ne pas les intégrer en amont de l'apprentissage, particulièrement lorsque ces connaissances sont notoires ou vérifiées. Quelle que soit la méthode de construction de la structure du graphe, le fait de prendre en compte les connaissances en amont permet de diriger la construction et de restreindre l'espace des graphes possibles.

A l'instar de l'utilisation de tests statistique d'indépendance, comme le test du χ^2 que nous venons de présenter plus haut, l'utilisation de l'information mutuelle pour rechercher les indépendances présente principalement les mêmes inconvénients. L'information mutuelle est une grandeur permettant de quantifier le degré de dépendance entre deux variables, en mesurant la quantité d'information qu'ils partagent (Latham & Roudi, 2009 ; Grandchamp, Alata, Olivier, Khoudeir, & Abadi, 2011). L'information mutuelle entre deux variables A et B, dans le cas discret, est donnée par (4.3) :

$$I(A,B) = \sum_{a \in A} \sum_{b \in B} P(a,b).log(\frac{\mathbb{P}(a,b)}{\mathbb{P}(a).\mathbb{P}(b)}) \tag{4.3}$$

Dans le cas d'évaluation d'indépendance conditionnelle, l'information mutuelle entre

deux variables A et B sachant C est donnée par (4.4) :

$$I(A, B|C) = \sum_{c \in C} \mathbb{P}(c). \sum_{a \in A} \sum_{b \in B} \mathbb{P}(a, b|c). \log \frac{\mathbb{P}(a, b|c)}{\mathbb{P}(a|c).\mathbb{P}(b|c)} \tag{4.4}$$

où $\mathbb{P}(a)$, $\mathbb{P}(b)$, et $\mathbb{P}(c)$ désignent les probabilités que les variables A, B, et C, prennent respectivement les valeurs a, b, et$c$.

Bien que cette grandeur puisse apporter une information sur la dépendance entre deux variables, elle reste à interpréter, puisqu'elle fournit uniquement un degré de dépendance : plus sa valeur est proche de 0, plus il est plausible que les variables étudiées soient indépendantes. Il faut par conséquent fixer un seuil en-dessous duquel le test permettra d'établir que les deux variables sont indépendantes. Ceci génère donc, en cas d'erreur, les mêmes répercussions en cascade des erreurs sur le reste du déroulement de la construction du graphe. D'autres tests d'indépendances peuvent également être adoptés par ce type d'algorithmes, mais leur utilisation partage les mêmes inconvénients dont nous venons de discuter.

Le deuxième type d'algorithmes pouvant être utilisé pour construire la structure causale est celui des algorithmes basés sur le calcul et l'optimisation d'un score. À l'inverse des algorithmes de recherche sous contraintes, ceux basés sur un score sont non déterministes, puisqu'ils peuvent fournir des résultats différents en fonction de la manière dont l'espace de solutions est exploré, et de l'importance que l'on choisi d'attribuer à la complexité de la structure. Parallèlement, pour le même jeu de données et pour un même test d'indépendance avec le même seuil, un algorithme de recherche sous contraintes fournira exactement le même PDAG, ou dans les cas les plus favorables (*i.e.* si l'algorithme parvient à propager les orientations sur toutes les arêtes), il fournira le même DAG. À l'opposé, les algorithmes basés sur l'optimisation d'un score ne garantissent pas l'obtention d'un même résultat, compte tenu de la taille de l'espace de recherche, et du fait que les graphes appartenant à une même classe d'équivalence obtiennent le même score. En outre, de par le nombre de graphes possibles qui augmente de manière super-exponentielle avec le nombre de nœuds, il n'est pas réalisable de calculer les scores de tous les graphes possibles en utilisant une recherche exhaustive (Margaritis, 2003). Par conséquent, la recherche du graphe ayant le score le plus optimale avec ce type d'algorithmes s'effectue de manière heuristique si l'on souhaite explorer tout l'espace de recherche. En tout état de cause, un graphe obtenu par apprentissage est toujours représentatif de sa classe d'équivalence, et tous les graphes appartenant à cette classe d'équivalence méritent d'être analysés puisqu'ils sont tous équivalents d'un point de vue probabiliste, mais un seul parmi eux encode les relations causales, et ce quelle que soit la méthode d'apprentissage adoptée. Les algorithmes basés sur la maximisation de score sont les plus utilisés pour apprendre les structures de réseaux Bayésiens (Scanagatta, Salmerón, & Stella, 2019), et leur objectif est donc de trouver le graphe G^* tel que :

$$G^* = \arg\max_{G \in \mathcal{G}} S_{G,\mathcal{D}} \tag{4.5}$$

Où $S_{G,\mathcal{D}}$ désigne le score global d'un graphe G parmi l'ensemble $\mathcal{G}$ des graphes candidats, et $\mathcal{D}$ la base de données étudiée. Les score utilisés pour les algorithmes de ce type dénotent la vraisemblance décrivant la plausibilité que la structure trouvée ait été

effectivement générée à partir du jeu de données étudié. Ceci a l'avantage de ne pas devoir spécifier un seuil déterministe acceptant ou réfutant catégoriquement un lien entre deux variables, mais de garder les liens d'une structure jusqu'à ce qu'une autre structure obtienne un score plus optimal. Un autre avantage qui rend ce type d'algorithmes intéressant réside dans le fait que plusieurs structures fournissant de bons résultats peuvent être combinées (Leray, 2006). Par ailleurs, la boucle modification du réseau / évaluation de la performance globale du réseau, s'avère plus intéressante que le trait local qui caractérise les tests d'indépendances, qui n'offrent pas la possibilité de retour en arrière pour remodifier ce qui a déjà été testé. Au vu des nombreuses combinaisons possibles pour un nombre donné de variables, la détermination et l'évaluation des scores de chaque structure candidate deviennent rapidement coûteuses. Ainsi, les scores les plus répandus pour ce type d'algorithmes sont décomposables localement (Scanagatta et al., 2019). Ils s'expriment donc en fonction des scores locaux attribués aux sous-graphes des structures candidates à évaluer. Outre leur caractère décomposable, ces scores sont dits équivalents, car ils ont la particularité d'être exactement égaux pour deux graphes appartenant à la même classe d'équivalence.

Dans la suite, nous opterons pour un algorithme basé sur le calcul et l'optimisation de score. Nous justifions cela par l'ensemble des inconvénients liés aux algorithmes de recherche sous contraintes, que nous avons préalablement cités, et que nous pouvons désigner comme étant un manque de robustesse : une proportion restreinte d'erreurs en entrée engendre une proportion élevée d'erreurs en sortie (*i.e.* une erreur dans l'acceptation de l'hypothèse nulle du test engendrera plusieurs erreurs liées à des absences / maintiens erronés d'arêtes, et des arcs orientés de manière fallacieuse).

4.2.2 Quelques scores usuels

Parmi les scores les plus communément mis en œuvre pour l'apprentissage de la structure de réseau Bayésien basé sur un score, nous pouvons citer le critère d'information d'Akaike AIC (Akaike Information Criterion) (Akaike, 1974), qui se base sur les concepts de la théorie de l'information. Ce score quantifie la qualité d'un modèle par rapport à un jeu de données, et comme pour la majorité des scores utilisés pour l'apprentissage de structure, il respecte le principe du rasoir d'Ockham en favorisant les modèles les plus simples pour garantir la parcimonie (Nguyen, 2012). Dans le cas des réseaux Bayésiens, le critère AIC s'écrit sous la forme donnée par (4.6) (Bouckaert, 2008 ; Prestat, 2010), et comprend donc un paramètre servant à pénaliser, dans le cas des réseaux Bayésiens, les structures complexes :

$$AIC_{G,\mathcal{D}} = \sum_{i=1}^{n}\sum_{j=1}^{p_i}\sum_{k=1}^{m_i} N_{ijk}\log\frac{N_{ijk}}{N_{ij}} - \sum_{i=1}^{n}(m_i-1).p_i \tag{4.6}$$

Le premier terme dans l'égalité 4.6 est relatif à l'entropie de la structure et représente une approximation de la log-vraisemblance du graphe G par rapport aux données $\mathcal{D}$ ($\log \mathbb{P}(\mathcal{D}|\mathcal{G})$). Il sert donc à décrire sa qualité, tandis que le deuxième terme est celui de la pénalisation, qui est positivement corrélé avec la complexité de la structure, et dont le calcul est basé sur le nombre de paramètres de la structure. n représente le nombre de variables étudiées, m_i dénote le nombre de modalités de la variable x_i, et p_i le nombre de

combinaisons des modalités des parents de la variable x_i ($i.e$ $p_i = \prod_{x_j \in pa(x_i)} m_j$). Lorsqu'une variable ne possède pas de parent, $p_i = 1$. N correspond au nombre d'observations dans le jeu de données $\mathcal{D}$, et N_{ijk} correspond au cardinal des observations pour lesquelles la variable x_i prend sa k^{eme} valeur, et les parents de x_i prennent les valeurs correspondant à la j^{eme} combinaison. Ainsi, $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$.

Un autre score fréquemment adopté est le critère d'information Bayésien BIC (Bayesian Information Criterion), qui est équivalent au MDL (Minimum Description Length), et qui tire son origine du score AIC. Il s'en distingue par la manière dont la complexité du graphe pénalise le score (Prestat, 2010) : le terme de pénalisation ne dépend pas uniquement de la complexité du graphe (*i.e.* le nombre de modalités variables et le nombre de combinaisons des modalités de leurs parents), mais également du nombre d'observations dans le jeu de données. Le score BIC s'exprime sous la forme suivante (4.7) :

$$BIC_{G,\mathcal{D}} = \sum_{i=1}^{n} \sum_{j=1}^{p_i} \sum_{k=1}^{m_i} N_{ijk} \log \frac{N_{ijk}}{N_{ij}} - \frac{1}{2} \log N \sum_{i=1}^{n} (m_i - 1).p_i \tag{4.7}$$

Nous pouvons remarquer que le score BIC pénalise plus sévèrement la complexité du graphe que le score AIC. Cette pénalisation reste toutefois moins pondérée que l'adéquation du graphe aux données exprimée par le premier terme, puisqu'elle dépend du nombre d'observations dans le jeu de données de manière logarithmique, tandis que l'adéquation du graphe aux données dépend du nombre d'observations de façon linéaire. Dans notre contexte, ce n'est pas tant la réduction de la complexité du modèle qui nous intéresse dans le terme de pénalisation, puisque nous favorisons l'obtention d'un modèle décrivant au mieux les relations causales qui existent entre les données, même dans les cas où elles s'avèrent nombreuses ; mais c'est plutôt le fait que ce terme de pénalisation est indispensable, dans le sens où aucun ajout d'arc ne peut diminuer la vraisemblance : il ne faut donc en ajouter que s'il participe considérablement à l'augmentation de la vraisemblance du graphe, afin d'éviter le sur-apprentissage. Autrement, le risque serait d'obtenir un graphe contenant certainement des arcs correctement ajoutés, mais également des arcs ajoutés de manière abusive.

Outre les scores basés sur les concepts de la théorie de l'information, il existe des scores dits Bayésiens comme le score BD (Bayesian Dirichlet) et le score BDe (Bayesian Dirichlet Equivalent), qui n'est autre qu'une amélioration du score BD qui s'exprime suivant la formule (4.8) (Cooper & Herskovits, 1992 ; François, 2006) :

$$\mathrm{BD}(G|\mathcal{D}) = \prod_{i=1}^{n} bd(x_i, Pa(x_i)|\mathcal{D}) \tag{4.8}$$

Où bd est le score local relatif à la variable x_i, dont le calcul peut être effectué selon la formule (4.9) :

$$bd(x_i, Pa(X_i)|\mathcal{D}) = \prod_{j=1}^{p_i} \frac{(m_i - 1)\,!}{(N_{ij} + m_i - 1)\,!} \prod_{k=1}^{m_i} N_{ijk}\,! \tag{4.9}$$

Le score BD est décomposable mais n'est pas équivalent (deux structures appartenant à la même classe d'équivalence n'obtiennent pas le même score BD). C'est pourquoi une amélioration de ce score a été proposée. Il s'agit du score BDe (Heckerman, Geiger, &

Chickering, 1995), défini par la formule (4.10) où $\mathbb{P}(\mathcal{G})$ représente l'information à priori sur la structure.

$$\mathrm{BDe}_{G,\mathcal{D}} = \mathbb{P}(\mathcal{G}) \prod_{i=1}^{n} \prod_{j=1}^{p_i} \frac{(m_i - 1)\,!}{(\frac{1}{q_i} + m_i - 1)\,!} \prod_{k=1}^{m_i} \frac{1}{m_i q_i}\,! \tag{4.10}$$

Bien que la façon de calculer le score BIC ne permet pas la prise en compte des connaissances à priori comme c'est le cas pour le score BDe, plusieurs études comparatives ont établi que l'utilisation du score BIC fait preuve de bonnes performances pour la recherche gloutonne (ChongYong & HongChoon, 2017 ; Liu, Malone, & Yuan, 2012). En outre, dans notre proposition, et comme nous l'expliquerons dans la sous-section suivante, nous prenons en compte les connaissances à priori sous forme d'un ensemble de contraintes sur la structure du graphe, que la recherche doit respecter du début jusqu'à la fin, et qui seront par conséquent prises en compte dans la structure finale. Dans la suite, nous choisissons donc d'évaluer les structures candidates en utilisant le score BIC.

4.2.3 Choix de l'algorithme d'apprentissage

Il existe plusieurs algorithmes basés sur le calcul et l'optimisation d'un score, qui peuvent être triés en deux catégories : ceux qui font une recherche gloutonne dans l'espace des DAGs, et ceux qui l'effectuent dans un sous espace, comme l'espace des arbres, ou en ordonnant les nœuds à priori. En raison de la complexité des relations entre les différentes entités dans un contexte industriel, et étant donné notre objectif de représentation des liens de causalité qui ne respectent pas forcément une structure non orientée arborescente, nous excluons les algorithmes de recherches dans l'espace des arbres (comme l'algorithme de l'arbre de recouvrement maximal MWST). Nous excluons également l'algorithme K2, puisqu'il est sensible, voire conditionné par l'ordonnancement des nœuds. Il en va de même pour l'algorithme K3, qui se distingue principalement de l'algorithme K2 par le score utilisé. En effet, l'ordonnancement nécessite de classer **tous** les nœuds, et ce classement conditionnera ensuite la possibilité d'ajout d'arc. À titre d'exemple, si nous considérons que nous recherchons la structure représentant au mieux les relations entre cinq variable $\{X_1, ..., X_5\}$, un ordonnancement O_1 tel que $O_1 = \{X_3, X_2, X_5, X_1, X_4\}$ signifie qu'il ne serait pas possible d'ajouter un arc $X_2 \rightarrow X_3$ ou un arc $X_4 \rightarrow X_2$, etc. De manière plus générale, si X_i est vient avant X_j dans l'ordonnancement, il n'est pas possible d'ajouter à la structure l'arc $X_j \rightarrow X_i$. Un autre ordonnancement O_2 tel que $O_2 \neq O_1$ donnera à *fortiori* un résultat différent. Or, dans la pratique, et dans un contexte similaire à celui qui nous intéresse, il est difficile, voire inenvisageable de connaître l'ordre de toutes les variables étudiées (*i.e.* le sens de la causalité entre toutes les variables est souvent inconnu). Nous écartons également l'algorithme SEM (EM structurel), qui est plus adapté dans les contextes où les données sont incomplètes (Francois & Leray, 2004), ce qui n'est pas le cas dans notre contexte, étant donné nos hypothèses de travail.

Nous nous orientons donc vers une recherche gloutonne, qui a montré son efficacité dans de nombreuses applications (ChongYong & HongChoon, 2017 ; Tsamardinos, Brown, & Aliferis, 2006). Avant de détailler l'algorithme que nous utiliserons, et les améliorations que nous y apporterons, nous allons à présent introduire la manière dont nous allons prendre en compte les connaissances des experts, de manière à ce que nous puissions en tirer profit pour orienter la recherche gloutonne et ainsi réduire son espace.

4.2.4 Prise en compte des connaissances à *priori* dans notre proposition

Comme nous l'avons expliqué précédemment, les connaissances expertes sont indispensables pour orienter un algorithme d'apprentissage de la structure d'un réseau Bayésien vers la structure qui décrit au mieux les relations causales, et pas seulement qui reflète les dépendances et indépendances d'un point de vue statistique. Par ailleurs, nous avons opté pour une recherche gloutonne, dont l'inconvénient principal est la convergence vers un optimum local, étant donné l'ampleur de l'espace de recherche. La prise en compte des connaissances à priori avant le début de la recherche gloutonne nous permettra alors de réduire cet espace. Nous proposons alors trois paramètres permettant de contraindre la recherche gloutonne, en prenant en compte, de manière réalisable, les connaissances des experts :

— **Les variables exogènes**

Nous définissons les variables exogènes dans notre contexte comme étant des variables ne pouvant pas être une conséquence d'une variable (cause) observée quelle qu'elle soit, étant donné le contexte étudié. Par exemple, si parmi les variables traitées, nous disposons de la variable "formation de l'opérateur", dans laquelle est renseignée, la formation scolaire ou universitaire dont l'opérateur est issu. Cette variable étant intrinsèquement inscrite dans le passé, et par conséquent impossible à modifier, il n'est pas possible qu'elle puisse être une conséquence dans le contexte étudié. Elle est donc exogène dans le contexte étudié. Quelques variables peuvent être exogènes quel que soit le contexte étudié. Par exemple, si parmi les variables traitées, nous disposons de la variable "jour de la semaine", dans laquelle est renseigné le jour de la semaine (du Lundi au Dimanche), auquel nous nous intéressons pour voir, à titre d'exemple, s'il a un effet sur la cadence, ou encore sur le nombre de commandes passées. Cette variable "jour de la semaine" est une variable exogène facilement reconnaissable, puisqu'elle l'est dans tous les contextes. Nous définissons ainsi un ensemble V_e, qui contiendra les variables exogènes désignées par l'utilisateur, et pouvant également être vide si l'utilisateur juge qu'il n'en existe pas, ou qu'il n'a pas les connaissances nécessaires. Si $x_i \in V_e$, alors il est impossible d'avoir un arc $x_j \rightarrow x_i$, quelle que soit la variable observée x_j. La recherche ne parcourra donc pas les graphes contenant un arc $x_j \rightarrow x_i$, pour tout $x_i \in V_e$;

— **Les liens causaux déjà connus :**

Un deuxième ensemble C peut être renseigné en y ajoutant des couples ordonnés de variables correspondant aux liens de causalités connus par les experts. Si l'ensemble C contient un couple (x_i, x_j), ceci contraindra l'algorithme de recherche à mettre l'arc $x_i \rightarrow x_j$ dans le graphe de départ, et à le garder durant toute la recherche (*i.e.* en faisant en sorte qu'il soit présent dans tous les graphes parcourus). Autrement dit, pour tout $\mathcal{G} \in \mathbb{B}$, si $(x_i, x_j) \in C$ et $x_i \rightarrow x_j \not\subseteq \mathcal{G}$, alors $\mathcal{G}$ ne sera pas parcouru. Comme pour l'ensemble V_e des variables exogènes, l'ensemble C peut rester vide si aucun lien causal n'est connu à priori ;

— **Les ordonnancements :**

L'idée ici est semblable à celle suggérée par les algorithmes K2 et K3 concernant l'ordonnancement des nœuds. Elle reste toutefois moins contraignante, dans le sens où l'ordonnancement de toutes les variables n'est pas requis. Il s'agit d'un ensemble O, composé de couples ordonnées, représentant les ordres entre variables déjà connus. Si O contient un couple (x_i, x_j), alors il ne peut y avoir un arc $(x_i \leftarrow x_j)$, ni de chemin dirigé $(x_i \dashleftarrow x_j)$. Une fois de plus, cet ensemble peut être vide, et s'il ne l'est pas, alors la recherche ne parcourt pas les graphes contenant un arc $(x_i \leftarrow x_j)$ ou un chemin dirigé $(x_i \dashleftarrow x_j)$, pour tout $(x_i, x_j) \in O$.

Nous considérons ces trois ensembles dans l'algorithme que nous implémentons pour l'apprentissage de la structure, comme étant des contraintes à définir par l'utilisateur, et par conséquent comme étant des paramètres en entrée de l'algorithme. Bien entendu, il ne faut pas que les variables et/ou les couples renseignés dans ces trois ensembles contiennent des informations paradoxales. Considérons que le jeu de données traité soit composé de quatre variables A, B, C, et D. Si $A \in V_e$, et $(B, A) \in C$, ces deux affirmations sont contradictoires, puisque $A \in V_e$ veut dire que A ne peut pas être l'extrémité d'un quelconque arc. Nous prenons donc en compte la vérification de la cohérence des contenus de ces trois ensembles dans l'implémentation de l'algorithme d'apprentissage, que nous détaillerons dans la suite. Cette vérification s'effectue par le biais de la règle suivante, qui se décompose en quatre axiomes :

Considérons le jeu de données $\mathcal{D} = \{x_1, ..., x_n\}$:

— Si $x_i \in V_e \Rightarrow \forall (x_j, x_k) \in C \;\; x_i \neq x_k$

— $\forall (x_i, x_j) \in C : x_j \notin V_e$

— $\forall (x_i, x_j) \in C, \forall (x_k, x_l) \in O$: si $x_i = x_l \Rightarrow x_j \neq x_k$

— $\forall (x_i, x_j) \in C, \forall (x_k, x_l) \in O$: si $x_j = x_k \Rightarrow x_i \neq x_l$

La validation de la cohérence des paramètres passés à l'algorithme requiert la vérification de ces quatre axiomes à la fois. Si un axiome est violé, la recherche ne peut pas démarrer et l'expert devra alors reconsidérer ses connaissances en fonction de l'erreur qui lui sera indiquée (*i.e.* en fonction de l'axiome ou les axiomes ayant été violés).

Une fois la cohérence validée, la recherche peut prendre deux points de départ différents : si $C \neq \emptyset$, le point de départ sera le sous-graphe contenant les arcs qui correspondent aux couples ordonnés appartenant à C ; si $C = \emptyset$, le point de départ sera un graphe vide (*i.e.* un graphe avec uniquement des sommets). Ensuite, quel que soit le point de départ de la recherche, cette dernière sera contrainte par les contenus des ensembles V_e et O, et C. Les connaissances *à priori* sont donc prises en compte de cette manière (*i.e.* sous forme de contraintes de recherche), d'une part parce qu'elle sont, le plus souvent, indispensables, et d'autre part afin de restreindre l'espace de la recherche gloutonne. En effet, lorsqu'aucune restriction n'est donnée pour la définition des graphes, le nombre de graphes voisins à calculer à chaque itération pour un graphe G à n variables est d'ordre $O(n(n-1))$ (Vandel, 2012). Par ailleurs, la manière dont nous demandons les connaissances à priori laisse place à la prise en compte de connaissances partielles, pouvant être faciles d'accès et de vérification, mais bénéfiques pour l'orientation de la recherche).

4.2.5 Algorithme d'apprentissage de la structure basé sur l'optimisation d'un score : adaptation du Hill Climbing

4.2.5.1 Algorithme du Hill Climbing classique

Dans les sous-sections 4.2.1 et 4.2.3, nous avons introduit et discuté des types les algorithmes d'apprentissage de structure de réseau Bayésien. Nous en avons conclu les algorithmes basés sur le calcul et l'optimisation d'un score s'avèrent être plus adaptés à notre contexte. Nous nous sommes également orientés vers la recherche gloutonne, guidée par les connaissances à *priori* des experts, pour parcourir les graphes et chercher le score optimal. L'algorithme de recherche gloutonne le plus utilisé dans le cadre de l'apprentissage de structure de réseau Bayésien est le *Hill Climbing (HC)* (Friedman, Nachman, & Pe'er, 2013), également appelé *Greedy Search (GS)* dans ce contexte, et ce en raison de son bon compromis entre les capacités de calcul qu'il mobilise et la qualité des modèles appris. Dans la suite, nous utiliserons l'appellation Hill Climbing pour s'y référer, pour éviter que la traduction de *Greedy Search* (recherche gloutonne) n'induise le lecteur en confusion avec les méthodes de recherche gloutonne dans leur globalité.

Comme son nom l'indique, le Hill Climbing est un algorithme d'escalade qui cherche à améliorer une fonction objective, en partant d'une structure donnée, et en évaluant le voisinage de la structure de l'itération en cours. Le voisinage est déterminé selon des opérations bien précises, et en l'occurrence ici, trois types d'opérations sont nécessaires pour définir le voisinage d'une structure donnée : l'ajout, la suppression, et l'inversion d'un arc à la fois. Chaque opération effectuée débouche en une nouvelle structure dont le score est calculé et stocké. Une fois toutes les opérations effectuées, la structure correspondante à l'opération ayant maximisé le score devient alors la structure de l'itération suivante, pour laquelle les scores des voisins vont être calculés, ainsi de suite. Si aucune opération n'augmente le score de l'itération en cours, la recherche s'arrête, et la structure de l'itération en cours devient ainsi la structure finale optimale apprise par l'algorithme. Il est à noter que pour chaque structure, le voisinage est souvent composé de bien plus de trois structures voisines. En effet, pour chaque type d'opérations (ajout / suppression / inversion), un arc est manipulé à la fois : par exemple, pour les opérations de type inversion, un arc de la structure de l'itération en cours est inversé à la fois et cette inversion singulière donne lieu à une nouvelle structure devant être évaluée (si l'inversion ne provoque pas de cycle) ; pour une structure donnée, et en supposant qu'aucune inversion ne provoquera de structure contenant un cycle, il y aura autant d'opérations de type inversion que d'arcs présents dans la structure de l'itération en cours. Ces détails sont expliqués à travers l'exemple illustré sur la figure 4.1. La fonction objective à optimiser est la fonction de score.

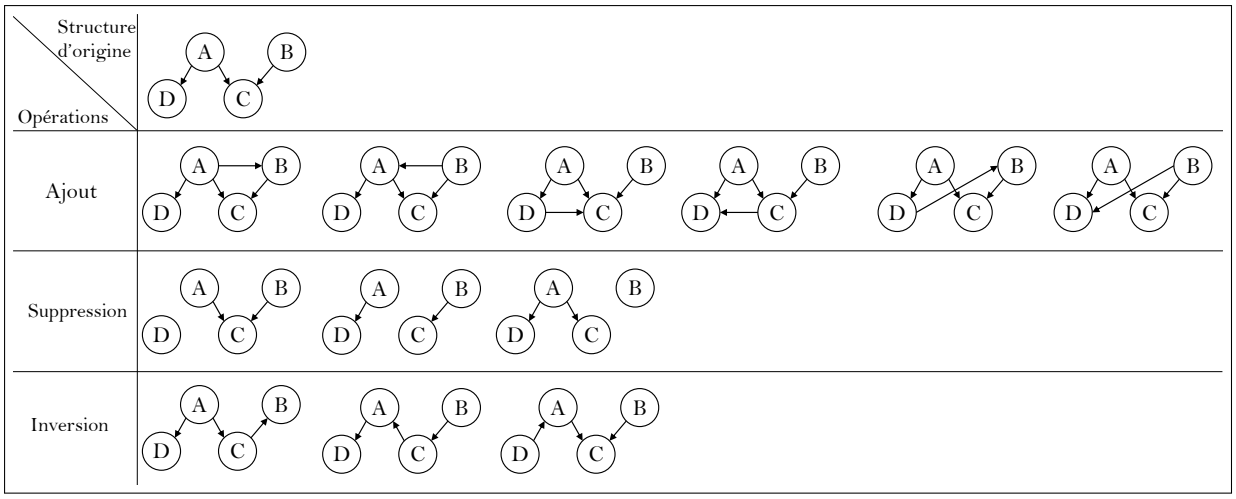


FIGURE 4.1 – Structures à évaluer, issues des différentes opérations possibles effectuées sur la structure d'origine.

Le déroulement de l'algorithme Hill Climbing original, qui est semblable à la descente du gradient mais dans le sens inverse et dans des contextes discrets, est décrit par l'algorithme 2. Le graphe de départ y est renseigné par l'utilisateur, ou généré aléatoirement à partir du graphe vide composé des sommets qui correspondent aux variables du jeu de données $\mathcal{D}$. Le graphe est mis à jour à chaque fois qu'un voisin obtienne un score meilleur, ce voisin même remplace le graphe, jusqu'à ce qu'à ce qu'il n'y ait plus de voisin qui augmente le score. La modification d'un arc à la fois à chaque étape justifie l'intérêt de l'utilisation d'un score décomposable, pour ne pas avoir à recalculer le score du graphe dans son ensemble.

Algorithme 2 : Algorithme Hill Climbing

- Soit $\mathcal{D} = X_1, ..., X_n$ le jeu de données d'intérêt ;
- Soit $\mathcal{G}$ un graphe de départ ;
- Soit $\mathcal{G}^*$ le graphe solution ;
- Soit $L = [\,]$ la liste destinée à stocker les structures voisines ;
- Soit $V(\mathcal{G})$ étant la fonction générant tous les DAGs $\mathcal{G}_i$ voisins de $\mathcal{G}$ par ajout, suppression, ou insertion d'un arc;

$\mathcal{G}^* \leftarrow \mathcal{G}$;
$score \leftarrow BIC_{\mathcal{G},D}$;
$score_{max} \leftarrow score$;
$ameliorer \leftarrow Vrai$;
tant que $ameliorer = Vrai$ **faire**
  $ameliorer \leftarrow Faux$;
  $L \leftarrow V(\mathcal{G}^*)$;
  pour chaque $\mathcal{G}_i \in L$ **faire**
    $score \leftarrow BIC_{\mathcal{G}_i,D}$;
    si $score > score_{max}$ **alors**
      $score_{max} \leftarrow score$;
      $\mathcal{G}^* \leftarrow \mathcal{G}_i$;
      $ameliorer \leftarrow Vrai$
    fin
  fin
  $L \leftarrow [\,]$
fin
Retourner $\mathcal{G}^*$

L'inconvénient majeur de cet algorithme est qu'il trouve souvent un optimum local, soit en raison de présence de plateaux, ou en raison de présence de voisins indirects augmentant le score (*i.e.* séparés de l'optimum local par des graphes diminuant le score). Par ailleurs, et comme son nom l'indique, cet algorithme n'évolue que dans une seule direction. C'est à dire que si un optimum est atteint par le graphe $\mathcal{G}_i^*$ dans l'itération i, il n'y a pas de retour en arrière : on ne peut donc pas revenir au graphe optimal précédent $\mathcal{G}_{i-1}^*$ de l'itération $i-1$ pour choisir un autre graphe voisin $\mathcal{G}_i$ dans l'itération i tel que $\mathcal{G}_i \neq \mathcal{G}_i^*$, qui augmente le score de $\mathcal{G}_{i-1}$ moins que le graphe $\mathcal{G}_i^*$, mais dont le meilleur voisin obtiendrait probablement un meilleur score à l'itération $i+1$ que le meilleur voisin du graphe $\mathcal{G}_i^*$. Autrement dit, soit $\mathcal{G}_{i-1}^*$ le meilleur graphe de l'itération $i-1$, soient $\mathcal{G}_i$ et $\mathcal{G}_i^*$ deux voisins de $\mathcal{G}_{i-1}^*$ tels que $BIC_{\mathcal{G}_{i-1}^*,\mathcal{D}} < BIC_{\mathcal{G}_i,\mathcal{D}} < BIC_{\mathcal{G}_i^*,\mathcal{D}}$, et soient $\mathcal{G}_{i+1}^*$ le meilleur voisin de $\mathcal{G}_i^*$ et $\mathcal{G}_{i+1}$ le meilleur voisin de $\mathcal{G}_i$ tels que $BIC_{\mathcal{G}_{i+1}^*,\mathcal{D}} < BIC_{\mathcal{G}_{i+1},\mathcal{D}}$ et $BIC_{\mathcal{G}_i^*,\mathcal{D}} < BIC_{\mathcal{G}_{i+1}^*,\mathcal{D}}$ et $BIC_{\mathcal{G}_i,\mathcal{D}} < BIC_{\mathcal{G}_{i+1},\mathcal{D}}$; à l'itération $i-1$, l'algorithme Hill Climbing choisira le graphe $\mathcal{G}_i^*$ pour en évaluer les voisins à l'itération i, puisque c'est celui qui maximise le score parmi tous les voisins de $\mathcal{G}_{i-1}^*$; à l'itération i, l'algorithme choisira le graphe $\mathcal{G}_{i+1}^*$, puisque c'est celui qui maximise le score parmi tous les voisins du graphe G_i^*. De cette manière, et puisque les voisins de $\mathcal{G}_i$ n'ont pas été évalués, l'algorithme ne trouvera pas la solution $\mathcal{G}_{i+1}$ qui maximise le score plus que la solution $\mathcal{G}_{i+1}^*$.

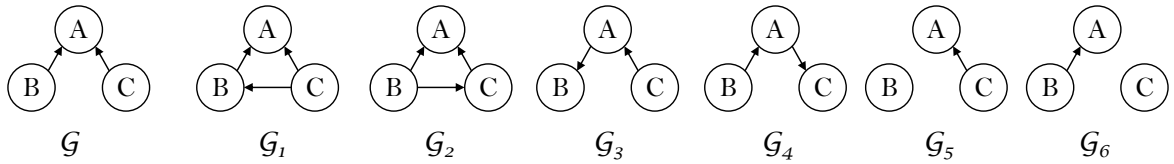
La solution la plus courante pour remédier au problème de convergence vers un optimum local est de relancer l'algorithme plusieurs fois en démarrant avec des graphes diffé-

rents générés aléatoirement, et de sélectionner à la fin des itérations, celui qui maximise le score. Cette solution est implémentée par l'algorithme IHC (Iterated Hill Climbing). Une autre solution est d'effectuer une recherche gloutonne tabou, en tolérant la diminution du score lorsque l'optimum est atteint. Cette démarche exige de stocker comme Tabou les dernières opérations effectuées, de manière à ne pas les refaire − puisqu'elles déjà ont été effectuées avant d'atteindre l'optimum auquel l'algorithme est arrivé − et par conséquent évaluer les voisinages des graphes issus d'autres opérations diminuant le score. Le nombre des dernières opérations Tabou à ne pas refaire doit être fixé et représente donc un paramètre décisif. L'augmentation du nombre d'opérations effectuées à la fois peut également être envisagée, mais la taille de voisinage augmente de manière exponentielle, étant donné le nombre de combinaisons possibles des opérations à effectuer à la fois.

L'adaptation que nous proposons pour l'algorithme Hill Climbing est inspirée de l'ensemble de ces solutions. Nous détaillons ci-dessous les limites liées au HC, avant de suggérer des modifications pour remédier à chaque inconvénient identifié. Afin de simplifier les explications, nous considérons que $\mathcal{D}$ contient trois variables A, B, et C.

— **Problème de scores égaux**

Considérons le graphe $\mathcal{G}$ suivant, et les structures composant son voisinage.



Admettons que le graphe $\mathcal{G}_3$ maximise le score. Ceci veut dire que le graphe $\mathcal{G}_4$ maximise également le score, puisqu'il appartient à la même classe d'équivalence que $\mathcal{G}_3$ et obtient donc le même score lorsqu'on utilise un score équivalent tel que le BIC. D'après le pseudo-code de l'algorithme Hill Climbing, le graphe qui sera retenu pour en évaluer le voisinage dans la prochaine itération est le graphe $\mathcal{G}_3$. Ainsi, le voisinage du graphe $\mathcal{G}_4$ ne sera jamais évalué. En utilisant le Hill Climbing avec une liste Tabou, si le voisinage de $\mathcal{G}_3$ permet d'augmenter le score, la recherche irait dans cette direction là, puisque la liste Tabou ne serait pas utilisée car le score augmente encore et il n'y a donc pas besoin de dégrader le score. Les voisinages de $\mathcal{G}_4$ ne seraient donc pas explorés, et par conséquent pas exploités. En autorisant une égalité dans la condition de mise à jour du score et du graphe maximal, ce sera le graphe $\mathcal{G}_4$ qui sera retenu, mais le même problème persistera cette fois à cause de la non-exploitation du graphe $\mathcal{G}_3$. L'autorisation d'égalité, bien qu'elle ne résout pas le problème que nous venons de décrire, permet tout de même d'aller au delà des plateaux, sans pour autant résoudre le problème de plateaux ;

— **Problème de plateaux**

Si nous admettons cette fois que les graphes $\mathcal{G}_1$ et $\mathcal{G}_2$ obtiennent le même score que $\mathcal{G}$, et que tous les autres graphes diminuent le score. Si l'on considère que la condition de mise à jour du score et du graphe dans le Hill Climbing autorise de continuer

en cas d'égalité, cela permettra à l'algorithme de poursuivre son exploration avec le graphe $\mathcal{G}_2$. Cependant, le problème précédent d'égalité de score persistera à cause de la non exploitation du graphe $\mathcal{G}_1$;

— **Contenu et utilisation de la liste Tabou**

Dans le cadre de la recherche de structure de réseau Bayésien, l'utilisation d'une liste Tabou consiste à stocker les dernières opérations effectuées (*i.e.* les derniers mouvements effectués). Cette liste est ensuite utilisée lorsque le score arrête d'augmenter : l'algorithme autorise une baisse de score, et explore quand même le voisinage, mais à condition que celui-ci soit généré sans refaire une opération qui appartient à la liste Tabou. Considérons l'exemple illustré sur la figure 4.2, qui décrit une une recherche effectuée par le Hill Climbing à partir d'un graphe vide, et admettons que le graphe maximisant le score auquel devrait aboutir la recherche est le graphe $\mathcal{G}^*$.

