COLLEGE ED MATHEMATIQUES
DOCTORAL STIC
PAYS DE LA LOIRE

université angers

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ D'ANGERS

ÉCOLE DOCTORALE N° 641
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Sciences et technologies de l'information et de la communication*

Par

Axel GUÉRIN

Extraction et gestion de la connaissance client par des méthodes d'apprentissage non supervisées

Thèse présentée et soutenue à Angers, le 19 janvier 2026
Unité de recherche : Laboratoire d'Etude et de Recherche en Informatique d'Angers

Rapporteurs avant soutenance :

Fadila BENTAYEB	Professeure des universités, Université de Poitiers
Allel HADJALI	Professeur des universités, Université Lumière Lyon 2

Présidente :	Florence SÈDES	Professeure des universités, Université de Toulouse
Examinateurs :	Fadila BENTAYEB	Professeure des universités, Université Lumière Lyon 2
	Allel HADJALI	Professeur des universités, Université de Poitiers
Dir. de thèse :	Frédéric SAUBION	Professeur des universités, Université d'Angers
Co-dir. de thèse :	Pierre CHAUVET	Professeur université catholique, Université Catholique de l'Ouest

Remerciements

Ces années de thèse ont été une expérience riche en travail et en réflexion.

Je remercie chaleureusement Frédéric Saubion et Pierre Chauvet, mes directeurs de thèse, pour leur confiance, leur encadrement et la qualité de leurs conseils tout au long de ce travail.

Je tiens à remercier Christophe Courtois, à l'origine de cette thèse, pour ses échanges constructifs et son regard complémentaire, à la fois scientifique et opérationnel.

Je remercie les membres du jury, Fadila Bentayeb, Allel Hadjali et Florence Sèdes, pour l'honneur qu'ils m'ont fait en acceptant d'évaluer ce travail.

Je tiens à remercier sincèrement Cloé Gelineau et Lisa Relion, avec qui j'ai eu le plaisir de travailler au quotidien, nos nombreuses discussions ont largement contribué à la qualité de ce travail.

Un grand merci aux doctorants et collègues pour les discussions scientifiques, les pauses café salvatrices et le soutien moral.

Je remercie l'ensemble des membres du LERIA pour leur accueil et l'environnement scientifique stimulant.

Je remercie le personnel administratif et technique pour son aide précieuse et son efficacité.

Enfin, je remercie ma famille, et en particulier mes parents, pour son soutien indéfectible tout au long de ce parcours.

TABLE DES MATIÈRES

Introduction **9**

1 Les cartes autoadaptatives **17**
1.1 L'Algorithme SOM 17
1.2 Visualisation 21
1.3 Extension des SOM en 3 dimensions 22
1.4 Distance 24
1.4.1 Distances classiques 24
1.4.2 Une nouvelle distance pondérée 25

2 Évolutions récentes des SOM **31**
2.1 Les cartes auto-organisatrices : la vision de T. Kohonen 31
2.1.1 L'article fondateur : The Self-Organizing Map 31
2.1.2 Impact dans la communauté scientifique 32
2.1.3 Avancées récentes dans le domaine des SOM 33
2.2 Gestion des données 36
2.2.1 Datasets semi-supervisés 37
2.2.2 Datasets perturbés 38
2.2.3 Données catégorielles 39
2.2.4 Distributions spécifiques des données 41
2.2.5 Datasets volumineux 42
2.2.6 Données temporelles et séquences 43
2.3 Topologies et Métriques 45
2.3.1 Topologies 45
2.3.2 Distances 48
2.3.3 Relations de voisinage 50
2.4 Techniques d'apprentissage 54
2.4.1 Approche générale d'apprentissage 54
2.4.2 Extraction de caractéristiques 56

2.4.3 Phase d'entraînement . . . 57
2.4.4 Clustering . . . 58
2.5 Visualisation . . . 60
2.6 Performances . . . 63
2.7 Hyperparamétrage . . . 65
2.7.1 Convergence . . . 66
2.7.2 Paramètres . . . 67
2.8 Conclusion . . . 70
2.9 Application aux produits clients . . . 71

3 Gestion des données **77**
3.1 Données manquantes . . . 77
3.2 Données qualitatives . . . 81
3.3 Données textuelles . . . 83
3.4 Données temporelles . . . 84

4 Hyperparamétrage **87**
4.1 Score . . . 87
4.1.1 Mesures classiques de distances . . . 87
4.1.2 Mesures spécifiques aux cartes SOM . . . 88
4.1.3 Erreur de quantification . . . 89
4.1.4 Distortion . . . 89
4.1.5 Erreur topographique . . . 90
4.1.6 Erreur combinée . . . 91
4.1.7 Produit topographique . . . 91
4.1.8 Erreur de Kruskal-Shepard . . . 92
4.1.9 C mesure . . . 93
4.1.10 Fonction topographique . . . 94
4.1.11 Normalisation . . . 95
4.1.12 Autre approche : analyse du rendu visuel . . . 96
4.2 Qualité de l'apprentissage . . . 96
4.2.1 Analyse expérimentale . . . 97
4.2.2 Extension aux SOM en 3 dimensions . . . 99
4.2.3 Mise en oeuvre d'un score combiné . . . 102
4.2.4 Étude du critère d'arrêt . . . 106

4.3 Paramétrage automatique de SOM via SMAC . . . 106
4.3.1 Expérimentation . . . 108
4.4 SOM 3D . . . 111

5 SOM alternative **115**
5.1 Introduction . . . 115
5.1.1 Problématique . . . 115
5.1.2 Approche générale . . . 116
5.2 Générateurs continus de SOM . . . 117
5.2.1 Cadre général de génération et optimisation . . . 117
5.2.2 MLP . . . 121
5.2.3 MLP avec Fourier features . . . 122
5.2.4 SIREN : réseaux à activations sinusoïdales . . . 123
5.2.5 Réseaux RBF . . . 125
5.2.6 Régression ridge à noyau (KRR) . . . 126
5.2.7 CNN CoordConv . . . 127
5.2.8 Comparaison des modèles . . . 130
5.2.9 Justification des modèles étudiés . . . 131
5.3 Protocole expérimental . . . 131
5.3.1 Optimisation . . . 131
5.3.2 Métriques . . . 132
5.3.3 Statistiques et protocole . . . 133
5.3.4 Résultats . . . 135
5.3.5 Analyse des résultats . . . 136

6 Application industrielle **139**
6.1 Contexte industriel et besoin spécifique de l'entreprise . . . 139
6.1.1 Objectifs . . . 139
6.1.2 Création et enrichissement des données . . . 142
6.2 Developpement de l'application . . . 146
6.2.1 Interface de Sélection des Clients . . . 146

Conclusion **151**

Bibliographie **155**

A Annexes **169**

A.1 Exemples de visualisation des cartes SOM 169

A.1.1 SOM bidimensionnelles 169

A.1.2 SOM tridimensionnelles 174

A.2 Qualité de l'apprentissage 176

A.2.1 S'_1 176

A.2.2 $S_{\Delta t}(S'_1)$ 180

A.3 Algorithmes générateurs 184

A.3.1 MLP 184

A.3.2 MLPFF 185

A.3.3 SIREN 186

A.3.4 RBF 187

A.3.5 KRR 188

A.3.6 CNN 189

A.4 Application Web 190

A.4.1 Accueil et Authentification 190

A.4.2 Sélection 191

A.4.3 Menu 193

A.4.4 Tableau de bord 194

A.4.5 Persona 206

A.4.6 ABC 207

A.4.7 RFM 208

A.4.8 SOM 209

INTRODUCTION

Contexte et problématique

L'Intelligence Artificielle (IA) est aujourd'hui un vecteur majeur de la transformation des processus commerciaux, notamment pour les entreprises B2B (Business-to-Business). Les solutions basées sur l'IA permettent aux entreprises de traiter des quantités massives de données, d'extraire des insights précis, et d'automatiser les processus décisionnels. Les applications de l'IA dans le B2B vont de la gestion des relations clients et des ventes à l'optimisation de la chaîne logistique, en passant par la personnalisation des offres.

L'entreprise Octave, spécialisée dans les solutions de gestion omnicanale, est un exemple pertinent de cette transformation numérique dans le B2B. Son modèle d'affaires repose sur le développement de solutions logicielles qui centralisent et connectent les différents canaux de vente (e-commerce, points de vente physiques, gestion de stocks, etc.) pour ses clients.

En intégrant des technologies d'IA, Octave peut exploiter les données collectées sur les comportements d'achat pour fournir des recommandations fondées sur des segments de clients, anticiper les tendances de consommation et adapter les stratégies de vente aux besoins spécifiques de chaque segment. Parmi ces approches, les modèles d'apprentissage non supervisé, tels que les Self-Organizing Maps (SOM), occupent une place centrale, car ils offrent une représentation visuelle et interprétable de structures complexes. En projetant des données multidimensionnelles sur une carte bidimensionnelle, les SOM permettent de rendre visibles les relations cachées entre clients et de faire émerger des motifs sous-jacents dans les comportements d'achat. Cette capacité de visualisation topologique facilite non seulement la classification des clients, mais aussi la compréhension des facteurs d'influence, des préférences et des dynamiques d'évolution de chaque segment. Les SOM deviennent ainsi un véritable outil d'exploration et d'explication des données, transformant des ensembles d'informations brutes en connaissances exploitables pour la stratégie commerciale.

La transformation numérique entraîne également de nombreux défis, en particulier en ce qui concerne la gestion et l'analyse des données. Pour les entreprises comme Octave,

les challenges liés à l'analyse des données sont multiples et complexes :

- **Gestion de Grandes Volumétries de Données** : Avec l'augmentation des données générées par les transactions, les interactions clients et les opérations commerciales, les entreprises doivent faire face à des volumes de données considérables.
- **Intégration de Données Hétérogènes** : Les données en entreprise proviennent de sources diverses, telles que les systèmes de gestion des stocks, les plateformes de vente en ligne, les caisses en magasin, et les outils de gestion de la relation client (CRM). Intégrer ces données provenant de formats, de structures et de fréquences différentes représente un défi technique majeur. La consolidation de ces données est essentielle pour créer une vue d'ensemble cohérente et précise des activités de l'entreprise et des comportements des clients.
- **Extraction d'Informations Exploitables** : Les entreprises B2B ont besoin d'outils d'analyse qui vont au-delà de la simple visualisation des données. Elles doivent être capables de transformer les données en informations exploitables qui soutiennent la prise de décision. Cela signifie fournir aux équipes commerciales et décisionnelles des insights détaillés, comme les tendances de vente, les prévisions de demande, et les profils de clients.
- **Précision et Fiabilité des Données** : Avec des données provenant de différentes sources, la précision des données devient cruciale. Les erreurs, les doublons et les valeurs manquantes doivent être identifiés et traités, car ils peuvent fausser les analyses et mener à des conclusions erronées.
- **Adaptabilité et Scalabilité des Outils** : Les modèles doivent être suffisamment flexibles pour s'adapter aux nouvelles données, aux évolutions des marchés et aux changements dans les comportements des clients. Dans le cas d'Octave, il est essentiel que les outils mis en place puissent évoluer au même rythme que les besoins des clients. Les SOM, par exemple, permettent une segmentation dynamique et évolutive, mais nécessitent un hyperparamétrage et une optimisation continue pour rester performants.

Ces défis montrent que l'analyse de données ne consiste pas uniquement à disposer de grandes quantités d'informations, mais aussi à pouvoir les structurer, les comprendre et les exploiter de manière efficace. Cela nécessite des outils avancés, des techniques d'optimisation et des méthodes de gestion de données robustes, capables de transformer les données brutes en ressources stratégiques.