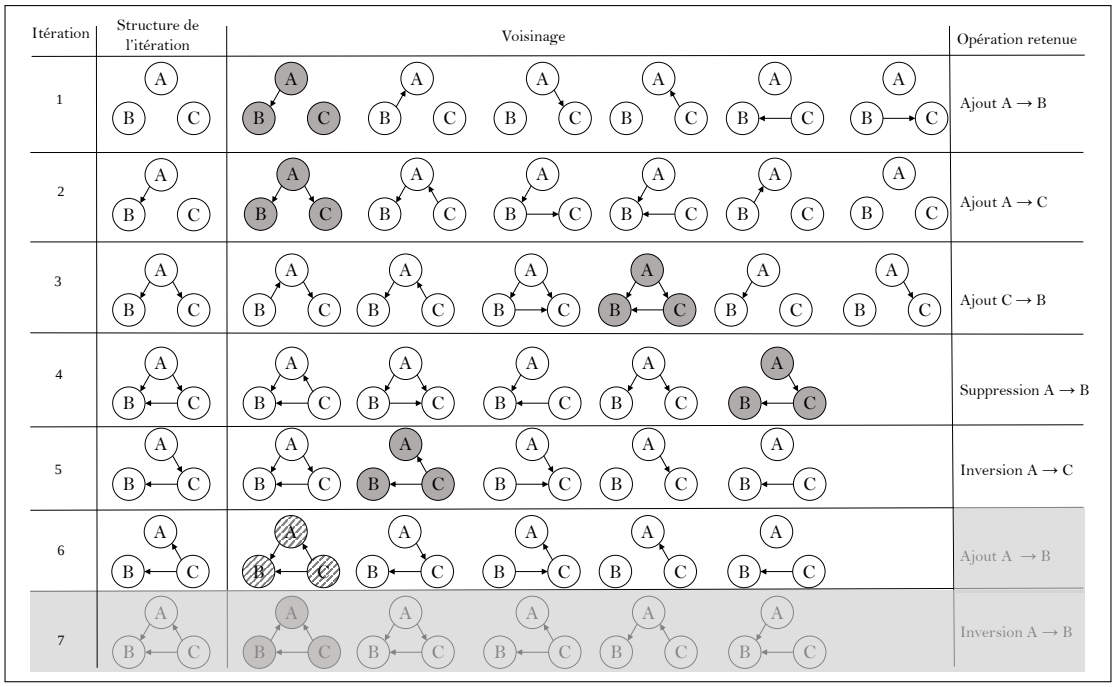


FIGURE 4.2 – Exemple décrivant le déroulement du Hill Climbing et mettant en avant l'inconvénient de stocker les mouvements en liste Tabou.

Sur cette figure, les graphes avec un fond gris uni représentent ceux qui maximisent le score à chaque itération et qui sont donc retenus pour l'itération suivante. Les parties grisées dans la dernière et l'avant dernière itération représentent les étapes que l'algorithme ne pourra pas effectuer en utilisant une liste Tabou contenant les derniers mouvements (opérations) effectués. Dans cet exemple, nous supposons que le graphe de départ est vide, et que nous stockons dans la liste Tabou un maximum de 10 opérations les plus récentes. Jusqu'à la cinquième itération, la liste Tabou contiendra les mouvements suivants : l'ajout de l'arc $A \rightarrow B$, l'ajout de l'arc $A \rightarrow C$, l'ajout de l'arc $C \rightarrow B$, la suppression de l'arc $A \rightarrow B$, et l'inversion de l'arc $A \rightarrow C$. Comme le montre la figure, nous admettons dans cet exemple que le score continue à augmenter jusqu'à la cinquième itération, où le graphe composé des arcs $C \rightarrow A$ et $C \rightarrow B$ est celui qui maximise le score et qui est donc retenu pour

en évaluer le voisinage à la sixième itération. À la sixième itération, aucun graphe appartenant au voisinage de la structure en cours n'augmente le score. Le principe de l'utilisation d'une liste Tabou est de permettre de dégrader le score et continuer l'exploration pour éviter les optimums locaux. L'utilisation d'une liste autorisant la dégradation du score mais interdisant les derniers mouvements effectués empêchera l'algorithme de sélectionner le premier graphe du voisinage de cette itération, composé des arcs $A \rightarrow B$, $C \rightarrow A$ et $C \rightarrow B$ (le graphe au fond rayé de la sixième itération). En effet, afin d'obtenir ce graphe il faut ajouter l'arc $A \rightarrow B$ au graphe retenu dans l'itération précédente. Or, le mouvement d'ajout de l'arc $A \rightarrow B$ fait partie de la liste Tabou, l'algorithme n'autorisera pas ce graphe et explorera donc un autre graphe qui dégrade le score. Bien que le graphe composé des arcs $A \rightarrow B$, $C \rightarrow A$ et $C \rightarrow B$ n'ait jamais été exploré dans les itérations précédentes, et malgré le fait qu'il soit celui qui minimise le score dans la sixième itération parmi les autres membres du voisinage, celui ci ne sera pas exploré. Par conséquent, le fait que la liste Tabou contienne uniquement les dernières opérations effectuées empêche d'outrepasser d'optimum local dans la meilleure direction.

Par ailleurs, l'utilisation d'une liste contenant des mouvements peut s'avérer contraignante, dans le sens où elle peut impliquer une stagnation généralisée de la recherche lorsque la liste comporte plusieurs mouvements interdits (Beretta, Castelli, Gonçalves, Henriques, & Ramazzotti, 2018). Il est facile de déduire que ce phénomène est d'autant plus remarquable lorsque le rapport $\frac{k}{N}$ augmente, où k est le cardinal de la liste Tabou, et N le nombre de sommets (*i.e.* le nombre de variables dans le jeu de données).

4.2.5.2 Adaptation de l'algorithme du Hill Climbing

Étant donné l'ensemble des remarques que nous venons de formuler, nous proposons les améliorations suivantes à l'algorithme Hill Climbing avec une recherche Tabou :

— **Prise en compte des connaissances à priori**

Dans l'algorithme du Hill Climbing classique, le graphe de départ peut être un graphe connu dont la véracité est admise, ou bien un graphe vide ou complet si aucun graphe ne peut être présupposé. Nous considérons que c'est une hypothèse assez forte, dans le sens où la véracité du graphe de départ conditionne tout le reste du déroulement de l'algorithme, et par conséquent l'exactitude du graphe final. Aussi, certaines connaissances partielles peuvent être connues, sans pour autant permettre la construction d'un graphe. C'est notamment le cas lorsque les connaissances ne sont pas "constructives" comme c'est le cas de la connaissance d'un ou plusieurs liens causaux, mais qu'elles sont plutôt "disqualifiantes", comme c'est le cas pour les variables exogènes et les ordonnancements, que nous avons précédemment identifiées. Bien que les connaissances "disqualifiantes" ne nous suggèrent pas dans quel sens il faut orienter la recherche, et ne nous proposent pas de graphe à proprement parler, elles nous suggèrent tout de même dans quel sens il ne faut pas l'orienter. Leur prise en compte peut donc être bénéfique pour la recherche, à plus forte raison lorsque les connaissances disponibles ne sont pas suffisantes pour construire un

graphe de départ. Nous proposons donc d'adapter le Hill Climbing en ajoutant plus de flexibilité quant au choix du graphe de départ. Nous proposons donc de donner la possibilité de renseigner des connaissances appartenant à une plusieurs des trois catégories que nous avons citées dans la sous-section 4.2.4 : les variables exogènes, les liens causaux déjà connus, et les ordonnancements. Ceci permettra d'abord de prendre en compte des connaissances partielles (disqualifiantes) même lorsqu'elles ne sont pas suffisantes pour construire un graphe de départ. Nous gardons tout de même la possibilité de démarrer avec un graphe vide si aucune connaissance n'est discernée.

— **Contenu de la liste Tabou**

Nous proposons d'utiliser le Hill Climbing avec une liste Tabou mémorisant les derniers graphes ayant déjà été visités et retenus dans les itérations précédentes, plutôt que les dernières opérations effectuées. Ceci permettra de remédier au problème de stagnation du processus de recherche, et d'éviter d'écarter l'exploration de nouvelles structures potentiellement intéressantes, pour la simple raison qu'elles soient issues d'une opération Tabou. La liste Tabou sera donc composée de listes contenant chacune les couples ordonnés représentant les arcs d'un graphe déjà exploré dans les itérations précédentes. Cette liste ne prendra donc pas en compte la manière avec laquelle les graphes des itérations passées ont été obtenus. En effet, ce n'est pas tant l'interdiction des mouvements qui nous intéresse, mais plutôt l'interdiction des graphes dont le voisinage a déjà été évalué, afin de ne pas augmenter le score en revenant à une structure qui a déjà été sélectionnée et donc créer une boucle. Nous parlons de boucle puisque l'utilisation d'une liste Tabou n'intervient qu'en cas d'atteinte d'optimum, elle n'est donc utilisée que pour autoriser le score à baisser. Et si nous baissons le score en revenant à l'exploration une structure déjà visitée, l'algorithme choisira son meilleur voisin, qui a lui aussi déjà été visité, et reviendra finalement au même graphe optimal, à partir duquel la baisse de score a été autorisée, ainsi de suite.

Par ailleurs, le paramètre k associé à la taille de la liste Tabou contenant les dernières opérations effectuées est décisif, et conditionne le déroulement du reste de l'algorithme. En effet, une liste de grande taille entraînera la stagnation de la recherche en interdisant beaucoup de mouvements, et empêchera l'exploration de plusieurs de structures n'ayant jamais été retenues. Une liste de faible taille entraînera également une stagnation puisque certains mouvements ne faisant pas partie de la liste peuvent être réeffectués, ce qui peut conduire à réévaluer une même structure plusieurs fois. Certains travaux recommandent l'utilisation d'une liste de taille 10, mais cette taille ne peut pas être valable pour tous les jeux de données. Le choix d'utiliser une liste de taille aléatoire après chaque itération reste également périlleuse puisqu'il s'agit d'un paramètre décisif dont dépend le comportement de l'algorithme. L'utilisation d'une liste mémorisant les graphes dont le voisinage a déjà été évalué permettra d'outrepasser le problème de définition de la taille de la liste Tabou, en s'affranchissant simplement de ce paramètre. En effet, même en stockant tous les graphes ayant été explorés depuis le début des itérations, cela ne mènera pas à la stagnation de la recherche : si nous reprenons notre exemple, stocker tous les graphes retenus à chaque itération n'empêchera pas de retenir le graphe composé des arcs $A \rightarrow B$,

$C \rightarrow A$, et $C \rightarrow B$ à la sixième itération, puis en évaluer le voisinage à septième itération, pour enfin trouver le graphe escompté (*i.e.* celui qui est composé des arcs $B \rightarrow A$, $C \rightarrow A$, et $C \rightarrow B$).

— **Problème de plateaux**

Comme nous l'avons précédemment évoqué, le problème de plateaux est partiellement résolu en autorisant simplement l'égalité dans la condition de sélection du graphe qui sera évalué à la prochaine itération. Avant de recourir à l'autorisation de la dégradation du score, et donc à l'utilisation de la liste Tabou, il serait logique d'explorer d'abord un graphe maintenant le score stable pour voir si son voisinage ne contient pas un graphe qui augmenterait le score. En effet, même dans le cas où deux graphes $\mathcal{G}$ et $\mathcal{G}'$ obtiennent le même score en raison de leur équivalence, leurs voisinages ne seraient pas identiques. Comme illustré sur la figure 4.3, les deux graphes $\mathcal{G}$ et $\mathcal{G}'$, qui appartiennent à la même classe d'équivalence, ne possèdent pas le même voisinage. Si l'on admet, par exemple que $\mathcal{G}$ est le graphe de départ, et qu'aucun graphe n'augmente le score, le fait d'autoriser l'égalité permettra de sélectionner le graphe $\mathcal{G}_2$, puisqu'il est le dernier graphe équivalent à avoir été généré dans le voisinage, et donc le dernier graphe à avoir obtenu le même score. La sélection de ce graphe − qui est le même que $\mathcal{G}'$− permettra de découvrir de nouvelles structures qui augmenteraient potentiellement le score.

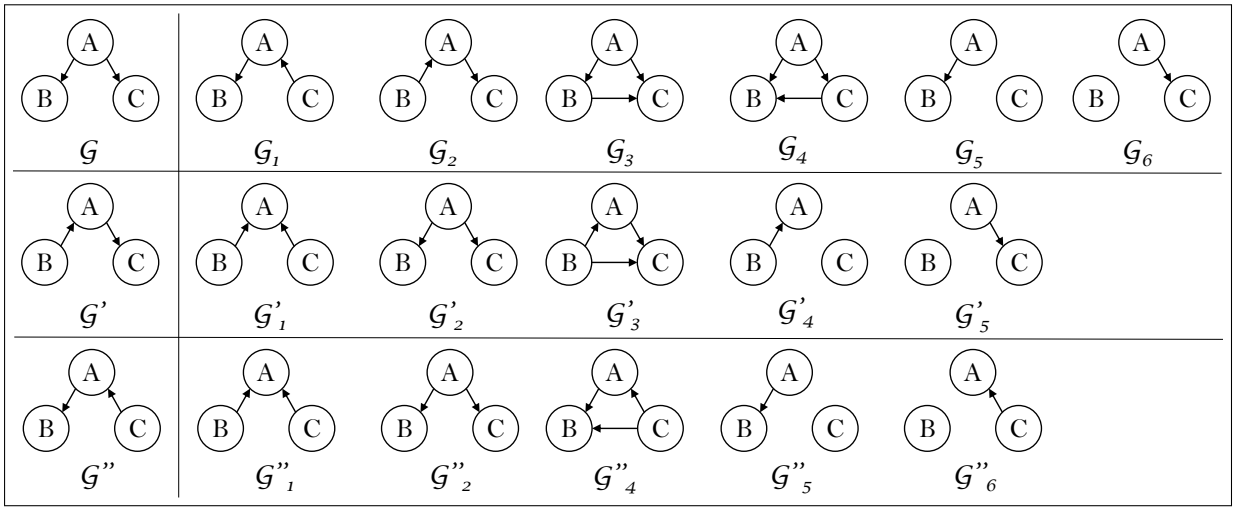


FIGURE 4.3 – Voisinage de deux graphes équivalents $\mathcal{G}$ et $\mathcal{G}'$

Cependant, ceci ne résout que partiellement le problème. Comme le montre la figure 4.3, le graphe équivalent $\mathcal{G}_1$− qui correspond au graphe $\mathcal{G}'$−, possède également des graphes voisins qui n'appartiennent pas aux voisinages de ses deux graphes équivalents $\mathcal{G}$ et $\mathcal{G}'$. Le choix du graphe $\mathcal{G}_1$ à la première itération aurait donc mené la recherche dans une autre direction que le choix du graphe $\mathcal{G}_2$. Dans notre version du Hill Climbing avec recherche Tabou, nous autorisons donc l'égalité des scores pour pouvoir passer à l'itération suivante en cas d'égalité. Bien que cela permette de résoudre le problème d'égalité de score entre le graphe évalué et les graphes de son voisinage, le problème d'égalité de score égaux persiste entre membres du voisinage.

— **Égalité des scores entre membres de voisinage, et écarts insignifiants entre scores**

Dans le voisinage de chaque graphe, il peut y avoir plusieurs graphes à score égaux. Ceci peut être rencontré dans deux situations différentes : la première, que nous avons citée plus haut, est celle de l'équivalence, qui a lieu lorsque les opérations appliquées au graphe de l'itération génèrent deux ou plusieurs graphes équivalents, comme c'est le cas dans la première itération de l'exemple illustré par la figure 4.3, où deux graphes équivalents ont été générés ($\mathcal{G}_1$ et $\mathcal{G}_2$) ; la deuxième situation est celle où, parmi les graphes du voisinage, deux ou plusieurs graphes non-équivalents obtiennent le même score. Bien que cette situation est moins fréquente que la première, elle n'est pas à exclure, en raison de la nature du score utilisé qui pénalise les graphes selon leur complexité. En effet, un graphe $\mathcal{G}$ représentant mieux les relations entre les données peut obtenir le même score qu'un autre graphe $\mathcal{G}'$ représentant moins fidèlement les relations entre les données, mais étant moins complexe que le graphe $\mathcal{G}$ qui est donc plus pénalisé sur sa complexité qui contrebalance sa vraisemblance (François, 2006).

Concernant la première situation, et étant donné la taille de l'espace de recherche (Cf. section 2.2.6.1), il n'est pas envisageable d'évaluer tous membres de tous les voisinages équivalents présentant des scores égaux et optimaux. Nous proposons de créer une liste, où seront stockés tous les graphes (sous forme de listes de couples ordonnés représentant leurs arcs) équivalents ayant obtenu un score optimal à une itération donnée, mais n'ayant pas été sélectionnés pour l'itération suivante. Nous soulignons que dans cette liste, seuls les graphes respectant les contraintes que nous avons définies dans la section 4.2.4 peuvent être stockés. Les graphes présents dans cette liste représenteront ensuite les graphes de départs pour d'autres itérations du Hill Climbing. Nous utiliserons donc un Iterated Hill Climbing, qui consiste à relancer l'algorithme plusieurs fois, mais au lieu d'utiliser uniquement des graphes de départs aléatoires, nous utiliserons également les graphes mémorisés dans la liste. Les graphes de départs générés aléatoirement doivent également respecter les mêmes contraintes que ceux issus de la liste.

Concernant la deuxième situation, nous proposons de calculer le rapport des taux d'évolutions afin de pouvoir élire, lors d'une itération, un graphe parmi plusieurs graphes candidats ayant des scores égaux, mais n'étant pas équivalents. Avant de recourir à ce choix, il faut d'abord vérifier que tous les graphes candidats respectent les contraintes du choix de graphe (Cf. sous-section 4.2.4). Si tel n'est pas le cas, les graphes ne respectant pas les contraintes sont systématiquement exclus, et le choix se fait uniquement entre les graphes candidats, s'il y en reste plus d'un. Considérons un graphe $\mathcal{G}$ ayant un score $BIC_{\mathcal{G}} = LL(\mathcal{G}) - p$, et contenant dans son voisinage deux graphes $\mathcal{G}_1$ et $\mathcal{G}_2$ ayant deux scores égaux $\text{BIC}_{\mathcal{G}_1}$ et $\text{BIC}_{\mathcal{G}_2}$ tels que $\text{BIC}_{\mathcal{G}_1} = LL(\mathcal{G}_1) - p_1$ et $BIC_{\mathcal{G}_2} = LL(\mathcal{G}_2) - p_2$. Où $LL(.)$ désigne le terme de la log-vraisemblance du graphe par rapport aux données, et le terme $p_.$ désigne le terme de la pénalité associée à la complexité du graphe. Le taux d'évolution permet de quantifier l'évolution d'une grandeur, relativement à une valeur de départ. Nous allons donc évaluer, pour chacun des deux graphes $\mathcal{G}_1$ et $\mathcal{G}_2$, dans un premier temps, les évolutions de leurs premiers termes reflétant les précisions des graphes, puis dans un deuxième temps l'évolution

de leurs deuxièmes termes reflétant les complexités des graphes, pour enfin comparer les évolutions des deux termes des deux graphes. Les valeurs suivantes devront alors être calculées :

$$E_{LL_1} = \frac{LL(\mathcal{G}_1) - LL(\mathcal{G})}{LL(\mathcal{G})} \tag{4.11}$$

$$E_{LL_2} = \frac{LL(\mathcal{G}_2) - LL(\mathcal{G})}{LL(\mathcal{G})} \tag{4.12}$$

$$E_{p_1} = \frac{p_1 - p}{p} \tag{4.13}$$

$$E_{p_2} = \frac{p_2 - p}{p} \tag{4.14}$$

Les rapports $\frac{E_{LL_1}}{E_{p_1}}$ et $\frac{E_{LL_2}}{E_{p_2}}$ sont ensuite calculés et comparés. Le graphe ayant obtenu le plus grand rapport est celui qui sera sélectionné, puisque ce sera celui qui aura eu une évolution plus importante de la précision par rapport à l'évolution de sa pénalité.

Par ailleurs, lorsque l'augmentation observée entre deux score Δ_{s_i,s_j} est insignifiant, ce calcul peut également être effectué, où s_i et s_j représentent respectivement les scores de deux graphes i et j, tels que $s_i > s_j$. En effet, lorsque $\Delta_{s_i,s_j} < 2$, cet écart est jugé insignifiant, et mérite d'être analysé de plus près (Guenter, 2012 ; Anderson, 2007). Considérons un graphe $\mathcal{G}$ et un graphe $\mathcal{G}_1$ appartenant au voisinage de $\mathcal{G}$, tel que $\mathcal{G}_1$ maximise le score et respecte les contraintes préalablement fixées. S'il existe un graphe $\mathcal{G}_2$ appartenant au voisinage de $\mathcal{G}$ et respectant les contraintes préalablement fixées, tel que $\Delta_{s_1,s_2} < 2$, il convient de le sélectionner comme un graphe candidat et d'effectuer une comparaison des évolutions des termes respectifs de précision et de pénalité des scores de $\mathcal{G}_1$ et $\mathcal{G}_2$ relativement à $\mathcal{G}_1$, selon les formules 4.11, 4.12, 4.13, et 4.14.

La figure 4.4 décrit le déroulement de notre version de l'algorithme Iterated Hill climbing avec liste Tabou. L'indice i correspond aux différentes exécutions de l'algorithme. L'indice r correspond aux itérations déroulées dans chacune des exécutions r. La liste L correspond à la liste Tabou contenant les graphes qui ont déjà été explorés et retenus dans une itération donnée r, quelle que soit l'exécution i. $\mathcal{S}_{\cdot}$ désigne le score du graphe renseigné en indice. La liste M correspond aux graphes ayant des scores égaux ou très proches du score du graphe maximisant le score, dans une itération donnée r. La liste P correspond aux graphes ayant eu des scores égaux ou très proches du score du graphe maximisant le score, dans une itération donnée r, et qui n'ont pas été retenus dans cette itération. S_max dénote le score maximal dans chaque itération r, et S_{max_i} le score maximal dans toutes les itérations r d'une exécution i. La liste B correspond à la liste de taille i, contenant les meilleurs graphes (le meilleur de chaque exécution i).

L'étape "Vérifier la cohérence des ensembles V_e, C, et O", est déroulée conformément aux règles énoncées dans la sous-section 4.4.2. L'étape "Générer le voisinage V_{G_r} de G_r tel que $\forall G_{rk} \in V_{G_r}$, G_r vérifie les règles de cohérence, et $G_r \notin L$" est similaire à la génération de voisinage de la version originale de l'algorithme Hill Climbing, en excluant en plus des graphes générant un cycles, tous les graphes Tabou, ainsi que tous les graphes ne respectant pas les contraintes définies dans V_e, C, et O.

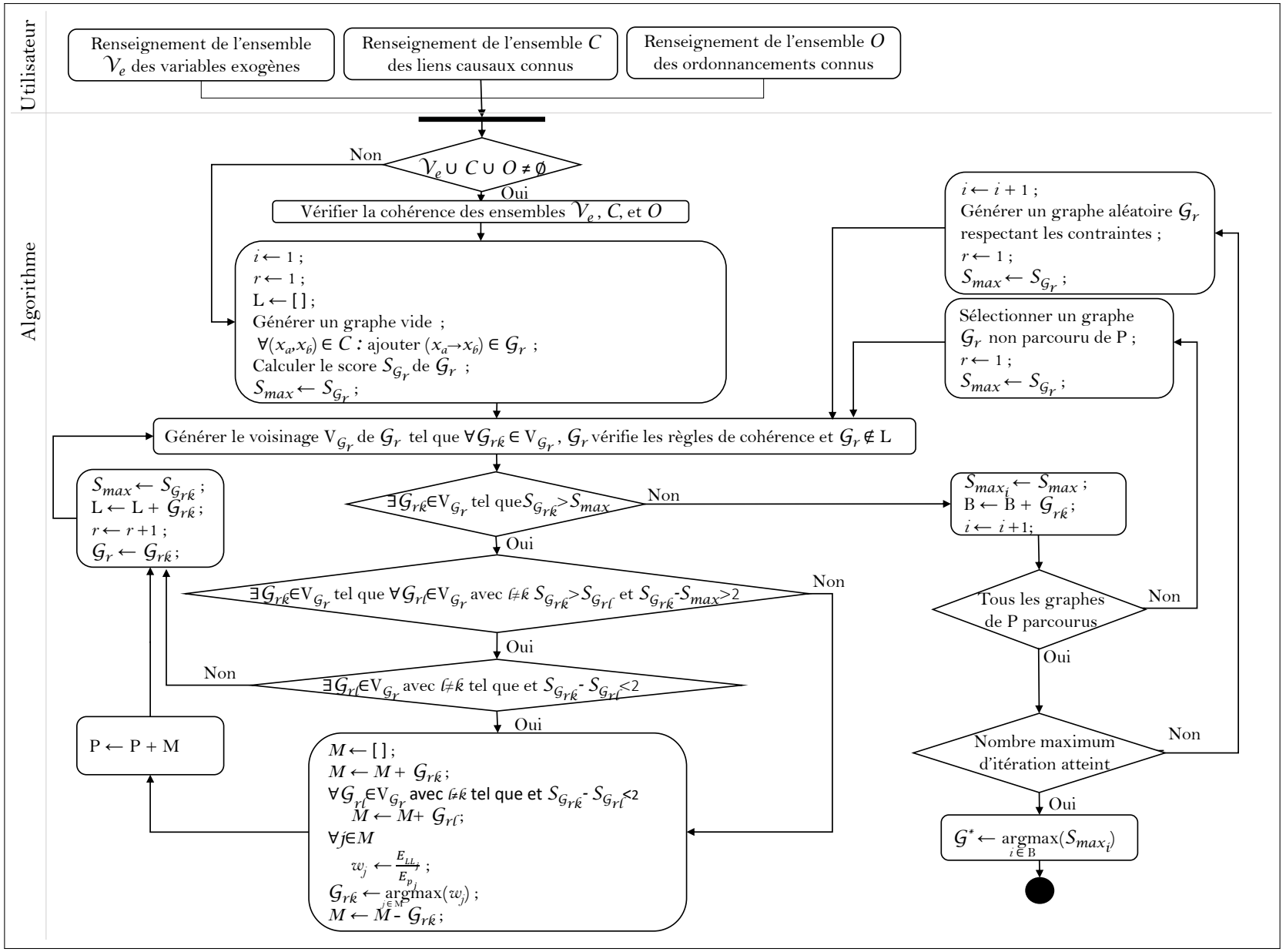


FIGURE 4.4 – Diagramme décrivant le déroulement de l'algorithme adapté de l'Iterated Hill Climbing avec liste Tabou.

4.3 Apprentissage des réseaux de neurones

Selon les besoins identifiés et les activités menées par une entreprise, ou au cours d'un projet, certains KPIs peuvent s'avérer plus pertinents que d'autres. Afin que leur exploitation soit utile, les KPIs qu'une entreprise ou une équipe choisit de suivre doivent s'aligner en tout point avec les objectifs souhaités. Or, ces choix peuvent être faits parmi un grand nombre de KPIs : il existerait au moins 75 KPIs différents prédéfinis (Marr, 2012). En outre, il existe d'autres KPIs développés singulièrement par une entreprise ou une équipe de manière à répondre à des besoins spécifiques (Marr, 2012). Par ailleurs, et comme nous l'avons souligné dans la section 1.2.2 du chapitre 1, l'anticipation de l'évolution de chaque KPI est primordiale à l'amélioration de la prise de décision. En effet, elle permettrait d'anticiper la recherche et la sélection d'alternatives afin d'éviter toute évolution défavorable de façon préventive si les causes sont identifiées, ou du moins avancer le début des réflexions, et par conséquent réduire le temps de rétablissement de la situation de façon corrective au cas où la prévention ne serait pas possible. En tout état de cause, avancer le moment de prise de conscience du besoin d'agir ne peut qu'être profitable, et ceci passe par la prédiction. Or, la construction des réseaux de neurones pour un nombre aussi important de KPIs peut vite devenir une lourde tâche. Par ailleurs, comme préalablement discuté dans les chapitres 2 et 3, nous avons fait le choix de répondre à la fonction de prédiction en utilisant des réseaux de neurones. En effet, afin de pouvoir prendre des décisions efficaces, nous souhaitons classer, par ordre d'importance, les causes identifiées grâce à l'analyse causale. Cette tâche n'est pas évidente en utilisant les tables de probabilités conditionnelles et marginales issues du réseau Bayésien, étant donné le nombre important de combinaisons à analyser. Nous nous orientons vers l'exploitation des poids finaux des réseaux de neurones ayant un bon pouvoir prédictif.

L'utilisation des réseaux de neurones nous permettra donc à la fois d'anticiper le moment de prise de conscience du besoin d'agir d'une part, et d'établir un classement des causes pour faciliter et fonder la sélection d'alternatives d'autre part. Pour ce faire, il est nécessaire de pouvoir d'abord construire un réseau de neurones ayant un bon pouvoir prédictif. Comme expliqué auparavant, cette construction nécessite du temps, à plus forte raison lorsque plusieurs réseaux, permettant chacun de prédire un KPI différent, doivent être construits. Cela s'avère être le cas dans notre contexte, comme nous venons de le souligner dans le paragraphe précédent. Les réseaux de neurones construits pourront donc, dans un premier temps, être employés dans la phase de développement pour classer les causes de leurs variables cibles, et dans un deuxième temps, ils serviront dans la phase d'utilisation pour les états futurs des variables cibles surveillées. La place importante que prend la construction de réseau de neurones, et la manière dont elle conditionne la qualité de la prédiction et la pertinence du classement des causes, ont été décrites dans la sous-section 3.2.3 du chapitre 3.

4.3.1 Apprentissage des réseaux de neurones par le biais de la neuroévolution

Les contraintes de temps liées au processus de construction des réseaux de neurones relèvent principalement son caractère itératif, que nous avons abordé dans la sous-section 3.2.3 du chapitre 3. Nous proposons ici d'automatiser ce processus, de manière à apporter de la rapidité et de la généricité à cette étape capitale qui conditionne la qualité de la

prédiction d'un côté, et qui facilite l'étape suivante d'un autre côte, comme nous le verrons dans la section 4.4 de ce chapitre.

Dans la littérature, de nombreux algorithmes sont proposés pour concevoir les réseaux de neurones. Les algorithmes les plus souvent utilisés sont les algorithmes constructifs, et les algorithmes d'élagage, aussi dits destructifs (Shih-Hung & Yon-Ping, 2012a). Les algorithmes constructifs, qui commencent avec une architecture minimale, puis ajoutent progressivement des nœuds pendant la phase d'apprentissage, souffrent de l'inconvénient lié à la difficulté de décider quand ajouter des neurones ou des connexions cachées et quand arrêter la mise à jour du réseau ((Shih-Hung & Yon-Ping, 2012a). À l'opposé des algorithmes constructifs, les algorithmes d'élagage partent d'un grand réseau de neurones, puis élaguent les neurones en utilisant l'un des critères de réductions existants (Tin-Yau & Dit-Yan, 1997). À l'inverse, ces algorithmes souffrent donc d'une contrainte forte liée au réseau de départ, qui est généralement définie par le concepteur du réseau de neurones. Le problème de la sous-estimation du problème peut alors se poser et par conséquent le réseau de départ serait insuffisant pour obtenir une bonne performance de prédiction. Subséquemment, si le réseau de départ est insuffisant, le réseau final construit aura également une performance faible, puisque ce type d'algorithme procède à la construction par réduction.

Pour un réseau de neurones destiné à effectuer une tâche spécifique, l'espace de recherche d'une topologie idéale et les hyper-paramètres qui lui sont associés, est infiniment grand, en raison des combinaisons possibles du nombre couches, du nombre de neurones par couches, et des hyper-paramètres. L'optimalité de la solution est d'autant plus difficile à atteindre du fait que plusieurs solutions différentes et éloignées les unes des autres, peuvent fournir des réseaux avec des performances très proches, la contraposée étant également vraie. Le caractère épars des solutions ayant de bonnes performances, rend alors les recherches de type constructif ou destructif peu adaptées à l'approximation de la solution optimale dans un tel espace. En effet, ces méthodes peuvent être assimilées à des recherches de type Hill Climbing, qui évoluent progressivement dans le voisinage de la meilleure solution trouvée. C'est cette non linéarité, entre les tailles des réseaux et leurs performances associées, qui rend ardue la tâche de la décision d'arrêter d'ajout et de suppression de neurones, respectivement pour les algorithmes constructifs et destructifs. En effet, il y a de fortes chances de tomber dans un minimum local si la recherche est interrompue dès la détérioration des performances. En outre, ces méthodes contraignent la recherche en lui imposant d'évoluer dans une seule et même direction tout au long du processus de recherche (Angeline, Saunders, & Pollack, 1994). Par ailleurs, la définition de la notion de voisin dans ce type d'algorithmes se restreint à la topologie du réseau, puisqu'il s'agit de méthodes purement structurelles. Or, deux réseaux ayant la mêmes topologie, et des hyper-paramètres différents, auraient des performances différentes, chose qui implique de rigueur la réitération du processus de recherche en utilisant des combinaisons différentes des hyper-paramètres possibles à chaque itération. Aussi, le caractère épars des solutions rend plus appropriées les recherches basées sur une population de solutions pertinentes plutôt que sur une solution unique, ce qui est notamment le cas pour les algorithmes génétiques (Angeline et al., 1994), qui permettent, de surcroît, une recherche non monotone. Le fait de conduire la recherche d'une population de solution ouvre également la possibilité d'élargir la notion de voisinage au delà de la topologie, en y incluant notamment les hyper-paramètres.

Dans notre proposition, nous suggérons donc d'utiliser une méthode de conception évolutionniste des réseaux de neurones, également appelée neuro-évolution. Celle-ci basée sur les algorithmes génétiques, qui ont montré un grand potentiel dans la conception des réseaux de neurones (Mantzaris, Anastassopoulos, & Adamopoulos, 2011b), et qui offrent l'avantage d'être moins sensibles aux réseaux initiaux que les algorithmes qui évoluent de façon monotone. Bien que de nombreux travaux aient eu recours aux algorithmes génétiques pour concevoir des réseaux de neurones (Shih-Hung & Yon-Ping, 2012b ; Ahmadizar, Soltanian, AkhlaghianTab, & Tsoulos, 2014 ; Stanley et al., 2019 ; Kwok & Yeung, 1997), à l'état actuel de nos connaissances, des problèmes liés à l'encodage et à l'application des opérations d'évolution persistent, et la prise en compte des paramètres d'apprentissage du réseau, simultanément à son architecture, n'a pas été abordé jusqu'à présent par les méthodes de construction des réseaux de neurones basées sur les algorithmes génétiques.

Dans la suite, nous présenterons brièvement les algorithmes génétiques, puis nous détaillerons l'encodage que nous avons établi pour adapter l'algorithme génétique au problème de construction des réseaux de neurones. Nous décrirons également comment nous avons pris en compte le problème du choix de la fonction d'activation, ainsi que les autres hyper-paramètres dans cet encodage.

4.3.2 Les algorithmes génétiques

Les algorithmes génétiques font partie des algorithmes évolutionnistes d'optimisation. Inspirés de la nature, ces algorithmes de recherche méta-heuristique sont fondés sur les théories de génétique évolutive et de sélection naturelle. Comme pour toutes les méthodes de recherche méta-heuristique, un algorithme génétique essaye de converger vers la meilleure solution possible pour résoudre un problème d'optimisation en explorant un espace de solutions. Dans notre cas, la meilleure solution correspond au réseau de neurones ayant le meilleur pouvoir prédictif. Pour ce faire, ces algorithmes se basent sur la théorie de Darwin sur l'évolution des espèces, qui rejette l'existence de systèmes naturels figés et déjà adaptés à n'importe quelle condition extérieure, et qui affirme que les êtres évoluent en s'adaptant progressivement à travers les processus de reproduction et de mutation. Uniquement les mieux adaptés adaptés aux besoins, selon le contexte, peuvent se reproduire pour transmettre leur patrimoine génétique et obtenir une progéniture adaptée.

La théorie de Darwin de laquelle sont inspirés les algorithmes génétiques repose sur trois principes :

— La variation : Pour une même espèce, les individus ne sont pas complètement similaires, chacun est plus ou moins unique. De génération en génération, des variations légères peuvent apparaître et se transmettre ;

— L'adaptation : Il s'agit de la lutte pour l'existence. Chaque individu se distingue par des caractéristiques qui forment le point de départ du processus de sélection naturelle, et qui peuvent être nocifs, favorables, ou sans incidence. Plus un individu est adapté à son environnement, plus il aura la chance d'atteindre l'âge adulte et de se reproduire ;

— L'hérédité : Les caractéristiques de chaque individu sont héréditaires et peuvent être transmises à ses descendants. En transmettant les caractéristiques favorables des antécédents, l'espèce évoluera progressivement.

À l'image de cette théorie, les algorithmes génétiques pour la construction des réseaux de neurones suivent le même principe : ils partent d'une population initiale composée de plusieurs réseaux (individus), et font évoluer ces réseaux au fil des générations, en favorisant ceux qui sont le mieux adaptés au problème (*i.e.* ceux qui ont un meilleur pouvoir prédictif) pour leur permettre de transmettre leurs caractéristiques au fil des générations, et en opérant des mutations pour permettre de mieux explorer l'espace de recherche.

Les notions et appellations de cette théorie sont retrouvées dans les algorithmes génétiques :

— Population : représente un ensemble de solutions possibles (chaque population est composée d'un ensemble de solutions appelés individus) ;

— Individu : représente une solution (optimale ou non), chaque solution est symbolisée par un ensemble de chromosomes ;

— Chromosomes : ce sont les chromosomes qui décrivent les caractéristiques de chaque solution. Chaque chromosome correspond à une caractéristique particulière de la solution. Afin de représenter nos solutions sous forme de génotype, il faut trouver un encodage pour les chromosomes. Dans notre proposition nous choisirons l'encodage binaire, en motivant notre choix, où chaque chromosome est composé de bits pouvant prendre la valeur 1 ou 0 ;

— Génotype : correspond à l'encodage d'un individu de la population (donc à l'encodage d'une solution potentielle), sur un ensemble de chromosomes ;

— Phénotype : correspond à l'individu dans son état apparent, et non codé. Il peut être confondu avec l'individu.