Objectifs de la thèse

Le principal objectif de cette thèse est de contribuer à l'optimisation et à la digitalisation des processus d'analyse de données dans un contexte commercial en utilisant les Self-Organizing Maps (SOM) et des techniques d'optimisation avancées. Ces approches, adaptées aux grands volumes de données complexes, permettent de mieux comprendre les comportements clients et d'optimiser les processus stratégiques des entreprises.

Les SOM, un type de réseau de neurones non supervisé, sont reconnus pour leur capacité à représenter visuellement des données multidimensionnelles et à identifier des segments naturels ou des motifs cachés. Leur application dans le cadre de cette thèse vise à démontrer leur utilité pour :

- **Segmenter des données de manière intuitive** : Les SOM regroupent les données en clusters visuels, facilitant ainsi l'identification des segments de marché et la compréhension des comportements clients [100, 17, 103].
- **Révéler des tendances et des comportements cachés** : Leur capacité à traiter des données hétérogènes et volumineuses permet de mettre en évidence des motifs difficiles à détecter avec des méthodes traditionnelles [87, 50].
- **Améliorer les décisions stratégiques** : En visualisant les clusters et les tendances [73, 81, 104], les entreprises peuvent adopter des stratégies mieux adaptées aux caractéristiques et aux besoins spécifiques de leurs segments clients.

Un autre axe majeur de la thèse concerne l'optimisation des SOM. Cela inclut le réglage fin des hyperparamètres, l'exploration d'architectures avancées telles que les SOM tridimensionnels pour une meilleure visualisation, et l'utilisation de techniques d'optimisation multicritères pour accroître la pertinence des segments identifiés.

Dans cette thèse CIFRE, les problématiques explorées incluent la segmentation des clients, l'enrichissement des données par des sources externes (comme les données météorologiques ou géographiques), et l'optimisation des prises de décision stratégiques.

Méthodologie et contributions

Pour atteindre les objectifs énoncés plus haut, cette thèse s'appuie sur une méthodologie rigoureuse combinant des aspects fondamentaux et pratiques, notamment :

- **Revue des fondements des SOM** : La première étape de la recherche consiste en une analyse approfondie des Self-Organizing Maps et des méthodes d'optimisa-

tion existantes. Ce cadre inclut une présentation des travaux de Teuvo Kohonen, le pionnier des SOM, et une exploration des extensions et améliorations algorithmiques qui ont été apportées à cette technique au fil des années.

— **Mise en œuvre des SOM dans un contexte de données hétérogènes** : La méthodologie inclut l'intégration de différentes sources de données, telles que des données géographiques, des données météorologiques, et des informations foncières, dans les SOM. La gestion de données de diverses natures (manquantes, qualitatives, temporelles, etc.) et leur transformation en formats exploitables pour les SOM sont des aspects clés du cadre méthodologique.

— **Optimisation des modèles SOM** : Une partie importante de la méthodologie consiste à optimiser les hyperparamètres des SOM et à explorer des architectures avancées, comme les SOM en trois dimensions. Des techniques d'optimisation multicritères, telles que SMAC (Sequential Model-based Algorithm Configuration), sont mises en œuvre pour maximiser la pertinence des modèles en fonction des objectifs commerciaux d'Octave.

— **Exploration de modèles alternatifs de SOM** : Une partie de la recherche s'attache à reformuler les cartes auto-organisatrices en adoptant une approche générative continue. Plutôt que d'ajuster les poids neuronaux de manière itérative, la carte entière est produite par un réseau de type MLP (ou ses variantes à encodage fréquentiel, activations sinusoïdales, modèles à noyaux ou CNN CoordConv). L'optimisation des paramètres repose sur des méthodes évolutionnaires, notamment CMA-ES, permettant de s'affranchir des contraintes de différentiabilité et d'utiliser des métriques globales. Cette approche offre une alternative robuste à la SOM classique, conciliant compacité, continuité et flexibilité topologique.

— **Validation empirique et application industrielle** : Enfin, la méthodologie prévoit une validation empirique des modèles développés dans le cadre du partenariat avec Octave. Des études de cas concrets, comme l'analyse des clients de Cash Piscine, permettent de tester et de démontrer l'utilité des modèles SOM pour des applications industrielles, tout en offrant des perspectives de recherche future.

En combinant ces éléments théoriques et pratiques, cette thèse vise à démontrer que les Self-Organizing Maps, optimisés et adaptés aux spécificités du secteur B2B, représentent une solution performante pour extraire des informations stratégiques à partir de données complexes et ainsi transformer les données en un véritable atout pour la prise de décision.

Données utilisées

Les données utilisées dans le cadre de cette thèse proviennent d'un environnement commercial réel, ce qui permet d'appliquer et de valider les méthodes proposées sur des cas concrets. Ces données incluent :

- **Données démographiques** : Informations telles que l'âge, la localisation, ou le type de client, qui permettent de caractériser les profils des utilisateurs.
- **Données transactionnelles** : Historique des achats comprenant les montants dépensés, la fréquence des transactions, et les catégories de produits achetés.
- **Données comportementales** : Analyse des préférences et des habitudes des clients, incluant la saisonnalité des achats et les comportements spécifiques.
- **Données externes** : Informations contextuelles enrichissant les analyses, telles que des données météorologiques ou des indicateurs géographiques.

Ces données, souvent hétérogènes et volumineuses, représentent un défi en termes d'intégration et d'analyse. Leur traitement, normalisation, et segmentation constituent des étapes clés pour exploiter pleinement leur potentiel. En utilisant des techniques d'intelligence artificielle comme les Self-Organizing Maps (SOM), cette thèse vise à extraire des insights stratégiques à partir de ces données, permettant d'optimiser les décisions commerciales. Ce corpus de données sert également à démontrer l'efficacité des modèles proposés dans des contextes industriels variés.

Résumé des Chapitres

Chapitre 1 : Les cartes autoadaptatives Le premier chapitre introduit les Self-Organizing Maps (SOM), en expliquant leur fonctionnement de base et leurs fondements théoriques. Développée par Teuvo Kohonen, cette méthode repose sur un réseau de neurones non supervisé, conçu pour structurer et visualiser des données multidimensionnelles. Les SOM sont particulièrement utiles pour identifier des motifs et des clusters naturels dans les données, sans recourir à des classifications prédéfinies. Ce chapitre explore également les différentes mesures de distance qui permettent aux SOM de regrouper les données selon leur similarité. En visualisant les données sous la forme de cartes topologiques, les SOM facilitent l'analyse et l'interprétation de données complexes.

Chapitre 2 : Evolutions récentes des SOM Ce chapitre présente une revue des travaux fondamentaux de Teuvo Kohonen et retrace les principales évolutions des algorithmes SOM. Des aspects tels que les extensions et améliorations méthodologiques, ainsi que les adaptations spécifiques à certaines applications, y sont explorés. Ce chapitre traite également de l'hyperparamétrage, un élément clé pour optimiser les performances des SOM et ajuster le modèle en fonction des spécificités des données. Enfin, les applications des SOM dans le secteur commercial sont discutées, mettant en évidence leur capacité à extraire des informations exploitables pour la prise de décision et la stratégie d'entreprise.

Chapitre 3 : Gestion des Données La gestion des données représente un défi central pour l'application des SOM en entreprise. Ce chapitre aborde les différentes problématiques de données rencontrées dans le projet, telles que la gestion des données manquantes, des données qualitatives, textuelles, et temporelles. Les stratégies pour traiter et enrichir ces données sont détaillées, en expliquant comment elles sont intégrées dans les modèles SOM pour en améliorer la pertinence et la robustesse. En utilisant des techniques de nettoyage, de normalisation et d'enrichissement, ce chapitre montre comment transformer les données brutes en informations utilisables dans les modèles d'analyse.

Chapitre 4 : Hyperparamétrage Ce chapitre se concentre sur l'optimisation des SOM pour maximiser leur efficacité. Les hyperparamètres des SOM, tels que le score, les paramètres temporels T_max et ϵ_0, ainsi que le critère d'arrêt, sont expliqués en détail. Le chapitre couvre également des techniques d'optimisation avancées, comme l'algorithme SMAC (Sequential Model-based Algorithm Configuration), qui permet d'ajuster automatiquement les paramètres en fonction des objectifs de performance. En outre, ce chapitre présente les SOM en trois dimensions (3D), qui offrent une représentation visuelle plus intuitive des clusters, améliorant ainsi l'interprétation des résultats et la prise de décision en entreprise.

Chapitre 5 : SOM alternative Ce chapitre reformule les SOM : plutôt que d'ajuster chaque neurone pas à pas, la carte est produite d'un bloc par une fonction générative continue (MLP, variantes avec encodage fréquentiel, réseaux à activations sinusoïdales, modèles à noyaux, CNN CoordConv). Les paramètres sont optimisés par une stratégie évolutionnaire (CMA-ES), ce qui contourne la dépendance au gradient et facilite l'usage de métriques globales. Un protocole commun compare ces approches à la SOM classique

selon trois critères complémentaires : qualité de quantification, préservation du voisinage et équilibre d'usage de la carte, agrégés dans un score global. À budget d'optimisation moyen ou élevé, les modèles génératifs sont des alternatives solides—souvent meilleures—tout en offrant compacité, continuité et flexibilité topologique.

Chapitre 6 : Application Industrielle Le dernier chapitre se concentre sur les applications pratiques des SOM dans un cadre industriel, grâce au partenariat avec Octave. Il présente le contexte industriel, les besoins spécifiques de l'entreprise et les défis rencontrés. Ce chapitre détaille les applications concrètes des modèles et méthodes développés, en montrant comment les SOM ont été utilisés pour enrichir les données clients, segmenter les comportements d'achat, et optimiser la prise de décision. En particulier, le cas de Cash Piscine est exploré pour illustrer l'impact des SOM sur l'identification des segments clients et l'amélioration des stratégies commerciales. Cette section montre comment les techniques développées tout au long de la thèse permettent de répondre aux problématiques industrielles réelles et de démontrer l'intérêt des SOM dans un contexte commercial.

Chapitre 1

Les cartes autoadaptatives

1.1 L'Algorithme SOM

L'algorithme SOM [61] (Self-Organizing Map ou carte auto-organisatrice) est un algorithme d'apprentissage non supervisé utilisé pour générer une représentation à faible dimension (généralement bi-dimensionnelle) d'un ensemble de données de dimension supérieur. Par exemple, si nous avons un ensemble de données avec p variables mesurées pour n observations, une carte SOM peut être utilisée pour regrouper les observations similaires et les représenter visuellement sur une carte.

Exemple : Considérons un ensemble de données comportant les caractéristiques de clients d'un site e-commerce, telles que l'âge, le revenu et le montant moyen dépensé. Avec une carte SOM, ces données multidimensionnelles peuvent être projetées sur une carte bidimensionnelle, où chaque point représente un client. Les clients présentant des comportements similaires (par exemple, des jeunes à revenu moyen dépensant beaucoup) seront regroupés dans des régions proches sur la carte. Cela permet d'identifier des segments de clientèle de manière intuitive.