Les algorithmes génétiques reposent sur trois opérateurs d'évolution pour faire converger le problème vers une solution optimale :

— Sélection : C'est l'opérateur qui correspond au principe d'adaptation de la théorie évolutionniste ; seules les meilleures solutions sont sélectionnées pour se "reproduire". Le but étant de faire converger la fonction d'évaluation vers un optimum global, les individus qui ont les valeurs les plus optimales de la fonction d'évaluation sont donc les plus adaptées. Pour sélectionner les individus qui se reproduiront, plusieurs techniques existent : la sélection élitiste, la roulette de casino, la sélection par tournoi, la sélection uniforme ;

— Croisement : qui est opéré entre les solutions les plus adaptées qui ont été sélectionnées, deux à deux. Les nouvelles solutions issues de la reproduction reprennent et mixent les chromosomes de leurs deux parents, ceci implémente le principe d'hérédité de la théorie de Darwin qui a lieu grâce au brassage génétique ;

— Mutation : Les génotypes du nouvel individu issu du croisement, peuvent être modifiés ou non selon une probabilité, appelée taux de mutation. La modification d'un chromosome s'effectue en substituant un gène de manière aléatoire (*i.e* en substituant 1 par 0 ou l'inverse dans le cas d'un codage binaire). Ceci correspond au

principe de variation de la théorie évolutionniste.

La figure 4.5 montre un aperçu du déroulement d'un algorithme génétique. Le critère d'arrêt peut correspondre à un nombre maximum d'itérations pour aboutir à une convergence de performance, ou à un seuil devant être franchi par la fonction d'évaluation (par exemple un seuil de précision dans le cas de recherche d'un modèle de prédiction).

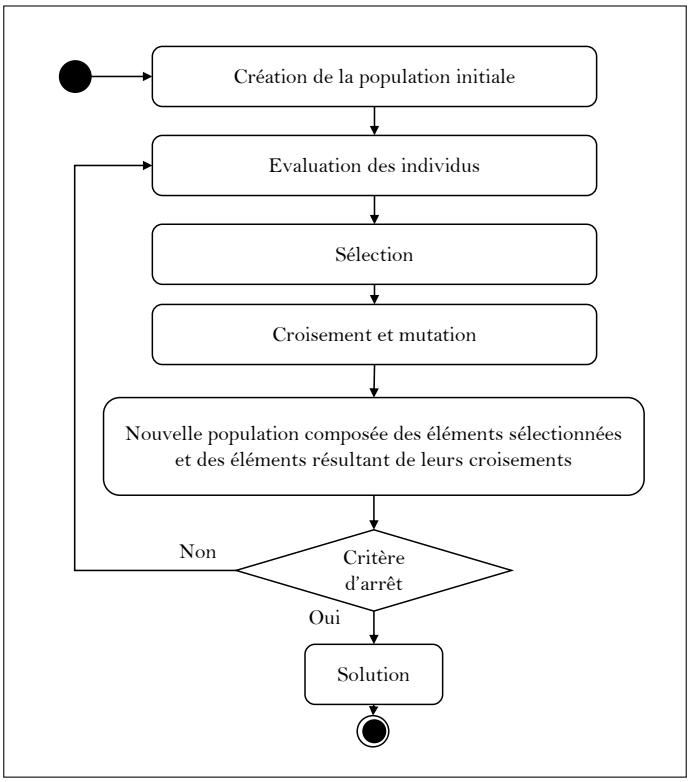


FIGURE 4.5 – Déroulement d'un algorithme génétique.

4.3.3 Algorithme de neuro-évolution

Dans le contexte de la construction de réseaux de neurones par neuro-évolution, les individus sont représentés par les réseaux de neurones. Afin de les faire évoluer, il est nécessaire de définir un encodage adapté qui puisse les représenter fidèlement sous forme de génotypes composés de chromosomes pour ensuite y appliquer les opérateurs d'évoluer. Avant de spécifier un tel encodage, il est nécessaire de spécifier ce que nous attendons exactement (*i.e.* les éléments que nous souhaitons faire évoluer, ainsi que ce qui les caractérise). L'encodage pourra ensuite être défini en adéquation avec ces spécifications. Cet encodage doit être assorti d'une fonction de transformation, ou d'un ensemble de règles permettant de le transformer en réseau de neurones testable et viable (*i.e.* pouvant être entraîné et testé). Ce processus est décrit sur la figure 4.6 pour le cas d'un encodage en chaînes binaires. Nous soulignons que les chaînes binaires apparaissant dans cette figure sont utilisées à une fin d'illustration, et ne possèdent pas de règles permettant une interprétation correcte pour arriver à leurs réseaux correspondants.

Une chaînes binaire encodant un réseau de neurones sous forme d'une série de chromosomes correspond au génotype d'une solution, et le réseau de neurones correspondant

représente son phénotype associé. L'algorithme génétique doit donc générer des populations, composées des génotypes des réseaux de neurones solutions. Les génotypes des solutions doivent ensuite être transformés en phénotypes associés, afin de pouvoir en évaluer les performances. C'est donc dans l'espace des génotypes que les opérations d'évolution doivent être mises en œuvre.

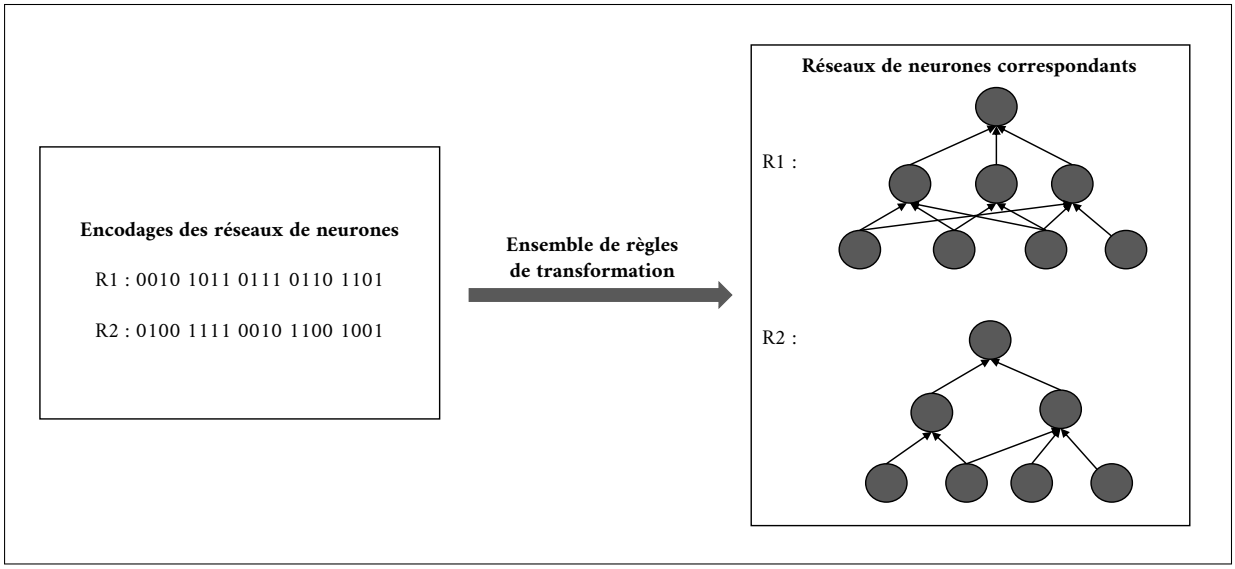


FIGURE 4.6 – Description du processus de passage d'un encodage de solutions représentant des réseaux de neurones, vers les réseaux de neurones correspondant aux solutions encodées, par le biais d'un ensemble de règles de transformation.

Par conséquent, notre proposition, la construction des réseaux de neurones par algorithme génétique se déroule en deux temps. Tout d'abord, il existe un fichier contenant l'algorithme génétique, dans lequel sont exécutées la création de la population initiale, la sélection, ainsi que les opérations de croisement et de mutation, dont les règles y sont également définies. Chaque solution (*i.e.* chaque réseau de neurones) créée lors de la génération de la population initiale, ou bien après les opérations d'évolution, est codée sous forme de génotype, en respectant des règles spécifiques que nous détaillerons dans la suite de cette section. Les génotypes de la population sont ensuite transmis, un par un, en entrée d'un autre fichier, servant à effectuer la transformation permet de retrouver la structure et les hyper-paramètres de la solution, et par conséquent, d'obtenir le phénotype correspondant pouvant être entraîné et testé. Suite à cette transformation, les réseaux de neurones correspondant aux solutions sont entraînés, testés et évalués. Les résultats de leurs évaluations respectives sont ensuite transmis au fichier initial de l'algorithme génétique, qui procède donc à comparaison des résultats et la sélection des meilleurs réseaux, puis au déroulement des opérations d'évolution sur les solutions sélectionnées. Les solutions composant la génération suivante repassent par les mêmes étapes, ainsi de suite, jusqu'à ce que le critère d'arrêt soit atteint.

La figure 4.7 illustre le processus d'évolution des réseaux de neurones que nous venons de décrire.

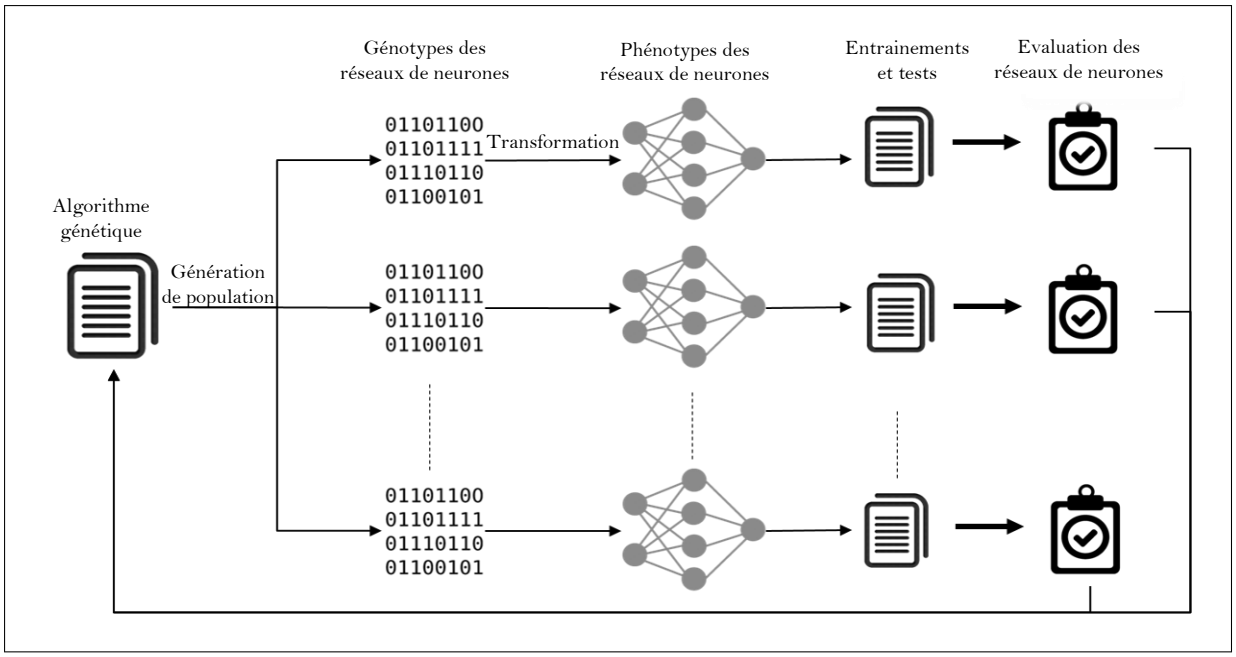


FIGURE 4.7 – Processus d'évolution des réseaux de neurones.

4.3.3.1 Caractérisation du problème et choix de l'encodage

La caractérisation du problème est indispensable avant de choisir l'encodage des génotypes. À l'instar du processus essai-erreur suivi lors d'une construction manuelle, nous souhaitons, dans notre contexte, faire évoluer simultanément la topologie du réseau (*i.e.* le nombre de couches intermédiaires et de nombre de neurones par couche), et les hyper-paramètres régissant son apprentissage, afin de trouver une combinaison d'architecture et d'hyper-paramètres fournissant un réseau ayant de bonnes performances. Certaines études considèrent que la topologie fait partie des hyper-paramètres (T. Yu & Zhu, 2020). Elles parlent donc uniquement des hyper-paramètres, en les catégorisant en deux groupes : les hyper-paramètres liés à la structure du réseau, et les hyper-paramètres liés à l'apprentissage du réseau. D'autres études distinguent topologie et hyper-paramètres, et utilisent le terme hyper-paramètres pour faire référence aux hyper-paramètres d'apprentissage (Smith, 2018). Dans notre contexte, nous parlerons de topologie ou architecture pour désigner le nombre des couches cachées ainsi que leur taille, et des hyper-paramètres pour désigner les hyper-paramètres d'apprentissage.

La topologie du réseau est déterminante pour sa capacité d'apprentissage, puisque la complexité d'un réseau relève de sa topologie. En effet, la complexité du réseau est liée à la complexité de la fonction globale permettant de calculer la ou les sortie(s) du réseau à partir de ses entrées. La complexité de cette fonction est, elle même, liée à la taille du réseau, à la répartition de ses nœuds sur les différentes couches intermédiaires, et aux connectivités ou non-connectivités entre les nœuds. La figure 4.8 illustre trois réseaux de neurones *feed-forward* différents complètement connectés, destinés à effectuer la même tâche en utilisant des topologies différentes ; et utilisant une même fonction d'activation $f(.)$ pour leurs couches cachées et une même fonction $g(.)$ pour la couche de sortie. Les couches cachées sont dénotées par h_i, la couche d'entrée étant h_0 ; un neurone j dans une couche h_i est dénoté h_{ij} ; le neurone de sortie est dénoté s ; le poids de connexion d'un neurone k vers un neurone l est dénoté w_{kl} ; et b_i désigne le biais associé à une couche

h_i. Les expressions des fonctions permettant de calculer la sortie pour chacun de ces trois réseaux, mettent en avant les liens entre la capacité d'un réseau de neurones à répondre à des tâches plus ou moins complexes, et les caractéristiques de sa topologie. En effet, c'est grâce à cette fonction globale que le réseau va d'approximer la fonction sous-jacente liant les entrées aux sorties, si elle existe, sinon de trouver la fonction la permettant de capturer au mieux le phénomène stochastique liant les entrées aux sorties. Si la fonction globale est moins complexe que le phénomène à modéliser, le réseau ne pourra pas réussir son apprentissage, dans le cas contraire, le réseau va sur-apprendre. Les difficultés de définir de façon empirique la profondeur du réseau, le nombre de neurones cachés, et leur répartition sur les couches, seront décrites plus tard dans ce chapitre. Pour le moment, nous nous contentons de conclure que ces éléments sont essentiels pour une profitable construction automatique des réseaux de neurones. Par conséquent, les deux composantes suivantes de la topologie feront partie des éléments que nous souhaitons faire évoluer par le biais de l'algorithme génétique : le nombre de couches, et le nombre de neurones par couches. Nous répertorions donc qu'il s'agit, pour le moment, d'un problème d'optimisation discrète combinatoire.

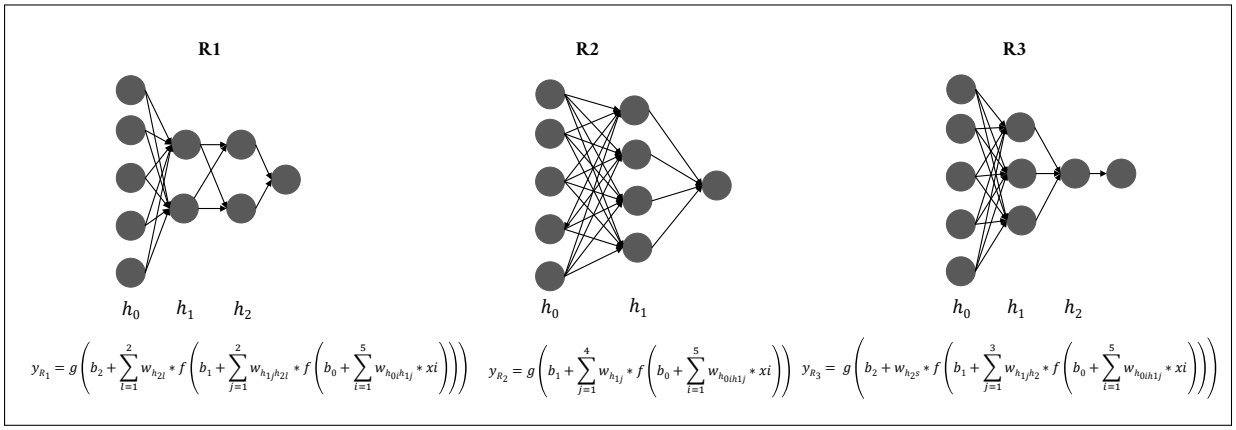


FIGURE 4.8 – Trois réseaux de neurones feed-forward complètement connectés ayant des topologies différentes.

La topologie est l'élément essentiel permettant l'approximation d'une fonction modélisant le mieux les liens entre entrées et la sortie, puisqu'elle détermine la capacité du réseau (Chollet, 2021). Le choix de la fonction d'activation, est également primordial à une telle approximation, puisqu'une simple succession de combinaisons linéaires ne permettrait pas d'approximer toutes les fonctions. Par exemple, pour le réseau R_2 de la figure 4.8, si on admet que $f(.)$ dans la couche cachée est une fonction identité (ce qui revient à ignorer $f(.)$ puisqu'elle n'aura aucun effet), cela aura pour effet capturer les relations en utilisant uniquement des combinaisons linéaires possibles des entrées dans un espace à 4 dimensions, chose qui restreindra la qualité de l'apprentissage. Le fait d'empiler plusieurs couches aura également le même effet. Le nombre de neurones dans une couche, quant à lui, peut augmenter cette dimensionnalité et palier au problème de linéarité, mais ceci aboutira à un sur-apprentissage puisque l'apprentissage s'adaptera de manière quasi exacte aux données d'entraînement (Chollet, 2021). Le rôle d'une fonction d'activation, est justement d'introduire de la non-linéarité lors l'approximation de la modélisation optimale du problème. Le choix de cette fonction doit être adapté à la nature de la tâche attendue du réseau de neurones, ainsi qu'aux données d'entrée. Ce choix n'est pas toujours évident,

et nécessite un certain degré d'expertise et un nombre d'itérations d'essais plus ou moins conséquent selon les applications, ce qui nous incite à prendre en compte, en plus du choix de la topologie, le choix de la fonction d'activation dans la construction automatique des réseaux de neurones par neuro-évolution, lequel s'effectue également dans un domaine discret (puisqu'il faudra choisir une fonction parmi plusieurs). Par conséquent, nous pouvons continuer à considérer notre problème comme étant un problème d'optimisation discrète combinatoire.

Outre l'approximation de la fonction de modélisation des liens entre les entrées et la sortie, le deuxième enjeu est celui de l'optimisation des paramètres de cette fonction. En effet, la topologie et la ou les fonctions d'activation déterminent la définitions de la fonction que l'on cherche. Cependant, afin de calculer une sortie à partir d'une observation (*i.e.* à partir d'un vecteur d'entrées $(x_1, x_2, ..., x_n)$), il faut définir les paramètres $w_{h_{i_j}}$ de la fonction de calcul, qui ne sont autres que les coefficients de la matrice des poids associés au réseau (*i.e.* les poids associés à chacun des neurones de la couche d'entrée et des couches cachées). Cette matrice est obtenue grâce à une optimisation, qui ne peut donc avoir lieu qu'une fois la topologie fixée. L'efficacité et la vitesse de convergence de cette optimisation sont liées à plusieurs hyper-paramètres. Bien que le temps requis pour la construction automatique des réseaux de neurones ne soit pas un critère discriminant dans notre contexte (puisqu'il s'agit d'une étape ponctuelle et non récurrente faisant partie de la phase de développement), la vitesse de convergence, et par conséquent le choix de ces hyper-paramètres aussi, restent tout de même importants, notamment lorsque la taille du jeu de données d'entraînement est limitée. Parmi ces hyper-paramètres d'optimisation (ou d'apprentissage), il y a d'abord l'algorithme d'optimisation, dont le principe est d'utiliser les données d'entrainement pour calculer l'erreur en sortie du réseau, puis de la rétropropager sur les poids afin de les mettre à jour, de façon à minimiser l'erreur en sortie, et ce sur plusieurs itérations. Il existe plusieurs algorithmes d'optimisation des poids, les plus connus étant la descente du gradient stochastique avec ou sans momentum et l'algorithme Adam, pour ce qui de l'optimisation avec condition de premier ordre, et l'algorithme LBFGS pour ce qui est de l'optimisation avec condition de second ordre (Karlsson & Bonde, 2020). Le choix parmi ces algorithmes peut dépendre de plusieurs facteurs, notamment la taille du jeu de données ainsi que la nature du problème (*i.e.* si c'est un problème de classification ou de régression). Dans l'algorithme génétique que nous développons, nous souhaitons également prendre en compte ce choix. Il s'agit donc encore d'un choix à effectuer sur un ensemble discret, chose qui nous permet de maintenir le caractère combinatoire discret de notre problème.

Selon l'algorithme d'optimisation choisi, d'autres hyper-paramètres liés à l'optimisation doivent être spécifiés, les plus importants étant la valeur du taux d'apprentissage et sa configuration (*i.e.* constant ou adaptatif), et le nombre d'itérations (*i.e.* le nombre d'epochs, c'est à dire le nombre de fois où le jeu d'entraînement va être parcouru) (T. Yu & Zhu, 2020). La configuration du taux d'apprentissage résulte d'un choix à faire sur un ensemble discret, comme c'est le cas pour les hyper-paramètres et les caractéristiques de topologie discutés jusqu'à présent. Cependant, la valeur du taux d'apprentissage est un nombre réel devant être choisi généralement entre $10^{-}4$ et 0.4 (Smith, 2018). Concernant le nombre d'epochs, il peut être choisi entre 100 et 1200, cet intervalle a été défini en se référant aux nombres d'epochs retrouvés de façon récurrente dans plusieurs applications ainsi que dans la littérature (Brownlee, 2018).

Étant donné que la majorité des caractéristiques de réseaux de neurones que nous souhaitons faire évoluer sont discrètes, nous optons pour un encodage binaire des génotypes qui représenteront les solutions. Bien que le taux d'apprentissage et le nombre d'epochs soient des valeurs à définir sur des intervalles, leurs précisions usuelles sont connues, ce qui rend la transformation du binaire vers le réel faisable, dans le sens où elle n'entraînera pas de perte d'information puisqu'il n'y aura *à priori* pas de perte de précision si l'encodage binaire est correctement défini.

Maintenant qu'il nous semble qu'un encodage binaire serait le plus approprié à notre besoin, nous devons définir l'architecture de cet encodage, afin que des phénotypes puissent être générés fidèlement aux génotypes correspondants. Puisque nous voulons faire évoluer plusieurs caractéristiques à la fois, nous avons opté pour un encodage par paliers, comme illustré sur la figure 4.9 : chaque génotype est représenté par une liste composée de plusieurs chromosomes représentés par des lignes, où chaque ligne fait référence à une caractéristique particulière identifiable grâce à un code unique. Comme le montre la figure, le génotype peut être perçu comme étant composé de deux parties, dont le début de chacune est reconnaissable grâce à une ligne à contenu inéchangeable : la partie commençant par une chaîne de bits, tous à 0, indique le début de la partie où seront codés toutes les caractéristiques propres au réseau dans son intégralité ; et la partie commençant par un bit à 1 suivi d'une chaîne de bits à 0 désigne le début des caractéristiques propres aux couches cachées. Cette liste, et plus précisément sa deuxième partie encodant les nombres de neurones par couche, est extensible : sa longueur dépend de la valeur que prend le deuxième chromosome du génotype (*i.e.* le nombre de couches cachées). Nous avons sciemment choisi de coder séparément les caractéristiques de l'ensemble du réseau et celles de ses couches, d'abord pour pouvoir faire évoluer ce codage avec un effort moindre en cas de besoin, notamment si l'on souhaite attribuer des fonctions d'activation différentes pour chacune des couches cachées, et également pour faciliter les opérations de croisement et faire en sorte qu'elles soient efficaces, comme nous allons l'expliquer dans la sous-section 4.3.3.4.

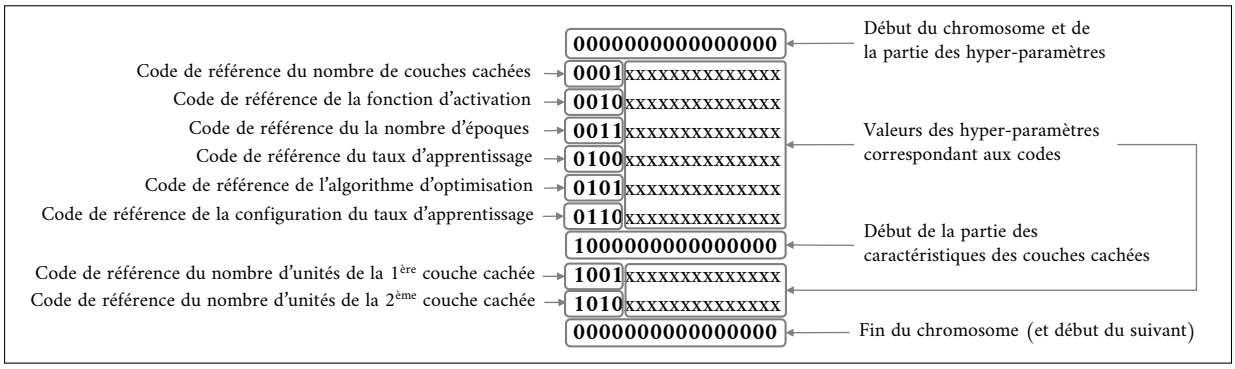


FIGURE 4.9 – Illustration de l'architecture d'encodage du génotype d'un réseau de neurones à deux couches cachées.

L'architecture de l'encodage des génotypes nous contraint à utiliser un "masque". En effet, nous souhaitons garder constantes toutes les lignes et les blocs de lignes qui nous permettront de retrouver les caractéristiques du réseau de neurones. Ce sont ces lignes et blocs de lignes là qui permettent d'appliquer les règles de transformation dont nous avons

parlé dans la section précédente (cf. la figure 4.6), et qui nous permettront de retrouver les phénotypes correspondants. Nous devons donc empêcher l'ensemble de ces lignes et blocs de lignes, mis en gras sur la figure 4.6, de toute modification, et ce tout au long du déroulement de l'algorithme génétique, que ce soit pendant les croisements ou les mutations. Ces blocs doivent être retrouvés intacts dans tous les génotypes générés automatiquement par l'algorithme génétique. Cette restriction sera mise en œuvre grâce au masque que nous utiliserons à chaque opération d'évolution. Le principe de fonctionnement de ce masque est simple : il est représenté par une liste de la même taille qu'un génotype, et contient des 0 dans les emplacements des bits ne devant pas être modifiés, et des 1 dans le reste des emplacements. Le génotype est en quelque sorte calqué sur le masque, afin de filtrer les changements acceptables de ceux qui ne le sont pas. Un exemple de ce processus, appliqué suite à une mutation, est décrit grâce aux figures 4.10 et 4.11.

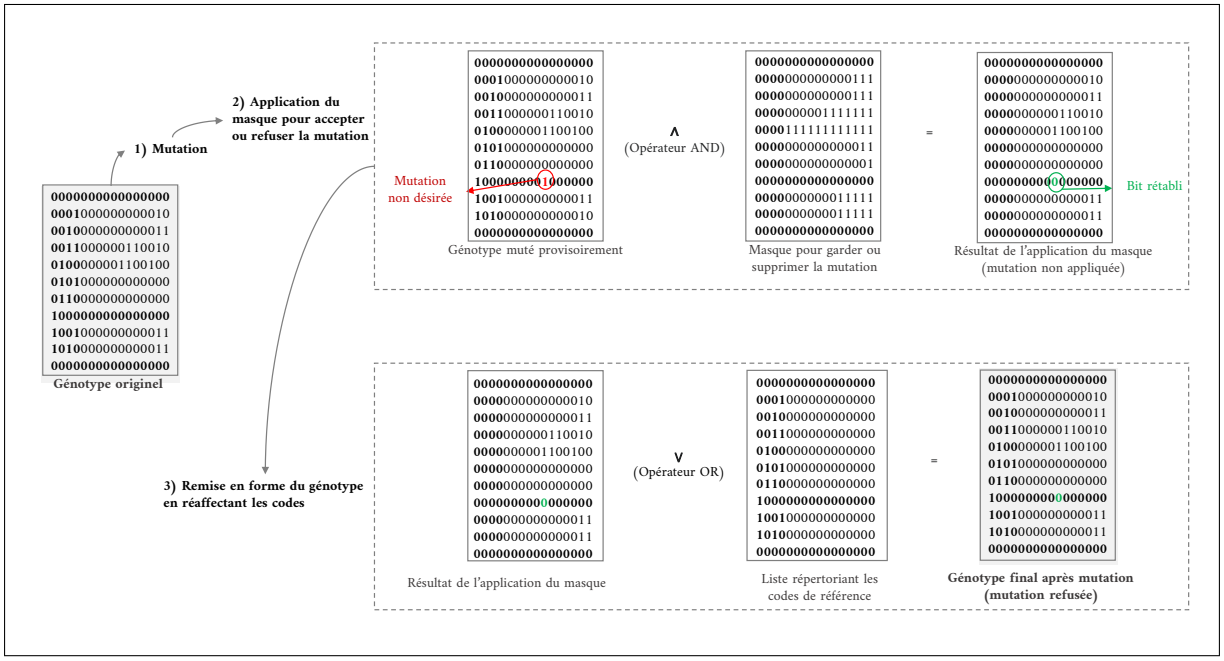


FIGURE 4.10 – Exemple d'une mutation refusée grâce à l'application du masque.

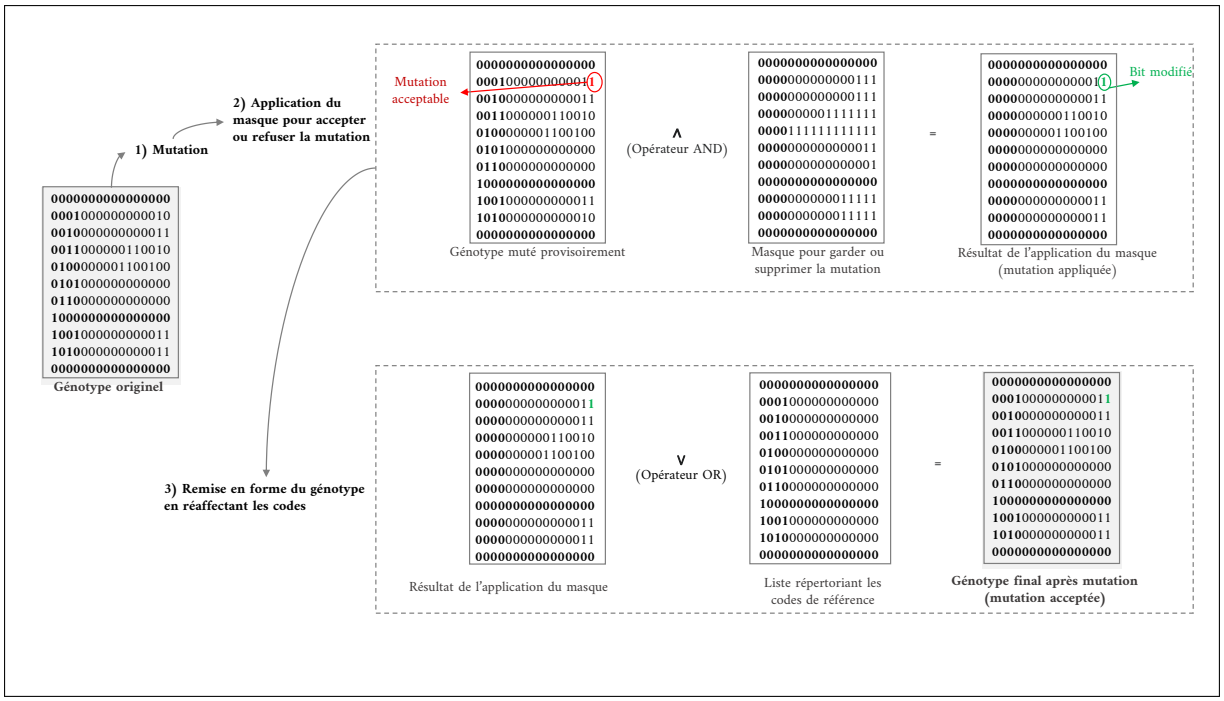


FIGURE 4.11 – Exemple d'une mutation acceptée après application du masque.

Ces deux figures mettent en avant la manière dont l'utilisation d'un même masque permet d'empêcher la création d'un génotype non viable. La définition de ce masque est étroitement liée aux caractéristiques que nous voulons faire évoluer dans les réseaux de neurones. Premièrement, comme pour les génotypes des réseaux de neurones, la deuxième partie du masque est extensible en fonction du nombre de couches du génotype sur lequel il sera appliqué. Ensuite, le choix des emplacements des 0 et des 1 n'y est pas laissé au hasard. En effet, comme nous l'avons expliqué précédemment, tous les chromosomes et les blocs de chromosomes servant à identifier les caractéristiques ne peuvent pas être modifiés. En outre, selon les valeurs ou les plages de valeurs autorisées pour chaque chromosome, certains bits des chromosomes servant à stocker les valeurs des caractéristiques ne peuvent pas prendre la valeur 1 afin de ne pas prendre des valeurs non autorisées, et sont donc inclues dans le masque, il s'agit en particulier des bits de gauches entraînant des valeurs élevées.

Les sections suivantes décrivent le déroulement de l'algorithme génétique étape par étape.

4.3.3.2 Génération de la population initiale

La population initiale est constituée d'un ensemble de génotypes représentant des réseaux de neurones MLP entièrement connectés. Les réseaux de neurones générés sont composés d'un maximum de cinq couches cachées. Le choix de cet intervalle est basé sur les recommandations disponibles dans la littérature (D. Yu & Seltzer, 2011). Il est vrai qu'en général, les réseaux de neurones à deux couches cachées sont suffisants pour approcher des fonctions de toute forme (Panchal, Ganatra, Kosta, & Panchal, 2011). Cependant, le nombre optimal de couches cachées dépend du nombre d'unités d'entrée et de sortie, de la taille de l'échantillon d'entrainement, de la quantité de bruit dans l'ensemble de données et de l'algorithme d'entrainement (Sheela & Deepa, 2013). Les paramètres

initiaux de l'algorithme génétique, à savoir la taille de la population et le nombre de générations, sont à spécifier par l'utilisateur qui souhaite construire le réseau de neurones. Ces paramètres conditionnent l'aboutissement à la solution optimale. Néanmoins, comme nous l'avons précédemment évoqué, l'étape de construction automatique d'un réseau de neurones destiné à effectuer une tâche particulière reste une étape ponctuelle à effectuer une fois en phase de développement, et à reconduire, au besoin, en cas de l'évolution du contexte (par exemple, en cas de disponibilité de nouvelles variables à intégrer dans la prédiction). Par conséquent, il revient à l'utilisateur de réitérer le déploiement de l'algorithme génétique avec des tailles de population et des nombres de générations différentes, jusqu'à obtention d'un réseau de neurones avec des performances satisfaisantes.

Dans notre proposition, afin d'éviter d'avoir beaucoup de réseaux de départ avec plus de deux couches cachées, et afin d'éviter de tomber dans un problème de sur-apprentissage, le nombre de couches cachées pour chaque réseau de neurones généré est défini aléatoirement selon la distribution de probabilité suivante, servant donc à faire une pondération : une probabilité de 0.2 pour que le réseau généré ait une couche cachée, 0.5 pour qu'il ait deux couches cachées, 0,1 pour qu'il ait trois couches cachées, 0.05 pour qu'il ait quatre couches cachées, et 0.05 pour qu'il ait cinq couches cachées. Ces probabilités ont été fixées compte tenu du fait que les réseaux de neurones comportant jusqu'à deux couches cachées sont capables d'approximer la plupart des problèmes complexes non linéaires (Karsoliya, 2012 ; Goodfellow, Bengio, & Courville, 2016).

Les fonctions d'activation des couches cachées de chaque réseau de neurones sont obtenues aléatoirement à partir d'une distribution discrète uniforme sur les éléments suivants : la fonction identité, la fonction Headvise, la sigmoïde, la tangente hyperbolique, l'unité de rectification linéaire (ReLU), softplus et identité courbée.

Concernant l'algorithme d'apprentissage, il est sélectionné de la même manière que la fonction d'activation, et ce parmi l'algorithme de descente de gradient stochastique (SGD), le quasi-Newton Broyden-Fletcher-Goldfarb-Shanno à mémoire limitée (LBFGS), et l'estimation du moment adaptatif (algorithme Adam). Un taux d'apprentissage compris entre 10^{-4} et 0,4 (Smith, 2018) est également défini pour les réseaux de neurones utilisant un algorithme d'apprentissage par descente de gradient stochastique (*i.e.* SGD ou Adam). Ce taux d'apprentissage peut être défini comme étant constant, ou bien adaptatif en restant constant tant que la perte diminue, et en diminuant dès que la perte ne diminue pas pendant deux epochs consécutifs.

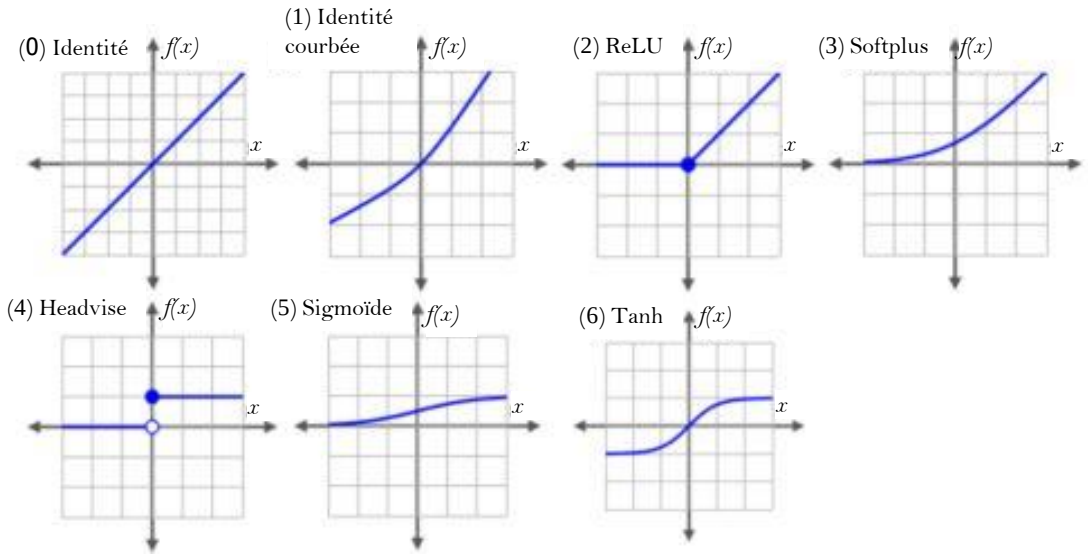
Le nombre d'epoch est choisi entre 200 et 1200 à partir d'une distribution discrète uniforme sur des multiples de 50. Concernant le nombre de neurones de chaque couche cachée, il fait l'objet de plusieurs travaux de recherche, dont certains tentent d'automatiser la recherche du nombre seuil à ne pas dépasser et à partir duquel le problème de sur-apprentissage surviendra. Dans notre proposition, c'est l'algorithme génétique qui fera évoluer le nombre de neurones par couches cachée. Cependant, afin de pouvoir orienter l'algorithme vers une bonne direction, nous suggérons de choisir ce nombre sur intervalle qui dépend de la taille de la couche d'entrée et de celle de la couche de sortie, selon les recommandations disponibles dans la littérature (Goodfellow et al., 2016 ; FronlineSolvers, 2020). Dans notre contexte, nous avons essentiellement besoin de construire des réseaux de neurones à sortie unique (*i.e.* un réseau de neurones par KPI). Généralement, il est

préférable que le nombre total des neurones cachés soit compris entre la taille de la couche d'entrée et N, qui est donné par (4.15), toujours selon les mêmes recommandations. Le terme 1 dans cette formule doit être substitué par le nombre de neurones de sortie s'il y en a plus d'une.

$$N = \frac{2}{3}.(1 + \text{Taille de la couche d'entrée}) \tag{4.15}$$

Le nombre de neurones par couche cachée sera donc choisi parmi l'ensemble suivant : $\left\{ \lfloor \frac{N}{P} \rfloor ; \ldots ; \lceil \frac{T_i}{P} \rceil \right\}$; où T_i, P dénotent respectivement la taille de la couche d'entrée, et le nombre de couches cachées. La notation $\lceil . \rceil$ et $\lfloor . \rfloor$ dénotent respectivement les opérations d'arrondissement vers le plus grand et le plus petit entier les plus proches.

Par ailleurs, afin de maintenir le trait de similarité (*i.e* hérédité) essentiel au bon fonctionnement des algorithmes génétiques, et que l'opération de croisement ait un sens, la fonction d'activation, considérée comme qualitative puisqu'il faut en choisir une parmi plusieurs, doit être codée de manière à ce que la fonction d'activation de la descendance hérite partiellement de certains aspects des parents. À cette fin, l'attribution des codes pour chaque fonction d'activation a été effectuée de manière à ce que le croisement génère une descendance dont la fonction d'activation est cohérente avec celles de ses parents. Par conséquent, leurs codes sont triés comme indiqué sur la figure 4.12 en fonction des spécificités caractérisant chacune des fonctions. Chaque fonction d'activation présente des similitudes avec ses voisins. L'utilité de ce classement est que lors d'une opération de croisement entre deux réseaux utilisant des fonctions d'activation différentes, le réseau descendant, issu de ce croisement, pourrait prendre soit la fonction d'activation de l'un de ses deux parents, soit une fonction dont les caractéristiques se rapprochent de celles des fonctions d'activation utilisées dans les deux parents, comme nous l'expliquerons dans la sous-section 4.3.3.4 destinée à expliquer les opérateurs d'évolution. On note qu'ici, nous parlons uniquement de la fonction d'activation des couches cachées, et non pas de celle de la couche de sortie. En effet, nous ne faisons pas évoluer la fonction d'activation de la couche de sortie, puisqu'elle dépend uniquement du type de la sortie attendue et de son intervalle, et doit être définie en amont et être la même pour tous les réseaux de neurones que nous faisons évoluer.



Fonction d'activation	Identité	Identité courbée	ReLU	SoftPlus	Headvise	Sigmoïde	Tanh
Intervalle	(-∞,+∞)	(-∞,+∞)	(0,+∞)	(0,+∞)	(0,1)	(0,1)	(0,1)
Dérivée monotone	Oui	Oui	Oui	Oui	Non	Non	Non
Identité à l'origine	Oui	Oui	Oui	Non	Non	Non	Oui
Monotone	Oui	Oui	Oui	Oui	Oui	Oui	Oui

FIGURE 4.12 – Caractéristiques et codes des fonctions d'activation.

Comme expliqué dans la sous-section précédente, tous les génotypes respectent un encodage binaire bien précis permettant de retrouver le phénotype correspondant. Ceci est rendu possible grâce à la définition d'un ensemble de règles de transformation adaptées à l'encodage que nous avons choisi, et décrites par la table 4.2. Pour la population initiale, les génotypes sont générés aléatoirement à partir d'un modèle respectant le masque dont nous avons discuté dans la sous-section précédente. Afin d'obtenir un génotype de la population initiale, ce modèle est modifié aléatoirement à plusieurs endroits, et les modifications effectuées doivent passer le test du masque pour être acceptées ou non, pour éviter que les chromosomes prennent des valeurs inattendues ou incohérentes, et s'assurer du maintien des chromosomes et parties de chromosomes permettant de retrouver les caractéristiques et de rendre la transformation en phénotype possible.

Dès que les génotypes de la population initiale obtenus , les mêmes règles de transformation sont suivies pour obtenir les phénotypes, aussi bien ceux de la population initiale que ceux des autres générations de l'algorithme. Par exemple, un chromosome commençant par 0001 en partant de la gauche sert à spécifier le nombre de couches cachées, et sa valeur est codée sur les trois premiers bits en partant de la droite. Un génotype à la fois est envoyé vers un fichier qui le transforme et qui retrouve ses caractéristiques grâce au règles décrites sur la table 4.2, puis qui passe ces caractéristiques en paramètres d'une fonction générique servant un entraîner les réseaux et les tester. Les caractéristiques retrouvées grâce aux transformations doivent pouvoir être reconnues par la fonction entraînant et testant le réseau, et doivent par conséquent être traduites en adéquation avec

les valeurs reconnaissables par la fonction d'entraînement et de test. Une fois le génotype transformé, et le phénotype correspondant entraîné et testé, les résultats sont renvoyés au premier fichier l'algorithme génétique, afin que ce dernier continue à effectuer les opérations d'évolution, et opère donc la sélection selon les résultats, puis le croisement et la mutation. La table 4.2 décrit les règles de transformation, à travers un exemple de transformation du génotype d'un réseau composé de deux couches cachées avec respectivement six et quatre neurones cachés, en utilisant une fonction d'activation sigmoïde, un algorithme d'optimisation SGD avec un taux d'apprentissage constant, et 500 epochs.

TABLE 4.2 – Description du génotype d'un réseau de neurones MLP.

Chromosome	Hyper-paramètre	Règle de transformation
0000000000000000	Début de la partie encodant les hyper-paramètre	
0001000000000010	Nombre de couches cachées	Transformation binaire en décimal.
0010000000000011	Fonction d'activation	Conversion de binaire en décimal, chaque fonction d'activation correspond à un numéro de code entier allant de 0 à 6, obtenu après la transformation.
0011000000110010	Nombre d'epochs	Conversion de binaire en décimal, le nombre décimal est ensuite multiplié par 10 et arrondi au multiple de 50 le plus proche.
0100000011010110	Taux d'apprentissage	Conversion de binaire en décimal, le nombre décimal est ensuite divisé par 10^4.
0101000000000001	Algorithme d'entraînement	Conversion de binaire en décimale, chaque algorithme d'entraînement correspond à un numéro de code entier allant de 1 à 3.
0110000000000000	Configuration du taux d'apprentissage	Conversion binaire en décimal, 0 correspond à un taux d'apprentissage constant et 1 à un taux d'apprentissage adaptatif.
1000000000000000	Début des couches cachées	
1001000000000110	Nombre de neurones de la 1^{ere} couche cachée	Conversion de binaire en décimal.
1010000000000100	Nombre de neurones de la 2^{eme} couche cachée	Conversion de binaire en décimal.

Le codage que nous avons défini nous permet d'outrepasser le problème de confusion qui implique que deux phénotypes différents aient le même génotype. Ce problème peut résulter, par exemple, du fait que deux réseaux de neurones différents aient des couches cachées de la même taille dans les deux réseaux, mais pas dans le même ordre. Par exemple, un réseau R_1 ayant une première couche cachée de 2 neurones et une deuxième

couche cachée de 3 neurones, et un réseau R_2 ayant une première couche cachée de 3 neurones et une deuxième couche cachée de 2 neurones. Ces deux réseaux auront des comportements différents et doivent par conséquent avoir des génotypes différents. Le codage des génotypes doit donc permettre de reconnaître les génotypes équivalents avant de pouvoir effectuer une opération de croisement, afin que celle ci soit efficace. Grâce aux blocs de chromosomes invariants référant chacun à un code de caractéristique, il est assuré que deux génotypes différents résultent en deux phénotypes différents à l'issu de la transformation, et que deux phénotypes différents sont représentés par deux génotypes différents.

4.3.3.3 Évaluation et sélection

À l'issue de la formation des génotypes, la transformation est opérée afin de retrouver les réseaux de neurones correspondants. Ces réseaux sont ensuite entraînés un par un, puis testés. Seuls les meilleurs, sont ensuite sélectionnés pour y appliquer les opérations d'évolution. La sélection des individus se fait en fonction de leurs performances de prédiction, afin de générer les individus de la génération suivante. Dans le cas d'une classification binaire, la performance d'un réseau de neurones peut être évaluée en calculant le F-score donné par (4.16), qui combine à la fois la précision et le rappel :

$$F_\beta = (1+\beta^2)\frac{Precision.Rappel}{\beta^2.Precision + Rappel} \tag{4.16}$$

où β est un coefficient de pondération du rappel par rapport à la précision, $\beta = 1$ signifie que la même importance est accordée à la précision et au rappel. La précision est calculée à partir de la formule (4.17), et le rappel à partir de (4.18).

$$Precision = \frac{VP}{VP + FP} \tag{4.17}$$

$$Rappel = \frac{VP}{VP + FN} \tag{4.18}$$

où VP, VN, FP, FN, dénotent respectivement les vrais positifs, les vrais négatifs, les faux positifs et les faux négatifs.

Dans notre contexte, nous privilégions le rappel par rapport à la précision, puisque dans notre contexte, nous souhaitons anticiper la prise de décision en avançant le moment de prise de conscience du problème. En effet, nous préférons sanctionner davantage les faux négatifs par rapport aux faux positifs quand il s'agit de prédire une déviation d'un KPI. En d'autres termes, nous préférons avoir une fausse alerte de future déviation, plutôt que de prédire une situation nominale lorsqu'une déviation aura effectivement lieu. Par conséquent, nous fixons β à la valeur de 1.5. Cependant, il faut noter que la pertinence de cette mesure est soumise à la condition de ne pas inverser la signification des classes. Ainsi, selon la perception que nous venons d'expliquer, le "positif" fait référence à la présence de déviation, et le "négatif" fait référence à l'absence de déviation.

En cas de prédiction de valeurs continues, la fonction d'évaluation à maximiser est le coefficient de détermination R^2, calculé par la formule (4.19).

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4.19}$$

où $\hat{y}_i$ désigne la valeur prédite, y_i désigne la valeur de sortie correcte, et $\bar{y}$ la moyenne des valeurs correctes de sortie.

Bien que ces mesures permettent d'évaluer les performances d'un réseau de neurones, la fonction d'évaluation de l'algorithme génétique ne doit pas se résumer au simple calcul du F-score en cas de classification, ni à celui du R^2 en cas de régression. En effet, nous rappelons que dans notre algorithme génétique, nous faisons évoluer la topologie et les hyper-paramètres liés à l'optimisation des poids, notamment grâce au choix de l'algorithme d'optimisation et de la fonction d'activation. Néanmoins, lors des entraînements respectifs de chaque réseau de neurones généré, les poids initiaux sont générés de façon aléatoire, ce qui implique que les entraînements des différents réseaux démarrent avec des poids initiaux différents. Cela peut entraîner un biais, à l'issue de l'entraînement et du test, lors de la sélection des meilleurs réseaux. Ce biais est dû au caractère stochastique du processus d'optimisation des poids, dont les initialisations aléatoires des poids sont à l'origine. Afin de pallier à cela et de rendre significative la comparaison des évaluations des réseaux, nous entraînons et testons plusieurs fois chaque réseau de la population. Ensuite, nous calculons, pour chaque réseau, la moyenne des F-scores ou des R^2, selon s'il s'agit d'un problème de classification ou de régression, issus des différents entraînements et tests d'un même réseau. Chaque réseau est entraîné 30 fois, puisqu'il est recommandé qu'un échantillon doit avoir au moins 30 observations pour qu'il soit statistiquement significatif[1] (*i.e.* pour que la distribution des variables aléatoires converge vers la loi normale, d'après le théorème central limite) (Hogg, Tanis, & Zimmerman, 1977 ; Kwak & Kim, 2017). Enfin, ce sont ces moyennes que nous utilisons pour comparer les performances des réseaux d'une population. Ce qui implique que pour les problèmes de classification, la fonction d'évatuation utilisée est donnée par (4.20), et celle utilisée pour les problèmes de régression est donnée par (4.21) :

$$f_{classification}(R) = \frac{1}{n}.\sum_{i=1}^{n} F_{\beta_i}(R) \tag{4.20}$$

$$f_{regression}(R) = \frac{1}{n}.\sum_{i=1}^{n} R_i^2(R) \tag{4.21}$$

où R dénote le réseau de neurones correspondant au génotype en cours d'évaluation, n dénote le nombre de fois où le réseau a été entraîné et testé, dont la valeur, dans notre cas, est fixée à 30.

À l'issue du calcul de la fonction d'évaluation de tous les réseaux d'une population, une sélection élitiste est appliquée, et ce en choisissant les k éléments maximisant la fonction d'évaluation, avec des probabilités proportionnelles à leurs adaptations (*i.e.* résultats de la fonction d'évaluation) ; k étant préalablement défini par l'utilisateur, de la même manière que le nombre de générations à produire. Les opérations de croisement sont ensuite appliquées aux individus sélectionnés, et donneront lieu à la population de la prochaine génération.

1. https ://sphweb.bumc.bu.edu/otlt/MPH-Modules/PH717-QuantCore/PH717-Module6-RandomError/PH717-Module6-RandomError11.html

4.3.3.4 Croisement et mutation

Suite à la sélection des n meilleurs individus de la génération i, les individus de la génération $i + 1$ seront obtenus grâce aux opérations d'évolution, à savoir le croisement et la mutation.

Dans le cadre de la construction de réseaux de neurones par algorithmes génétiques, la manière d'effectuer le croisement de façon à ce qu'il soit bénéfique est souvent remise en question (Ding, Li, Su, Yu, & Jin, 2013). La difficulté de trouver un encodage permettant de rendre les opérations de croisement utiles (García-Pedrajas, Ortiz-Boyer, & Hervás-Martínez, 2006 ; Ahmadizar et al., 2015) est telle que beaucoup d'applications de l'apprentissage des réseaux de neurones par algorithmes génétiques font abstraction de cette étape (Volna, 2010 ; Ahmadizar et al., 2015). Ceci est équivalent à l'implémentation d'un recuit simulé (Yao, 1993). Or, le pouvoir des algorithmes génétiques réside justement dans la combinaison du croisement et de la mutation, qui permettent respectivement l'exploitation des meilleures solutions, et l'exploration de l'espace de recherche (Spears & Anand, 1991). La raison du manque d'efficacité souvent observé suite au croisement entre réseaux de neurones relève de plusieurs aspects. Premièrement, il y a l'incompatibilité éventuelle des chromosomes. Par exemple, si un premier réseau de neurones utilise l'algorithme SGD pour la mise à jour des poids et spécifie donc le taux d'apprentissage à y appliquer, et qu'un deuxième réseau utilise l'algorithme LBFGS pour la mise à jour des poids, alors un croisement qui mixe le chromosome caractérisant l'algorithme d'optimisation et le chromosome caractérisant le taux d'apprentissage donnera lieu à un réseau non viable qui applique un taux d'apprentissage à un algorithme quasi-Newtonien (LBFGS) de mise à jour des poids. La deuxième difficulté relève de la redondance pouvant apparaitre dans certains encodages (Volna, 2010). En effet, deux réseaux de neurones partageant exactement la même architecture peuvent fournir des performances différentes, puisqu'ils peuvent utiliser des hyper-paramètres différents. Ce problème est souvent rencontré lorsque l'encodage choisi définit uniquement l'architecture du réseau. Enfin, la troisième difficulté relevée pour certains encodages, est également observée lorsque les réseaux élus pour le croisement ont les mêmes architectures et paramètres. En effet, lorsque l'encodage spécifie l'existence de connexions entre neurones sous forme de chromosome traduisant une matrice d'adjacence. Ceci peut rendre possible la formation de génotypes différents pour un même réseau si les neurones ayant des connexions équivalentes dans un réseau et dans l'autre sont placés à des endroits différents du chromosome (Hancock, 1992). Par ailleurs, la représentation distribuée caractérisant les réseaux de neurones réduit la probabilité que le croisement produise une descendance adaptée (Volna, 2010).

Concernant la première difficulté, elle peut être surmontée grâce aux chaînes de bits que nous avons incluses dans l'encodage proposé, et qui permettent de retrouver le chromosome de chaque caractéristique. En effet, une recherche, effectuée grâce à ces blocs d'identification permet d'empêcher la formation de génotype avec des chromosomes incompatibles lors du croisement, en référence à une liste où sont répertoriés tous les chromosomes incompatibles. Par ailleurs, l'encodage, que nous avons proposé dans la sous-section 4.3.3.1 de ce chapitre, prévoit également de surmonter la deuxième difficulté que nous avons citée dans le paragraphe précédent. En effet, cet encodage prend en compte les hyper-paramètres associés à chaque réseau. Par conséquent, grâce à cela, deux réseaux différents partageant la même architecture auront forcément des génotypes différents, puisqu'ils n'utiliseront pas les mêmes hyper-paramètres d'apprentissage, et auront par conséquent des chromo-

somes différents pour leurs hyper-paramètres respectifs. Aussi, la troisième difficulté n'est plus présente avec l'encodage que nous proposons. En effet, comme nous utilisons exclusivement des réseaux de type feed-forward complètement connectés, l'ordre des nœuds dans une même couche n'a pas d'importance. Nous avons donc proposé de remplacer l'utilisation de matrices d'adjacence souvent utilisée pour exprimer la topologie, par une représentation du nombre de neurones par couche, en utilisant un chromosome par couche, qui renseigne sur le nombre de neurones de la couche en question.

Enfin, en raison de la représentation distribuée des réseaux de neurones, nous interdisons les croisements entre les réseaux ayant des profondeurs différentes (*i.e.* des nombres différents de couche cachées). Par conséquent, une opération de croisement se déroule en deux temps : d'abord, la mise en "couples" des génotypes élus pour le croisement s'effectue en fonction du chromosome définissant le nombre de couches cachées (les génotypes sur lesquels le croisement est effectué doivent avoir en commun le même chromosome dénotant le nombre de couches cachées), le croisement peut ensuite avoir lieu entre les génotypes de chaque couple pour fournir deux nouveaux réseaux. Trois cas de figures peuvent alors se présenter :

— Les deux génotypes ont exactement les mêmes chromosomes dans la partie des caractéristiques des couches cachées. Dans ce cas, seuls sont croisés, un par un, les chromosomes de la partie des hyper-paramètres, hormis les chromosomes renseignant sur le nombre de couches cachées pour chacun des deux réseaux ;

— Les deux génotypes ont exactement les mêmes chromosomes dans la partie des hyper-paramètres. Dans ce cas, le croisement s'effectue uniquement sur les chromosomes, un par un, de la partie des caractéristiques des couches cachées ;

— Les deux génotypes diffèrent dans la partie des caractéristiques des couches cachées et dans la partie des hyper-paramètres. Dans ce cas, le croisement s'effectue sur tous les chromosomes, un par un, hormis le chromosome renseignant sur le nombre de couches cachées.

Les croisements entre les chromosomes s'effectuent selon les règles suivantes :

— Nombre de neurones d'une couche cachée : dans le génotype résultant, le chromosome codant le nombre de neurones pour la couche cachée en question aura une probabilité de 0.5 de garder le chromosome entier de l'un des deux parents, avec des chances égales (*i.e.* une probabilité de 0.25 de garder le chromosome entier codant le nombre de neurones dans la couche cachée en question du premier parent, et une probabilité de 0.25 de garder le chromosome entier du deuxième parent) ; et une probabilité de 0.5 d'effectuer un croisement arithmétique permettant d'obtenir un nombre de neurones correspondant à la moyenne arrondie des nombres de neurones de la couche cachée de chaque parent. Par exemple, si le premier parent contient 4 neurones pour la première couche cachée, et que le deuxième parent en contient 6, le descendant aura 25% de chances de contenir 4 neurones dans sa première couche cachée, 25% de chances d'en contenir 6, et 50% d'en contenir 5 ;

— Fonction d'activation : un croisement arithmétique est effectué en convertissant en décimal, puis en sélectionnant aléatoirement un entier positif dans l'intervalle $[c_1 - 0.5(c_2 - c_1), c_2 + 0.5(c_2 - c_1)]$ où c_1 et c_2 sont les plus petits et les plus grands codes des fonctions d'activation des parents. L'individu résultant du croisement a une probabilité de 0.5 d'avoir la fonction d'activation correspondant au code choisi,

et une probabilité de 0.5 d'avoir la fonction d'activation d'un de ses deux parents, avec des chances égales pour chaque parent. Cet intervalle permet d'obtenir un code compris entre les codes des fonctions d'activations des deux parents, mais aussi un code voisin au-delà de l'intervalle des codes des deux parents, c'est à dire à droite du plus grand code et à gauche du plus petit code. L'intervalle de sélection est réduit à l'intervalle des valeurs possibles dans le cas où son extension implique des codes non applicables ;

— Nombre d'epochs : le chromosome résultant a une probabilité de 0.5 de garder le chromosome entier de l'un des deux parents, avec des chances égales, et une probabilité de 0.5 d'effectuer un croisement arithmétique permettant d'obtenir un nombre d'epochs correspondant à la moyenne arrondie du nombre d'epochs de chaque parent ;

— Algorithme d'entraînement : le chromosomes résultant a une probabilité de 0.5 de reprendre le chromosome entier du premier parent, et une probabilité de 0.5 de reprendre le chromosome entier du deuxième parent ;

— Taux d'apprentissage : cette étape est effectuée après le croisement des chromosomes de l'algorithme d'entraînement. Si l'algorithme issu de ce croisement est le LBFGS, le croisement des chromosomes du taux d'apprentissage n'a pas lieu, et ce chromosome prend une chaîne de bits tous à 0, traduisant qu'il n'y a pas de taux d'apprentissage. Si l'algorithme issu du croisement est SGD ou Adam, le croisement des chromosomes du taux d'apprentissage a lieu, et le chromosome résultant a une probabilité de 0.5 de reprendre le chromosome entier de l'un des deux parents, avec des chances égales, et une probabilité de 0.5 d'effectuer un croisement arithmétique permettant d'obtenir un taux d'apprentissage correspondant à la moyenne des taux d'apprentissage de chaque parent dans le cas où les deux parents en ont. Dans le cas où un seul parent possède un taux d'apprentissage, le descendant reprend le même taux ;

— Configuration du taux d'apprentissage : cette étape est effectuée après le croisement des chromosomes du taux d'apprentissage. Si le chromosome issu de ce croisement est différent d'une chaîne de 0, le croisement des chromosomes de la configuration du taux d'apprentissage a lieu, et le chromosome résultant a une probabilité de 0.5 de reprendre la configuration du taux d'apprentissage du premier parent, et de 0.5 de reprendre celle du deuxième parent. Nous notons que lorsque nous parlons d'une chaîne de 0, nous faisons référence à la chaîne modifiable du chromosome (*i.e.* tout le chromosome, sauf les quatre premiers bits en commençant par la gauche).

Afin d'enrichir l'exploration, une mutation est appliquée aléatoirement, avec un taux de 0,005 à un chromosome parmi ceux autorisés à muter. Aussi, un chromosome est obligatoirement muté aléatoirement lorsque deux réseaux de neurones se distinguent uniquement par le chromosome définissant le nombre d'epochs. À l'issue de ces opérations, une nouvelle population est ainsi générée, elle est composée des réseaux de neurones les plus performants de la population précédente et de leur descendance. Le processus est réitéré jusqu'à ce que le critère d'arrêt soit atteint. Ainsi, le réseau de neurones à retenir est celui qui, à l'issue de toutes les itérations de l'algorithme génétique, maximise la fonction d'évaluation.

4.4 Classement des causes par le biais des paramètres finaux d'un réseau de neurones

Suite aux deux étapes décrites dans les deux sections précédentes, à savoir la construction d'une structure causale décrivant les liens causaux entre un KPI d'intérêt et les variables contextuelles, et la construction d'un réseau de neurones ayant pour tâche de prédire l'état ou la valeur futur(e) du KPI d'intérêt, nous allons maintenant nous positionner dans une situation ou une déviation du KPI est prédite. Le fait de disposer au préalable d'une structure causale nous permet de restreindre le spectre d'actions, et d'avoir une idée plus précise sur les actions préventives envisageables. Dans cette section, nous allons tenter d'organiser ce spectre d'actions, de manière à prioriser les actions selon les pertinences de leurs effets.

Dans le chapitre 2, nous avons établi que la proposition (4.22) est vraie :

$$C \text{ est une cause de } E \Longrightarrow \mathbb{P}(E|C) > \mathbb{P}(E) \tag{4.22}$$

Nous avons également montré que pour deux évènements E et C, la proposition (4.23) est vraie :

$$\mathbb{P}(E|C) > \mathbb{P}(E) \Leftrightarrow \mathbb{P}(E|C) > \mathbb{P}(E|\neg C) \tag{4.23}$$

À partir des assertions (4.22) et (4.23), nous pouvons déduire la proposition (4.24) :

$$C \text{ est une cause de } E \Longrightarrow \mathbb{P}(E|C) > \mathbb{P}(E|\neg C) \tag{4.24}$$

Ceci revient à dire que si nous connaissons la cause d'une future potentielle déviation, supprimer cette cause revient à amoindrir la probabilité de l'apparition de la déviation. Par conséquent, l'action sur les causes d'une déviation permettrait d'accroître la probabilité de l'éviter. Par ailleurs, nous avons également établi que la force de l'association entre un effet et ses causes est un attribut sous-jacent à l'aspect probabiliste qui caractérise la notion de la causalité. Dit autrement, si C_1 et C_2 sont deux causes différentes de l'effet E, et que la force d'association entre C_1 et E est supérieure à la force d'association de C_2 et E, alors $\mathbb{P}(E|\neg C_2) > \mathbb{P}(E|\neg C_1)$. Ceci veut dire que la suppression de la cause C_1 augmentera la probabilité d'éviter la déviation plus que la suppression de C_2 ne le ferait. Ainsi, classer les causes par ordre de force d'association permettrait d'accroître l'efficacité de l'action préventive engagée pour éviter la déviation prédite. Pour obtenir un tel classement, il faut d'abord pouvoir quantifier cette force d'association pour chacune des causes liées au KPI d'intérêt.

Comme précédemment mentionné, les tables de probabilités conditionnelles associées au réseau Bayésien permettent de capturer ces forces d'association. Cependant, il s'avère souvent difficile d'exploiter correctement les tables de probabilités conditionnelles lorsqu'il s'agit d'un problème complexe (Castellani, 2013), d'autant plus que la plupart des personnes ne peuvent pas interpréter les informations au-delà de quatre dimensions (Pollino & Henderson, 2010). Par ailleurs, le déroulement des analyses de sensibilité sur les différentes combinaisons des états des causes liées au KPI prennent du temps et requièrent un certain niveau d'expertise (Kjærulff & van der Gaag, 2013). De plus amples explications ont été données dans la sous-section 3.2.2 du chapitre 3.

Étant donné que pour chaque KPI d'intérêt, nous disposons d'un réseau de neurones capable de le prédire correctement, nous nous penchons vers l'exploitation des poids finaux de ce réseau, qui sont à l'origine de sa puissance. En effet, les poids finaux d'un réseau de neurones remplissant correctement sa tâche représentent les forces des connexions entre ses neurones, et mettent en évidence les forces d'association entre les entrées et la sortie (Han et al., 2015).

Plusieurs travaux se sont penchés sur l'extraction des importances respectives des entrées sur la ou les soties moyennant les réseaux de neurones. Les approches suivies par ces travaux peuvent être répertoriées en deux grandes catégories : les approches utilisant des analyses de sensibilités consistant à opérer des modifications sur les entrées et analyser les effets de ces modifications sur les sorties ; et les approches se basant essentiellement sur les paramètres intrinsèques du réseau de neurones (De Oña & Garrido, 2014). Dans notre proposition, nous utilisons les deux approches, mais dans deux buts différents : nous adoptons une approche basée essentiellement sur les poids des réseaux de neurones pour quantifier les forces d'associations et classer les causes, puis nous adoptant une analyse de sensibilité pour valider ce classement.

4.4.1 Exploitation des poids d'un réseau de neurones

La méthode la plus connue pour fournir des degrés d'importance aux entrées d'un réseau de neurones en adéquation avec ses sorties est celle proposée par Garson (Garson, 1991). Cette méthode propose de fournir des degrés d'influences relatives, calculés à partir de la formule (4.25) :

$$D_i = \frac{\sum_{j=1}^{m} \frac{|w_{ij}|.|v_{jk}|}{\sum_{i=1}^{n}|w_{ij}|}}{\sum_{i=1}^{n}\sum_{j=1}^{m} \frac{|w_{ij}|.|v_{jk}|}{\sum_{i=1}^{n}|w_{ij}|}} \tag{4.25}$$

Cette formule calcule l'importance d'une entrée i sur une sortie k si le réseau possède plusieurs sorties. n et m dénotent respectivement le nombre de neurones d'entrées (le nombre de variables en entrée), et le nombre de neurones cachés, w_{ij} désigne un poids connectant l'entrée x_i et le j^{eme} neurone caché, et v_{jk} désigne un poids connectant le j^{eme} neurone caché à la k^{eme} sortie. Dans notre contexte, nous utilisons principalement les réseaux de neurones à sortie unique : nous proposons de prévoir un réseau à sortie unique par KPI. La formule (4.25) peut donc être simplifiée en enlevant l'indice k qui symbolise la sortie pour laquelle nous voulons classer les entrées par force d'association.

Bien que cette formule ait pu fournir des classements exacts dans plusieurs applications (Zhang et al., 2018), elle souffre tout de même de quelques inconvénients qui remettent en question sa fiabilité (Olden & Jackson, 2002 ; Olden, Joy, & Death, 2004). Le premier inconvénient réside dans son instabilité, qui est due au caractère stochastique des poids finaux d'un réseau de neurones à un autre. Or, les forces d'associations entre les causes et les effets ont en réalité, dans la majorité des cas, un ordre constant. La deuxième limite réside dans son invalidité dans certains cas, notamment dans les cas où les poids des connexions reliant une entrée à une sortie se contrebalancent les uns les autres. Ceci est dû notamment à l'application de la valeur absolue sur tous les poids. En effet, dès lors qu'il y a un mélange de poids positifs et négatifs dans le réseau, cette formule peut devenir trompeuse.

4.4.2 Hypothèses de travail

Dans notre contexte, nous utilisons exclusivement les réseaux de neurones de type feed-forward, il est donc inutile de prendre en compte un même poids plus d'une fois, puisque l'erreur en sortie est retro-propagée dans un sens unique de la sortie vers les entrées. De plus, comme nous avons pu le voir dans la section précédente, tous les réseaux de neurones générés à partir de l'algorithme génétiques n'utilisent que des fonctions d'activation monotones. Nous soulignons donc ici, que nous n'avons pas pu garantir que la méthode que nous proposons soit applicable avec des réseaux qui utilisent des fonctions non monotones. Il en va de même pour l'algorithme d'optimisation des poids : les réseaux que nous utilisons peuvent utiliser l'algotithme SGD, Adam, ou LBFGS.

4.4.3 Méthode d'exploitation des poids

Dans un réseau de neurones, chaque neurone opère des calculs sur ses entrées, et le résultat de ces calculs représente une donnée d'entrée pour chaque neurone suivant lié au neurone en question. La figure 4.13 montre comment les entrées sont transformées dans un neurone pour fournir un résultat.

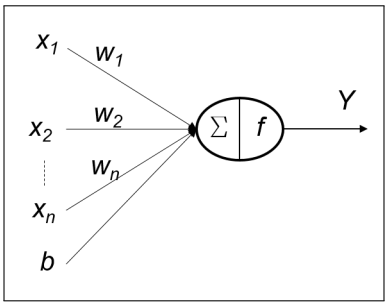


FIGURE 4.13 – Transformation des entrées en sortie dans un neurone.

Comme illustré sur la figure, une somme des valeurs des entrées de l'observation x, pondérées par les poids associées, est d'abord effectuée, puis additionnée à la valeur du biais associé à la couche, ensuite, une fonction d'activation est appliquée au résultat pour obtenir la sortie Y :

$$Y = f(\sum_{i=1}^{n} w_i x_i + b) \tag{4.26}$$

où x_i est la valeur de la i^{eme} entrée, w_i le poids connectant la $i^e me$ entrée à la sortie Y, et b le biais associé à la couche d'entrée.

Nous notons ici que nous ne parlons pas spécifiquement de la sortie du réseau de neurones, mais plus généralement de la sortie d'un neurone. Si par exemple, le réseau contient deux neurones cachées dans une même couche cachée, l'opération que nous venons de décrire sera appliquée séparément pour chacun des deux neurones, et les résultats constitueront les entrées pour les neurones la couche suivante (*i.e.* la couche de sortie, s'il s'agit d'un réseau à une couche cachée). L'erreur en sortie est ensuite calculée puis rétro-propagée sur les poids pour les mettre à jour. Les mêmes calculs sont ré-effectués au sein des neurones, ainsi que le calcul et la rétropagation de l'erreur, et la mise à jour des poids, jusqu'à ce que le réseau soit entraîné.

De même que la méthode de Garson, nous n'exploiterons pas la dans notre méthode la fonction d'activation. Cependant, dans notre proposition, ceci est conditionné par les hypothèses que nous avons émises dans la sous-section précédente, et plus précisément par la monotonie de la fonction d'activation utilisée. En effet, le fait que l'estimation de la force d'association d'une entrée à une sortie revienne à observer les écarts en sortie que provoquent les écarts en entrées, et le fait que la sortie globale ne puisse pas être correctement approchée sans fonction d'activation, doit persuader qu'il est indispensable d'intégrer la fonction d'activation pour évaluer correctement les forces d'associations. Cependant, l'utilisation d'une fonction d'activation monotone permet de s'affranchir de sa prise en compte, chose que nous faisons afin d'éviter d'ajouter à la méthode une complexité qui ne changera finalement pas le classement des forces d'associations. En effet, l'utilisation d'une fonction d'activation monotone croissante, comme c'est le cas ici, permet d'affirmer l'équivalence (4.27) :

$$\sum_{i=1}^{n} w_i x_i + b > \sum_{j=1}^{n} w_j x_j + b_1 \iff f(\sum_{i=1}^{n} w_i x_i + b) > f(\sum_{j=1}^{n} w_j x_j + b_1) \tag{4.27}$$

Par conséquent, la proposition (4.28) est également vraie :

$$\begin{aligned} &\sum_{j=1}^{n}(w_j x_j + b_1) - \sum_{i=1}^{n}(w_i x_i + b) > \sum_{k=1}^{n}(w_k x_k + b_2) - \sum_{i=1}^{n}(w_i x_i + b) \iff \\ &f(\sum_{j=1}^{n}(w_j x_j + b_1) - f(\sum_{i=1}^{n}(w_i x_i + b)) > f(\sum_{i=k}^{n}(w_k x_k + b_2)) - f(\sum_{j=1}^{n}(w_i x_i + b)) \end{aligned} \tag{4.28}$$

L'utilisation d'une fonction d'activation croissante implique l'obtention d'une dérivée positive sur tout le domaine de définition de la fonction. Par conséquent, l'évolution de la sortie d'un neurone dans le sens positif (respectivement négatif), veut systématiquement dire que la combinaison linéaire passée en paramètre de la fonction a évolué aussi dans le sens positif (respectivement négatif). Cependant, nous soulignons que nous tentons ici est de classer les causes par ordre d'importance, sans spécifier la direction de l'association. L'aide à la décision fournie à l'expert lui permettrait donc d'identifier sur quoi il faudrait agir pour en priorité, sans définir comment cette action devrait être conduite.

Par ailleurs, lors du calcul de la sortie d'un neurone, le terme du biais b peut être ajouté à chaque combinaison linéaire de deux façon différentes : il est soit traité comme un neurone supplémentaire ajouté à chaque chaque couche avec une valeur b et un poids de 1, ou bien ajouté de manière individuelle à chaque neurone. Ces deux méthodes se valent, et nous allons considérer la première pour expliquer comment nous pouvons nous s'affranchir du biais dans le calcul des forces d'association des entrées représentant les causes. En effet, nous souhaitons connaître l'influence que chaque entrée a sur la sortie. Or, nous avons mentionné préalablement que nous utilisons des réseaux de neurones feed-forward complètement connectés. Si le biais est perçu comme étant un neurone supplémentaire, nous ne souhaitons pas l'inclure dans le classement des causes puisqu'il ne s'agit pas d'une entrée réelle faisant partie des observations. En même temps, sa valeur intervient dans le calcul de la sortie. Au regard du type de réseaux que nous utilisons, nous pouvons donc répartir la valeur finale d'un biais d'une couche sur l'ensemble des neurones de la même couche en le divisant simplement par le nombre des neurones de la couche (Hornik, 1993).