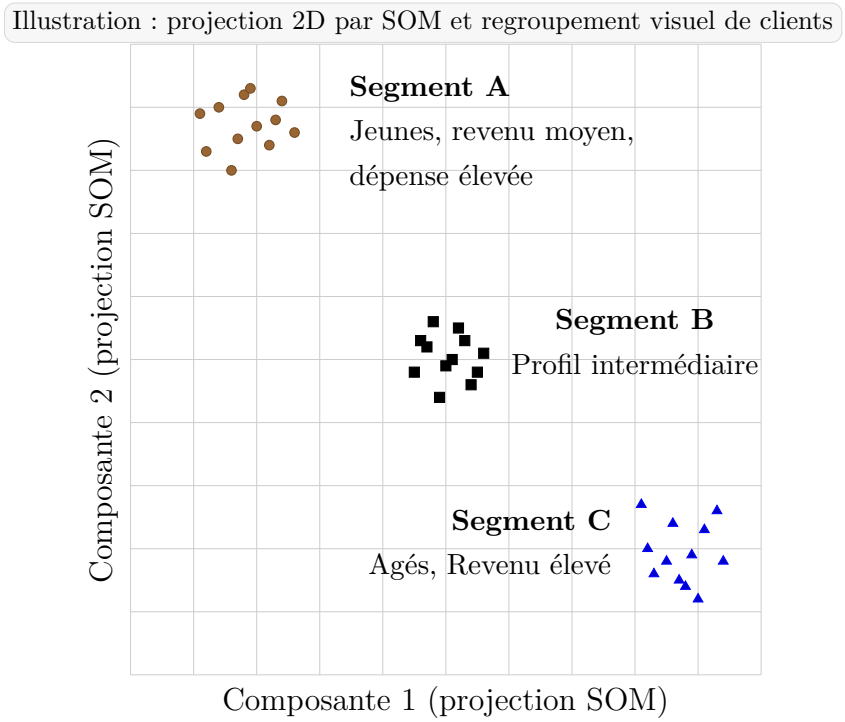


FIGURE 1.1 – Exemple illustratif : des clients aux profils proches se projettent dans des régions voisines de la carte, ce qui fait émerger des segments lisibles d'un coup d'œil.

Également nommées cartes de Kohonen ou réseau de Kohonen, le modèle a été introduit en 1982 par le chercheur finlandais Teuvo Kohonen. Le processus de formation d'une SOM repose sur des principes compétitifs plutôt que sur la correction d'erreurs, comme c'est le cas pour d'autres types de réseaux de neurones artificiels. La conceptualisation de la carte de Kohonen s'appuie sur des modèles inspirés de la biologie, notamment des modèles neuronaux des années 1970 et des théories de la morphogenèse formulées par Alan Turing dans les années 1950. En somme, les cartes auto-organisées sont des outils puissants pour la visualisation et l'analyse de données complexes. Elles permettent de regrouper les données similaires et de représenter ces regroupements de manière visuelle, facilitant ainsi la compréhension des structures sous-jacentes et des relations entre les données multidimensionnelles.

Les cartes auto-organisées, tout comme la plupart des architectures de réseaux neuronaux artificiels, opèrent dans deux modes distincts : l'entraînement et la cartographie. Dans la phase d'entraînement, un ensemble de données d'entrée (représentant l'"espace d'entrée") est utilisé pour générer une représentation de dimension réduite des données initiales (définissant l'"espace de la carte"). Par la suite, lors de la phase de cartographie,

des données d'entrée supplémentaires sont classées en utilisant la carte ainsi générée.

Dans la plupart des scénarios, l'objectif de l'entraînement consiste à projeter un espace d'entrée de dimension p sur un espace cartographique de deux dimensions. Plus précisément, si l'espace d'entrée comprend p variables, il est dit avoir une dimension p. L'espace cartographique est structuré en unités appelées "nœuds" ou "neurones", disposés en une grille hexagonale ou rectangulaire à deux dimensions. Préalablement déterminé en fonction des objectifs globaux d'analyse et d'exploration des données, le nombre et la disposition des nœuds sont spécifiés.

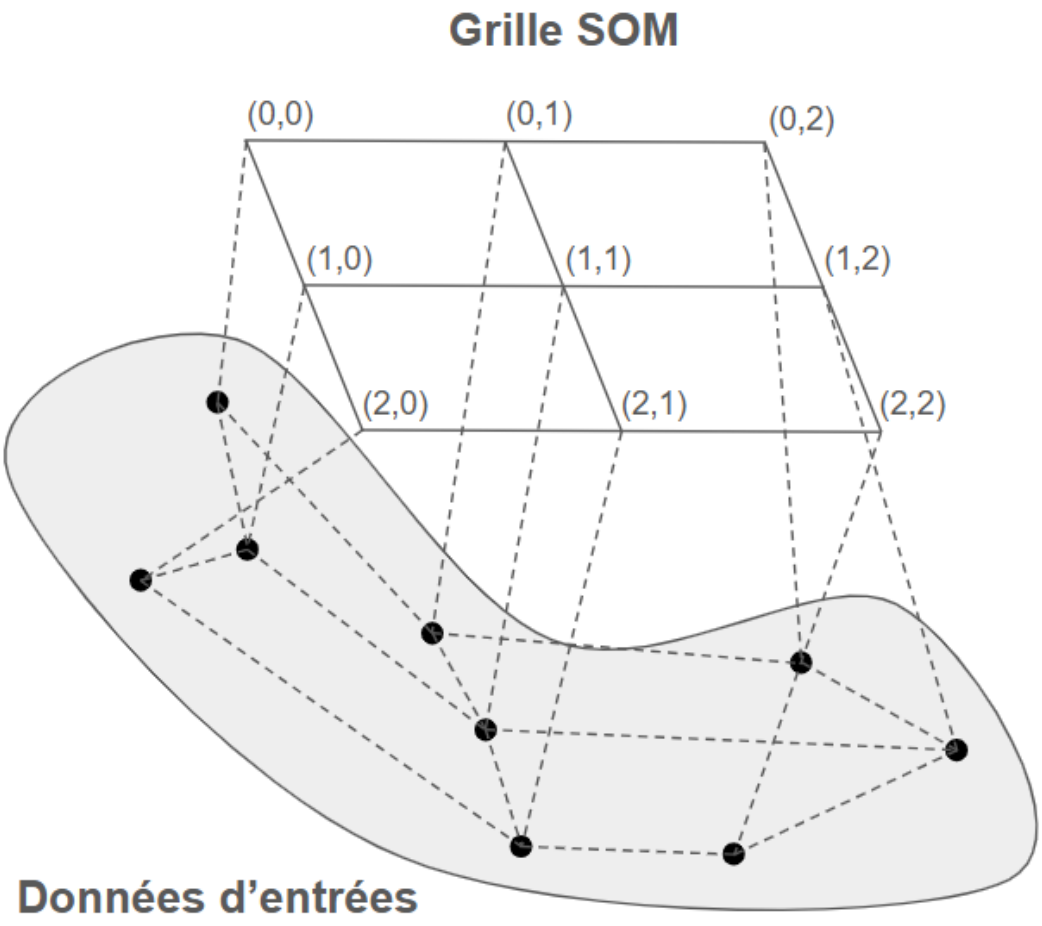


FIGURE 1.2 – Schema d'une grille SOM

Chaque nœud au sein de l'espace cartographique est associé à un vecteur de "poids", lequel reflète la position du nœud dans l'espace d'entrée. Bien que les nœuds au sein de

l'espace cartographique demeurent fixes, la phase d'entraînement implique le déplacement des vecteurs de poids vers les données d'entrée (en minimisant une mesure de distance telle que la distance euclidienne), tout en préservant la topologie qui découle de la structure de l'espace cartographique. Une fois l'entraînement accompli, la carte est en mesure de classifier des observations additionnelles dans l'espace d'entrée en identifiant le nœud dont le vecteur de poids est le plus proche (selon la plus petite mesure de distance) du vecteur associé à l'observation en question.

Une carte auto-organisatrice est définie comme un couple

$$\mathcal{M} = (E, W)$$

où :

— $E = \{e_1, e_2, \ldots, e_K\}$ est l'ensemble des *nœuds* ou *neurones*, disposés selon une topologie donnée (souvent une grille régulière 2D) ;
— $W = \{w_i \in \mathbb{R}^p \mid i \in [\![1, K]\!]\}$ est l'ensemble des *vecteurs de poids* associés à ces nœuds, définissant leur position dans l'espace d'entrée $\mathbb{R}^p$.

L'ensemble E est muni :

— d'une *topologie discrète* : tout sous-ensemble de E est un ouvert ;
— d'une *distance de carte* $d^* : E \times E \to \mathbb{R}^+$, correspondant au nombre minimal de polygones reliant deux nœuds voisins sur la grille (selon la topologie choisie — carrée, triangulaire ou hexagonale).

Différentes formes géométriques peuvent être utilisées pour représenter le réseau de neurones. Les trois pavages les plus courants sont les grilles de triangles, de carrés ou d'hexagones, qui garantissent une couverture régulière de l'espace cartographique.

L'algorithme ci-dessous est celui utilisé le plus couramment, des adaptations ou variations de ce dernier existent en fonction des cas d'usages.

Algorithm 1 Self-Organizing Maps

Initialisation
$\epsilon_0 \leftarrow$ pas de l'algorithme
$\sigma_0 \leftarrow$ rayon de la carte
$T_{max} \leftarrow$ nombre d'itérations
$n \leftarrow$ nombre de données d'entrées
$p \leftarrow$ nombre de variables
$X \leftarrow$ matrice des entrées normalisées de taille $n \times p$
$d \leftarrow$ distance entre 2 vecteurs
$d^* \leftarrow$ distance entre 2 polygones sur la carte (dépend de la topologie)
$E \leftarrow$ carte des poids, $E = \{W_i, i \in [\![1, 4\sigma_0^2]\!]\}$, initialisée aléatoirement
for $t \leftarrow 1$ **to** T_{max} **do**
    Prendre X^* aléatoirement, vecteur de X
    Chercher W^* le noeud de la carte le plus proche de X^*
    $W^* = Inf(d(W, X^*), W \in E)$
    Mise à jour des poids des neurones voisins de W^*
    for $i \leftarrow 1$ **to** $4\sigma_0^2$ **do**
        $W_i(t+1) \leftarrow W_i(t) + \Theta(t)\epsilon(t)(X^* - W_i(t))$, avec :
        $\Theta(t) = e^{-d^*(W^*, W_i)/2\sigma^2(t)}$
        $\sigma(t) = \sigma_0 e^{-t/T_{max}}$
        $\epsilon(t) = \epsilon_0 e^{-t/T_{max}}$
    end for
end for

1.2 Visualisation

Le résultat de l'algorithme SOM est la carte des poids E, contenant pour chaque neurone un vecteur de dimension p correspondant aux p variables d'entrée. Chaque observation peut être associée à son neurone le plus proche, de la même manière que durant la phase d'entraînement. Ainsi, chacune des variables peut être représentée sous forme de carte, où une couleur est associée à chaque nœud en fonction de la valeur correspondante du poids.

La visualisation constitue un des principaux atouts des cartes auto-organisatrices. Elle permet de représenter des données multidimensionnelles de manière intuitive et interpré-

table, en facilitant l'identification des zones homogènes et des frontières de clusters. Selon la topologie choisie (carrée, triangulaire ou hexagonale), la carte prend des formes différentes mais conserve la même logique de représentation topologique.

Les exemples de cartes illustrant ces différents pavages, ainsi que les panels complets correspondant à plusieurs variables d'un jeu de données, sont présentés en **annexe A**.

1.3 Extension des SOM en 3 dimensions

Les cartes auto-organisées (SOM) sont traditionnellement conçues comme des cartes bi-dimensionnelles, offrant un moyen visuellement intuitif de représenter des données complexes. Cependant, il est possible d'étendre les SOM à des dimensions supérieures, notamment à trois dimensions. Cette extension augmente la capacité de la carte à modéliser des relations plus complexes entre les données, mais elle présente également des défis significatifs.