Les poids finaux associés au neurones d'une couches se voient donc ajouter une même quantité, ce qui n'altère pas la participation de chaque neurones au calcul de la sortie si la fonction d'activation est monotone.

Intéressons nous à présent à la façon dont les poids finaux, qui seront à l'origine de l'évaluation des forces d'associations, ont été obtenus. Ces poids sont issus de plusieurs successions de calculs d'erreur, et la mise à jour des poids à chaque époch est basée sur la valeur de l'erreur que l'algorithme d'apprentissage tente de minimiser pendant l'entraînement en mettant à jour les poids. Pour chaque observation du jeu d'entraînement, l'erreur quadratique est donnée par (4.29) :

$$e(W) = (\hat{y}_i - y)^2 \tag{4.29}$$

où W est la matrice des poids de l'itération en cours, y la sortie attendue, et $\hat{y}$ la sortie estimée. Après l'entraînement du réseau avec la matrice de poids W, la fonction de perte associée à W est donnée par (4.30) :

$$E(W) = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y)^2 \tag{4.30}$$

où N est la taille de l'échantillon d'apprentissage, qui est constante.

Pendant l'apprentissage, l'objectif est de minimiser $(\hat{y}_i - y)^2$, où $\hat{y}_i = f\sum_{i=1}^{n}(w_i x_{ji})$, dans le cas d'un seul neurone, si l'on néglige le biais, avec n le nombre d'entrées, et x_{ji} la j^{me} valeur d'entrée de l'observation i. Ainsi, nous pouvons observer que la fonction d'erreur dépend à la fois des poids et des valeurs d'entrée. En effet, dans le cas d'un algorithme d'apprentissage standard utilisant la méthode de descente de gradient standard, les poids sont mis à jour est effectuée selon la règle donnée par 4.31 :

$$W_t = W_{t-1} - \lambda \nabla E(W_{t-1}) \tag{4.31}$$

où W_{t-1} est la matrice des poids de l'époch précédente, et λ est le taux d'apprentissage. Cette règle de mise à jour peut être substituée par une mise de la matrice des poids après chaque observation et non pas après le passage de toutes les observations dans une epoch, si l'apprentissage est opéré en ligne, comme c'est le cas pour les algorithme utilisant une descente de gradient stochastique, où la règle de mise à jour est la donnée par (4.32) :

$$W_t = W_{t-1} - \lambda \nabla e(W_{t-1}) \tag{4.32}$$

où W_{t-1} est la matrice des poids à l'issue de l'entraînement avec l'observation précédente. Concernant l'optimisation par méthodes quasi Newton, comme c'est le cas de l'algorithme LBFGS, les poids sont mis à jours suivant la règle donnée par (4.33) (Moré, 1978) :

$$W_t = W_{t-1} - \mu H_{t-1} \nabla E(W_{t-1}) \tag{4.33}$$

où H_{t-1} désigne une approximation de la matrice Hessienne de la fonction de coût, et μ le pas de la descente.

Une constatation évidente et commune à ces trois règles de mise à jour des poids, est que le gradient de l'erreur y est toujours impliqué. Cela qui signifie clairement que les valeurs des entrées y sont également impliquées, puisque la fonction d'erreur dépend de

$\hat{y}_i$, qui dépend, à son tour des valeurs des entrées. Il est donc intuitif de conclure que pour deux prédicteurs x_1 et x_2 en entrée du réseau, ayant tous les deux exactement la même force d'association à la sortie, si x_i varie dans une plage de valeurs très grandes, et x_2 dans une plage de valeurs très petites par rapport à x_1, les poids affectés à x1 auront une valeur absolue beaucoup plus petite que ceux affectés à x_2. Les plages de variation des deux matrices de poids finaux W_1 et W_2 seront inversement corrélées avec les plages de variation des entrées x_1 et x_2.

Bien que les données doivent être mises à l'échelle avant l'entraînement du réseau, il n'est pas garanti que cela soit suffisant pour établir que l'on peut négliger les valeurs des entrées lors de l'évaluation de leurs forces respectives d'association à la sortie. En effet, les méthodes usuelles couramment utilisées pour mettre les données à l'échelle présentent des limites [2]. La méthode min-max, donnée par (4.34) a l'inconvénient de ne pas être robuste face aux valeurs aberrantes.

$$x'_i = \frac{x_i - \min\limits_{x_1,\dots x_n}}{\max\limits_{x_1,\dots x_n} - \min\limits_{x_1,\dots x_n}} \tag{4.34}$$

où $\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ représentent toutes les valeurs de la variable x, et x'_i la nouvelle valeur de x_i après la mise à l'échelle. D'autres méthodes, telles que le z-score, ou la mise à l'échelle décimale, présentent également le même problème de sensibilité aux valeurs aberrantes, la méthode du z-score est aussi sensible à la distribution initiale des données puisque son utilisation suppose que les données suivent une distribution Gaussienne.

Par ailleurs, si la matrice des poids contient des petites valeurs avec des précisions qui font toutes la différence, les valeurs des entrées seront nécessaires pour ne pas perdre cette précision lors de l'évaluation des forces d'association. Par conséquent, il nous semble indispensable de les introduire lors d'une telle évaluation.

Afin de classer les entrées par ordre d'importance selon les poids finaux d'un réseau de neurones multi-couches, nous introduisons d'abord, à des fins d'explication, le réseau de neurones à une couche cachée comportant deux neurones, illustré sur la figure 4.14, puis nous généraliserons pour le cas d'un réseau de neurones à n couches cachées.

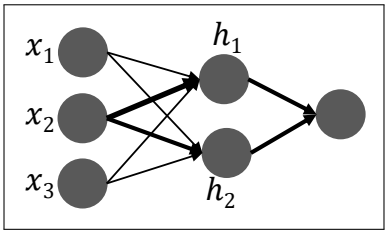


FIGURE 4.14 – Réseau de neurones à deux neurones cachés sur une même couche.

En nous intéressant de plus près à une entrée en particulier, en l'occurrence à l'entrée x_2 pour l'illustration, nous voyons qu'il participe aux calculs des sorties des deux neurones

2. https ://www.cs.ccu.edu.tw/ wylin/BA/Fusion_of_Biometrics_II.ppt

cachés h_1 et h_2, qui participent à leurs tours au calcul de la sortie finale. Néanmoins l'entrée x_2 n'est pas la seule à contribuer aux sorties de h_1 et h_2 : les x_1 et x_3 y participent également, avec des poids plus ou moins élevés, à chaque neurone cachés h_1 et h_2 qui, à leurs tours on des poids de connexion différents les reliant à la sortie. Par conséquent, nous commençons d'abord par quantifier les contributions des neurones cachés au neurone de la couche de sortie.

Soient w_{21} et w_{22} les poids liant l'entrée x_2 respectivement aux neurones h_1 et h_2, et w_{1s} et w_{2s} les poids liant respectivement les neurones h_1 et h_2 à la sortie. Nous pouvons par conséquent établir que les neurones h_1 et h_2 participent respectivement à $\frac{|w_{1s}|}{|w_{1s}|+|w_{2s}|}\%$ et à $\frac{|w_{2s}|}{|w_{1s}|+|w_{2s}|}\%$ au calcul de la sortie.

Par ailleurs, pour l'entrée h_1, les entrées x_1, x_2, et x_3 sont toutes impliquées dans le calcul de sa sortie, par conséquent, les participations respectives de chacune des trois entrées sont a participation de x_2 à la sortie h_1 est de $\frac{|w_{21}|}{|w_{11}|+|w_{21}|+|w_{31}|}\%$, $|w_{11}|$ et $|w_{31}|$ étant respectivement les valeurs absolues des poids liant x_1 et x_3 à h_1. Il en va de même pour la contribution de x_2 à la sortie h_2, qui est de $\frac{|w_{22}|}{|w_{12}|+|w_{22}|+|w_{32}|}\%$, $|w_{12}|$ et $|w_{32}|$ étant respectivement les valeurs absolues des poids liant x_1 et x_3 à h_2.

Par conséquent, nous pouvons faire une conclusion provisoire, selon laquelle la contribution de l'entrée x_2 serait de :

$$\frac{|w_{21}|}{|w_{11}|+|w_{21}|+|w_{31}|} * \frac{|w_{1s}|}{|w_{1s}|+|w_{2s}|} + \frac{|w_{22}|}{|w_{12}|+|w_{22}|+|w_{32}|} * \frac{|w_{2s}|}{|w_{1s}|+|w_{2s}|} \tag{4.35}$$

À ce stade, cette contribution ne peut évidemment pas être quantifiée en pourcentage de contribution à la sortie finale, puisque nous ne disposons pas encore des valeurs, analogues à celles que nous venons de décrire, associées aux deux autres entrées x_1 et x_3.

En outre, bien que dans notre périmètre, le sens de l'influence ne nous importe pas, l'utilisation des valeurs absolues peut engendrer une erreur d'interprétation. À titre d'illustration, si le neurone x_2 a un poids le connectant à h_1, et un poids positif le connectant à h_2, l'utilisation des valeurs absolues en dominateur fera écran du contre-balacement qui a lieu lors du calcul de la sortie. Nous proposons donc d'utiliser les valeurs signées en numérateur, et de garder les valeurs absolues en dénominateur.

Enfin, comme nous l'avons précédemment expliqué, nous souhaitons prendre en compte les données d'entrée afin d'avoir une évaluation plus exacte des forces d'association, surtout lorsque la matrice finale des poids contient des valeurs très petites dont la précision peut être perdue suite aux calculs des forces d'association. Pour cela, nous proposons d'utiliser des mesures résumant les valeurs d'entrée et leurs dispersions, de manière à pondérer les forces d'association. Nous utiliserons donc simplement le rapport entre la moyenne et l'écart-type, pour chacune des entrées. En effet, le rapport entre l'écart-type et la moyenne, connu sous le nom de coefficient de variation, donne une idée sur la dispersion des valeurs autour de la moyenne, permettant la comparaison entre deux distributions même lorsqu'elles n'évoluent pas dans la même échelle. Il est obtenu, pour une les valeurs

d'une entrée donnée, en calculant le rapport $\frac{\sigma}{\mu}$, où σ est l'écart-type, et μ la moyenne. Plus il est faible, mieux les données sont dispersées autour de la moyenne. Par conséquent, nous introduisons dans notre calcul des forces d'association, la grandeur inverse au coefficient de variation. Plus cette grandeur sera grande, mieux les valeurs seront dispersées autour de la moyenne, ce qui nous permettra de compenser les irrégularités introduites dans les valeurs des poids à cause des valeurs aberrantes.

Dans l'exemple que nous venons de décrire, nous pouvons faire une conclusion définitive sur la force d'association du nœud x_2 à la sortie finale, qui est calculée selon la formule 4.36.

$$c_{x_2} = \frac{\mu_{x_2}}{\sigma_{x_2}} * (\frac{w_{21}}{|w_{11}|+|w_{21}|+|w_{31}|} * \frac{w_{1s}}{|w_{1s}|+|w_{2s}|} + \frac{w_{22}}{|w_{12}|+|w_{22}|+|w_{32}|} * \frac{w_{2s}}{|w_{1s}|+|w_{2s}|}) \quad (4.36)$$

Nous pouvons à présent généraliser cette conclusion à un réseau de neurones à n couches cachées, et qui rempli nos hypothèses de travail, en donnant la formule 4.37.

$$d_i = |\frac{\mu_i}{\sigma_i}(\sum_{j=1}^{m} \frac{w_{ij}}{\sum_{i=1}^{n}|w_{ij}|} * \prod_{k=1}^{m} \frac{h_{jk}}{\sum_{j=1}^{m} h_{jk}})| \quad (4.37)$$

où i est l'entrée pour laquelle nous évaluons la force d'association à la sortie, n est le nombre des entrées du réseau, m désigne le nombre de neurones cachés connectés à l'entrée i, μ_i et σ_i désignent la moyenne et l'écart type des valeurs de l'entrée i.

La force d'association de chaque entrée peut ensuite être évaluée en utilisant la formule ci-dessus, et un classement peut ensuite être fourni sur la base des valeurs d_i de chaque variable i.

Afin de réduire l'instabilité d'un tel classement, nous proposons de réitérer les calculs en utilisant plusieurs réseaux entraînés et ayant de bonnes performances avec des écarts de précisions minimaux, pour ensuite effectuer une moyenne des forces d'association, et corriger le classement si besoin. En effet, cette instabilité peut provenir de deux sources différentes : la première source d'instabilité est intrinsèquement liée aux poids utilisés dans les calculs, plus précisément, de la performance du réseau qui conditionne la fiabilité de ses poids finaux. En effet, un réseau ayant une grande marge d'erreur fournira un classement qui reflètera moins fidèlement la réalité, vu que les poids utilisés pour obtenir ce classement ne modélisent pas bien le problème. Un classement issu d'un réseau de neurones avec une moindre marge d'erreur devrait à priori souffrir de moins d'instabilité lorsqu'il est comparé au classement réel. Il faut tout de même noter que, même en utilisant des poids finaux issus d'un réseaux ayant des performances élevées, le risque d'instabilité est moins probable mais reste toujours présent, en raison de la présence, bien que moindre, d'erreurs de prédiction. La deuxième source d'instabilité est liée au caractère stochastique du processus les ayant générés. Il s'agit plus précisément de la génération aléatoire des poids initiaux à l'origine de l'entraînement. Deux réseaux de neurones destinés à effectuer les mêmes tâches avec les mêmes données en entrée, ayant la même topologie et les mêmes hyper-paramètres, et fournissant des pouvoirs de prédictions très proches, peuvent avoir des poids finaux très différents, en raison de l'initialisation aléatoire. Ceci implique que le classement peut être différent entre ces deux réseaux, cependant il ne peut pas être

complètement différent, puisque les relativités entre les poids dans le premier réseau sont en principe similaires à celles des poids du deuxième réseau. L'instabilité pouvant être relevée est observable surtout lorsque les poids d'un réseau évoluent dans une échelle très grande par rapport aux poids de l'autre réseau : la perte de précision lors du calcul des forces d'association sur la base du réseau ayant des petits poids, perte qui n'a pas lieu lors du calcul basé sur les grands poids du deuxième réseau. C'est ce qui peut expliquer les quelques écarts de classement dus à ce type d'instabilité, qui serait moins probable si les pertes avaient eu lieu dans les deux calculs, ou n'avaient pas eu lieu dans les deux calculs.

Concernant le premier type d'instabilité, si le réseau a été mal entrainé, il est simplement inenvisageable de pouvoir retrouver un classement stable par rapport à la réalité. Néanmoins, l'instabilité entre deux classements issus de deux réseaux différents peut être réduite en utilisant des réseaux ayant un même pouvoir de prédiction : pour qu'une comparaison puisse être crédible, elle doit être faite sur des éléments comparables, or, des poids n'ayant pas appris les mêmes choses ne peuvent simplement pas rendre compte d'un même classement. Concernant ce deuxième type d'instabilité, nous devons donc réitérer les calculs des forces d'association plusieurs fois et en calculer la moyenne, afin de contrecarrer le caractère stochastique de l'apprentissage. Les réseaux entrainés doivent évidemment avoir les mêmes performances, comme nous venons de l'expliquer. Nous choisissons de réitérer le calcul au moins 30 fois, afin que les moyennes des forces d'associations de chaque variable soient statistiquement significatives.

Lors de la phase de développement de la méthode, il est nécessaire d'effectuer cette opération afin de pouvoir fournir un classement fiable, puisqu'il constituera l'aide à la décision offerte aux experts pour choisir une alternative parmi plusieurs. Nous pouvons à ce stade, et toujours lors de la phase de développement, faire appel à l'algorithme génétique, afin de pouvoir générer l'échantillon de réseaux de neurones dont nous avons besoin pour gérer l'instabilité du classement.

La figure 4.15 fournit un aperçu sur le processus de sélection des réseaux et la gestion des stabilités intrinsèques et stochastiques.

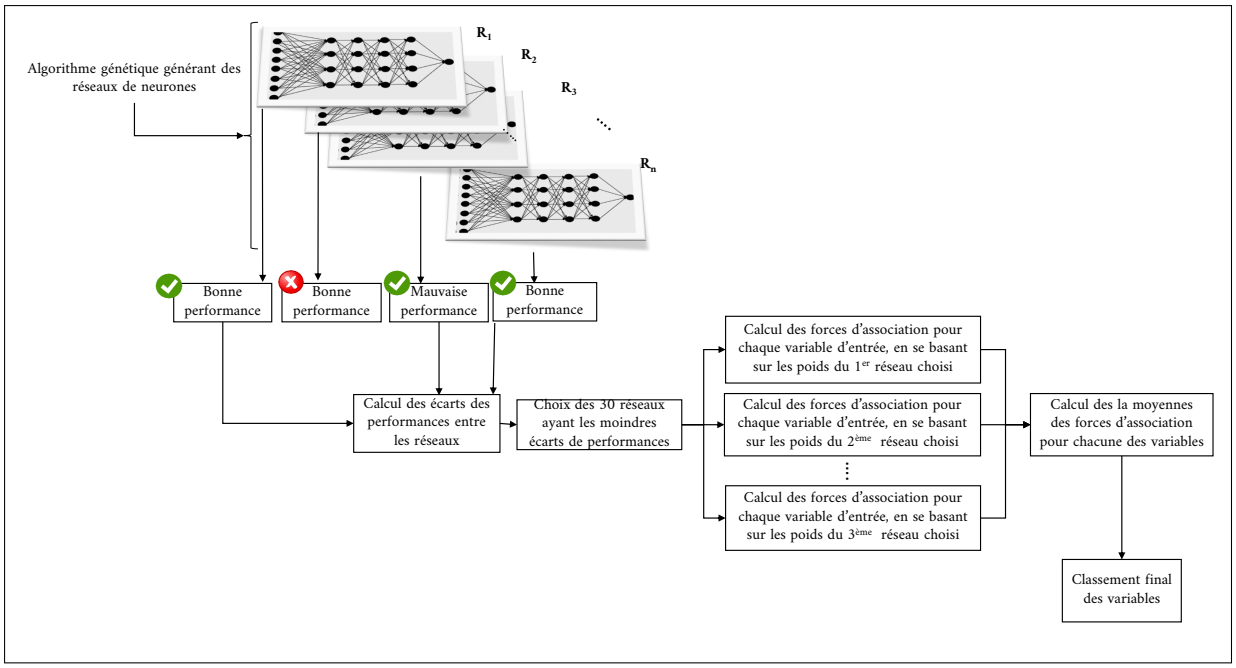


FIGURE 4.15 – Processus de sélection des réseaux et de gestion des stabilités intrinsèque et stochastique lors du classement des forces d'associations entre les entrées d'un réseaux de neurones et sa sortie.

4.5 Conclusion

Dans ce chapitre, nous avons détailler chacune des briques que nous avions introduit auparavant dans le chapitre 3, et qui constituent, ensemble l'approche que nous proposons. Nous avons d'abord commencé par décrire la brique de l'analyse causale, dont l'objectif est de construire, pour chaque KPI d'intérêt, un graphe décrivant les relations causales entre le KPI d'intérêt, et les données contextuelles. Cette brique répond à notre fonction F1, et son intérêt réside, d'un côté dans la réduction de la durée de prise de décision, et plus précisément le temps de recherche d'alternatives. D'un autre côté, l'automatisation de la construction des structures causales permet également de limiter les biais cognitifs liés à la prise de décision, et qui tirent leur origine de l'expérience de l'expert. En effet, cette brique permettra, outre le gain de temps, l'invalidation de certaines croyances et la découverte de nouvelles relations causales jusqu'alors inconnues. Nous avons fait le choix d'implémenter cette brique en utilisant une méthode de construction basée sur le calcul d'un score. Nous avons également identifié les limites du Hill Climbing classique avec ou sans Tabou, qu'elle tente de combler avant de détailler la version que nous proposons.

Afin de répondre aux deux fonctions F2 et F3, nous avons fait le choix d'utiliser les réseaux de neurones. La deuxième section de ce chapitre n'a pas répondu à une fonction principale à proprement parler, mais a plutôt servi d'outil de travail pour remplir les fonctions F2 et F3. L'objectif de cette section était de développer un algorithme génétique automatisant le processus expérimental de la construction des réseaux de neurones. Dans cette section, nous avons décrit nos motivations quant à l'utilisation des algorithmes génétiques à cette fin, avant de caractériser le problème et de détailler l'approche que nous avons adoptée pour encoder les réseaux de neurones, de manière à pouvoir retrouver un réseau par encodage, par le biais de blocs indiquant des codes de références pour chaque caractéristique que nous voulions faire évoluer. Enfin, nous avons ensuite décrit le déroule-

ment de l'algorithme génétique, ainsi que la manière d'évaluer les réseaux qu'il génère. Les réseaux résultant nous permettent par conséquent de répondre partiellement à la fonction F3, qui envisage de prédire les KPIs et d'alerter en cas de déviation, puisque les modèles de prédiction peuvent être construits grâce à la brique de construction automatique des réseaux de neurones. L'alerte des déviations nécessite, quant à elle, une utilisation de ces réseaux en temps réel.

Enfin, dans la troisième section de ce chapitre, nous avons décrit comment nous implémentons la troisième et dernière brique de notre proposition (que nous avions présentée en deuxième au chapitre 3), qui consiste à l'exploitation des poids finaux des réseaux de neurones afin de classer leurs entrées en adéquation avec les forces d'association respectives qu'elles entretiennent avec la sortie. Cette brique nous permet de répondre à la fonction F2, qui envisage de hiérarchiser les causes d'un KPI par ordre d'importance, et qui permet donc également de réduire le temps de prise de décision, mais en agissant cette fois sur la phase de sélection d'alternatives. Cette brique permet également, comme la première, de limiter les biais cognitifs liés à l'expérience ou à des croyances fallacieuses. Dans cette section, nous avons développé une formule permettant de rendre compte, pour chaque entrée, la force de son association à la sortie, par le biais des poids finaux d'un réseau de neurones entraîné, à condition que ce dernier remplisse les conditions émises sous forme d'hypothèses de travail dans cette même section. Dans cette formule, nous avons tenté de tenir compte des interactions des poids des couches cachées, en mettant également une attention sur les contrebalancements que les poids peuvent s'exercer mutuellement, ainsi que sur l'importance de prendre en compte les valeurs de chaque entrée du réseau. Enfin, nous avons décrit un processus permettant de contrecarrer les fluctuations du classement dues aux instabilités intrinsèques et stochastiques de tout classement basé sur les poids d'un réseau de neurones, afin de fournir un classement final fiable.

Chapitre 5

Application de la proposition et validation

Dans le présent chapitre, nous présenterons deux cas d'études permettant d'illustrer tout ou une partie des briques de notre proposition. Un premier cas d'étude académique complet, que nous avons construit en simulant des données à partir d'un étalon, introduira l'utilisation de toutes les briques de la proposition, une par une. Il permettra de comparer les résultats obtenus à des résultats attendus d'un côté, notamment pour la première et la deuxième brique. D'un autre côté, il permettra de comparer les résultats obtenus à ceux issus d'un autre outil commercial (le logiciel Bayesialab [1]), utilisant une autre technique mais ayant les mêmes objectifs, en particulier ceux de la troisième brique. Un deuxième cas d'étude, utilisera cette fois des données que nous n'avons pas simulées, mais pour lesquelles nous disposons quand même d'un étalon. Il permettra également d'un coté de vérifier la cohérence de nos résultats, en particulier ceux de la première brique l'apprentissage des réseaux Bayésien, puisque nous disposerons d'un étalon, et de confronter les résultats à ceux d'autres algorithmes ayant les mêmes ambitions que les nôtres.

5.1 Cas d'étude académique avec étalon simulé

5.1.1 Implémentation de la méthode proposée

Afin de mettre en œuvre la méthodologie proposée, et d'évaluer la cohérence de nos résultats, nous avons construit un cas d'étude académique, dans lequel le KPI d'intérêt est le TRS sur une opération d'assemblage. Ce cas d'étude est basé sur un échantillon représentatif de données sommaires que nous avons simulées à partir d'une structure causale, qui nous servira d'étalon.

Afin de générer les données, nous avons d'abord défini et construit manuellement une structure de réseau Bayésien, avant d'y associer des tables de probabilités conditionnelles et marginales. La structure que nous avons choisie pour cette structure de réseau est décrite grâce à la figure 5.1. Comme le montre cette figure, nous avons supposé que la déviation du TRS est causée par les trois variables discrètes suivantes, de manière directe : l'indisponibilité de l'outil de travail qui est causée par l'ordonnancement choisi, les

1. www.bayesia.com

ralentissements qui sont causés par la température ambiante dans l'atelier, et les pertes, qui sont causées par la qualité du guide d'assemblage. Le résultat attendu est une classification permettant de savoir si le TRS déviera ou non.

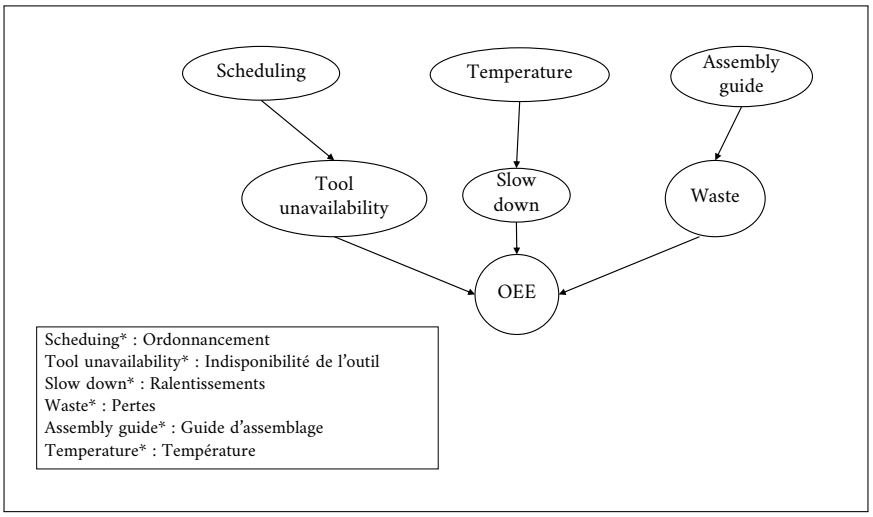


FIGURE 5.1 – Structure causale à partir de laquelle les données du cas d'étude ont été générées, et traduction des nœuds en légende.

L'échantillon de données étudié a été obtenu à partir d'une simulation de données basée sur ce réseau Bayésien causal, ainsi que les tables de probabilité marginales et conditionnelles correspondantes, qui sont décrites sur la figure 5.2. Afin d'associer des tables de probabilités à cette structure de réseau Bayésien causal, les bibliothèques `pgmpy` et `bnlearn` de python ont été utilisées. Ensuite, les données ont été simulées, en adéquation avec les liens causaux présents sur le graphes et les probabilités fixées, en utilisant les mêmes bibliothèques.

	Assembly guite (1)	Assembly guite (2)
Waste (no)	0.1	0.9
Waste (yes)	0.9	0.1

	Temperature (notmal)	Temperature (high)
Slow down (no)	0.1	0.9
Slow down (yes)	0.9	0.1

Assembly guide (1)	0.5
Assembly guide (2)	0.5

	Scheduling (1)	Scheduling (2)
Tool unavailability (no)	0.1	0.9
Tool unavailability (yes)	0.9	0.1

Temperatre (high)	0.5
Temperature (normal)	0.5

Scheduling (1)	0.5
Scheduling (2)	0.5

	Tool unavailability (no)	Tool unavailability (no)	Tool unavailability (no)	Tool unavailability (no)	Tool unavailability (yes)	Tool unavailability (yes)	Tool unavailability (yes)	Tool unavailability (yes)
	Waste (no)	Waste (no)	Waste (yes)	Waste (yes)	Waste (no)	Waste (no)	Waste (yes)	Waste (yes)
	Slow down (no)	Slow down (yes)	Slow down (no)	Slow down (yes)	Slow down (no)	Slow down (yes)	Slow down (no)	Slow down (yes)
OEE (normal)	1	0.86	0.9	0.2	0.7	0.15	0.1	0
OEE (Deviation)	0	0.14	0.1	0.8	0.3	0.85	0.9	1

FIGURE 5.2 – Tables de probabilités conditionnelles et marginales associées au graphe causal du TRS (OEE).

Afin de conduire notre méthode d'aide à la décision du début jusqu'à la fin, nous avons parcouru, dans l'ordre les trois briques composant notre proposition Nous avons donc commencé par déployer notre algorithme d'apprentissage des réseaux Bayésien, afin de vérifier

si la structure réelle à partir de laquelle les données ont été générées serait retrouvée correctement sans aucun apport de connaissances. Si tel n'est pas le le cas, nous procèderons au calcul du nombre de liens non retrouvés, inversés, ou ajoutés fallacieusement, afin de connaître à quel degré notre méthode retrouve correctement les causes. Aussi, si la structure causale n'est pas retrouvée, il serait pertinent de quantifier les connaissances minimum nécessaires afin de trouver correctement le reste de la structure.

Pour ce qui est du présent cas d'étude, toutes les relations ont été retrouvées comme illustré sur la figure 5.3, sans aucun apport d'informations préalable.

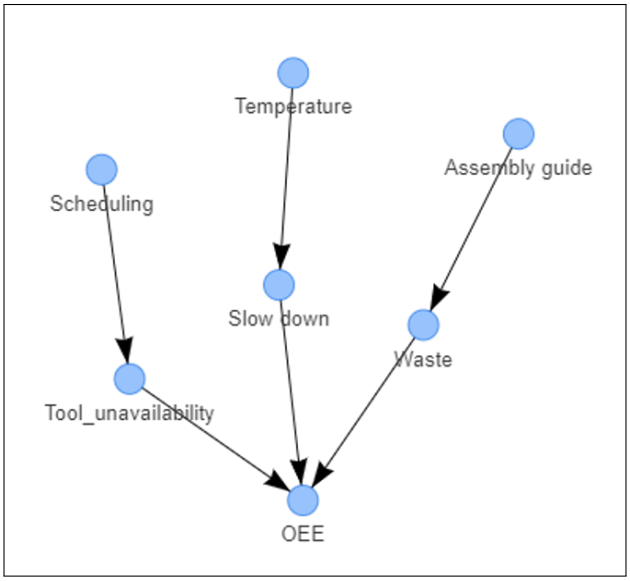


FIGURE 5.3 – Réseau Bayésien obtenu suite à l'apprentissage avec notre version du Hill Climbing

Suite à cela, nous avons déroulé l'algorithme génétique développé, afin d'obtenir un réseau ayant une haute performance prédictive pour la déviation du TRS. Pour ce cas d'étude, une taille de population de 28 individus nous a permis de retrouver des réseaux ayant des bons pouvoirs prédictifs, au bout de la 17^{eme} génération. Lors du passage d'une génération à la suivante, les sept réseaux ayant obtenu la meilleure moyenne des F-scores obtenus après 30 entraînements et tests, sont retenus pour produire les nouveaux individus introduits dans la génération suivante, et remplaçant les moins performants. La table 5.1 récapitule les F-scores des sept meilleurs réseaux de la population initiale d'un coté, et les F-scores des sept meilleurs réseaux de la population de la 17^{eme} génération de l'autre.

L'observation de cette table, nous permet d'établir que l'algorithme génétique a en effet été utile pour faire évoluer les réseaux de neurones, et substituer la démarche itérative expérimentale. Le meilleur réseau généré fournit des prédictions avec une certitude moyenne à hauteur de 0.8786%. La figure 5.4 illustre la matrice de confusion normalisée associée à ce réseau, et permettant de mesurer la qualité de la classification effectuée à l'issue de l'apprentissage.

Au vu des performances des réseaux finaux issus de d'algorithmes génétiques, nous pouvons affirmer que cet algorithme a, pour ce cas d'étude, rempli sa fonction, puisque nous estimons que ces performances sont bonnes. En effet, en regardant de plus près les tables de probabilités conditionnelles ayant été utilisées pour générer les données, nous

pouvons remarquer qu'une assertion selon laquelle le TRS déviera ou non, ne peut être certaine, en moyenne, qu'à hauteur de 0.87625%. Cette estimation peut être obtenue en faisant la moyenne des probabilités les plus élevées pour chacune des combinaisons des causes directes du TRS. Par conséquent, nous estimons que le réseau de neurones généré a atteint des performances aussi élevées que possible, puisqu'il ne peut pas outrepasser la certitude contenue dans les données sur lesquelles il a appris, et quand c'est le cas, l'écart ne peut être que minime et dû au hasard.

TABLE 5.1 – Évolution des performances des réseaux de neurones entre la première et la 17^{eme} génération.

Fitness de la première population	Fitness de la quinzième population
0.48836666666666667	0.8785666666666665
0.5108666666666667	0.8686666666666667
0.4902666666666667	0.8745333333333334
0.48673333333333335	0.87648
0.7764666666666666	0.8757333333333334
0.6852	0.8738666666666667
0.7964666666666666	0.8782333333333333

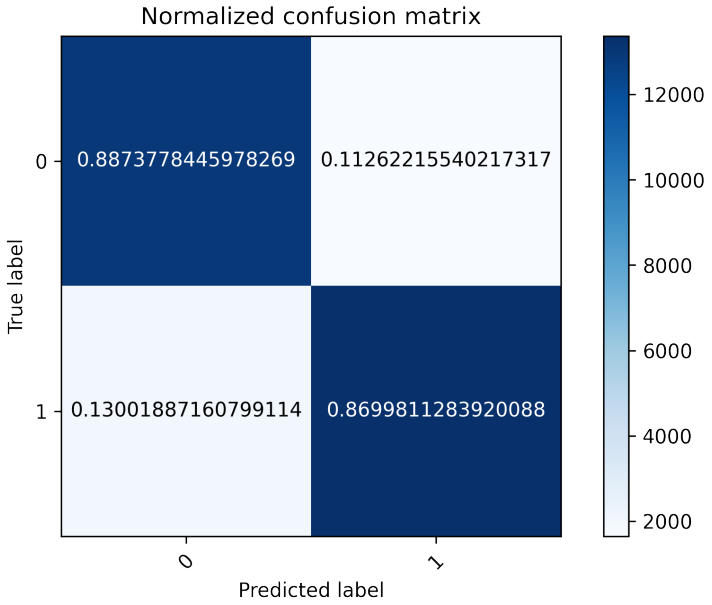


FIGURE 5.4 – Matrice de confusion normalisée associée au meilleur réseau généré par l'algorithme génétique.

Passons à présent à l'évaluation de la troisième brique de la proposition sur ce même cas d'étude. Nous notons que pour fournir le classement des entrées du réseau de neurones tout en palliant à l'instabilité intrinsèque et stochastique, nous en avons généré une

TABLE 5.2 – Classement général des forces d'association des entrées au TRS.

Classement	Valeur de la force d'association	Entrée
#1	16.945206257137187	Indisponibilité de l'outil
#2	10.060149902028803	Pertes
#3	9.808118368056034	Ralentissements
#4	9.500956633913196	Température
#5	9.232680220598699	Guide d'assemblage
#6	9.14470575536555	Ordonnancement

TABLE 5.3 – Classement général des forces d'association des entrées au TRS en utilisant des réseaux différents de ceux de la première itération.

Classement	Valeur de la force d'association	Entrée
#1	14.600708117842089	Indisponibilité de l'outil
#2	12.892838321870572	Pertes
#3	11.623951703243291	Ralentissements
#4	8.885434822003866	Température
#5	8.862161240203546	Guide d'assemblage
#6	8.805144359822698	Ordonnancement

trentaine grâce à l'algorithme génétique, tous ayant des performances avec des écarts inférieurs à 10^{-2}. Le classement général obtenu à partir des calculs des forces d'associations des six entrées confondues, indépendamment de leurs ordres de liaisons au TRS (*i.e.* s'il s'agit de causes directes ou non), est décrit sur la table 5.2.

Nous avons réitéré une seconde fois ce même processus, que nous avons détaillé dans la sous-section 4.4.3 du chapitre 4. Le classement résultant est décrit par la table 5.3, qui reporte le classement issu d'un ensemble de réseaux relativement moins performants que ceux qui ont permis d'obtenir le premier classement. Le classement est tout de même resté intact, en raison des bonnes performances des réseaux générés, dans l'absolu. Ceci dit, si une déviation du TRS est prévue durant la phase d'utilisation de la méthode que nous proposons, il serait préférable de se concentrer sur l'analyse de l'indisponibilité de l'outil, si nous ne considérons que les causes directes.

En effet, l'analyse des forces d'association doit prendre en compte l'ordre du lien de causalité d'une cause par rapport au KPI d'intérêt (*i.e.* s'il s'agit d'une cause directe, indirecte d'ordre 1 ou 2, etc.). Dans ce cas d'étude, l'*indisponibilité de l'outil*, les

pertes et les *ralentissements* sont les causes directes, tandis que *le guide d'assemblage*, l'*ordonnancement* et la *température* sont au premier ordre des causes indirectes. Ces niveaux, ou ordres des liaisons causales, sont issus de la structure causale apprise lors du déploiement de la première brique de la proposition, et ne doivent pas être négligés lors de l'interprétation des résultats, sauf si l'on s'intéresse uniquement aux causes directes : leurs classements respectifs entre elles sont les seuls à pouvoir être interprétés directement du classement général que nous proposons. Cependant, lors de l'analyse du classement d'une cause indirecte, il est indispensable de prendre en compte les classements des causes directes et des causes indirectes descendantes qui y sont liées. En effet, les forces d'association donnent une idées sur la manière dont une entrée influe la sortie. Néanmoins, la manière dont ces forces ont été obtenues considère toutes les variables comme étant toutes au même niveau (*i.e.* les six variables ont été passées à la fois au réseau de neurones).

Avant de présenter l'analyse des causes indirectes et directes confondues, qui sera décrite dans la section suivante, nous nous intéressons dans un premier temps au classement des causes appartenant à un même niveau, afin de voir sa cohérence. Pour illustrer les influences des causes directes sur le TRS, une série de quatre réseaux de neurones différents a été construite, afin d'évaluer l'impact de la modification d'une cause directe sur le TRS. Les performances des réseaux de neurones constituant cette série ont été comparés en utilisant les aires sous les courbes ROC de chaque réseau de neurones (Receiver Operating Characteristic) (Fawcett, 2006). L'utilisation de la courbe ROC est une technique permettant de visualiser et de sélectionner des classificateurs en fonction de leurs performances. La performance est évaluée en calculant l'aire sous la courbe (AUC). L'AUC est une mesure de performance très largement utilisée pour les règles de classification et de diagnostic (Airola, Pahikkala, Waegeman, De Baets, & Salakoski, 2011), elle est particulièrement utile lorsque les courbes ROC se croisent.

La manière d'évaluer les influences des causes directes sur les TRS reprend le principe d'une analyse de sensibilité, et est effectuée comme suit : un premier réseau de neurones a été construit en utilisant toutes les causes directes. Les trois autres réseaux de neurones ont été construits en omettant, successivement et pour chacun des réseau, une cause différente parmi les trois causes directes initialement utilisées en entrée du réseau initial. Ensuite, nous avons comparé les performances des prédictions exécutées en superposant leurs courbes ROC, et les performances de chaque prédiction ont été évaluées en calculant les AUC. D'après les courbes ROC de la figure 5.5, nous pouvons clairement voir que lorsque nous modifions la cause directe la plus influente, *i.e* l'*indisponibilité de l'outil*, la prédiction est plus modifiée que lorsque nous modifions le *ralentissement* ou les *pertes*.

En ce qui concerne la comparaison des causes indirectes entre elles, il faut tenir compte de leurs scores dans les nœuds descendants. En effet, en considérant l'analyse des scores donnés par la table 5.2, et en ne tenant compte que des scores des causes directes, nous pouvons conclure que l'*indisponibilité de l'outil* a une contribution de 46% sur la prédiction de l'état du TRS, le gaspillage y contribue à 27,3%, et le ralentissement à 26,7%. Les scores des causes indirectes de premier niveau doivent tenir compte de ces contributions, puisque leurs effets directs respectifs, qui sont également les causes directes du TRS, ne contribuent pas de la même façon à leur effet final qu'est le TRS. Par conséquent, ces contributions indirectes doivent être pondérées en fonction de leurs contributions aux causes de niveau inférieur. Par conséquent, lors de l'évaluation du classement des causes indirectes de pre-

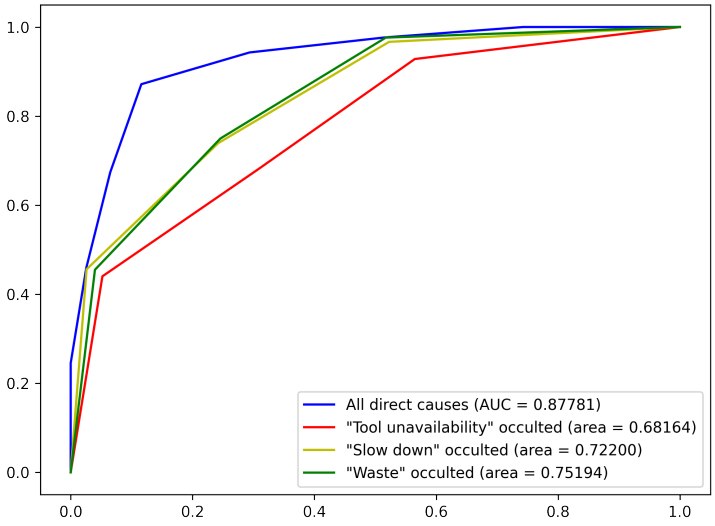


FIGURE 5.5 – Courbe ROC associée au réseau prenant toutes les causes en entrée, et courbes ROC associées aux réseaux omettant chacun une cause directe différente

mier niveau entre elles, les scores à comparer deviennent les suivants : $9.14 \times 0.46 =$**4.2044** pour l'*ordonnancement*, puisqu'il est le nœud parent de l'*indisponibilité des outils*, qui elle même, contribue à hauteur de 46% à l'effet final ; de la même manière, le score à utiliser lors de la comparaison de la force d'association de la cause *guide d'assemblage*, avec celles des autres causes indirectes du même niveau, sera de $9.23 \times 0.27 =$ **2.49**, puisqu'il est le nœud parent du gaspillage, enfin le score de comparaison, dans cette même situation, est de $9.5 \times 0.26 =$ **2.47** pour la *température*, puisqu'elle représente le nœud parent du *ralentissement*. Cela place l'*ordonnancement* en tête du classement des causes indirectes, et la *température* en fin de classement.

5.1.2 Comparaison de la hiérarchisation des causes avec une analyse de sensibilité Bayésienne.

L'objectif de cette section est de comparer le classement des forces d'associations obtenues grâce à l'exploitation des poids finaux d'un réseau de neurones ayant un bon pouvoir de prédiction, avec les pourcentages des contributions résultant de l'analyse de sensibilité Bayésienne. Nous cherchons donc à évaluer la cohérence des résultats de la proposition, en particulier de la troisième brique, en les comparant à une autre méthode ayant le même objectif que notre troisième brique. La méthode proposée sera confrontée avec les résultats de l'analyse de sensibilité qui est basée sur les tables de probabilités marginales et conditionnelles associées au réseau bayésien original. Cette section met également en évidence les difficultés et les erreurs liées à l'interprétation des tables de probabilités conditionnelles associées à un réseau Bayésien.

Le classement général a déjà été donné par la table 5.2. Deux autres réseaux de neurones supplémentaires ont été construits : un premier avec les causes directes uniquement, et un second avec les causes indirectes uniquement. Les tables 5.4 et 5.5 rendent compte des classements des causes directes et indirectes considérées séparément, selon les deux nouveaux réseaux de neurones que nous venons de décrire. Comme le montrent ces tables, les classements sont cohérents avec l'analyse que nous avons faite dans la section précé-

dente.

TABLE 5.4 – Classement des causes directes issu d'un réseau de neurones n'ayant pour entrées que les causes directes

Classement	Force d'association	Cause directe
#1	14.196932965129683	Indisponibilité de l'outil
#2	12.500569730345381	Pertes
#3	11.050863041452459	Ralentissements

TABLE 5.5 – Classement des causes directes issu d'un réseau de neurones n'ayant pour entrées que les causes indirectes

Classement	Force d'association	Cause indirecte
#1	12.290120535676731	Ordonnancement
#2	9.69051603768232	Guide d'assemblage
#3	9.6223850678669	Température

Afin de conclure correctement sur le classement des forces d'association des causes basé sur les tables de probabilités sans tomber dans des interprétations erronées, une analyse de sensibilité doit être menée. Les contributions résultantes serviront de référence pour vérifier la cohérence des classements obtenus par notre approche. Le logiciel Bayesialab contient la fonctionnalité d'analyse de sensibilité et peut nous fournir une analyse complète qui répond à la problématique du classement des facteurs d'influence. Comme notre objectif est de pouvoir engager des actions sur la cause racine si possible, nous commençons par évaluer les contributions directes de sur le TRS, avant d'analyser les contributions des causes de la cause directe la plus influente.

Considérons dans un premier temps les causes directes et indirectes séparément. Les contributions directes sur l'état du TRS, selon l'analyse de sensibilité Bayesialab sont présentées dans la figure 5.6. La dernière colonne de la figure 5.6 montre que l'i*ndisponibilité de l'outil* a plus d'impact sur le TRS (40,08%) que les *pertes* et les *ralentissements*, qui ont respectivement des contributions de 30,16% et 29,75%.

Direct Effects on Target OEE						
Node	Prior Value/Mean	Standardized Direct Effect	Direct Effect	Elasticity	Overall Elasticity	Contribution
Tool unavailability	0.4991	0.5026	0.5025	50.2469%	50.2469%	40.0834%
Waste	0.4994	0.3781	0.3780	37.8045%	37.8045%	30.1578%
Slow down	0.4983	0.3731	0.3730	37.3047%	37.3047%	29.7589%

FIGURE 5.6 – Contributions directes sur l'État du TRS selon Bayesialab

Afin de comparer de manière significative les résultats de Bayesialab à ceux fournis par l'approche de notre proposition, les classements doivent être exprimés en pourcentages.

Les pourcentages des contributions des causes directes selon l'approche de classement que nous proposons peuvent être dérivés de la table 5.4, en divisant le score de chaque cause par la somme des scores des causes directes. Par conséquent, les contributions obtenues des causes directes sont les suivantes : l'*indisponibilité de l'outil* contribue à 37,61%, les *pertes* contribuent à 33,11%, et les ralentissements à 29,38%. Ces résultats sont cohérents avec les résultats fournis par l'analyse Bayesialab. D'abord, le classement est le même, ensuite les pourcentages de contributions fournis selon notre approche sont proches de ceux fournis selon l'analyse Bayesialab, même s'ils ne sont pas les mêmes. Ces écarts sont dus à l'instabilité intrinsèque due aux poids finaux sur lesquels les classements, ces poids sont issus issus d'un réseau de neurone ayant une marge d'erreur qui s'approche de 14%. En effet, la méthode proposée repose sur les poids finaux d'un réseau de neurones, mais nous devons garder à l'esprit que ce réseau de neurones n'est pas précis à 100 %, il a donc appris des erreurs dans la phase d'apprentissage. De plus, le réseau a pas basé le calcul de ses poids finaux en s'appuyant uniquement sur le jeu de données d'entraînement, puisque lors de l'entraînement, les mises à jours des poids n'ont pas tenu compte des observations présentes dans le jeu de test, contrairement à l'analyse de Bayesialab, qui est basée sur l'ensemble du jeu de données.

De la même manière, les influences des causes indirectes peuvent être prises séparément afin de comparer leurs influences. Les contributions indirectes sur l'état du TRS, selon l'analyse de sensibilité Bayesialab sont présentées dans la figure 5.7. La dernière colonne de la figure 5.7 montre que l'ordonnancement a plus d'impact sur le TRS (39,99%) que le guide d'assemblage et la température (respectivement 30,19% et 29,81%). Ces contributions sont conformes au classement donné par la table 5.5, qui attribue des contributions de 38,9% pour l'*ordonnancement*, 30,65% pour le *guide d'assemblage* et 30,45% pour la *température*. Ces contributions ont été calculées en divisant les forces d'association de chaque cause indirecte par la somme des forces d'association de toutes les causes indirectes, en se basant sur la table 5.5, qui relate les forces d'association issues d'un réseau de neurones prédisant uniquement avec les causes indirectes. Les résultats du classement de Baysialab sont également cohérents avec l'analyse menée dans le dernier paragraphe de la section précédente.

Direct Effects on Target OEE						
Node	Prior Value/Mean	Standardized Direct Effect	Direct Effect	Elasticity	Overall Elasticity	Contribution
Scheduling	0.5010	-0.4015	-0.4014	-40.1439%	-40.1439%	39.9973%
Assembly guide	0.4999	-0.3031	-0.3030	-30.3037%	-30.3037%	30.1930%
Temperature	0.5019	-0.2993	-0.2992	-29.9192%	-29.9192%	29.8097%

FIGURE 5.7 – Contributions indirectes de l'état du TRS selon Bayesialab

À ce stade de l'analyse, nous avons uniquement traité séparément les causes directes et les causes indirectes. Considérons maintenant le classement général des contributions de toutes les causes sur l'état du TRS, quel que soit leurs ordres. L'analyse du classement général est importante pour la phase de sélection des alternatives dans les processus déci-

Total Effects on Target OEE									
Node	Prior Value/Mean	Standardized Total Effects	Total Effects	G-test	df	p-value	G-test (Data)	df (Data)	p-value (Data)
Tool unavailability	0.4991	0.5026	0.5025	79,353.4513	1	0.0000%	79,552.5125	1	0.0000%
Scheduling	0.5010	-0.4017	-0.4016	49,795.2717	1	0.0000%	49,777.0789	1	0.0000%
Waste	0.4994	0.3781	0.3780	43,984.1295	1	0.0000%	43,901.9330	1	0.0000%
Slow down	0.4983	0.3731	0.3730	42,799.9576	1	0.0000%	42,882.2319	1	0.0000%
Assembly guide	0.4999	-0.3020	-0.3020	27,801.9960	1	0.0000%	27,810.6592	1	0.0000%
Temperature	0.5019	-0.2983	-0.2982	27,106.6241	1	0.0000%	27,043.3632	1	0.0000%

FIGURE 5.8 – Effets de toutes les causes sur le TRS selon Bayesialabb

sionnels. En effet, les décideurs peuvent vouloir comparer des causes de niveaux différents, notamment lorsqu'ils jugent qu'agir sur une cause racine donnée nécessite la mobilisation de beaucoup d'efforts ou de ressources, et qu'il serait plus intéressant de comparer, par exemple son descendant avec une autre cause racine. La figure 5.8 montre, dans la quatrième colonne, les effets respectifs de chaque cause, tous niveaux confondus, sur le TRS. En comparant avec notre méthodologie, nous considérons la valeur absolue de ces effets, puisque, comme nous l'avons décrit dans les chapitres précédents, nous ne traitons pas dans notre périmètre les directions d'influence des causes sur le KPI adressé, nous orientons la décision uniquement sur l'objet de l'action.

Afin de comparer ces contributions avec notre analyse de classement, nous avons d'abord ramené les valeurs des effets totaux, reportés sur la figure 5.8 en pourcentages, en se référant à leurs valeurs absolues, ce qui a donné les contributions suivantes : ***22,28% pour l'indisponibilité de l'outil, 17% pour l'ordonnancement, 16,76% pour les pertes, 16,53% pour les ralentissements, 13,4% pour le guide d'assemblage et 13,22% pour la température.***

Puisque le classement général donné par le tableau 5.2 implique des causes directes et indirectes ensemble, la structure causale joue un rôle clé dans son analyse. En effet, les poids sont mis à jour de manière à minimiser l'erreur en sortie. Cependant, étant donné la structure causale, si un nœud possédant un parent et un descendant causaux, il est évident que l'attribution d'un poids élevé à la connexion liant ce nœud à la sortie lors de l'entrainement du réseau de neurones entraînera l'attribution d'un poids faible à la connexion reliant le parent de ce même nœud à la sortie, et inversement. À titre d'illustration, puisque l'*indisponibilité de l'outil* contribue fortement au changement d'état du TRS, son parent, l'ordonnancement, n'est plus déterminant pour la tâche de prédiction, si et seulement si l'indisponibilité de l'outil est présente dans l'ensemble de données ; mais cela ne signifie pas que l'ordonnancement n'est pas fortement lié au TRS de manière causale. Par conséquent, lorsqu'il s'agit de hiérarchiser les causes de différents niveaux, la structure causale est indispensable pour analyser le classement donné par la méthodologie proposée.

Dans notre cas d'étude, cela signifie que dans le classement général, nous calculerons d'abord les contributions des causes de différents niveaux séparément en fonction des scores du classement général donné par le tableau 5.2. Ces contributions nous permettront ensuite d'échelonner tous les scores, afin de fournir les contributions finales de toutes les causes de niveaux mixtes, comme si elles appartenaient au même niveau. En ce qui concerne les causes directes, nous avons déjà obtenu, dans la section précédente et selon notre classement général, les pourcentages des contributions

suivants : L'*indisponibilité de l'outil* contribue à 46%, les *pertes* contribuent à 27.3%, et les *ralentissements* à 26.7%. Comme nous l'avons précédemment expliqué, le classement des scores indirects donné par le tableau 5.2 doit être pondéré par les contributions des causes directes liées à chaque cause indirecte, avant de comparer les contributions des causes indirectes entre elles, comme expliqué précédemment. Les nouveaux scores pondérés des causes indirectes sont : **4,2044 pour l'*ordonnancement***, **2,49 pour le *guide d'assemblage***, et **2,47 pour la *température***. L'opération inverse doit ensuite être effectuée afin de prendre en compte les contributions des causes indirectes de premier niveau sur les causes directes. Le classement des forces d'associations directes donné par le tableau 5.2 doit être pondéré par les contributions des causes indirectes liées à chaque cause directe. Tout d'abord, les contributions des causes indirectes doivent être comparées entre elles, indépendamment des causes directes. Ces contributions peuvent être extraites du tableau 5.2 en divisant le score de chaque cause indirecte par la somme des scores des causes indirectes, ce qui donne : $9,5 \div (9,5 + 9,23 + 9,14) = 34,09\%$ pour la température, $9,23 \div (9,5 + 9,23 + 9,14) = 33,12\%$ pour le guide d'assemblage, et $9,14 \div (9,5 + 9,23 + 9,14) = 32,8\%$ pour l'ordonnancement. Les nouvelles notes pondérées des causes indirectes sont les suivantes :$16,94 \times 0,328 =$ **5,554 pour l'*indisponibilité des outils*** puisqu'il s'agit du descendant de l'*ordonnancement*, $10,06 \times 0,3312 =$3,332 **pour les *pertes*** puisqu'il s'agit du descendant causal du *guide d'assemblage*, et $9,81 \times 0,3409 =$ **3,344 pour les *ralentissements*** puisqu'il s'agit du descendant causal de la *température*.

Une fois que les forces d'association du classement général sont pondérées en fonction de la structure causale, leurs pourcentages de contributions peuvent être calculés en divisant chaque score pondéré par la somme de tous les autres scores pondérés, ce qui donne les contributions finales suivantes *: **25,96 % pour l'indisponibilité de l'outil, 19,65 % pour l'ordonnancement, 15,63 % pour les ralentissements, 15,57 % pour les pertes, 11,64 % pour le guide d'assemblage et 11,54 % pour la température.***

Ce classement, ainsi que les pourcentages des forces d'associations qui lui sont associés sont cohérents ceux donnés par l'analyse des effets fournie par Bayesialab. Les positions des *pertes* et des *ralentissements* dans l'analyse de classement général proposée et dans l'analyse Bayseialab sont inversées. Cependant, les valeurs des contributions des *pertes* et des *ralentissements* sont très proches les unes des autres dans les deux analyses (15,63%) pour les ralentissements et 15,57% pour les déchets dans l'analyse de classement général proposée, et 16,76% pour les déchets, 16,53% pour les ralentissements). Cette inversion n'est donc pas critique, puisque l'écart entre les contributions des pertes et des ralentissements dont de l'ordre de 10^{-2}.

Compte tenu de ce classement final, il est possible de comparer entre elles les causes de différents niveaux intervenant dans l'état de du TRS. Par exemple, si nous comparons l'*ordonnancement*, les *pertes* et les *ralentissements*, l'*ordonnancement* s'avère être plus influent que les *pertes* ou les *ralentissements*, même s'il s'agit d'une cause indirecte. Afin de vérifier cette affirmation, un réseau de neurones prenant comme entrées uniquement l'*ordonnancement*, les *pertes*, et les *ralentissements* a été construit à l'aide de l'algorithme génétique de construction de réseaux de neurones. Trois autres réseaux de neurones ont également été générés en omettant, à chaque fois, une entrée à la fois parmi ces trois. La figure 5.9 illustre l'impact de l'omission de l'*ordonnancement*, des *pertes* et des *ra-*

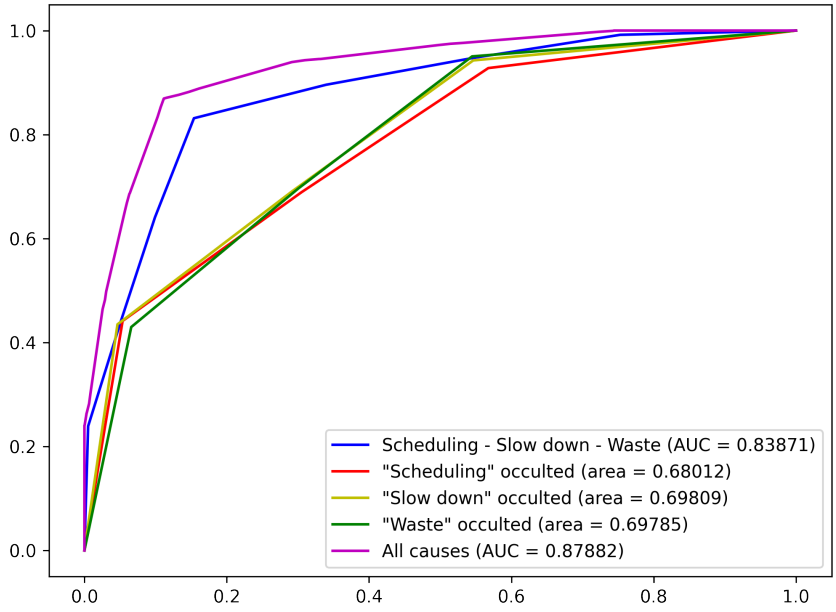


FIGURE 5.9 – Comparaison des contributions de l'*ordonnancement*, des *pertes* et des *ralentissements* au pouvoir prédictif.

lentissements dans les entrées. Les courbes ROC et les AUC présentées dans cette figure montrent que l'omission de l'*ordonnancement* a un impact plus important que l'omission des *pertes* ou du *ralentissement*.

5.1.3 Conclusion

Le déploiement de l'algorithme d'apprentissage des réseau Bayésien nous a permis de construire une structure causale exacte sans aucun apport d'information. Pour ce qui est de l'algorithme génétique, nous avons pu voir qu'il a réussi à trouver un réseau avec des performances aussi élevées que possible, au vu de l'incertitude contenue dans les données. De plus, nous en avons perçu l'utilité lors de nos différents tests, que ce soit pour générer des réseaux de neurones qui nous ont servi pour estimer les forces d'association des causes, ou pour générer des réseaux ayant pour objectif de valider le classement à travers une analyse de sensibilité, en omettant à chaque fois une entrée différente. Concernant, la méthode d'exploitation des poids pour hiérarchiser les causes, elle nous a permis d'obtenir des pourcentages cohérents avec ceux fournis par l'analyse de sensibilité de BayesiaLab. Les classements finaux des causes, qu'elles fassent partie d'un même niveau ou non, sont exactement les même que celui issu de l'analyse de sensibilité de Bayesialab. Bien que des écarts aient pu être observés entre les valeurs des pourcentages issues des deux méthodes, les tendances demeurent cohérentes.Les écarts peuvent être expliqués par l'instabilité intrinsèque liée aux erreurs apprises, et par le fait que l'analyse de BayesiaLab s'est appuyée sur un ensemble de données plus riche, puisqu'elle a utilisé l'ensemble des données générées, tandis que notre méthode s'est appuyé uniquement sur les données d'entraînement pour fournir les poids finaux qui sont à l'origine des calculs de ces pourcentage. Les résultats sont globalement cohérents, mais une comparaison plus équitable peut être effectuée en fournissant à l'analyse Bayesialab un échantillon de la même taille que celui sur lequel notre réseau de neurones s'est basé. Notre objectif initial étant d'informer les décideurs sur les causes les plus pertinentes sur lesquelles agir, nous considérons que la proposition

donne des résultats capables de remplir cette fonction, et qu'elle offre une généricité et une applicabilité facile sur différents KPIs.

Par ailleurs, cette méthode laisse le choix de classer des causes soit d'un même ordre, soit de mélanger des causes d'ordres différents dans un même classement.

Nous concluons par dire que l'interprétation directe de l'ensemble des forces d'associations calculées peut être trompeuse, à cause des relations causales entre les prédicteurs à l'entrée du réseau de neurones qui a généré le classement. Il faut donc prendre en considération, lors de l'interprétation des forces d'association, la structure causale des variables qu'on souhaite classer. Selon le classement souhaité, il suffit d'échelonner les forces d'associations obtenues du classement général en prenant en compte les contributions des forces d'associations des causes directes et indirectes. Bien que cette opération d'échelonnage ne demande à priori aucune expertise, et bien qu'elle soit une opération ayant lieu de façon ponctuelle lors de la phase du développement, son automatisation peut avoir une réel valeur ajoutée.

Ce cas d'étude nous aura permis de dérouler toute la phase de développement de notre proposition, qui a pour objectif de configurer les trois briques autour d'un même KPI. À l'issue de cette phase de développement, nous sortons avec deux trois éléments exploitables et prêts à être utilisés en cas de besoin : une structure causale pour mieux cerner le périmètre de recherche de solution lors d'un processus de décision, et pour mieux comprendre comment un KPI interagit avec le contexte dans lequel il évolue ; un réseau de neurones performant permettant de prédire les futures déviations du KPI pour lequel le réseau a été construit, afin d'anticiper les actions à entreprendre ; et enfin un classement des forces d'associations des entités liées causalement au KPI.

5.2 Deuxième cas d'étude avec étalon

5.2.1 Méthode et outils de comparaison des algorithmes pour la construction d'une structure de réseau Bayésien causal

Pour ce deuxième cas d'étude, nous utilisons un jeu de données, et un graphe étalon [2] issus du domaine médical, et dont la structure causale décrivant les interactions causales entre ses variables, est déjà connue. Nous nous en servons, en particulier, pour comparer l'efficacité de notre algorithme d'apprentissage de structure Bayésienne à d'autres algorithmes que nous avons pu tester.

Le graphe étalon duquel nous devons nous approcher au maximum, est celui donné par la figure 5.10. Ce graphe contient 11 variables discrètes. Nous pouvons faire l'analogie entre les variables de ce graphe, et les variables contextuelles et les KPIs dans notre contexte, en considérant, par exemple, que la variable Cancer du poumon est notre KPI. En effet, l'application de notre proposition sur ce jeu de données, peut témoigner de la généricité de notre proposition, qui peut être applicable dès lors que nous nous intéressons aux liens de causalité.

2. http ://www.causality.inf.ethz.ch/data/LUCAS.html

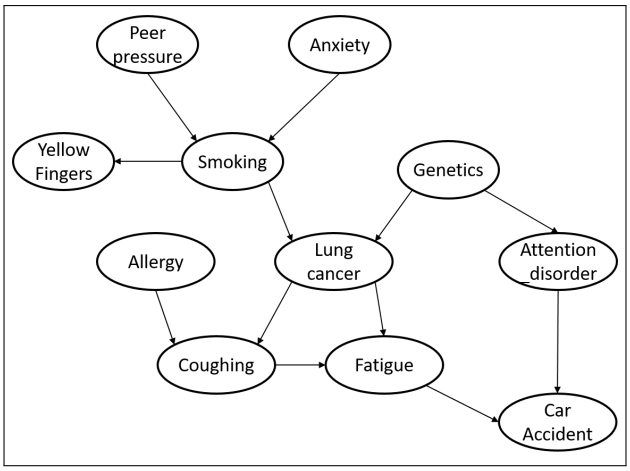


FIGURE 5.10 – Structure du réseau Bayésien causal du jeu de données étudié.

La comparaison de l'algorithme que nous proposons se fera avec les algorithmes suivants, dans des conditions différentes :

— L'algorithme Hill Climbing Search de la bibliothèque `pgmpy` de Python ;

— L'algorithme Taboo implémenté sous BayesiaLab ;

— L'algorithme EQTaboo implémenté sous BayesiaLab ;

— L'algorithme des classes d'équivalences implémenté sous BayesiaLab ;

Afin de tester les algorithmes implémentés sous BayesiaLab, nous étions contraints de réduire notre jeu de données à 10 variables. En effet, nous avons fait nos expérimentations sur la version d'essai du logiciel, qui ne tolère pas plus de 10 variables. Par conséquent, afin que les comparaisons puissent être significatives, nous avons déployé notre algorithme avec 10 variables également, pour avoir les mêmes données de départ. Concernant l'algorithme Hill Climbing Search, le problème ne se posait pas, et nous l'avons donc comparé avec notre algorithme en maintenant l'ensemble du jeu de données pour les deux algorithmes.

Dans la procédure de comparaison, nous avons commencé par tester l'apprentissage de la structure sans aucune connaissance *à priori*. Nous n'avons donc spécifié aucune relation causale déjà connue, ni aucun ordre de précédence entre les variable, ni aucune variable exogène. Nous avons ensuite commencé à injecter des connaissances progressivement, pour pouvoir s'apercevoir des quantités d'information *à priori* nécessaires pour chacun des algorithmes. Les résultats sont décrits dans la section suivante.

5.2.2 Résultats des comparaisons

Nous avons commencé par comparer notre algorithme avec l'algorithme Hill Climbing Search. Les résultats respectifs de l'algorithmes Hill Climbing Search, et ceux de notre proposition, sont illustrés sur les figures 5.11 et 5.12. Les constats qui ressortent de cette comparaison sont les suivants :

— **Ajouts fallacieux d'arcs :** 2 arcs fallacieux ont été ajouté par le Hill Climbing Search, et un seul arc fallacieux par l'algorithme proposé ;

— **Inversions erronées des d'arcs :** 5 arcs inversés de façon erronée par le Hill Climbing Search, et 3 arcs fallacieusement inversés par l'algorithme proposé ;

— **Ajouts d'arcs sémantiquement corrects :** 2 arcs supplémentaires ont été trouvés par l'algorithme proposé. Ces arcs représentent des fermetures transitives qui été ajoutées entre un nœud et son enfant indirect. Ces ajouts n'altèrent pas la représentation des causalités, dans le sens où ils n'invalident aucune information, et n'affirment aucune contradiction, ils font simplement une sorte de raccourci entre deux nœuds. Cependant, cet ajout peut altérer la quantification de la causalité, dans le sens où l'ajout d'un tel arc laisserai présager que le nœud à l'origine de l'arc contribue doublement à l'effet. Dit autrement, lors de la prise de décision, de tels ajouts n'altéreront pas la phase de recherche des solutions, mais plutôt la phase de sélection d'une solution parmi plusieurs.

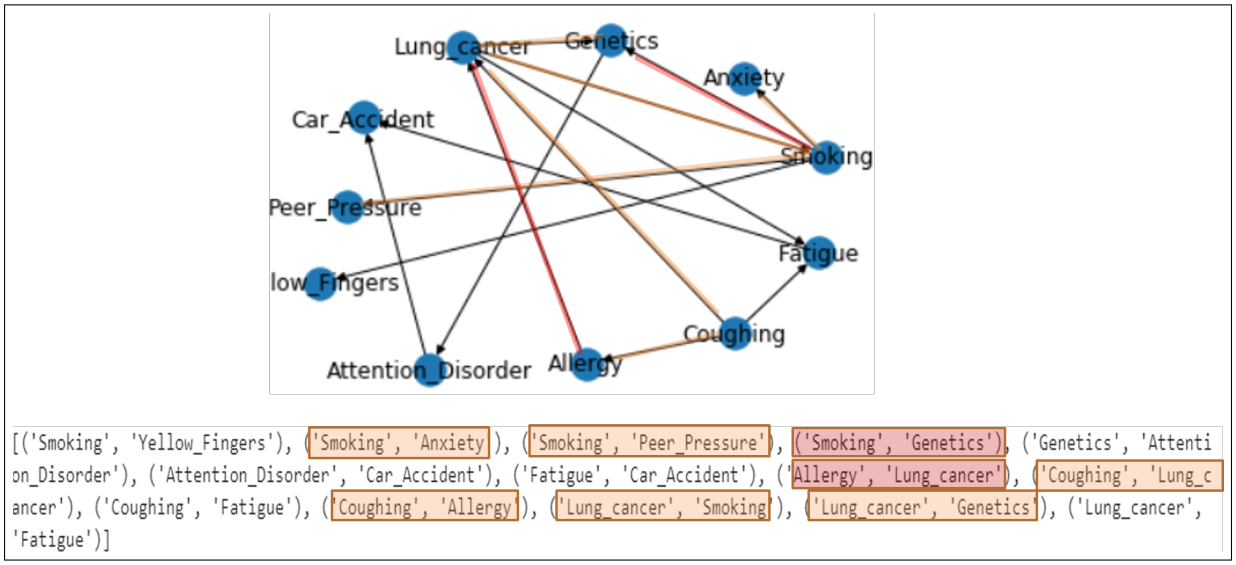


FIGURE 5.11 – Structure de réseau Bayésien apprise en utilisant l'algorithme Hill Climbing Search.

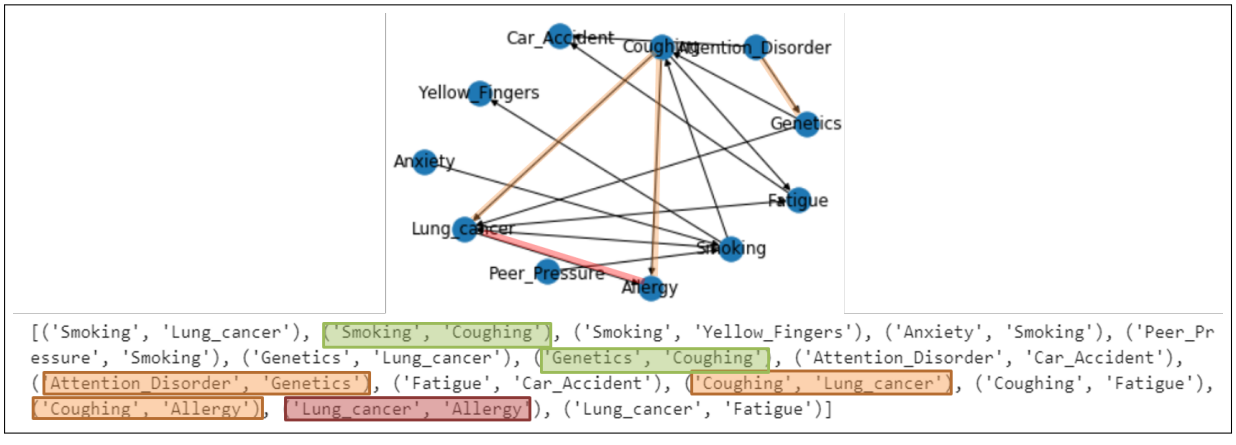


FIGURE 5.12 – Structure de réseau Bayésien apprise en utilisant l'algorithme de la proposition.

Nous en déduisons que sans connaissances à priori, notre algorithme apprend une structure causale plus proche de la réalité que le Hill Climbing Search. Cependant, il contient tout de même des erreurs. Nous introduisons donc un minimum de connaissances

aux deux algorithmes, pour savoir si les connaissances à priori qu'ils nécessitent pour retrouver la bonne structure sont conséquentes. Comme nous l'avons spécifié dans le chapitre 4, l'algorithme que nous proposons permet à l'utilisateur d'introduire trois types de connaissances : des liens causaux, des variables exogènes, ou des ordonnancements entre variables. Nous décidons d'introduire la connaissance la plus minimaliste, qui consiste, selon nous, à ajouter une variable exogène. Nous considérons que l'ajout d'une telle variable nécessite moins de connaissances que l'ajout d'une relation, puisqu'on sait simplement, que la variable exogène en question ne peut pas être un effet, sans rien connaître des particularités des autres variables, ou des liens qu'elles entretiennent. La variable exogène la plus intuitive à introduire, dans ce cas d'étude, est la variable *Genetics*, puisque nous pouvons affirmer que parmi les variables du jeu de données, aucune ne pourra modifier les gènes d'une personnes. Les résultats obtenus suite à l'ajout de cette variable sont illustrés sur la figure. Concernant l'algorithme Hill Climbing Search, l'introduction d'une telle connaissance (*i.e.* une variable exogène) n'y est pas prévue, nous ne pouvons par conséquent pas comparer son effet. La seule connaissance *à priori* pouvant être ajoutée est de définir un graphe de départ. La connaissance minimale que nous pouvons donc ajouter est une relation causale (*i.e.* qui correspond à définir un graphe avec un seul arc comme graphe de départ). Comme en témoigne la figure 5.13, nous pouvons établir que l'ajout de la connaissance de variable exogène à notre algorithme a été efficace, vu que le réseau appris cette fois est complètement fidèle au graphe étalon. Concernant le Hill Climbing Search, afin de retrouver la structure initiale, il a fallu au moins deux relations causales connues.

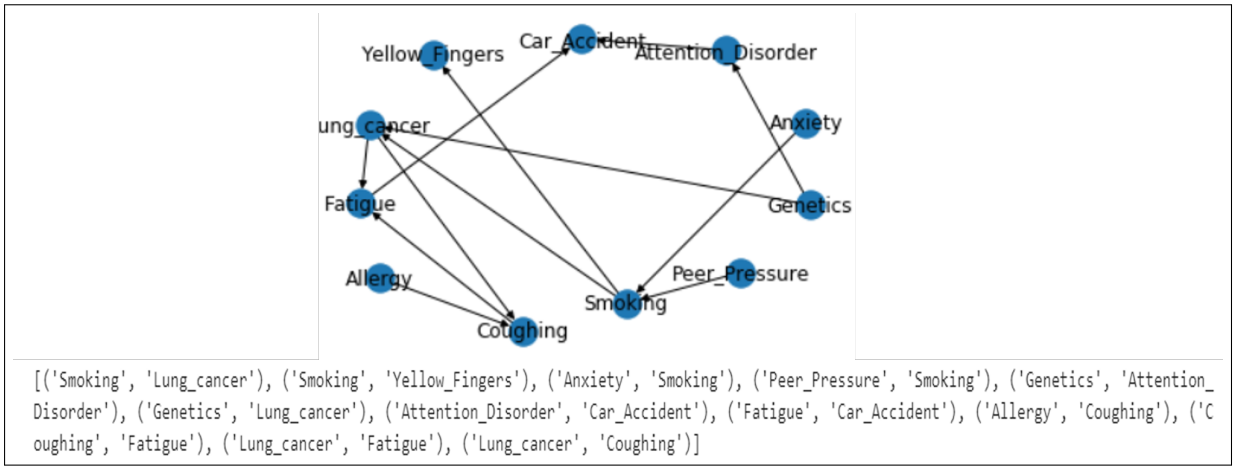


FIGURE 5.13 – Structure de réseau Bayésien apprise en utilisant l'algorithme de la proposition, suite à l'ajout d'une connaissance sur une variable exogène.