Lorsqu'on augmente le nombre de dimensions d'une carte SOM à trois, l'aspect visuel et l'interprétabilité peuvent se compliquer. Bien que les cartes tridimensionnelles permettent une représentation plus riche des données, elles perdent en clarté visuelle, rendant l'analyse et l'interprétation moins directes. De plus, la visualisation de telles cartes nécessite souvent des outils et des techniques spécifiques pour explorer efficacement l'espace tri-dimensionnel.

Par ailleurs, le nombre de neurones dans une carte SOM augmente exponentiellement avec l'ajout de dimensions. Dans une carte tridimensionnelle, ce phénomène se traduit par une augmentation rapide du nombre de neurones nécessaires, ce qui peut entraîner des défis computationnels et une complexité accrue dans l'entraînement et l'optimisation du modèle.

Concernant le pavage dans une carte tridimensionnelle, la structuration de l'espace devient plus complexe. L'utilisation de cubes comme unités de base pour le pavage est une approche courante, car elle offre une manière structurée et systématique de diviser l'espace tridimensionnel. Cependant, cette méthode simplifie la représentation de l'espace de données et peut ne pas capturer entièrement la complexité inhérente à certaines distributions de données.

Des exemples illustrant ces cartes tridimensionnelles et les panels associés sont présentés en **annexe A.1.1**, afin d'illustrer la structure et la visualisation des SOM dans un espace 3D.

L'introduction d'une dimension supplémentaire dans les cartes auto-organisatrices, passant de configurations bidimensionnelles (SOM 2D) à tridimensionnelles (SOM 3D), offre une capacité accrue à capturer et à modéliser des structures de données plus complexes. Cette caractéristique rend les SOM 3D particulièrement adaptées pour analyser des jeux de données où les relations spatiales ou les modèles intrinsèques ne peuvent être pleinement appréhendés dans un espace à deux dimensions. En effet, la dimension additionnelle permet une séparation plus fine et une représentation plus riche des clusters de données, facilitant ainsi la découverte de insights plus profonds.

Cependant, cette augmentation de la dimensionnalité n'est pas sans conséquences. Le temps de calcul requis pour l'entraînement et la convergence des SOM 3D est considérablement plus long par rapport aux configurations 2D, en raison de la complexité accrue des calculs et du nombre de neurones impliqués. Cette réalité pose des défis significatifs, surtout dans des contextes où les ressources computationnelles sont limitées ou lorsque le temps de réponse rapide est crucial.

En outre, un autre aspect qui se complexifie avec les SOM 3D est la visualisation des résultats. Si les cartes 2D peuvent être facilement représentées et interprétées à l'aide de graphiques ou de heatmaps, la visualisation des SOM 3D nécessite des outils plus avancés, comme des visualisations 3D interactives, qui peuvent ne pas être immédiatement accessibles ou intuitifs pour tous les utilisateurs. Cette difficulté dans la visualisation directe peut rendre l'analyse et l'interprétation des résultats plus ardues, exigeant ainsi des approches créatives ou des méthodes de réduction dimensionnelle pour présenter les données de manière compréhensible.

En dépit des défis inhérents à la visualisation des résultats des SOM 3D, on peut envisager le développement d'outils de visualisation avancés et interactifs comme une voie prometteuse pour surmonter ces obstacles. Ces outils, en exploitant des techniques de visualisation tridimensionnelle et d'interaction utilisateur sophistiquées, pourraient grandement faciliter l'exploration et l'analyse des cartes 3D, rendant les structures complexes non seulement visibles mais aussi intuitivement compréhensibles.

L'introduction de l'interactivité, par exemple, permettrait aux utilisateurs de manipuler la visualisation en temps réel, de zoomer sur des zones d'intérêt, de tourner la vue pour explorer les données sous différents angles, et potentiellement même de filtrer dynamiquement des clusters ou des caractéristiques spécifiques. Ces capacités enrichiraient considérablement l'expérience analytique, permettant une immersion plus profonde dans les données et facilitant la découverte de patterns et d'insights cachés.

Cependant, le développement de tels outils de visualisation interactifs représente un investissement substantiel en termes de temps et de ressources. La conception d'interfaces utilisateur intuitives, le codage de fonctionnalités interactives avancées, et l'optimisation des performances pour gérer des ensembles de données volumineux en 3D sont des tâches complexes qui exigent une expertise spécialisée en développement logiciel et en visualisation de données. De plus, il faut également considérer l'accessibilité de ces outils pour s'assurer qu'ils soient utilisables par une large gamme d'utilisateurs, y compris ceux qui n'ont pas de compétences techniques approfondies.

En conclusion, bien que les SOM 3D offrent des avantages indéniables pour l'analyse de données complexes grâce à leur capacité à modéliser des structures tridimensionnelles, elles s'accompagnent de défis notables en termes de temps de calcul et de visualisation. Ces considérations doivent être soigneusement pesées lors du choix de la configuration des SOM pour une tâche d'analyse de données spécifique.

1.4 Distance

Au cœur des cartes auto-organisatrices (SOM) réside le principe fondamental de l'organisation spatiale des données à travers une grille de neurones. Comme présenté dans l'algorithme 1, chaque itération d'apprentissage repose sur la mesure d'une distance [101, 99] entre un vecteur d'entrée X^* et les vecteurs de poids W_i associés aux neurones de la carte. Cette mesure conditionne la sélection du neurone gagnant ainsi que la mise à jour des poids de ses voisins. Le choix de la distance $d(\cdot, \cdot)$ est donc un élément déterminant du comportement et des performances du modèle.

1.4.1 Distances classiques

Différentes mesures de distance peuvent être employées selon la nature des données et les objectifs de l'analyse. Elles influencent directement la capacité des SOM à modéliser la structure sous-jacente des données et à générer une classification pertinente. Les plus courantes sont :

— **Distance euclidienne (L_2)** :

$$d_{eucl}(x, y) = \sqrt{\sum_{i=1}^{p}(x_i - y_i)^2}$$

C'est la distance la plus utilisée dans les SOM, appréciée pour sa simplicité et son interprétation géométrique naturelle.

— **Distance de Manhattan (L_1)** :

$$d_{man}(x, y) = \sum_{i=1}^{p} |x_i - y_i|$$

Moins sensible aux valeurs extrêmes, elle mesure la dissimilarité par la somme des écarts absolus entre composantes.

— **Distance de Minkowski** :

$$d_{mink}(x, y) = \left(\sum_{i=1}^{p} |x_i - y_i|^q \right)^{1/q}, \quad q \geq 1$$

Elle généralise les distances précédentes ($q = 1$ donne la Manhattan, $q = 2$ la Euclidienne) et permet de moduler la sensibilité aux grandes différences.

— **Distance de Hamming** :

$$d_{ham}(x, y) = \sum_{i=1}^{p} \mathbf{1}_{x_i \neq y_i}$$

Adaptée aux données catégorielles ou binaires, elle compte le nombre de positions où les deux vecteurs diffèrent.

Une revue plus complète de ces distances et de leurs effets sur la qualité topologique et quantitative des SOM sera présentée dans le chapitre suivant, consacré à l'état de l'art.

1.4.2 Une nouvelle distance pondérée

Dans cette thèse, nous introduisons une **nouvelle variante pondérée de la distance de Minkowski**, conçue pour mieux capturer la structure intrinsèque des données complexes. L'idée consiste à attribuer un poids λ_i à chaque dimension afin de moduler son importance relative dans le calcul de la distance. Cette formulation permet de mettre davantage l'accent sur certaines variables jugées plus pertinentes selon le contexte applicatif, tout en conservant la souplesse de la métrique de Minkowski classique.

Cette approche, à notre connaissance, n'a pas encore été explorée dans le cadre des SOM. Elle constitue donc une **contribution originale de ce travail**, visant à améliorer la capacité du modèle à représenter fidèlement la structure multidimensionnelle des

données d'entrée et à optimiser la segmentation obtenue.

Avant d'utiliser cette mesure dans le cadre des SOM, il est nécessaire de vérifier qu'elle définit bien une distance au sens mathématique du terme. Nous présentons donc ci-dessous une démonstration rigoureuse établissant que la fonction d satisfait les propriétés requises d'une métrique sur l'espace $E = [0,1]^n$.

Soit l'espace $E = [0,1]^n$, on définit la distance suivante :

Soient $a \in E$, $b \in E$, $\lambda \in E$, $n \in \mathbb{N}^*$, $p \geq 1$

Soit $d : E \times E \to \mathbb{R}^+$

$$d(a,b) = \left(\frac{\sum_{i=1}^{n} |a_i - b_i|^p \lambda_i}{\sum_{i=1}^{n} \lambda_i} \right)^{\frac{1}{p}}$$

Preuve

Montrons que (E,d) est un espace métrique. C'est à dire :

$$\forall (a,b) \in E^2 d(a,b) = d(b,a)$$

$$\forall (a,b) \in E^2, d(a,b) = 0 \iff a = b$$

$$\forall (a,b,c) \in E^3, d(a,c) \leq d(a,b) + d(b,c)$$

Soient $a \in E$, $b \in E$, $\lambda \in E$, $n \in \mathbb{N}^*$, $p \geq 1$

$$d(a,b) = \left(\frac{\sum_{i=1}^{n} |a_i - b_i|^p \lambda_i}{\sum_{i=1}^{n} \lambda_i} \right)^{\frac{1}{p}}$$

$$\iff d(a,b) = \left(\frac{\sum_{i=1}^{n} |-(a_i - b_i)|^p \lambda_i}{\sum_{i=1}^{n} \lambda_i} \right)^{\frac{1}{p}}, \text{ car } |x| = |-x|$$

$$\iff d(a,b) = \left(\frac{\sum_{i=1}^{n} |b_i - a_i)|^p \lambda_i}{\sum_{i=1}^{n} \lambda_i} \right)^{\frac{1}{p}}$$

$$\iff d(a,b) = d(b,a)$$

Soient $a \in E$, $b \in E$, $\lambda \in E$, $n \in \mathbb{N}^*$, $p \geq 1$

$$\text{On pose } d(a,b) = 0$$

$$\left(\frac{\sum_{i=1}^{n}|a_i-b_i|^p\lambda_i}{\sum_{i=1}^{n}\lambda_i}\right)^{\frac{1}{p}}=0$$

$$\iff \frac{\sum_{i=1}^{n}|a_i-b_i|^p\lambda_i}{\sum_{i=1}^{n}\lambda_i}=0, \text{ car } p>0$$

$$\iff \sum_{i=1}^{n}|a_i-b_i|^p\lambda_i=0, \text{ car } \sum_{i=1}^{n}\lambda_i>0$$

$$\iff \forall i\in[\![1,N]\!],\ |a_i-b_i|^p\lambda_i=0, \text{ car } |a_i-b_i|^p\lambda_i>0$$

$$\iff \forall i\in[\![1,N]\!],\ |a_i-b_i|^p=0$$

$$\iff \forall i\in[\![1,N]\!],\ |a_i-b_i|=0, \text{ car } p>0$$

$$\iff \forall i\in[\![1,N]\!],\ a_i=b_i$$

$$\text{Ainsi, } d(a,b)=0\iff a=b$$

Soient $a\in E$, $b\in E$, $c\in E$, $\lambda\in\mathbb{R}^n$, $n\in\mathbb{N}^*$, $p\geq 1$

$$\text{On a, } |a_i-c_i|=|a_i-b_i+b_i-c_i|, \text{ pour } i\in[\![1,N]\!]$$

$$\implies |a_i-c_i|\leq|a_i-b_i|+|b_i-c_i|, \text{ car } |x+y|\leq|x|+|y|, \forall(x,y)\in\mathbb{R}^2$$