Concernant la comparaison avec les algorithmes implémentés sous BayesiaLab, nous avons supprimé la variable *anxiété* du jeu de données, afin de pouvoir faire les expérimentations avec 10 variables. La suppression d'un nœud n'ayant pas de parent, ou d'un nœud feuille et n'étant pas une cause commune à deux autres nœuds, ne doit à priori pas altérer les liens entre les autres nœuds. Le reste du graphe doit donc être retrouvé tel qu'il est dans le graphe étalon. Il en va de même lors de la suppression d'un nœud n'ayant pas d'enfant, et n'étant pas un effet commun de deux autres nœuds. Les graphes résultants des apprentissages avec Taboo, EQTaboo, et l'algorithme des classes d'équivalences sont illustrés, respectivement, sur les figures 5.14, 5.15, et 5.16. Le résultats de l'algorithme

de notre proposition est illustré sur la figure 5.17. Les constats qui ressortent de cette comparaison sont les suivants :

— **Ajouts fallacieux d'arcs :** pour tous les algorithmes comparés ici, aucun arc fallacieux n'a été ajouté, ;

— **Inversions erronées des d'arcs :** 2 arcs inversés de façon erronée par les algorithmes Taboo et EQTaboo de Bayesialab, 1 arc a été inversé par l'algorithme des classes d'équivalence de BayesiaLab, et 1 arc a été inversé par l'algorithme proposé (le même que celui inversé par l'algorithme des classe d'équivalence).

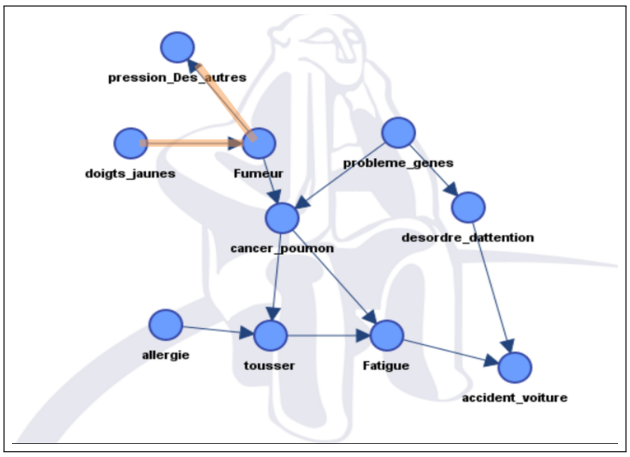


FIGURE 5.14 – Structure de réseau Bayésien apprise en utilisant l'algorithme Taboo de BayesiaLab, en omettant la variable "*anxiété*".

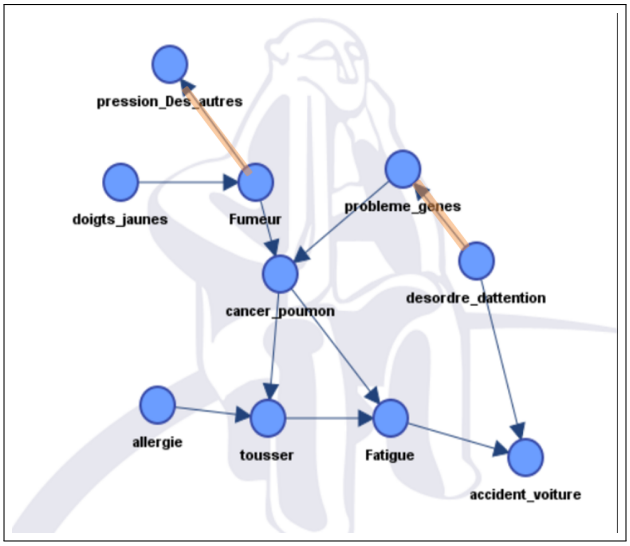


FIGURE 5.15 – Structure de réseau Bayésien apprise en utilisant l'algorithme EQTaboo de BayesiaLab, en omettant la variable "*anxiété*".

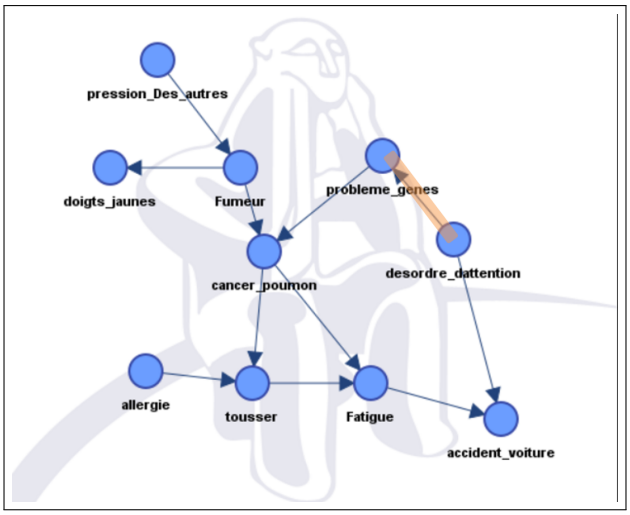


FIGURE 5.16 – Structure de réseau Bayésien apprise en utilisant l'algorithme des classes d'équivalences de BayesiaLab, en omettant la variable "*anxiété*".

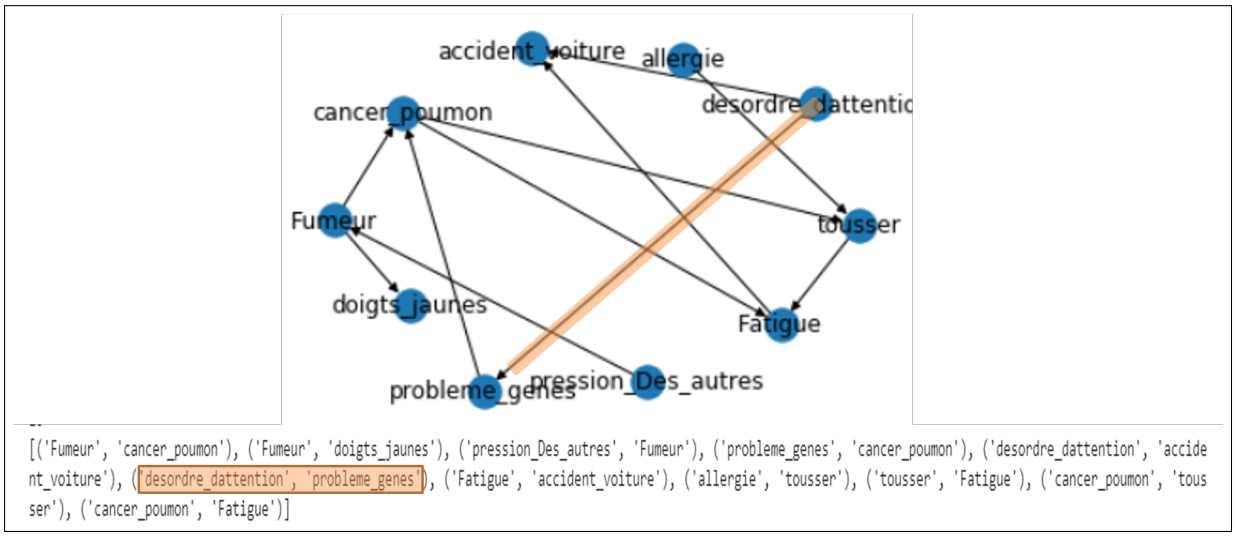


FIGURE 5.17 – Structure de réseau Bayésien apprise en utilisant l'algorithme de la proposition, en omettant la variable "*anxiété*".

Par ailleurs, l'ajout d'un seul arc, a permis de retrouver, pour les quatre algorithmes que nous venons de comparer, la structure de l'étalon. En outre, pour l'algorithme que nous avons proposé, l'ajout d'une variable exogène ou d'un arc a également permis de retrouver la structure d'origine.

La même comparaison que celle que nous venons de décrire sera maintenant réitérée, mais cette fois, en ôtant un nœud enfant au lieu d'un nœud parent : nous remettons la variable *anxiété* dans le jeu de données, et enlevons la variable *doigts jaunes*. La figure 5.18 illustre le graphe obtenu par l'algorithme Taboo de BayesiaLab, la figure 5.19 illustre le graphe obtenu par les deux algorithmes EQTaboo et classes d'équivalences, enfin la figure 5.20 montre le graphe obtenu par notre algorithme.

Cette comparaison montre que l'algorithme Taboo de BayesiLab et l'algorithme de notre proposition se valent lorsque nous éliminons la variable *doigts jaunes*, et fournissent

la structure attendue. Les algorithmes EQTaboo et classes d'équivalences se valent également à leur tour, mais produisent une erreur chacun, se manifestant en l'inversion d'un arc.

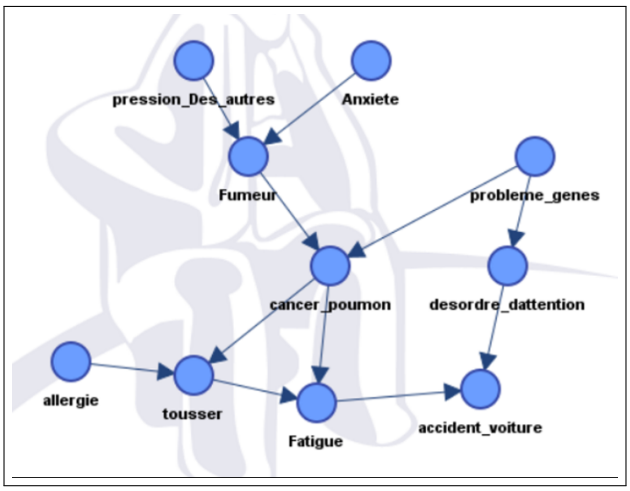


FIGURE 5.18 – Structure de réseau Bayésien apprise en utilisant l'algorithme Taboo de BayesiaLab, en omettant la variable "*doigts jaunes*".

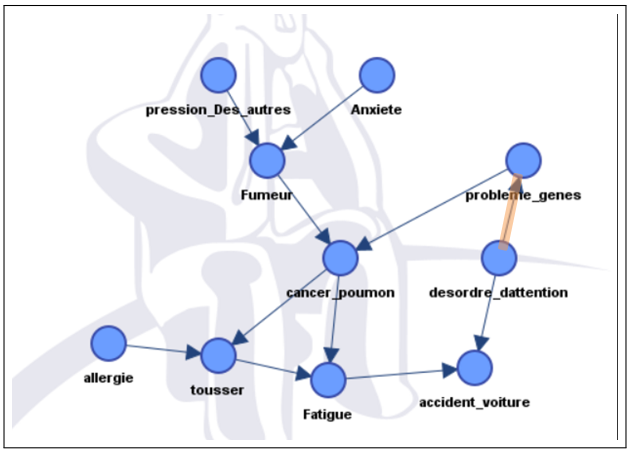


FIGURE 5.19 – Structure de réseau Bayésien apprise en utilisant l'algorithme EQTaboo ou celui des classe d'équvalence de BayesiaLab, en omettant la variable "*doigts jaunes*".

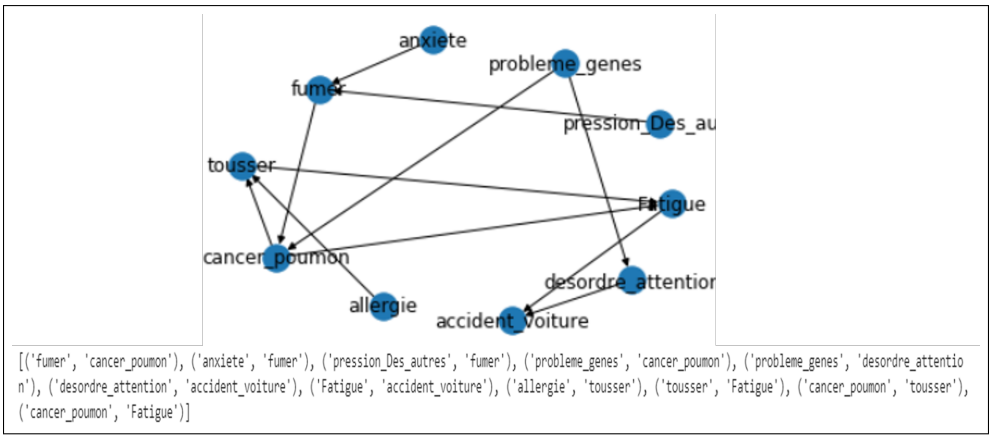


FIGURE 5.20 – Structure de réseau Bayésien apprise en utilisant l'algorithme Taboo de BayesiaLab, en omettant la variable *doigts jaunes*".

5.2.3 Conclusion

Dans cette section, nous avons comparé notre algorithme d'apprentissage de structure de réseau Bayésien avec d'autres algorithmes remplissant la même fonction, en comparant les structure respectives que chaque algorithme fournit. Étant donné les contraintes de limitation du nombre de nœuds dans la version de BayesiaLab dont nous disposons, nous avons entrepris cette comparaison en deux temps. D'abord, nous avons comparé notre algorithme avec le Hill Climbing Search classique, il en est ressorti que notre algorithme fait moins d'erreurs, aussi bien pour l'ajout des arcs fallacieux que pour l'inversion des sens des arcs. Nous soulignons tout de mêmes que notre algorithme a retrouvé des arcs supplémentaires par rapport au graphe étalon, mais qui ne compromettent pas l'interprétation des liaisons causales dans le sens logique du terme, puisque les deux ajouts observés concernent des fermetures transitives. Cependant, ces ajouts peuvent compromettre l'interprétation des forces d'associations.

Nous avons ensuite voulu comparer notre algorithme avec les algorithmes implémentés par BayesiaLab. Pour ce faire, nous avons du supprimer une variable de notre jeu d'entrée, avant de procéder aux comparaisons. Le choix de la variable à supprimer s'est fait en adéquation avec la structure causale originelle. En effet, nous voulions supprimer les variables influant le moins sur la structure causale d'origine, afin que les résultats des algorithmes puissent être comparés moyennant une référence qui continue à refléter le plus fidèlement la réalité. De plus, supprimer une variable centrale de l'étalon, comme la variable *cancer du poumon*, reviendra à supprimer de l'étalon, et donc des données d'apprentissage aussi, les variables *fumeur*, *anxiété*, et *pression*. Ceci aurait diminué la complexité de l'étalon, et nous aurait empêcher de voir bien apercevoir les différences de comportements des différents algorithmes testés. Nous avons donc supprimé dans un premier temps la variable *anxiété*, notre algorithme a eu des performances similaires à celles de l'algorithme classe d'équivalences, fournissant les meilleures performances parmi les algorithmes de BayesiaLab. Nous avons ensuite effectuer une deuxième comparaison avec ces mêmes algorithmes, mais cette fois en supprimant la variable *doigts jaunes*, une fois encore, notre algorithme a rivalisé avec le meilleur parmi les trois testés de BayesiaLab, qui a été, cette fois, l'algorithme Taboo. Bien que les meilleurs résultats de BayesiaLab aient provenus de deux algorithmes différents pour les deux jeux de données différents,

notre algorithme avait, pour les deux jeux de données, un résultat équivalent au meilleur à chaque fois : il n'a inversé qu'un seul arc lors de la première comparaison, et n'a eu aucune erreur lors de la deuxième comparaison, tandis que Taboo, qui n'a eu aucune erreur lors de la deuxième comparaison, en avait quand même commis deux lors de la première comparaison. Par ailleurs, dans notre algorithme, nous donnons à l'utilisateur l'opportunité d'intégrer des connaissances de trois types différents, dont les connaissances sur les variables exogènes pouvant être faciles à reconnaître tout en s'avérant âtre d'une grande aide à l'apprentissage. Aussi, pour tous les cas où notre algorithme s'est trompé, une seule connaissance aurait suffit pour qu'il soit exact, du moins pour ce cas d'étude.

5.3 Validation par rapport aux caractéristiques attendues

Dans la section 2.2.3 du chapitre 2, nous avions établi une liste des caractéristiques qu'une analyse causale devrait satisfaire afin qu'elle soit profitable à la prise de décision. Pour rappel, ces caractéristiques sont les suivantes :

- Objectivité ;
- Implication minimale des ressources humaines ;
- Rapidité ;
- Exhaustivité ;
- Prise en compte des variables de contexte ;
- Généricité ;
- Prise en compte de l'importance des causes ;
- Respect des caractéristiques de causalité (précédemment identifiées dans la section 2.1 du chapitre 2).

Concernant l'objectivité, la méthode que nous avons proposée est basée sur les données. Bien qu'elle puisse nécessiter un apport de connaissances humaines, elle contient tout de même moins de subjectivité qu'une analyse descriptive effectuée manuellement, et reste moins sujette aux biais cognitifs qui sont à l'origine des analyse erronées. L'objectivité de la méthode est, selon nous, étroitement liée au degré d'apport des connaissances humaines. Or, nous avons vu lors des comparaisons que nous avons effectuées avec d'autres algorithmes, que notre méthode fournit les bons résultats avec un minimum de connaissances humaines, relativement aux autres algorithmes. À moins de vérifier matériellement (*via* des interventions réelles) une à une toutes les relations causales possibles entre toutes les variables étudiées, l'objectivité est un critère qui ne peut jamais être atteint de façon absolue, puisque, comme nous l'avons vu au chapitre 2, aucune méthode basée sur les données ne peut identifier correctement toutes les relations causales. Même en cas de vérification matérielle, l'affirmation ou l'infirmation de lien causal entre deux variables ne peut être admissible que si les vérifications sont répétées un grand nombre de fois. En effet, les liens causaux sont caractérisés par leur aspect probabiliste, et une modification ou suppression de la cause n'entraîne pas forcément une modification ou suppression de l'effet : elle en augmente la probabilité seulement. Il faut donc effectuer des vérifications répétées, pour un même lien, afin que la probabilité calculée à posteriori soit significative.

Par conséquent, à défaut de pouvoir parler d'objectivité absolue, nous avons évalué l'objectivité relative (par rapport à d'autres méthodes), en référence à un étalon, et qui s'est avérée supérieure, puisque l'apport en connaissances y a été minimisé.

Pour ce qui est de l'implication minimale des ressources humaines, elle est d'une par satisfaite par le niveau d'apport en connaissances humaines dont nous venons de discuter. En outre, la construction de la structure causale, ainsi que la hiérarchisation des causes, sont des étapes qui sont effectuées lors de la phase de développement. Ceci veut dire que lorsqu'une décision doit être prise concernant un KPI dont le graphe causal et la hiérarchisation des causes ont déjà été définis en phase de développement, les ressources humaines interviendront uniquement pour consulter les résultats, et éventuellement les interpréter avant de décider d'agir ou ne de ne pas agir sur la première cause suggérée par le système.

Pour la rapidité, elle va de paire avec l'implication des ressources humaines, si l'on parle de rapidité de la prise de décision au moment du besoin. En effet, la durée du processus décisionnel est réduite dans les phases de recherche et de sélection de solutions. La rapidité de l'analyse causale en phase de développement, quant à elle, est assez correcte par rapport au gains que peuvent apporter ses résultats : pour le cas d'étude que nous avons introduit dans la section 5.2, l'apprentissage de la structure causale a pris entre une et quatre minutes, en fonction du nombre de variables impliquées, le nombre d'observations, et les connaissances renseignées en entrée. Ces durées sont plus conséquentes que celles observées pour les exécutions des algorithmes de Bayesialab qui ont pris moins de deux secondes chacune. Malgré cela, nous estimons que par rapport aux gains que la méthode apporte, et à la réduction d'apport en connaissances à priori, la rapidité d'apprentissage est satisfaisante. Par ailleurs, la rapidité de l'établissement d'un classement final des causes peut quant à elle être améliorée. En effet, même si l'obtention des forces d'association est automatisé dans la troisième brique, ces forces nécessitent encore d'être relativisées les unes aux autres, par rapport à la structure causale, chose qui nuit à la rapidité quand ces opérations sont effectuées manuellement. Il n'en reste pas moins, encore une fois, qu'il s'agit ici d'une étape ponctuelle servant à préparer les éléments qui serviront d'aide à la décision.

Concernant l'exhaustivité, elle ne peut être atteinte que dans la limite des variables impliquées dans l'analyse. La possibilité d'inclure plusieurs variables rend la proposition plus exhaustive qu'une analyse descriptive. Le degré d'exhaustivité est toutefois conditionné par les données disponibles, mais l'hypothèse H1 que nous avons énoncées au premier chapitre, et qui postule la disponibilité des données issues de plusieurs systèmes d'acquisition, rend cette condition favorable, ce qui favorise donc l'exhaustivité également.

Concernant la prise en compte des variables de contexte, nous avons vu que la méthode que nous proposons peut prendre en compte autant de variables que souhaité. Aussi, nous avons vu dans le chapitre 3, que lorsque le nombre de variables est très important, l'analyse causale peut être décomposée, pour que les résultats de chaque analyse soient ensuite recombinés.

La généricité a également été satisfaite, dans le sens où la structure causale peut être apprise par l'algorithme que nous proposons quel que soit le KPI d'intérêt, à partir du moment où les données satisfont les hypothèses de travail que nous avons émises. Aussi,

la généricité de la hiérarchisation des causes est satisfaite, puisque les réseaux de neurones fournissant les poids utilisés dans la hiérarchisation sont générés automatiquement grâce à la neuro-évolution. Cette généricité peut être améliorer en automatisant la pondération de la hiérarchisation pour classer en tenant compte de la structure causale.

La prise en compte de l'importance des causes est satisfaite grâce à la troisième brique qui permet de calculer les forces d'association entre les causes et le KPI d'intérêt, et d'en déduire une hiérarchisation, moyennant l'algorithme d'exploitation des poids et la relativisation de ses résultats.

Concernant le respect des caractéristiques des liens causaux, nous rappelons que dans la section 2.1 du chapitre 2, nous en avons identifié sept, au regard de notre contexte. Ces caractéristiques étaient les suivantes :

— Aspect probabiliste ;
— Transitivité ou intransitivité selon les cas ;
— Dépendance de l'espace de recherche ;
— Présence possible de cycles ;
— Antécédence temporelle en instanciation ;
— Vérifiabilité ;
— Plausibilité.

L'aspect probabiliste des liens causaux identifiés est intrinsèquement satisfait par la méthode, puisque l'apprentissage de la structure causale est basé sur un calcul probabiliste. Concernant la transitivité, nous avons vu au chapitre 2, que l'approche contre-factuelle ne perçoit pas la causalité comme un lien transitif, tandis que l'approche de production conçoit que la causalité est transitive. Dans notre contexte, nous traitons de la causalité dans le but de pouvoir agir sur les causes afin de modifier les effets (*i.e.* les déviations), nous percevons donc la causalité comme un lien de production (une cause aide son effet à se produire). La transitivité est donc acceptable, dans notre contexte, à partir du moment où liens intermédiaires de cette transitivité restent présents présents dans la structure, puisque nous ne voulons pas toujours agir que sur les causes racines. Pour résumer, les conditions de l'interprétation logique de la transitivité ont été respectées, puisque les arcs intermédiaires ont été maintenus lors des fermetures transitives, cependant, la transitivité au sens quantitatif n'a pas été respectée, dans le sens où les calculs des degrés d'influence sur l'effet seront erronés. Concernant la dépendance à l'espace de recherche, nous avons pu voir, à travers le cas d'étude présenté dans la sous-section 5.2, que pour un même cas d'étude et un même nombre de variables contextuelles, l'omission d'une variable différente à chaque fois donne des structures causales différentes (en l'occurrence, l'omission de la variable *anxiété* et celle de la variable *doigts jaunes* donnent lieu à des résultats différents. Pour ce qui est de la présence possible de cycles, notre méthode ne permet pas d'inclure des cycles dans une structure causale. Il revient donc à l'utilisateur d'analyser les résultats et d'observer le contexte pour ajouter éventuellement des cycles. Ceci peut être fait notamment en vérifiant, matériellement et à de multiples reprises, les effets mutuels qu'ont les deux variables soupçonnées de causer un cycle l'une sur l'autre. Concernant l'antécédence temporelle et la plausibilité, nous n'avons pu vérifier que partiellement si les résultats obtenus sont plausibles, et s'ils respectent l'ordre d'antécédence temporelle.

En effet, l'ordre des variables, ainsi que la présence ou l'absence des liens entre elles, sont cohérents avec les graphes étalons, ce qui valide implicitement la plausibilité et l'antécédence temporelle. En effet, ces graphes étalons sont considérés comme étant avérés, et par conséquent plausibles et respectant les antécédences temporelles. De ce fait, tout graphe cohérent avec le graphe étalon est également plausible et respecte les antécédences temporelles. Nous n'avons malheureusement pas pu appliquer la proposition sur un cas d'étude réel dont la structure vraie est inconnue, pour vérifier la plausibilité des résultats par rapport aux avis des utilisateurs, et pour vérifier la pertinence des ordres des variables (*i.e.* les orientations des arcs) par rapport à la réalité. Il en va de même pour la vérifiabilité.

5.4 Conclusion

Dans cette section, nous avons introduit deux cas d'études servant à la fois à illustrer le déroulement de la méthode proposée, à vérifier que nous apportons des réponses aux exigences que nous nous sommes fixées sous forme de fonctions lors du chapitre 1, et à valider que les réponses que nous y apportons sont correctes.

Nous avons donc commencé par un cas d'étude académique construit par le biais d'une simulation de données. Nous avons utilisé ce cas d'étude pour mettre en avant le déroulement des trois briques que notre méthode, et pour vérifier que chaque brique répond bien à sa fonction. Dans ce cas d'étude, la première brique a été implémenté pour apprendre une structure causale, et vérifier si elle s'accorde avec la structure de référence qui a été utilisée pour générer les donner. Nous avons pu retrouver cette structure. À ce stade, nous pouvions établir que cette brique répond bien à la fonction d'identification des liens causaux pour ce cas d'étude, mais gardons à l'esprit que cette brique ne pourrait jamais y répondre infailliblement, à elle seule, dans toutes les circonstances. Raison pour laquelle un deuxième cas d'études a été introduit plus tard. Nous avons ensuite poursuivi notre démarche en vérifiant que la deuxième brique répond bien à sa fonction de construction automatique de réseaux de neurones performants. Là encore, il serait présomptueux de croire que l'on pourrait prouver que l'algorithme génétique, utilisé pour construire les réseaux de neurones, a effectivement convergé vers un optimum global. Cependant, étant donné que nous disposions des tables de probabilités associées aux réseaux, nous pouvions juger si les performances atteintes sont correctes ou non. Il en est sorti que cette brique a bien rempli sa tâche, et a permis de bien faire évoluer la performance de prédiction, permettant de répondre à la fonction F3 si la phase d'utilisation est implémentée. Enfin, la troisième brique a été détaillée, pour le même cas d'études, et les conditions d'interprétations des forces d'association issues de cette brique y ont été expliquées. Grâce à la comparaison avec l'analyse de sensibilité de BayesiaLab, nous avons pu montré qu'elle remplit bien sa tâche, qui correspond à la fonction F2. En effet, cette brique a permis, en conjugaison avec la première, de fournir un ordre avéré de hiérarchisation des causes. Nous avons souligné pourquoi elle ne doit pas âtre utilisée toute seule à cette fin quand des causes de plusieurs ordres y sont mélangées. Nous avons ensuite expliqué comment de simples mises à l'échelle en fonction des ordres des causes, se prêtant bien à une automatisation, peuvent directement fournir une hiérarchisation finale fiable des causes.

Suite à ce premier cas d'études, nous avons donc pu montrer que nous répondons correctement à fonction F2, et que nous fournissons les outils nécessaires (*i.e.* des réseaux de neurones performants, pour répondre à la fonction F3 lors de la phase d'utilisation).

Quant à la fonction F1, nous avons fait état, dans le chapitre 2, qu'il est simplement impossible de pouvoir y répondre de façon complètement automatisée, et que les connaissances humaines, peuvent s'avérer indispensables dans certaines situations. Le rôle de cette fonction F1 s'est donc précisé au chapitre 2, et le but est devenu de trouver ou d'approcher une structure causale avec un minimum de connaissances humaines. Nous avons, dans ce but, introduit le deuxième cas d'étude où une structure causale, plus complexe, est avérée et sert de référence. Nous avons donc déroulé l'algorithme d'apprentissage de structure Bayésienne en utilisant les données associées à cette structure, pour comparer son résultat, d'abord à la structure de référence pour évaluer à quel point nous approchons la structure causale, et ensuite à d'autres algorithmes partageant notre objectifs pour évaluer à quel point nous minimisons les connaissances requises.

Grâce à ces deux cas d'études, nous avons donc pu vérifier que la phase de développement de notre approche répond aux attentes, et comment les résultats de chaque brique, prise individuellement, se positionnent par rapport à d'autres méthodes répondant aux mêmes objectifs que la brique en question.

Enfin, nous avons détaillé les forces et faiblesses de notre proposition. Pour cela, nous avons utilisé les critères que nous nous étions fixés au chapitre 2, et qui nous permettent de vérifier si l'analyse causale complète que nous proposons répond bien aux caractéristiques attendues pour une analyse causale dans un contexte d'aide à la décision.

Conclusion générale et perspectives

Conclusion générale

Tous les industriels aspirent à l'amélioration des performances, qui est étroitement liée à l'amélioration de la prise de décision. L'optimisation des processus décisionnels peut être vue sous deux angles différents : l'amélioration de la durée de prise de décision, et l'amélioration de la pertinence du contenu de la décision. L'objectif des travaux de cette thèse était de pouvoir améliorer la prise de décision sous ces deux angles. Une décision pertinente repose sur un enchaînement de réflexions sur les tenants et les aboutissants de la situation : elle repose sur la notion de la causalité. L'objectif était donc de développer une méthode d'analyse de la causalité, de manière à améliorer la prise de décision, ainsi que la durée qu'elle engage. Nous nous sommes intéressés plus particulièrement aux KPIs, qui sont au cœur des processus décisionnels dans la majorité des entreprises. Une décision basée sur un KPI, nécessite de pouvoir comprendre et décrire ce qui régit le comportement du KPI, c'est à dire ses causes. Nous avons donc souhaité construire un système qui puisse, premièrement, identifier, pour chaque KPI, l'ensemble de ses causes parmi les variables contextuelles de l'environnement où il évolue. Ceci participera notamment à l'amélioration de la pertinence de la décision, ainsi qu'à la réduction du temps de recherche des solutions. Deuxièmement, afin de réduire le temps de sélection de la solution, nous avons souhaité que ce système puisse, pour chaque KPI, hiérarchiser les causes qui lui sont liées. Enfin, nous souhaitions être proactifs, et disposer d'un moyen pour prédire les déviations, et ce pour chaque KPI. Afin de développer notre proposition nous avons émis une hypothèse forte sur la disponibilité des données contextuelles en large quantité, ainsi que celle de leurs valeurs historiques et celles des KPIs.

Pour construire un tel système, nous nous sommes d'abord intéressés à la notion de causalité dans son sens le plus large. Nous avons donc effectué une revue de littérature sur les différentes perceptions de la causalité dans divers domaines. Il en est sorti qu'il n'y a à ce jour pas de définition universelle de la notion de causalité. Nous avons alors listé plusieurs caractéristiques, qui cumulées les une aux autres, permettent d'approcher la présomption d'existence d'un lien causal. Nous avons ensuite passé en revue les différentes techniques et méthodes utilisées pour l'analyse causale dans le contexte de prise de décision. Cela nous a permis souligner les forces et faiblesses de chaque méthode, et nous a servi à identifier les caractéristiques que notre proposition devrait satisfaire. Au vu des caractéristiques identifiées caractérisant les liens causaux, nous avons choisi d'utiliser les réseaux Bayésiens, que nous avons donc présentés. Plus précisément, nous avons opté pour l'apprentissage des réseaux Bayésien. Concernant le besoin de prédiction, nous avons fait le choix d'utiliser des réseaux de neurones, en raison des nombreux avantages qu'ils présentent. Enfin, nous avons justifié notre choix d'utiliser également les réseaux de

neurones pour répondre au troisième besoin de classement des causes par ordre de force d'association au KPI qu'elles affectent.

Suite à cela, nous avons construit, pas à pas, et en adéquation avec les opportunités et les besoins identifiés dans l'état de l'art, un diagramme SADT décrivant l'architecture globale de notre méthode, et qui est composée de quatre brique : une brique pour l'analyse causale servant à identifier les causes d'un KPI donné ; une brique pour la construction automatique des réseaux de neurones dont nous aurons besoin pour faire la prédiction, et dont nous allons également nous servir pour la hiérarchisation des causes, qui elle, représente la troisième brique ; et enfin une quatrième brique compilant l'ensemble de ces éléments pour permettre une prise de décision proactive face à une déviation grâce à la prédiction, et pour permettre une prise de décision efficace, puisque cette brique indiquera les causes sur lesquelles il serait plus efficace d'agir, selon la hiérarchisation des causes. Le déroulement de la méthode proposée se fait donc en deux temps : pour chaque KPI d'intérêt, une première phase de développement est d'abord conduite. Cette phase inclue : la spécification des causes liées au KPI ainsi que la structure de ces liaisons, le classement de ces causes par ordre d'importance, et la spécification le réseau de neurones qui servira plus tard à prédire le KPI en question. A l'issu de cette première phase de développement, on aura construit une sorte de boîte à outils pour le KPI d'intérêt, qui peut être utilisée dès lors qu'une décision doit être prise concernant ce KPI. La phase d'utilisation comprend simplement le déploiement du réseau de neurones construit, afin d'effectuer une prédiction en temps réel, et dès qu'une déviation est prédite, les outils construits lors de la phase de développement peuvent être utilisés pour orienter la décision vers les actions les plus prometteuses.

Une fois l'architecture de la méthode fixée, nous avons procédé au développement interne à chaque brique. Pour la première brique, nous avons proposé d'apprendre la structure Bayésienne causale entre les données utilisant un algorithme Hill Climbing, auquel nous avons apporté des modifications pour palier aux problèmes de plateaux, de stagnation de la recherche, et de scores égaux. Nous avons également spécifié trois types de connaissances que l'utilisateur peut fournir pour démarrer et enrichir la recherche : les variables exogènes, les liaisons causales déjà connues, et les ordres d'antériorité entre les variables. Concernant la construction automatique des réseaux de neurones, nous avons choisi de faire de la neuro-évolution, et nous avons proposé un codage permettant de retrouver les caractéristiques des réseaux correctement, et d'appliquer correctement les opérateurs d'évolution. Pour la troisième brique, nous avons proposé une formule de classement des entrées d'un réseau de neurones, en adéquation avec les forces d'association entre chaque nœud en entrée du réseau et la sortie du réseau. Le calcul de ces forces d'association est basé sur les nœuds finaux d'un réseau de neurones ayant un bon pouvoir de prédiction, construit grâce à la brique précédente, et doit être interprété en adéquation avec la structure causale.

Enfin, nous avons validé notre méthode en utilisant deux cas d'études différents. Nous avons évalué les algorithmes développés pour chaque brique par rapport à un étalon, afin de pouvoir vérifier que la méthode répond bien à ses fonctions. Nous avons également évaluer les performances de notre méthode en la comparant sa première et troisième briques à d'autres méthodes que nous avons testées sur les mêmes cas d'étude, et qui tentent de répondre chacune, séparément, à une fonction. Enfin, nous avons détaillé les forces et

faiblesses de notre proposition. Pour cela, nous avons utilisé les critères que nous nous étions fixés au chapitre 2, et qui nous permettent de vérifier si l'analyse causale complète que nous proposons répond bien aux caractéristiques attendues pour une analyse causale dans un contexte d'aide à la décision.

Perceptives

A l'issue de ces travaux de thèse, plusieurs perspectives peuvent être envisageables pour enrichir la méthode proposée ainsi que sa validation :

— La phase d'utilisation doit pouvoir être implémentée sur dans un contexte impliquant des données issues de systèmes d'acquisition en temps réel, afin de mieux percevoir l'utilité de l'aide à la décision fournie et/ou les limites de l'utilisation d'une telle méthode sur un cas d'application réel ;

— L'apprentissage de l'analyse causale devrait également être effectué sur un cas réel, où la structure causale n'est pas connue, afin de vérifier si les liens causaux remplissent bien les caractéristiques identifiées pour la causalité ;

— Dans le chapitre 5, nous avons expliqué que lors de l'exploitation de la formule développée dans la section 4.4 du chapitre 4, il faut tenir compte des ordre des causes par rapport à l'effet. Il serait intéressant de pouvoir automatiser les transformations à faire pour relativiser les forces d'associations les unes aux autres, en adéquation avec la structure causale ;

— Pour la construction des réseaux de neurones avec les algorithmes génétiques, nous faisons évoluer une seule et même fonction d'activation pour toutes les couches cachées. Il serait probablement pertinent de prévoir des réseaux utilisant des fonctions d'activation différentes dans les différentes couches cachées, et de voir si cela favorise davantage la construction des réseaux de neurones performants ;

— Il serait également intéressant de prendre en compte les retours d'expérience pour enrichir les classements des causes. Ceci est néanmoins conditionné par le déploiement de la phase d'utilisation. En effet, l'idée serait que lorsqu'un utilisateur est confronté à la déviation d'un KPI, et qu'il décide de d'agir sur la cause qui lui a été proposée par le système, il pourra ensuite attribuer une "note" au lien entre le KPI en question et la cause qui lui a été proposée. Cette valeur permettra de pénaliser le lien si le système s'est trompé (*i.e.* si l'action sur la cause proposé n'a pas amélioré la situation ou si elle l'a empirée).

Références

Abellán, J., Gómez-Olmedo, M., & Moral, S. (2006). Some variations on the pc algorithm. In *Probabilistic graphical models.*

Ahmadizar, F., Soltanian, K., AkhlaghianTab, F., & Tsoulos, I. (2014). Artificial neural network development by means of a novel combination of grammatical evolution and genetic algorithm. *Engineering Applications of Artificial Intelligence*, *39*, 1–13.

Ahmadizar, F., Soltanian, K., AkhlaghianTab, F., & Tsoulos, I. (2015). Artificial neural network development by means of a novel combination of grammatical evolution and genetic algorithm. *Engineering Applications of Artificial Intelligence*, *39*, 1-13.

Airola, A., Pahikkala, T., Waegeman, W., De Baets, B., & Salakoski, T. (2011). "An experimental comparison of cross-validation techniques for estimating the area under the ROC curve". *Computational Statistics & Data Analysis*, *55*(4), 1828–1844.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716-723. doi: 10.1109/TAC.1974.1100705

Allain, P. (2013). La prise de décision : aspects théoriques, neuro-anatomie et évaluation : Prise de décision. *Revue de neuropsychologie*, *5*(2), 69–81. doi: 10.1684/nrp.2013.0257

Anderson, D. R. (2007). *Model based inference in the life sciences : a primer on evidence.* Springer Science & Business Media.