$$\implies |a_i-c_i|^p\leq(|a_i-b_i|+|b_i-c_i|)^p, \text{ car } f(x)=x^p \text{ est croissante sur } \mathbb{R}^+ \text{ pour } p>0$$

$$\implies |a_i-c_i|^p\leq(|a_i-b_i|+|b_i-c_i|)^p\leq|a_i-b_i|^p+|b_i-c_i|^p, \text{ car } p\geq 1$$

$$\implies |a_i-c_i|^p\leq|a_i-b_i|^p+|b_i-c_i|^p$$

$$\Longrightarrow |a_i - c_i|^p \lambda_i \leq |a_i - b_i|^p \lambda_i + |b_i - c_i|^p \lambda_i, \text{ car } \lambda_i \geq 0$$

$$\Longrightarrow \sum_{i=1}^{n} |a_i - c_i|^p \lambda_i \leq \sum_{i=1}^{n} |a_i - b_i|^p \lambda_i + |b_i - c_i|^p \lambda_i$$

$$\Longrightarrow \sum_{i=1}^{n} |a_i - c_i|^p \lambda_i \leq \sum_{i=1}^{n} |a_i - b_i|^p \lambda_i + \sum_{i=1}^{n} |b_i - c_i|^p \lambda_i$$

$$\Longrightarrow \frac{\sum_{i=1}^{n} |a_i - c_i|^p \lambda_i}{\sum_{i=1}^{n} \lambda_i} \leq \frac{\sum_{i=1}^{n} |a_i - b_i|^p \lambda_i}{\sum_{i=1}^{n} \lambda_i} + \frac{\sum_{i=1}^{n} |b_i - c_i|^p \lambda_i}{\sum_{i=1}^{n} \lambda_i}, \text{ car } \frac{1}{\sum_{i=1}^{n} \lambda_i} > 0$$

$$\Longrightarrow \left(\frac{\sum_{i=1}^{n} |a_i - c_i|^p \lambda_i}{\sum_{i=1}^{n} \lambda_i} \right)^{\frac{1}{p}} \leq \left(\frac{\sum_{i=1}^{n} |a_i - b_i|^p \lambda_i}{\sum_{i=1}^{n} \lambda_i} + \frac{\sum_{i=1}^{n} |b_i - c_i|^p \lambda_i}{\sum_{i=1}^{n} \lambda_i} \right)^{\frac{1}{p}},$$

$$\text{car } f(x) = x^{\frac{1}{p}}, \text{ avec } 0 < \frac{1}{p} \leq 1 \text{ est croissante}$$

$$\text{On a, } 0 \leq |a_i - b_i| \leq 1, \text{ pour } i \in [\![1, N]\!]$$

$$\Longleftrightarrow 0 \leq |a_i - b_i|^p \leq 1, \text{ car } p > 0]\!]$$

$$\Longleftrightarrow 0 \leq |a_i - b_i|^p \lambda_i \leq \lambda_i, \text{ car } \lambda_i \geq 0]\!]$$

$$\Longleftrightarrow 0 \leq \sum_{i=1}^{n} |a_i - b_i|^p \lambda_i \leq \sum_{i=1}^{n} \lambda_i$$

$$\Longleftrightarrow 0 \leq \frac{\sum_{i=1}^{n} |a_i - b_i|^p \lambda_i}{\sum_{i=1}^{n} \lambda_i} \leq 1, \text{ car } \frac{1}{\sum_{i=1}^{n} \lambda_i} > 0$$

De la même manière, on a :

$$0 \leq \frac{\sum_{i=1}^{n} |b_i - c_i|^p \lambda_i}{\sum_{i=1}^{n} \lambda_i} \leq 1$$

$$\text{Or, } (x+y)^{\frac{1}{p}} \leq x^{\frac{1}{p}} + y^{\frac{1}{p}}, \text{ pour } 0 \leq x \leq 1,\ 0 \leq y \leq 1 \text{ et } 0 < \frac{1}{p} \leq 1$$

$$\Longrightarrow \left(\frac{\sum_{i=1}^{n} |a_i - c_i|^p \lambda_i}{\sum_{i=1}^{n} \lambda_i}\right)^{\frac{1}{p}} \leq \left(\frac{\sum_{i=1}^{n} |a_i - b_i|^p \lambda_i}{\sum_{i=1}^{n} \lambda_i}\right)^{\frac{1}{p}} + \left(\frac{\sum_{i=1}^{n} |b_i - c_i|^p \lambda_i}{\sum_{i=1}^{n} \lambda_i}\right)^{\frac{1}{p}},$$

$$\Longleftrightarrow d(a,c) \leq d(a,b) + d(b,c)$$

Ainsi d est une distance et (E, d) est un espace métrique.

Dimensionnalité : n représente le nombre de dimensions des vecteurs a et b, indiquant que la distance est calculée en considérant l'ensemble des caractéristiques disponibles.

Exposant p : Ce paramètre détermine la "dureté" de la mesure de distance. Par exemple, $p = 2$ conduit à une distance euclidienne, tandis que $p = 1$ donne une distance de Manhattan. Cela permet d'ajuster la sensibilité de la distance vis-à-vis des grandes différences entre les caractéristiques.

Poids λ_i : Ces poids permettent de moduler l'importance relative de chaque dimension dans le calcul de la distance, offrant la possibilité d'accorder plus ou moins de poids à certaines caractéristiques en fonction de leur pertinence.

Normalisation : Le dénominateur $\sum_{i=1}^{n} \lambda_i$ assure que la distance est normalisée par rapport à la somme des poids, garantissant ainsi que l'échelle de la distance reste cohérente quelles que soient les valeurs de λ_i.

La puissance de cette mesure de distance réside dans sa flexibilité et sa capacité à s'adapter à divers contextes d'application en ajustant ses paramètres (p et λ_i). Elle permet de tenir compte à la fois de la nature des données (à travers le choix de p) et de l'importance relative des différentes dimensions (via les poids λ_i). Cette adaptabilité fait de la distance définie une outil précieux pour la construction et l'optimisation des cartes auto-organisatrices, en permettant une segmentation et une visualisation des données qui reflètent fidèlement la structure et la dynamique sous-jacentes de l'ensemble de données analysé.

Chapitre 2

Évolutions récentes des SOM

2.1 Les cartes auto-organisatrices : la vision de T. Kohonen

Les Cartes Auto-Organisatrices (SOM), représentent une avancée significative dans le domaine des réseaux de neurones artificiels. L'objectif de ce chapitre est de présenter les principales évolutions de l'algorithme originel des cartes auto-organisatrices, en mettant l'accent sur les avancées de la dernière décennie. Ces travaux récents illustrent la vitalité du domaine et la capacité des SOM à s'adapter à la complexité croissante des données et des usages.

2.1.1 L'article fondateur : The Self-Organizing Map

Dans son article fondateur de 1990 intitulé "The self-organizing map" [61], Teuvo Kohonen a posé les bases de ce qui allait devenir un pivot majeur dans le domaine des réseaux neuronaux artificiels.

L'aspect révolutionnaire des SOM réside dans leur capacité à créer des représentations internes organisées spatialement de diverses caractéristiques des signaux d'entrée et de leurs abstractions. Kohonen démontre que cette propriété permet à la SOM de découvrir des relations sémantiques dans les données, un concept qui était à l'avant-garde à l'époque. Cette capacité de découvrir des structures inhérentes dans des ensembles de données complexes a ouvert de nouvelles voies dans la compréhension et l'interprétation de données de haute dimension.

Dans cet article, Kohonen passe également en revue les travaux antérieurs sur l'apprentissage compétitif et les cartes cérébrales, jetant les bases théoriques des SOM. Il détaille ensuite l'algorithme des SOM, en se concentrant sur la sélection de la cellule la mieux adaptée et l'adaptation des vecteurs de poids, deux composants clés de la méthode.

Kohonen illustre également comment appliquer l'algorithme des SOM, en montrant

des démonstrations du processus d'ordonnancement et un exemple de regroupement hiérarchique des données. Un point important de l'article est le raffinement de la carte par l'apprentissage vectoriel quantique, une technique d'affinement qui améliore la précision de la SOM.

Enfin, l'article explore l'utilisation des SOM dans des applications pratiques telles que la reconnaissance vocale et présente une expérience de simulation sur la cartographie sémantique. Cette exploration précoce des applications pratiques témoigne de la polyvalence des SOM et de leur potentiel pour transformer divers domaines d'application.

Après son article pionnier de 1990, Teuvo Kohonen a continué à développer et à affiner sa théorie des Self-Organizing Maps (SOM), culminant avec la publication de son livre influent "Self-Organizing Maps" en 1995 [57]. Ce livre représente un jalon significatif dans la littérature sur les réseaux neuronaux et l'intelligence artificielle, offrant une exploration complète et détaillée de la théorie, des algorithmes et des applications des SOM. Dans cet ouvrage, Kohonen étend sa recherche initiale sur les Self-Organizing Maps, en fournissant une exploration détaillée et complète de cette technologie innovante.

2.1.2 Impact dans la communauté scientifique

Depuis sa publication, "Self-Organizing Maps" est devenu un ouvrage de référence et la première réédition de "Self-Organizing Maps" datant de 1997 [59] a permis d'intégrer de nouveaux développements significatifs.

— Introduction de l'ASSOM : Une innovation majeure dans cette édition est l'introduction de l'Adaptive-Subspace SOM (ASSOM). Cette nouvelle architecture, où des détecteurs de caractéristiques invariants émergent via un processus d'apprentissage non supervisé, représente une avancée importante dans la compréhension et l'application des SOM.

— Révision et Extension des Sections : Des sections clés du livre ont été révisées et étendues, notamment la section 5.9 entièrement réécrite, qui aborde plus en détail l'ASSOM, ainsi que la section 1.4 sur les classificateurs d'espaces adaptatifs, offrant ainsi une base théorique plus solide pour ces concepts.

— Intégration du WEBSOM : Une autre addition notable est la présentation du WEBSOM, une architecture à deux couches conçue pour organiser de vastes collections de documents textuels. Ce développement reflète l'adaptabilité des SOM à l'ère numérique et à la gestion de données massives.

La réédition de 2001 [60] témoigne de la l'intérêt continu pour les SOM, avec une

augmentation substantielle du nombre de publications scientifiques sur le sujet.

- Approche Plus Détaillée de l'Analyse des Modèles : Kohonen a enrichi l'approche de l'analyse statistique des modèles dans cette édition, avec une discussion plus détaillée sur les vecteurs propres et valeurs propres des matrices symétriques.
- Inclusion de Nouveaux Concepts Probabilistes : L'ajout de concepts tels que l'analyse factorielle dans la section 1.3.1 et une enquête sur les méthodes de projection dans la section 1.3.2, relie les SOM à des paradigmes classiques, élargissant ainsi leur contexte théorique.
- Extension sur la Quantification Vectorielle : La discussion sur la quantification vectorielle a été consolidée en une section principale, incluant une dérivation de la densité ponctuelle des vecteurs du codebook, ce qui aide à une meilleure compréhension de cette analyse statistique complexe.
- Perspective Historique et Philosophique Élargie : Avec une revue historique dans la section 2.2 et une discussion élargie sur la philosophie de la modélisation neurale, Kohonen vise à fournir aux nouveaux venus dans le domaine une meilleure orientation parmi la multitude de modèles neuronaux contemporains.

Ces rééditions montrent l'engagement continu de Kohonen à développer et à affiner le concept des SOM. Elles reflètent non seulement les progrès dans la recherche, mais aussi l'évolution des applications pratiques des SOM.

Pour finir, dans son article de 2013, "Essentials of the Self-Organizing Map" [56], Teuvo Kohonen étend la possibilité de représenter un élément d'entrée de manière encore plus précise par un mélange linéaire de plusieurs modèles correspondants. Kohonen introduit une procédure de moindres carrés avec des contraintes de non-négativité pour les coefficients, améliorant ainsi la finesse de la représentation des données dans les SOM. Cet article de 2013 constitue un pont entre les travaux antérieurs de Kohonen et les applications modernes des SOM.

2.1.3 Avancées récentes dans le domaine des SOM

Dans ce chapitre consacrée à l'état de l'art, nous nous limiterons à examiner les publications et les recherches menées au cours des dix dernières années. Cette période, allant de 2014 à aujourd'hui, représente une phase restant dynamique et innovante dans le développement des cartes SOM. En nous concentrant sur cette fenêtre temporelle, nous pourrons saisir les tendances actuelles, les avancées technologiques, et les applications émergentes les plus pertinentes. Cette approche nous permettra d'offrir une perspective à jour et

approfondie sur les dernières évolutions dans ce domaine, tout en mettant en lumière les directions futures de la recherche sur les cartes SOM.

Ce travail a été mis à disposition de la communauté sous la forme d'un prépublication Arxiv : [31]

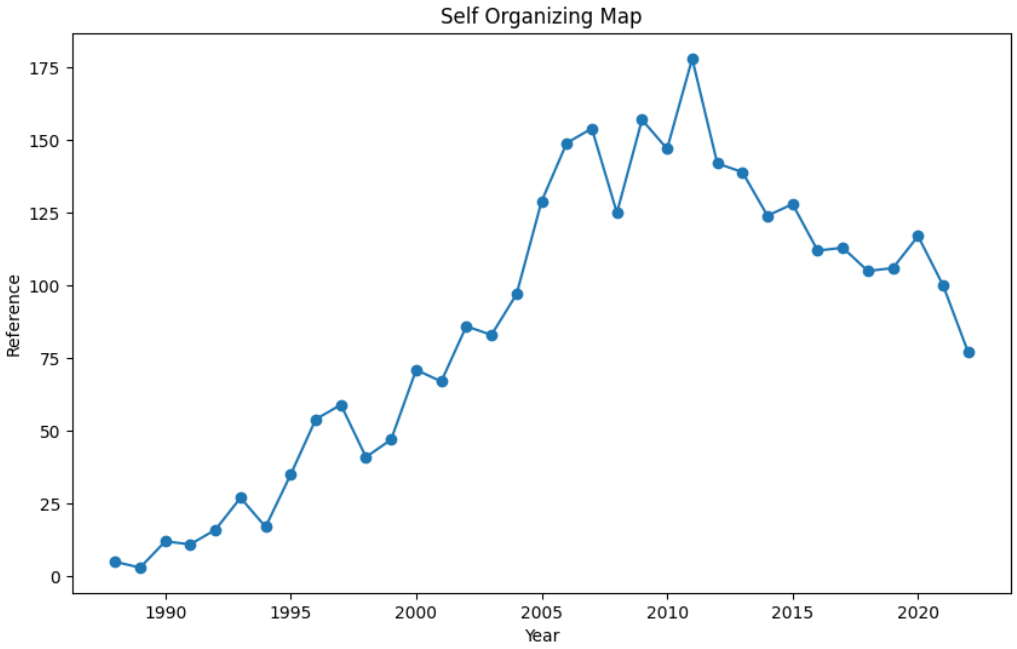


FIGURE 2.1 – Evolution du nombre de référence par année pour la recherche "Self Organizing Map" sur la base DBLP (`https://dblp.org/`)

Pour examiner certains des travaux les plus représentatifs de cette période, nous avons organisé notre exploration selon quatre axes distincts d'amélioration des algorithmes SOM, comme illustré dans la Figure 2.2. Cette figure détaille l'organisation des prochaines sous-sections de cette revue.

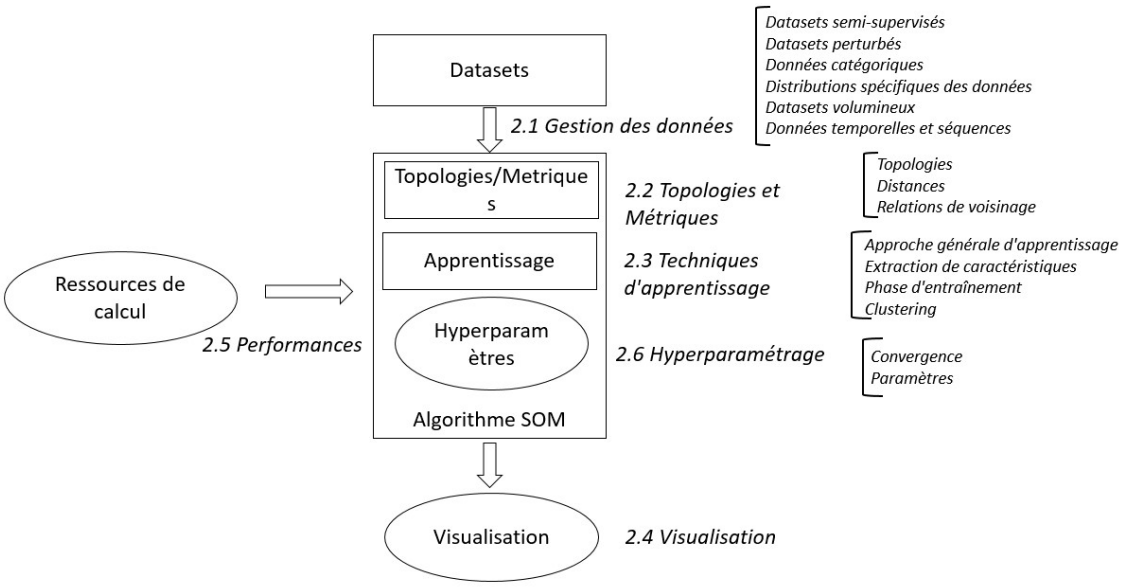


FIGURE 2.2 – Organisation de l'état de l'art selon les différents composants et caractéristiques clés des algorithmes SOM.

— **Gestion des données :** Les SOM peuvent être utilisées pour visualiser différents types de données (numériques, catégoriques, etc.). De plus, le jeu de données d'entrée peut présenter des difficultés dans divers cas réels (valeurs manquantes, hétérogénéité, etc.).
— **Topologies et Métriques :** Les algorithmes SOM nécessitent la sélection d'une topologie pour représenter la carte, ainsi que des distances et des relations de voisinage pour la construction de cette carte.
— **Techniques d'apprentissage :** Étant donné que les algorithmes SOM appartiennent aux techniques d'apprentissage automatique, ils impliquent des choix concernant le réglage du processus d'apprentissage, ainsi que pour les différentes phases d'apprentissage, incluant l'entraînement et la préparation des données.
— **Visualisation :** La visualisation des données est évidemment l'objectif principal de l'algorithme SOM. Par conséquent, différentes techniques et options peuvent être sélectionnées pour améliorer les cartes obtenues.
— **Performances :** Comme de nombreuses approches d'apprentissage automatique, l'algorithme SOM peut être très coûteux en termes de calcul. Des propositions ont été faites pour améliorer le processus de classification, y compris des solutions

matérielles dédiées.

— **Hyperparamétrage :** L'hyperparamétrisation désigne le processus de sélection et d'ajustement des hyperparamètres, qui sont des paramètres de configuration influençant le comportement et la performance de l'algorithme SOM pour améliorer ses résultats.

2.2 Gestion des données

La gestion des données représente un défi central dans l'application des cartes auto-organisatrices, compte tenu de la diversité des types de données et des contraintes associées. Les SOM, conçues initialement pour traiter des ensembles de données numériques, ont vu leur champ d'application s'élargir pour répondre à des besoins spécifiques dans des contextes variés. Ce chapitre explore les avancées récentes dans l'adaptation des SOM pour gérer des caractéristiques distinctes des données.

Nous commencerons par examiner les datasets semi-supervisés, où les échantillons étiquetés sont rares, nécessitant des méthodes hybrides combinant apprentissage supervisé et non supervisé. Ensuite, nous aborderons les datasets perturbés, qui incluent des données bruitées ou incomplètes, posant des défis importants pour l'entraînement et la précision des modèles.

Les données catégorielles, historiquement difficiles à intégrer dans les SOM, feront également l'objet d'une attention particulière avec des solutions innovantes pour combiner données numériques et catégorielles de manière cohérente. Nous explorerons ensuite les approches pour traiter des distributions spécifiques de données, où les variables sont représentées par des distributions plutôt que des valeurs fixes, nécessitant des métriques adaptées pour capturer la complexité des relations.

L'analyse des datasets volumineux mettra en lumière les défis liés à la scalabilité des SOM face à des volumes massifs de données, tandis que la gestion des données temporelles et séquences démontrera comment les SOM peuvent être adaptées pour capturer les dynamiques et les relations évolutives dans les données. Cette exploration globale illustrera les innovations et les ajustements nécessaires pour maximiser l'efficacité des SOM dans un large éventail de contextes et de types de données.

2.2.1 Datasets semi-supervisés

Les ensembles de données semi-supervisés, caractérisés par une quantité limitée d'étiquettes, nécessitent des approches adaptées combinant apprentissage supervisé et non supervisé. Les SOM ont été étendues pour exploiter efficacement ces données partiellement étiquetées, tout en tirant parti des informations disponibles dans les échantillons non étiquetés. Cette section explore des algorithmes novateurs qui intègrent des contraintes et des stratégies hybrides pour améliorer le clustering et la classification dans des contextes de données semi-supervisées, tout en maintenant des performances robustes sur des ensembles massifs ou en streaming.

"A Semi-Supervised Self-Organizing Map for Clustering and Classification" [9] présente le SS-SOM, une méthode innovante pour gérer des ensembles de données semi-supervisés. Face à la prévalence des données avec peu d'échantillons étiquetés, le SS-SOM offre une solution flexible, alternant entre apprentissage supervisé et non supervisé en fonction de la disponibilité des étiquettes. Cette approche se distingue par sa capacité à exceller dans les conditions où les données étiquetées sont limitées, tout en restant efficace pour des ensembles entièrement étiquetés. Le SS-SOM se révèle ainsi particulièrement utile pour les scénarios de clustering et de classification dans des contextes de données semi-supervisées.

Dans "Constrained Semi-Supervised Growing Self-Organizing Map" [3], l'algorithme CS2GS est introduit comme une solution pour le clustering semi-supervisé en ligne. Ce modèle se concentre sur l'utilisation de contraintes dans les échantillons non étiquetés, une stratégie utile lorsque les données étiquetées sont rares ou difficiles à obtenir. CS2GS modifie l'apprentissage en ligne des Self-Organizing Maps semi-supervisés et l'adapte en un problème d'apprentissage métrique contraint résolu par les projections itératives de Bregman.

Cette approche répond aux défis posés par les ensembles de données massifs ou en streaming, où les méthodes hors ligne et par lots sont limitées. Les tests effectués avec des données synthétiques et réelles, y compris des jeux de données UCI (Université de Californie, Irvine) et un corpus bilingue pour la traduction automatique, démontrent l'efficacité de CS2GS dans le clustering semi-supervisé en ligne. Les résultats soulignent

l'avantage d'intégrer des échantillons non étiquetés pour améliorer la précision du système dans des scénarios de données réalistes.