Andersson, S. A., Madigan, D., & Perlman, M. D. (1997). A characterization of markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, *25*(2), 505–541.

Angeline, P. J., Saunders, G. M., & Pollack, J. B. (1994). An evolutionary algorithm that constructs recurrent neural networks. *IEEE transactions on Neural Networks*, *5*(1), 54–65.

Ariely, D., & Zakan, D. (2001). A timely account of the role of duration in decision making. *Acta Psychologica*, *108*(2), 187-207. (Time, Judgement and Decision Making) doi: https://doi.org/10.1016/S0001-6918(01)00034-8

Atallah, C. (2014). *Analyse de relations de discours causales en corpus : étude empirique et caractérisation théorique* (Theses, Université Toulouse le Mirail - Toulouse II). Consulté sur `https://tel.archives-ouvertes.fr/tel-01239326`

Barberousse, A., Bonnay, D., Cozic, M., et al. (2011). La causalité. *Précis de philosophie des sciences*, 100–140.

Bard, D., Barouki, R., Benhamou, S., Bénichou, J., Clavel, J., Jougla, E., & Launoy, G. (2005). *Cancer : approche méthodologique du lien avec l'environnement.*

Belis, M. (1995). Causalité, propension, probabilité. Consulté sur `https://www.persee.fr/doc/intel_0769-4113_1995_num_21_2_1501` doi: 10.3406/intel.1995.1501

Bellot, D. (2002). *Fusion de données avec des réseaux bayésiens pour la modélisation des systèmes dynamiques et son application en télémédecine* (Thèse de doctorat non publiée). Université Henri Poincaré-Nancy 1.

Ben Amor, N. (1988, sep). Réseaux bayésiens..

Benoît, C. (2011). *Prendre la bonne décision avec la méthode des 4 Éléments*. Gereso.

Beretta, S., Castelli, M., Gonçalves, I., Henriques, R., & Ramazzotti, D. (2018). Learning the structure of bayesian networks : A quantitative assessment of the effect of different algorithmic schemes. *Complexity*, *2018*.

Besneux, J.-M. (2017). *Usages de la causalité dans l'argumentation* (Theses, Normandie Université). Consulté sur `https://tel.archives-ouvertes.fr/tel-01743775`

Blanchard, T. (2018). Causalité (a). *l'Encyclopédie philosophique*. Consulté sur `https://encyclo-philo.fr/causalite-a`

Bonnefond, M., Castelain, T., Cheylus, A., & Van der Henst, J.-B. (2014, 10). Reasoning from transitive premises : An eeg study. *Brain and Cognition*, *90*. doi: 10.1016/j.bandc.2014.06.010

Bordeleau, F.-È., Mosconi, E., & Santa-Eulalia, L. A. (2018). Business Intelligence in Industry 4.0 : State of the art and research opportunities. *Proceedings of the 51st Hawaii International Conference on System Sciences*, *9*, 3944–3953. doi: 10.24251/hicss.2018.495

Boreux, J.-J., Parent, E., & Bernier, J. (2009). *Pratique du calcul bayésien (statistique et probabilités appliquées) (french edition)* (1st Edition. éd.). Springer.

Bouckaert, R. R. (2008). Bayesian network classifiers in weka for version 3-5-7. *Artificial Intelligence Tools*, *11*(3), 369–387.

Boularas, N., & Djalab, C. (2020). *Système de la prévision mensuelle de la consommation d'énergie électrique basée sur les réseaux de neurones artificielle-etude cas la ville de m'sila* (Rapport technique). Univ M'sila.

Bourgeois-Gironde, S. (2002). Causalité et probabilité. *Philosophies*, 37–54.

Bousdekis, A., Lepenioti, K., Apostolou, D., & Mentzas, G. (2021). A review of data-driven decision-making methods for industry 4.0 maintenance applications. *Electronics*, *10*(7), 828.

Brownlee, J. (2018, juillet). *Difference between a batch and an epoch in a neural network.*

Cagnetti, C., Gallo, T., Silvestri, C., & Ruggieri, A. (2021). Lean production and industry 4.0 : Strategy/management or technique/implementation ? a systematic literature review. *Procedia Computer Science*, *180*, 404-413. Consulté sur `https://www.sciencedirect.com/science/article/pii/S1877050921002994` (Proceedings of the 2nd International Conference on Industry 4.0 and Smart Manufacturing (ISM 2020)) doi: https://doi.org/10.1016/j.procs.2021.01.256

Castellani, M. (2013). Evolutionary generation of neural network classifiers—an empirical comparison. *Neurocomputing*, *99*, 214–229.

Chollet, F. (2021). *Deep learning with python.* Simon and Schuster.

ChongYong, C., & HongChoon, O. (2017). Comparison of scoring functions on greedy search bayesian network learning algorithms. *Goal of Pertanika*, 719.

Colombo, D., & Maathuis, M. H. (2014, janvier). Order-independent constraint-based causal structure learning. *J. Mach. Learn. Res.*, *15*(1), 3741–3782.

Combacau, M., Berruet, P., Zamai, E., Charbonnaud, P., & Khatab, A. (2000). Supervision and Monitoring of Production Systems. *IFAC Proceedings Volumes*, *33*(17), 849–854. doi: 10.1016/s1474-6670(17)39514-9

Cooper, G. F., & Herskovits, E. (1992). A bayesian method for the induction of probabilistic networks from data. *Machine learning*, *9*(4), 309–347.

Dash, A. K., Bhattacharjee, R. M., Singh, C. S., Aftab, A., & Sagesh, K. M. R. (2017). A decision can be a disaster : A descriptive analysis of a case study. *International Journal of Applied Environmental Sciences*, *12*(10), 1803–1820.

Davis, W. A. (1988). Probabilistic theories of causation. In *Probability and causality* (pp. 133–160). Springer.

De Campos, L. M. L., Roisenberg, M., & de Oliveira, R. C. L. (2011). Automatic design of neural networks with l-systems and genetic algorithms-a biologically inspired methodology. In *The 2011 international joint conference on neural networks* (pp. 1199–1206).

De Oña, J., & Garrido, C. (2014). Extracting the contribution of independent variables in neural network models : a new approach to handle instability. *Neural Computing and Applications*, *25*(3), 859–869.

Ding, S., Li, H., Su, C., Yu, J., & Jin, F. (2013). Evolutionary artificial neural networks : a review. *Artificial Intelligence Review*, *39*(3), 251–260.

Doncieux, S. (2010). *Robotique évolutionniste : conception orientée vers le comportement* (Habilitation à diriger des recherches, Université Pierre et Marie Curie - Paris VI). Consulté sur `https://tel.archives-ouvertes.fr/tel-00547778`

Dreyfus, G., Martinez, J.-M., Samuelides, M., Gordon, M. B., Badran, F., & Thiria, S. (2008). *Apprentissage statistique.* Editions Eyrolles.

Drouet, I. (2007). *Causalité et probabilités : réseaux bayésiens, propensionnisme* (Theses, Université Panthéon-Sorbonne - Paris I). Consulté sur `https://tel.archives-ouvertes.fr/tel-00265287`

Duret, D., & Pillet, M. (2011). *Qualité en production : de l'iso 9000 à six sigma.* Editions Eyrolles.

Esfeld, M. (2010). Les fondements de la causalité. *Matière première*, *1*, 199–222.

Faghraoui, A. (2013). *Modélisation de causalité et diagnostic des systèmes complexes de grande dimension.* (Theses, Université de Lorraine). Consulté sur `https://tel.archives-ouvertes.fr/tel-01750529`

Falk, R., & Bar-Hillel, M. (1983). Probabilistic dependence between events. *The Two-Year College Mathematics Journal*, *14*(3), 240–247.

Faucher, J. (2009). *Pratique de l'amdec-2e édition : Assurez la qualité et la sûreté de fonctionnement de vos produits, équipements et procédés.* Dunod.

Fawcett, T. (2006). "An introduction to ROC analysis". *Pattern recognition letters*, *27*(8), 861–874.

François, O. (2006). *De l'identification de structure de réseaux bayésiens à la reconnaissance de formes à partir d'informations complètes ou incomplètes.* (Thèse de doctorat non publiée). INSA de Rouen.

Francois, O., & Leray, P. (2004). Evaluation d'algorithmes d'apprentissage de structure pour les réseaux bayésiens. In *Proceedings of 14ème congrès francophone reconnaissance des formes et intelligence artificielle, rfia* (pp. 1453–1460).

Freeman, C., & Louça, F. (2001). *As time goes by : from the industrial revolutions to the information revolution.* Oxford University Press.

Friedman, N., Nachman, I., & Pe'er, D. (2013). Learning bayesian network structure from massive datasets : The" sparse candidate" algorithm. *arXiv preprint arXiv :1301.6696*.

FronlineSolvers. (2020). *Training an artificial neural network - intro.* `https://www.solver.com/training-artificial-neural-network-intro`. (Accessed : 2021-01-09)

Gamache, S. (2019). *Stratégies de mise en oeuvre de l'industrie 4.0 dans les petites et moyennes entreprises manufacturières québécoises* . Université du Québec à Chicoutimi en extension avec l'Université du Québec à Trois-Rivières.

Gamache, S., Abdulnour, G., & Baril, C. (2019). Étude du potentiel de l'industrie 4.0 quant à la transformation de la pme manufacturière québécoise : Une analyse littéraire et expérimentale. *Génie industriel et productique*, *2*(Numéro Spécial Lean et industrie du futur). Consulté sur https://www.openscience.fr/Etude-du-potentiel-de-l-Industrie-4-0-quant-a-la-transformation-de-la-PME doi: 10.21494/ISTE.OP.2019.0427

García-Pedrajas, N., Ortiz-Boyer, D., & Hervás-Martínez, C. (2006). An alternative approach for neural network evolution with a genetic algorithm : Crossover by combinatorial optimization. *Neural Networks*, *19*(4), 514–528.

Garson, D. (1991). Interpreting neural network connection weights.

Georgakopoulos, D., Jayaraman, P. P., Fazia, M., Villari, M., & Ranjan, R. (2016). Internet of things and edge cloud computing roadmap for manufacturing. *IEEE Cloud Computing*, *3*(4), 66–73.

Giorgi, R. (2013, apr). *Traitement des données manquantes* (Rapport technique). Marseille : SESSTIM - Faculté de Médecine - Aix-Marseille Université.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Goupy, J. (2000). *Plans d'expérience : Les mélanges*. Dunod.

Grandchamp, E., Alata, O., Olivier, C., Khoudeir, M., & Abadi, M. (2011). Sélection des variables optimales par optimisation multi-objective de l'information mutuelle. In *Gretsi* (p. 1).

Grislin, M. (1995). *Définition d'un cadre pour l'évaluation à priori les interfaces homme-machine dans les systèmes industriels de supervision* (Theses, Université de Valenciennes et du Hainaut-Cambrésis). Consulté sur https://tel.archives-ouvertes.fr/tel-01441585

Guenter. (2012). *How to correctly choose model based on bic ?* Cross Validated (https ://stats.stackexchange.com/users/296955/guenter). Consulté sur https://stats.stackexchange.com/q/491009 (URL :https ://stats.stackexchange.com/q/491009 (version : 2020-10-08))

Güss, C. D., & Robinson, B. (2014). Predicted causality in decision making : the role of culture. *Frontiers in Psychology*, *5*, 479.

Hall, N. (2004). Two concepts of causation. *Causation and counterfactuals*, 225–276.

Halpern, J. Y. (2016). Sufficient conditions for causality to be transitive. *Philosophy of Science*, *83*(2), 213–226.

Han, S., Pool, J., Tran, J., & Dally, W. J. (2015). Learning both weights and connections for efficient neural networks. *arXiv preprint arXiv :1506.02626*.

Hancock, P. J. (1992). Genetic algorithms and permutation problems : a comparison of recombination operators for neural net structure specification. In *[proceedings] cogann-92 : International workshop on combinations of genetic algorithms and neural networks* (pp. 108–122).

Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning bayesian networks : The combination of knowledge and statistical data. *Machine learning*, *20*(3), 197–243.

Hill, A. B. (1965). *The environment and disease : association or causation ?* Sage Publications.

Hippert, H. S., & Taylor, J. W. (2010). An evaluation of bayesian techniques for controlling model complexity and selecting inputs in a neural network for short-term load forecasting. *Neural Networks*, *23*(3), 386-395. Consulté sur https://www.sciencedirect.com/science/article/pii/S0893608009003189 doi: https://

doi.org/10.1016/j.neunet.2009.11.016
Hoffmann Souza, M. L., da Costa, C. A., de Oliveira Ramos, G., & da Rosa Righi, R. (2020). A survey on decision-making based on system reliability in the context of industry 4.0. *Journal of Manufacturing Systems*, *56*, 133-156. Consulté sur https://www.sciencedirect.com/science/article/pii/S0278612520300807 doi: https://doi.org/10.1016/j.jmsy.2020.05.016
Hogg, R. V., Tanis, E. A., & Zimmerman, D. L. (1977). *Probability and statistical inference* (Vol. 993). Macmillan New York.
Hornik, K. (1993). Some new results on neural network approximation. *Neural networks*, *6*, 1069–1072.
Houde, L. (2014). *Tests du khi-deux* (Rapport technique). Université du Québec à Trois-Rivières – Département de Mathématiques et d'informatique.
Hourbracq, M., Wuillemin, P.-H., Gonzales, C., & Baumard, P. (2018). Apprentissage et sélection de réseaux bayésiens dynamiques pour les processus online non stationnaires. *Revue d'Intelligence Artificielle*, *32*(1), 75.
Jensen, F. V., & Nielsen, T. D. (2001). *Bayesian networks and decision graphs* (Vol. 2). Springer.
John-Mathews, J.-M. (2017, dec). *Causalité et condition de markov* (Rapport technique).
Johnson, S. G., & Keil, F. C. (2017). Statistical and mechanistic information in evaluating causal claims. In *Cogsci*.
Juhel, J. (2015). Modèles structuraux et inférence causale. *Différences et variabilités en psychologie*, 309–352.
Kagermann, H., Wahlster, W., & Helbig, J. (2013, apr). *Recommendations for implementing the strategic initiative industrie 4.0 – securing the future of german manufacturing industry* (Final Report of the Industrie 4.0 Working Group). München : Acatech – National Academy of Science and Engineering. Consulté sur http://forschungsunion.de/pdf/industrie_4_0_final_report.pdf
Kamienski, C., Soininen, J.-P., Taumberger, M., Dantas, R., Toscano, A., Salmon Cinotti, T., ... Torre Neto, A. (2019). Smart water management platform : Iot-based precision irrigation for agriculture. *Sensors*, *19*(2), 276.
Karlsson, L., & Bonde, O. (2020). *A comparison of selected optimization methods for neural networks.*
Karsoliya, S. (2012). Approximating number of hidden layer neurons in multiple hidden layer bpnn architecture. *International Journal of Engineering Trends and Technology*, *3*(6), 714–717.
Kayser, D., & Levy, F. (2004). Modélisations symboliques du raisonnement causal. *Intellectica*, *38*(1), 291–323.
Kechaou, F. (2020). *Construction d'un système d'aide à la décision statistico-cognitive pour le pilotage des processus d'entreprise* .
Kenny, D. A. (1979). *Correlation and causality* (1St Edition éd.). John Wiley Sons Inc.
Kiel, D., Müller, J. M., Arnold, C., & Voigt, K. I. (2017). Sustainable industrial value creation : Benefits and challenges of industry 4.0. *International Journal of Innovation Management*, *21*(8), 0–34. doi: 10.1142/S1363919617400151
Kistler, M. (2011). *La causalité.* Vuibert.
Kjærulff, U., & van der Gaag, L. C. (2013). Making Sensitivity Analysis Computationally Efficient. , 317–325. Consulté sur http://arxiv.org/abs/1301.3868
Koehler, J. J. (1996). The base rate fallacy reconsidered : Descriptive, normative, and methodological challenges. *Behavioral and Brain Sciences*, *19*(1), 1–17. doi: 10

.1017/S0140525X00041157
Kwak, S. G., & Kim, J. H. (2017). Central limit theorem : the cornerstone of modern statistics. *Korean journal of anesthesiology*, *70*(2), 144.
Kwok, T.-Y., & Yeung, D.-Y. (1997). Constructive algorithms for structure learning in feedforward neural networks for regression problems. *IEEE transactions on neural networks*, *8*(3), 630–645.
Lacour, P. (2017). Pourquoi cela est-il arrivé ? l'explication causale de l'événement chez paul ricœur. *Methodos. Savoirs et textes*(17).
Latham, P. E., & Roudi, Y. (2009). Mutual information. *Scholarpedia*, *4*(1), 1658.
Laudon, J. P., & Laudon, K. C. (2019). *Management Information Systems : Managing the Digital Firm* (16e éd.). Pearson.
Lavigne, S. E., & Forrest, J. L. (2020). Un examen-cadre des revues systématiques des preuves d'une relation causale entre les microbes parodontaux et les maladies respiratoires : Exposé de position de l'association canadienne des hygiénistes dentaires.
Leray, P. (2006). *Réseaux bayésiens : Apprentissage et diagnostic de systemes complexes* (Habilitation à diriger des recherches, Université de Rouen). Consulté sur `https://tel.archives-ouvertes.fr/tel-00485862`
Lewis, D. K. (1986). Causation.
Liebetruth, T. (2017, jan). Sustainability in Performance Measurement and Management Systems for Supply Chains. *Procedia Engineering*, *192*, 539–544. doi: 10.1016/J.PROENG.2017.06.093
Liliana, L. (2016). A new model of ishikawa diagram for quality assessment. In *Iop conference series : Materials science and engineering* (Vol. 161, p. 012099).
Liu, Z., Malone, B., & Yuan, C. (2012). Empirical evaluation of scoring functions for bayesian network model selection. In *Bmc bioinformatics* (Vol. 13, pp. 1–16).
Mackie, J. L. (1965). Causes and conditions. *American philosophical quarterly*, *2*(4), 245–264.
Mantzaris, D., Anastassopoulos, G., & Adamopoulos, A. (2011a). Genetic algorithm pruning of probabilistic neural networks in medical disease estimation. *Neural Networks*, *24*(8), 831–835.
Mantzaris, D., Anastassopoulos, G., & Adamopoulos, A. (2011b). Genetic algorithm pruning of probabilistic neural networks in medical disease estimation. *Neural Networks*, *24*(8), 831–835.
Margaritis, D. (2003). *Learning bayesian network model structure from data* (Rapport technique). Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science.
Marie, S., & Poindextre, R. (2020, dec). *L'industrie 4.0 à l'heure du plan de relance : espoirs ou désillusions ?* (Rapport technique). Wavestone. Consulté sur `https://www.wavestone.com/app/uploads/2020/11/Barometre-Industrie-2020.pdf`
Marr, B. (2012). *Key performance indicators (kpi) : The 75 measures every manager needs to know*. Pearson UK.
Martin, T. H., Howard, B. D., Mark, H. B., & Orlando, D. J. (2014). *Neural network design* (2e éd.).
Masood, T., & Sonntag, P. (2020). Industry 4.0 : Adoption challenges and benefits for SMEs. *Computers in Industry*, *121*, 103261. doi: 10.1016/j.compind.2020.103261
Mechri, W., Simon, C., & Morel, D. (2017). Retour d'expérience et modèle graphique probabiliste pour l'isolation de défaillances. In *12eme congrès international pluridisciplinaire en qualité, sureté de fonctionnement et développement durable, qualita'2017.*

Messerli, F. H. (2012). Chocolate consumption, cognitive function, and nobel laureates. *New England Journal of Medicine*, *367*(16), 1562-1564. Consulté sur https://doi.org/10.1056/NEJMon1211064 doi: 10.1056/NEJMon1211064

Microsoft. (2019). *2019 manufacturing trends report* (Rapport technique). Auteur. Consulté sur https://info.microsoft.com/rs/157-GQE-382/images/EN-US-CNTNT-Report-2019-Manufacturing-Trends.pdf

Minini, P., & Chavance, M. (2004). Observations longitudinales incomplètes : de la modélisation des observations disponibles à l'analyse de sensibilité. *Journal de la Société française de Statistique*, *145*(2), 5–18.

Mirdamadi, S. (2009). *Modélisation du processus de pilotage d'un atelier en temps réel à l'aide de la simulation en ligne couplée à l'exécution* (Thèse de doctorat). Consulté sur https://oatao.univ-toulouse.fr/7831/

Moeuf, A. (2018). *Identification des risques, opportunités et facteurs critiques de succès de l'industrie 4.0 pour la performance industrielle des PME.* (Theses, Université Paris Saclay (COmUE)). Consulté sur https://tel.archives-ouvertes.fr/tel-01849981

Moeuf, A., Pellerin, R., Lamouri, S., Tamayo-Giraldo, S., & Barbaray, R. (2018). The industrial management of smes in the era of industry 4.0. *International Journal of Production Research*, *56*(3), 1118–1136.

Moré, J. J. (1978). The levenberg-marquardt algorithm : implementation and theory. In *Numerical analysis* (pp. 105–116). Springer.

Mortureux, Y. (2002). Arbres de défaillance, des causes et d'événement.

Mouret, J.-B. (2015). *Evolutionary adaptation in natural and artificial systems* (Thèse de doctorat non publiée). Université Pierre et Marie Curie.

Naïm, P., Wuillemin, P.-H., Leray, P., Pourret, O., & Becker, A. (1999). Réseaux bayésiens. *Eyrolles, Paris*, *3*, 120.

Nakache, J.-P., & Confais, J. (2004). *Approche pragmatique de la classification : arbres hiérarchiques, partitionnements.* Editions Technip.

Neely, A., Gregory, M., & Platts, K. (1995). Performance measurement system design : a literature review and research agenda. *International journal of operations & production management*.

Neely, A., Gregory, M., & Platts, K. (2005). "Performance measurement system design : A literature review and research agenda". *International Journal of Operations and Production Management*, *25*(12), 1228–1263. doi: 10.1108/01443570510633639

Newman, J. W. (1971). *Management applications of decision theory.* HarperCollins Publishers.

Nguyen, H.-T. (2012). *Réseaux bayésiens et apprentissage ensembliste pour l'étude différentielle de réseaux de régulation génétique* (Thèse de doctorat non publiée). Université de Nantes.

Nudurupati, S., Arshad, T., & Turner, T. (2007, sep). Performance measurement in the construction industry : An action case investigating manufacturing methodologies. *Computers in Industry*, *58*(7), 667–676. doi: 10.1016/J.COMPIND.2007.05.005

Olden, J. D., & Jackson, D. A. (2002). Illuminating the "black box" : a randomization approach for understanding variable contributions in artificial neural networks. *Ecological Modelling*, *154*(1), 135-150.

Olden, J. D., Joy, M. K., & Death, R. G. (2004). An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. *Ecological modelling*, *178*(3-4), 389–397.

Panchal, G., Ganatra, A., Kosta, Y. P., & Panchal, D. (2011). Behaviour Analysis of Multilayer Perceptronswith Multiple Hidden Neurons and Hidden Layers. *International Journal of Computer Theory and Engineering*, *3*(2), 332–337. doi: 10.7763/ijcte.2011.v3.328

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems.*

Pearl, J. (1995). From bayesian networks to causal networks. In *Mathematical models for handling partial knowledge in artificial intelligence* (pp. 157–182). Springer.

Pearl, J. (2000). *Causality : Models, reasoning, and inference.* Cambridge University Press. Consulté sur `https://books.google.fr/books?id=wnGU\_TsW3BQC`

Pearl, J. (2009a). Causal inference in statistics : An overview. *Statistics surveys*, *3*, 96–146.

Pearl, J. (2009b). *Causality : Models, reasoning, and inference. second edition.* Cambridge university press.

Pillet, M. (2004). *Six sigma, comment l'appliquer.* Editions d'organisation.

Pollino, C. A., & Henderson, C. (2010). A guide for their application in natural resource management and policy. *A Technical Report No. 14. Integrated Catchment Assessment and Management Centre, Fenner School of Environment and Society, Australian National University, Canberra*(14), 48. Consulté sur `http://www.utas.edu.au/{\_}{\_}data/assets/pdf{\_}file/0009/588474/TR{\_}14{\_}BNs{\_}a{\_}resource{\_}guide.pdf`

Pouget, C., Caylou, D., & Doignies, P. (2019). Les nouveaux modes de consommation. In *Conférence printemps des etudes.*

Prestat, E. (2010). *Les réseaux bayésiens : classification et recherche de réseaux locaux en cancérologie* (Thèse de doctorat non publiée). Université Claude Bernard-Lyon I.

Prinz, A. L. (2020). Chocolate consumption and noble laureates. *Social Sciences Humanities Open*, *2*(1), 100082. Consulté sur `https://www.sciencedirect.com/science/article/pii/S2590291120300711` doi: https://doi.org/10.1016/j.ssaho.2020.100082

Pujo, P., & Kieffer, J.-P. (2002). *Fondements du pilotage des systèmes de production.*

Pérez-Alvarez, J. M., Maté, A., Gó mez López, M. T., & Trujillo, J. (2018). Tactical Business-Process-Decision Support based on KPIs Monitoring and Validation. *Computers in Industry*, *102*, 23–39. doi: 10.1016/j.compind.2018.08.001

Rabenoro, T. (2015). *Outils statistiques de traitement d'indicateurs pour le diagnostic et le pronostic des moteurs d'avions* (Theses, Université Paris 1 Panthéon Sorbonne). Consulté sur `https://hal.archives-ouvertes.fr/tel-01225739`

Reichenbach, H. (1956). *The direction of time.* Dover Publications.

Rennell, T. (2021). *Data vs metric vs kpi vs report.* Consulté sur `https://tel.archives-ouvertes.fr/tel-02968936`

Robinson, R. W. (1977). Counting unlabeled acyclic digraphs. In *Combinatorial mathematics v* (pp. 28–43). Springer.

Scanagatta, M., Salmerón, A., & Stella, F. (2019). A survey on bayesian network structure learning from data. *Progress in Artificial Intelligence*, *8*(4), 425–439.

Sheela, K. G., & Deepa, S. N. (2013). Review on methods to fix number of hidden neurons in neural networks. *Mathematical Problems in Engineering*, *2013*. doi: 10.1155/2013/425740

Shih-Hung, Y., & Yon-Ping, C. (2012a). An evolutionary constructive and pruning algorithm for artificial neural networks and its prediction applications. *Neurocomputing*,

86. doi: 10.1016/j.neucom.2012.01.024

Shih-Hung, Y., & Yon-Ping, C. (2012b). An evolutionary constructive and pruning algorithm for artificial neural networks and its prediction applications. *Neurocomputing*, *86*. doi: 10.1016/j.neucom.2012.01.024

Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters : Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv :1803.09820*.

Spears, W. M., & Anand, V. (1991). A study of crossover operators in genetic programming. In *International symposium on methodologies for intelligent systems* (pp. 409–418).

Spirtes, P., Glymour, C., & Scheines, R. (2000). *Causation, prediction, and search, 2nd edition*. MIT press.

Stamatis, D. H. (2003). *Failure mode and effect analysis - fmea from theory to execution* (2nd Edition Revised and Expanded éd.). American Society for Quality (ASQ).

Stanley, K. O., Clune, J., Lehman, J., & Miikkulainen, R. (2019). Designing neural networks through neuroevolution. *Nature Machine Intelligence*, *1*(1), 24–35.

Stark, J. (2016). *Product lifecycle management (volume 2) : The devil is in the details* (3e éd., Vol. 2). Springer International Publishing.

Stathakis, D. (2009). How many hidden layers and nodes ? *International Journal of Remote Sensing*, *30*(8), 2133–2147.

Stuart J. Russell, P. N. (2016). *Artificial intelligence : A modern approach, 3rd edition* (3e éd.). Pearson Education.

Suppes, P. (1968). *A probabilistic theory of causality*. Amsterdam : North-Holland Pub. Co.

Suppes, P. (1970). *A probabilistic theory of causality*. Amsterdam : North-Holland Pub. Co.

Talvitie, T., Eggeling, R., & Koivisto, M. (2019). Learning bayesian networks with local structure, mixed variables, and exact algorithms. *International Journal of Approximate Reasoning*, *115*, 69–95.

Taroni, F., Aitken, C. G., Garbolino, P., & Biedermann, A. (2006). *Bayesian networks and probabilistic inference in forensic science*. Wiley Chichester.

Tin-Yau, K., & Dit-Yan, Y. (1997, 5). Constructive algorithms for structure learning in feedforward neural networks for regression problems. *IEEE Transactions on Neural Networks*, *8*. doi: 10.1109/72.572102

Tsamardinos, I., Brown, L. E., & Aliferis, C. F. (2006). The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, *65*(1), 31–78.

Tu, J. V. (1996). Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical Epidemiology*, *49*. doi: 10.1016/s0895-4356(96)00002-9

Tufféry, S. (2011). *Data mining and statistics for decision making*. John Wiley & Sons.

Vandel, J. (2012). *Apprentissage de la structure de réseaux bayésiens. application aux données de génétique-génomique* (Thèse de doctorat non publiée). Université Toulouse III-Paul Sabatier.

Verriere-cuenot, P., & De Noinville, M. (2019). *Digitalisation de l'industrie française : de la performance à la croissance ?* (Rapport technique). Wavestone. Consulté sur `https://www.wavestone.com/app/uploads/2019/11/2019-Wavestone-BAROMETRE-INDUSTRIE-4.0.pdf`

Volna, E. (2010). *Neuroevolutionary optimization.*

Von Sydow, M., Hagmayer, Y., & Meder, B. (2016). Transitive reasoning distorts induction in causal chains. *Memory & cognition*, *44*(3), 469–487.

Williamson, J., et al. (2005). *Bayesian nets and causality : philosophical and computational foundations*. Oxford University Press.

Wright, S. (1934). The method of path coefficients. *The annals of mathematical statistics*, *5*(3), 161–215.

Yang, S.-H., & Chen, Y.-P. (2012). An evolutionary constructive and pruning algorithm for artificial neural networks and its prediction applications. *Neurocomputing*, *86*, 140-149. Consulté sur `https://www.sciencedirect.com/science/article/pii/S0925231212001154` doi: https://doi.org/10.1016/j.neucom.2012.01.024

Yao, X. (1993). A review of evolutionary artificial neural networks. *International journal of intelligent systems*, *8*(4), 539–567.

Yin, S., Zhu, X., & Kaynak, O. (2015). "Improved PLS focused on key-performance-indicator-related fault diagnosis". *IEEE Transactions on Industrial Electronics*, *62*(3), 1651–1658. doi: 10.1109/TIE.2014.2345331

Yu, D., & Seltzer, M. L. (2011). Improved bottleneck features using pretrained deep neural networks. In *Twelfth annual conference of the international speech communication association.*

Yu, T., & Zhu, H. (2020). Hyper-parameter optimization : A review of algorithms and applications. *arXiv preprint arXiv :2003.05689*.

Zeng, L., Lingenfelder, C., Lei, H., & Chang, H. (2008). Event-driven quality of service prediction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5364 LNCS*, 147–161. doi: 10.1007/978-3-540-89652-4-14

Zhang, Z., Beck, M. W., Winkler, D. A., Huang, B., Sibanda, W., Goyal, H., et al. (2018). Opening the black box of neural networks : methods for interpreting neural network models in clinical applications. *Annals of translational medicine*, *6*(11).

Zheng, Z., & Pavlou, P. A. (2010). Research note—toward a causal interpretation from observational data : A new bayesian networks method for structural models with latent variables. *Information Systems Research*, *21*(2), 365–391.

Annexe

Cet annexe a pour objectif d'éclaircir, ou de compléter les explications relatives à certaines notions mathématiques abordées ou utilisées dans ce manuscrit de thèse.

Théorème de Bayes

Soient A et B deux évènements de probabilités non nulles, on a :

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(B|A).\mathbb{P}(A)}{\mathbb{P}(B)} \tag{5.1}$$

Formule des probabilités totales

Considérons un système exhaustif fini d'évènements $(B_i)_{i\in I}$, tel que $\forall i \in I : \mathbb{P}(B_i) \neq 0$, alors, pour tout évènement A, on a :

$$\mathbb{P}(A) = \sum\nolimits_{i\in I} \mathbb{P}(A|B_i).\mathbb{P}(B_i) = \sum\nolimits_{i\in I} \mathbb{P}(A \cap B_i) \tag{5.2}$$

Théorème central limite

Soit X_n une suite de variables indépendantes identiquement distribuées. On note $S_n = \sum_{i=1}^{n} X_i, m = \mathbb{E}[X_1]$ et $\sigma^2 = Var(X_1) > 0$. Alors on a :

$$\frac{S_n - nm}{\sqrt{n\sigma^2}} \to \mathcal{N}(0,1) \tag{5.3}$$

Test d'hypothèse

Il s'agit d'une procédure permettant d'obtenir une règle de décision, afin de faire un choix entre deux hypothèses statistiques.

Hypothèse statistique

Il s'agit d'un énoncé émis sur les caractéristiques d'une population (distribution, paramètres, etc). Deux principaux types d'hypothèses peuvent être distingués : hypothèse nulle, et hypothèse alternative.

Hypothèse nulle

Souvent notée H_0, il s'agit de l'hypothèse considérée comme étant vraie à priori, et mise à l'épreuve par le test d'hypothèse, afin qu'elle puisse être affirmée ou infirmée en fonction d'un seuil prédéfini ; on parle alors d'acceptation ou de rejet de l'hypothèse nulle. C'est une hypothèse selon laquelle on fixe à priori un paramètre de la population à une valeur particulière.

Hypothèse alternative

Toute autre hypothèse que l'hypothèse nulle est appelée hypothèse alternative, et est souvent notée H_1.

Seuil de signification d'un test d'hypothèse

Il s'agit du risque pour lequel on accepte de se tromper, et est souvent noté α. Autrement dit, c'est le risque de rejeter à tort l'hypothèse nulle lorsqu'elle est vraie. Il peut alors être noté comme suit :

$$\alpha = \mathbb{P}(\text{Rejet de } H_0 | H_0 \text{ est vraie}) \tag{5.4}$$

Matrice Hessienne

La matrice Hessienne $H(f)$ d'une fonction f est une matrice carrée dont les coefficients correspondent aux dérivées partielles secondes de f. Considérons la fonction $f : \mathbb{R}^n \to \mathbb{R}$; $(x_1, x_2, ..., x_n) \mapsto f(x_1, x_2, ..., x_n)$. La matrice Hessienne de f est :

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \tag{5.5}$$

Méthode d'analyse de la causalité par apprentissage automatique, pour l'aide à la décision dans un contexte de supervision pour l'industrie 4.0.

RÉSUMÉ : Avec l'avènement de l'industrie 4.0, l'accélération des processus qui l'accompagne, et la prolifération des données, l'enjeu pour les processus décisionnels évoluant dans un tel contexte est d'assurer des prises de décisions rapides et fiables. Les indicateurs clé de performances (KPI) sont étroitement liés à la prise de décision : ils en sont aussi bien déclencheurs que pilotes. Ceci nous persuade que pour améliorer les processus décisionnels, une attention particulière devrait être portée sur les KPIs. Lorsqu'un KPI révèle une situation anormale, la compréhension de l'origine de cette déviation est indispensable pour rechercher des solutions, et pour en sélectionner une parmi plusieurs. Dans ces travaux de thèse, nous nous intéressons à cette compréhension, en particulier à l'identification des liens causaux entre un KPI d'intérêt et les variables contextuelles manipulables, et à la quantification de ces liens causaux. À cette fin, nous proposons un système d'aide à la décision orienté causalité, répondant à trois fonctions : l'identification des variables contextuelles liées causalement à un KPI sous forme structure causale ; la hiérarchisation de ces variables selon leurs forces respectives d'association au KPI d'intérêt ; et la possibilité de prédiction du KPI pour des fins de proactivité. La première fonction a pour objectif de permettre une meilleure compréhension des déviations du KPI. Elle est mise en œuvre grâce à un algorithme d'apprentissage des réseaux Bayésiens causaux. La deuxième fonction permet une meilleure sélection de la meilleure solution, et est implémentée grâce un calcul que nous proposons de faire sur les poids finaux d'un réseau de neurones ayant un bon pouvoir de prédiction du KPI d'intérêt. La troisième fonction permettant de prendre de décision de façon proactive, est rendue possible grâce à ce même réseau de neurones. La méthode a été validée en utilisant deux jeux de données étalons, puis comparée à d'autres techniques ayant les mêmes objectifs.

Mots clés : Causalité, apprentissage des réseaux Bayésiens, hiérarchisation des causes, réseaux de neurones, aide à la décision.

Method of causality analysis by machine learning, for decision support in a supervisory context for Industry 4.0.

ABSTRACT : With the advent of Industry 4.0 and the accompanying acceleration of processes and data proliferation, the challenge for decision-making processes is to ensure rapid and reliable decision making. Key Performance Indicators (KPIs) are closely linked to decision making : they are both triggers and drivers. Thus, in order to improve decision-making processes, focus should be on KPIs. When a KPI reveals an abnormal situation, understanding the origin of the deviation is essential to look for solutions, and to select one among several. In this thesis, we are interested in this understanding, in particular in identifying the causal links between a KPI of interest and the manipulatable contextual variables, and in quantifying these causal links. To this end, we propose a causality-oriented decision support system that fulfils three functions : the identification of contextual variables causally linked to a KPI in the form of a causal structure ; the prioritisation of these variables according to their respective strengths of association with the KPI of interest ; and the possibility of predicting the KPI for proactive purposes. The first function aims to provide a better understanding of KPI deviations. It is implemented through a causal Bayesian network learning algorithm. The second function allows a better selection of best solutiond. It is implemented thanks to a computing that we propose to make on the final weights of a neural network having a good predictive power of the KPI. The third function, allows proactive decision making, it is made possible by the same neural network. The method was validated using two benchmarks and compared to other techniques with the same objectives.

Key words : Causality, Bayesian network learning, cause ranking, neural networks, decision support.