2.2.2 Datasets perturbés

Les ensembles de données perturbés, comprenant des valeurs manquantes ou des données bruitées, posent des défis majeurs dans l'entraînement et la précision des modèles d'apprentissage automatique. Les SOM ont été étendues et adaptées pour mieux gérer ces perturbations, en utilisant des approches comme le traitement des données manquantes et la pondération des caractéristiques pour minimiser l'impact des anomalies. Cette section met en lumière des méthodes et outils innovants qui permettent aux SOM de maintenir leur efficacité, même dans des contextes où les données sont incomplètes ou altérées, tout en améliorant la qualité de la classification et de la visualisation des données.

"Semi-automated data classification with feature weighted self organizing map" [94] introduit le Feature Weighted Self-Organizing Map (FWSOM). Le FWSOM utilise les informations de topologie d'un SOM standard pour guider automatiquement la sélection des entrées importantes durant l'entraînement, améliorant ainsi la classification des données comportant des entrées non pertinentes.

Cette méthode a démontré une précision de classification accrue par rapport au SOM standard et à d'autres classificateurs existants, sur des jeux de données synthétiques et réels. Un atout notable du FWSOM est sa capacité à identifier les caractéristiques pertinentes, améliorant ainsi la performance de classification d'autres méthodes.

"IntraSOM : A comprehensive Python library for Self-Organizing Maps with hexagonal toroidal maps training and missing data handling" [30] présente IntraSOM, une nouvelle bibliothèque Python dédiée à la mise en œuvre des Self-Organizing Maps (SOM). Cette bibliothèque se distingue par sa capacité à supporter des grilles hexagonales et une topologie toroïdale, offrant ainsi une plus grande flexibilité dans la modélisation et l'analyse des données complexes. Un atout majeur d'IntraSOM réside dans sa gestion efficace des données manquantes durant le processus d'entraînement, une problématique fréquente dans les ensembles de données réels.

En outre, IntraSOM propose des outils de visualisation avancés et des algorithmes de regroupement (clustering) efficaces, ce qui en fait un outil précieux pour les chercheurs et les praticiens souhaitant explorer et analyser des ensembles de données complexes de manière intuitive. La bibliothèque se veut accessible et complète, dotée d'un cadre extensible qui facilite son intégration avec d'autres algorithmes et bibliothèques Python.

2.2.3 Données catégorielles

L'intégration des données catégorielles dans les SOM représente un défi en raison des différences fondamentales entre les données numériques et catégorielles. Les SOM, initialement conçues pour traiter des données numériques, doivent être adaptées pour gérer ces données discrètes tout en préservant leur structure sémantique. Cette section explore plusieurs approches novatrices pour surmonter ces limitations, en introduisant des techniques permettant de mélanger efficacement les données numériques et catégorielles, d'améliorer leur visualisation, et de préserver la structure des relations au sein des données catégoriques. Ces avancées élargissent considérablement les capacités des SOM pour des applications dans des domaines variés, allant de l'analyse de données biomédicales à l'étude des comportements sociaux.

"Unsupervised distance learning for extended self-organizing map and visualization of mixed-type data" [43] aborde une problématique importante dans le traitement des données catégorielles avec les Self-Organizing Maps (SOM). Initialement, les SOM étaient conçues pour traiter des données numériques, mais cette étude étend leur applicabilité aux données catégorielles en intégrant des hiérarchies de distances qui reflètent la structure sémantique de ces données.

L'intégration de hiérarchies de distances apprises avec le SOM étendu permet une meilleure gestion et visualisation des ensembles de données mixtes, comprenant à la fois des éléments numériques et catégoriels. Les expériences menées dans l'étude pour vérifier la faisabilité et comparer les performances des différentes méthodes d'apprentissage non supervisé démontrent l'efficacité de l'approche.

"Mixing numerical and categorical data in a Self-Organizing Map by means of frequency neurons" [18] apporte une contribution notable à l'adaptation des Self-Organizing Maps (SOM) pour le traitement des données catégorielles, un défi courant dans le domaine de la reconnaissance de formes et de l'exploration de données.

Traditionnellement, les SOM sont conçues pour traiter des données numériques, et l'intégration de données catégorielles implique souvent leur conversion en codes binaires. Cependant, cette transformation peut conduire à une distorsion lors de l'entraînement du réseau et de l'analyse ultérieure des données. Pour surmonter cette limitation, l'article propose une architecture SOM innovante qui peut traiter directement les valeurs catégorielles sans nécessiter de conversion préalable en codes binaires.

Un aspect clé de cette nouvelle architecture est sa capacité à mélanger efficacement les données numériques et catégorielles, en attribuant un poids équivalent à toutes les caractéristiques. Cette approche assure une représentation plus fidèle et équilibrée des différentes types de données au sein du réseau. De plus, l'architecture proposée est décrite comme étant évolutive, ce qui suggère sa capacité à gérer des ensembles de données de grande taille et diverses.

L'analyse et la visualisation des données multidimensionnelles représentent un défi majeur dans le domaine de l'apprentissage automatique, notamment lorsque ces données incluent des variables catégorielles. "hSOM : Visualizing Self-Organizing Maps to Accommodate Categorical Data" [53] aborde cette problématique.

Un aspect souvent sous-estimé dans l'utilisation des SOM est la visualisation effective des données elles-mêmes, en particulier lorsque ces données sont catégorielles. Pour répondre à ce défi, les auteurs proposent une approche innovante en intégrant un histogramme directement dans la visualisation SOM classique. Cette méthodologie vise à améliorer la représentation des constituants d'une SOM, permettant une meilleure compréhension des données catégorielles regroupées par une variable discrète.

L'approche proposée, hSOM, évalue l'utilité de cette intégration d'histogramme en examinant la façon dont elle peut enrichir la compréhension de la distribution des données catégorielles au sein de la SOM. Cette innovation offre une nouvelle dimension à la visualisation SOM, augmentant non seulement la clarté de la représentation des données,

mais aussi permettant une analyse plus intuitive et informative.

En facilitant la compréhension des données catégorielles, hSOM rend les SOM plus accessibles et utiles dans des contextes où la catégorisation des données joue un rôle crucial, comme en recherche biomédicale, en marketing, ou dans l'étude des comportements sociaux.

2.2.4 Distributions spécifiques des données

Les données décrites par des distributions de probabilité ou contenant des valeurs aberrantes posent des défis uniques dans l'analyse et le clustering. Les SOM ont été adaptées pour gérer ces complexités en incorporant des mesures spécialisées et des mécanismes robustes d'apprentissage. Cette section explore deux approches distinctes qui repoussent les limites des SOM traditionnelles : l'intégration de métriques adaptées aux données distributionnelles et le lissage des cartes pour une meilleure gestion des données atypiques. Ces innovations illustrent comment les SOM peuvent être adaptées pour traiter des distributions spécifiques et améliorer la qualité des résultats dans des contextes complexes.

"Batch Self-Organizing Maps for Distributional Data with an Automatic Weighting of Variables and Components" [1] introduit un algorithme de Self-Organizing Map par lots (DBSOM) adapté aux variables décrites par des distributions de probabilité ou de fréquence. Le DBSOM utilise la distance de Wasserstein $L2$ [10], couramment appliquée dans l'analyse de données distributionnelles, comme fonction de perte. Une innovation clé réside dans l'introduction de poids de pertinence appris automatiquement pour chaque variable à valeur distributionnelle, ainsi que pour les composantes de la distance de Wasserstein, qui se décompose en moyennes et en taille/forme des distributions.

Cette approche permet d'accentuer l'importance des différentes caractéristiques des distributions dans la valeur de la distance. Les algorithmes proposés ont été validés sur des ensembles de données réelles à valeurs distributionnelles, démontrant l'efficacité de la méthode DBSOM pour gérer et interpréter des données complexes dans un contexte multi-dimensionnel.

"Smoothed self-organizing map for robust clustering" [20] présente une version améliorée de la SOM, appelée Smoothed SOM (S-SOM), conçue pour offrir une robustesse accrue face à la présence de valeurs aberrantes ou de données atypiques. Ce développement vise à surmonter une limitation courante des SOM traditionnelles, qui peuvent être sensibles à ces valeurs aberrantes, affectant ainsi la qualité de la cartographie de densité des entrées, de la quantification vectorielle et du clustering.

L'innovation clé de la S-SOM réside dans la modification de la règle d'apprentissage. Cette modification permet de lisser la représentation des vecteurs d'entrée aberrants sur la carte, en utilisant une distance exponentielle complémentaire entre le vecteur d'entrée et son codebook le plus proche. Cette approche permet à la S-SOM de mieux gérer les données atypiques et de produire une carte plus représentative de la structure réelle des données.

2.2.5 Datasets volumineux

L'analyse de grands ensembles de données constitue un défi majeur en raison de la complexité computationnelle et de la nécessité de préserver la qualité de la représentation. Les SOM sont particulièrement bien adaptées pour la visualisation et le clustering, mais leur applicabilité à des datasets volumineux nécessite des adaptations. Cette section explore des approches novatrices, notamment des méthodes de visualisation optimisées et des algorithmes de clustering adaptés à des données de haute dimensionnalité, démontrant comment les SOM peuvent évoluer pour relever les défis posés par les grands ensembles de données.

"A New Method for Emulating Self-Organizing Maps for Visualization of Datasets" [46] introduit une méthode innovante pour la visualisation rapide de grandes quantités de données. La complexité computationnelle des SOM le rend moins adapté pour les grands ensembles de données. Pour surmonter ce défi, les auteurs proposent une méthode de visualisation dirigée par la force qui imite les capacités des SOMs, tout en réduisant significativement la complexité à $o(n)$. Cette approche repose sur un ajustement fin par la force de la représentation 2D des données. Pour démontrer l'efficacité et le potentiel de cette méthode comme outil de visualisation rapide, elle est appliquée à la projection 2D

de l'ensemble de données des chiffres manuscrits MNIST. Ce travail suggère une avancée prometteuse pour les applications nécessitant une synthèse et une visualisation rapides et précises des données, telles que les tendances commerciales, les réponses aux catastrophes et les flambées de maladies.

"Local adaptive receptive field dimension selective self-organizing map for multi-view clustering" [4] aborde un problème crucial dans le traitement de données complexes : la gestion de données à haute dimensionnalité, souvent caractéristiques de domaines variés tels que la bioinformatique, l'imagerie médicale, la recherche marketing ou l'analyse de réseaux sociaux.

Les auteurs mettent en évidence que les algorithmes de clustering traditionnels voient souvent leurs performances diminuer face à l'augmentation des dimensions des données. Pour surmonter ce défi, ils proposent une approche novatrice de clustering multi-vue basée sur une SOM adaptative et évolutive dans le temps. Cette méthode se distingue par sa capacité à traiter efficacement des données réelles caractérisées par une grande éparsité et une dimensionnalité élevée, ainsi que par des représentations diverses.

Leur solution innovante utilise une approche de clustering en sous-espaces, s'adaptant ainsi dynamiquement aux différentes "vues" ou perspectives des données. Les résultats expérimentaux obtenus démontrent que leur modèle offre des performances supérieures à celles d'autres modèles de pointe spécifiquement conçus pour le clustering multi-vue.

2.2.6 Données temporelles et séquences

Les données temporelles et séquentielles posent des défis uniques pour les SOM, en raison de leur nature dynamique et évolutive. Pour relever ces défis, des approches innovantes ont été développées pour adapter les SOM à l'analyse des séquences et à la capture des dynamiques temporelles. Cette section explore des contributions récentes, notamment les cartes auto-organisatrices temporelles pondérées et les mécanismes de codage basés sur les cycles limites, qui permettent une représentation plus précise des motifs évolutifs et des relations complexes dans les données temporelles.

"Automated and weighted self-organizing time maps" [89] propose des schémas innovants pour des cartes auto-organisatrices temporelles automatisées et pondérées (SOTM). Ces cartes offrent une méthode visuelle pour le clustering évolutif, qui consiste à produire une séquence de solutions de regroupement au fil du temps, un processus appelé clustering dynamique visuel.

L'automatisation des SOTM implique une paramétrisation pilotée par les données, ainsi qu'une adaptation de l'entraînement aux caractéristiques des données à chaque étape temporelle. L'objectif des SOTM pondérées est d'améliorer l'apprentissage en accordant plus d'importance aux données jugées fiables ou importantes. Cette approche offre un poids variable selon chaque instance de données.

Pour illustrer l'efficacité de ces schémas, les auteurs appliquent les SOTM automatisées et pondérées à deux ensembles de données du monde réel : des indicateurs de risque par pays pour mesurer l'évolution des déséquilibres mondiaux, et des données de demandeurs de crédit pour évaluer l'évolution des risques de crédit au niveau des entreprises.

"Self-organizing maps based on limit cycle attractors" [44] explore une nouvelle approche dans le développement des SOM en intégrant des cycles limites comme mécanisme de codage pour représenter des motifs ou séquences d'entrée. Cette méthode s'écarte des représentations de codage statique traditionnelles utilisées dans les SOMs, où chaque motif d'entrée est représenté par un schéma d'activation de point fixe sur la carte, un concept qui n'est pas en accord avec l'activité oscillatoire rythmique observée dans le cerveau.

Dans cette étude, les auteurs développent et examinent un schéma de codage alternatif qui utilise des cycles limites codés de manière clairsemée pour représenter des motifs ou séquences d'entrée externes. Ils établissent des conditions sous lesquelles les représentations de cycle limite apprennent de manière fiable et dominent la dynamique dans un SOM. Ces cycles limites tendent à être relativement uniques pour différents inputs, robustes aux perturbations, et assez insensibles au timing.

2.3 Topologies et Métriques

Les cartes auto-organisatrices se distinguent par leur capacité à représenter et organiser des données de haute dimension dans un espace de plus faible dimension tout en préservant leur structure topologique. Un élément central de cette performance repose sur la topologie utilisée pour structurer les neurones et sur les métriques qui définissent les relations entre eux. La topologie, qu'elle soit basée sur des grilles carrées, hexagonales ou sur des approches plus dynamiques et adaptatives, joue un rôle déterminant dans l'efficacité et l'adéquation des SOM à des tâches spécifiques telles que le clustering, la classification ou la visualisation de données.

De même, les métriques utilisées pour évaluer les distances entre les points de données et les neurones influencent directement la qualité des représentations générées. En intégrant des métriques alternatives, telles que les distances non euclidiennes ou spécifiques à des types de données complexes, les SOM peuvent s'adapter à des applications plus variées et à des données plus hétérogènes.

Cette section explore les avancées récentes dans les topologies et les métriques des SOM, en mettant en lumière des approches innovantes qui optimisent la structure et la performance des cartes. Nous analyserons également l'importance des relations de voisinage dans le processus d'apprentissage, en présentant des méthodes adaptatives qui améliorent la qualité des clusters et la préservation de la topologie. Ces développements renforcent l'efficacité des SOM et élargissent leurs possibilités d'application à des problématiques toujours plus complexes.

2.3.1 Topologies

La topologie des cartes SOM est un facteur clé influençant leur capacité à représenter efficacement les données. Bien que les topologies classiques, telles que les grilles carrées ou hexagonales, soient largement utilisées, les recherches récentes ont exploré des alternatives et des ajustements dynamiques pour mieux s'adapter aux besoins spécifiques des données et des tâches. Cette section examine des approches novatrices, notamment des topologies adaptatives, aléatoires ou basées sur des structures dynamiques, visant à améliorer la précision, la flexibilité et la préservation topologique des SOM dans divers contextes.

"Grid topologies for the self-organizing map" [68] examine les diverses extensions des cartes auto-organisatrices, en se concentrant spécifiquement sur les topologies de grille. Traditionnellement, les SOM sont structurées en grilles carrées ou hexagonales, mais cette étude propose des alternatives dérivées de la théorie géométrique des tessellations. L'objectif est d'évaluer l'efficacité de ces nouvelles topologies dans différents domaines d'application comme le clustering non supervisé, la segmentation d'images en couleur et la classification.

Les résultats expérimentaux révèlent des différences statistiquement significatives entre les topologies, soulignant que la topologie optimale varie en fonction du problème spécifique traité. Ce constat est crucial car il suggère que la sélection de la topologie de la grille ne devrait pas être une décision standardisée, mais plutôt adaptée à chaque cas d'utilisation spécifique pour maximiser les performances.

En outre, l'article propose une interprétation théorique de ces résultats, offrant une compréhension plus profonde des mécanismes sous-jacents qui rendent certaines topologies plus efficaces que d'autres dans des contextes spécifiques. Cette recherche ouvre la voie à une exploration plus approfondie des topologies de grille dans les applications des SOM, encourageant une personnalisation basée sur les besoins spécifiques de chaque tâche.

"Self Organizing Maps Whose Topologies Can Be Learned With Adaptive Binary Search Trees Using Conditional Rotations" [5] introduit une amélioration des cartes auto-organisatrices (SOM) à travers l'usage de structures d'arbres binaires de recherche (BST) dynamiques. Cette approche, nommée TTOCONROT, permet une adaptation et une transformation dynamique de la topologie des SOM, optimisant ainsi la représentation des données. L'innovation clé réside dans le concept de "Neural Promotion", où les neurones s'ajustent en fonction de leur importance croissante, reflétant plus précisément la distribution stochastique des données. Les résultats expérimentaux montrent que cette méthode améliore significativement la précision et la dynamique des SOM, sans nécessiter une compréhension approfondie des propriétés topologiques des données par l'utilisateur.

"AMSOM : Adaptive Moving Self-organizing Map for Clustering and Visualization" [93] présente AMSOM, une variante innovante du modèle SOM. Alors que les SOM traditionnels ont une structure fixe durant l'entraînement, AMSOM introduit une flexibilité significative en permettant des ajustements dynamiques de la position des neurones, ainsi que l'ajout et la suppression de neurones pendant l'entraînement.

Cette approche répond à deux limitations majeures des SOM classiques : la rigidité de leur grille de neurones et l'incapacité à supprimer des neurones une fois introduits. En offrant une structure plus adaptable, AMSOM améliore non seulement la performance d'entraînement, mais aussi la qualité de la visualisation des données. Les expérimentations réalisées sur divers ensembles de données ont montré que l'AMSOM conduit à une meilleure représentation des données d'entrée et fournit un cadre utile pour déterminer le nombre optimal de neurones et leur structure, améliorant ainsi l'efficacité du clustering et de la visualisation.

Dans "Randomized Self Organizing Map" [85], introduit le placement aléatoire des neurones sur une surface bi-dimensionnelle. Cette approche, qui suit une distribution de bruit bleu, permet de créer diverses topologies avec des discontinuités aléatoires mais contrôlables. Cette flexibilité est particulièrement avantageuse pour l'organisation de données de haute dimension. L'algorithme a été testé sur des tâches en une, deux et trois dimensions, ainsi que sur l'ensemble de données de chiffres manuscrits MNIST. Les résultats, validés par analyse spectrale et analyse topologique des données, montrent que cette carte auto-organisatrice randomisée peut se réorganiser de manière efficace en cas de lésion neuronale ou de neurogenèse, démontrant ainsi une capacité d'adaptation remarquable.

"A directed batch growing approach to enhance the topology preservation of self-organizing map" [99] présente une avancée significative dans le domaine des cartes auto-organisatrices en croissance (GSOM). L'approche innovante, baptisée DBGSOM (Directed Batch Growing Self-Organizing Map), vise à optimiser la qualité de préservation de la topologie lors de la croissance des cartes SOM.

Contrairement aux stratégies d'apprentissage incrémentiel habituelles des GSOM, le

DBGSOM introduit une stratégie d'apprentissage par lots qui oriente le processus de croissance en se basant sur l'erreur accumulée autour du neurone candidat à la frontière. Cette méthode permet d'ajouter un seul nouveau neurone autour de chaque neurone candidat, réduisant ainsi le risque de déformation et de torsion de la carte, problèmes courants dans les GSOM traditionnels dus à une croissance réseau imprévue et à une mauvaise initialisation des poids des neurones.

2.3.2 Distances

Les métriques de distance jouent un rôle fondamental dans le fonctionnement des SOM, influençant directement leur capacité à organiser et à représenter les données d'entrée. Cette section explore des avancées récentes dans l'utilisation et l'adaptation des métriques pour les SOM. Des alternatives aux distances euclidiennes classiques, telles que les métriques non-euclidiennes, les distances tensoriales et les métriques multi-échelles, sont examinées pour leur potentiel à améliorer la précision, la flexibilité et la robustesse des SOM face à des données complexes et hétérogènes.

Dans l'article "Metrics and the Cooperative Process of the Self-organizing Map Algorithm" [101], l'impact de l'utilisation de différentes mesures de distance sur le processus coopératif de l'algorithme SOM est exploré. Alors que la distance euclidienne est généralement employée dans les implémentations standards de l'algorithme, l'étude révèle que d'autres métriques, telles que la distance de Manhattan et celles de la même famille que la métrique euclidienne, peuvent également converger et produire des résultats comparables. Cependant, l'article souligne que le simple fait d'être une métrique ne suffit pas à garantir des résultats satisfaisants, et présente des exemples de mesures qui ne produisent pas de cartes adéquates malgré leur analogie avec la métrique euclidienne. Cette recherche éclaire sur la nécessité de choisir judicieusement les mesures de distance dans les SOM, en tenant compte de leur compatibilité avec la structure et les objectifs des données traitées.

"Self-organizing maps with information theoretic learning" [13] explore l'amélioration SOM en utilisant une mesure de similarité nommée métrique induite par la correntropie (CIM). Cette approche vise à améliorer le facteur de grossissement du mappage, souvent déformé dans les SOM standards en raison de l'usage de l'erreur quadratique moyenne qui tend à suréchantillonner les régions de faible probabilité.

L'étude démontre que l'adaptation du SOM selon le CIM est équivalente à la réduction du potentiel d'information croisée localisé, une fonction théorique de l'information qui quantifie la similarité entre deux distributions de probabilité. En exploitant cette propriété, les auteurs proposent un algorithme d'adaptation de la largeur de bande du noyau pour les noyaux gaussiens. Le modèle proposé atteint un mappage avec un grossissement optimal et adapte automatiquement les paramètres de la fonction du noyau, offrant ainsi une amélioration significative par rapport aux approches SOM traditionnelles, sans complexifier excessivement l'algorithme.

"Non-Euclidean Self-Organizing Maps" [12] propose un cadre généralisé pour les SOM non-euclidiennes. Traditionnellement, les SOM sont basées sur une géométrie euclidienne, ce qui limite leur capacité à modéliser les relations de similarité dans les données complexes. En adoptant une géométrie non-euclidienne, les auteurs ouvrent de nouvelles perspectives pour la réduction de dimension, le clustering et la découverte de similarités dans de grands ensembles de données. Cette approche innovante offre une nouvelle liberté dans la traduction des similarités en relations de voisinage spatiales, améliorant ainsi significativement la capacité des SOM à modéliser des données complexes et hiérarchiques.

"On Tensor Distances for Self Organizing Maps : Clustering Cognitive Tasks" [23] explore le traitement de données de haute complexité telles que des images, graphiques, matrices et séries temporelles. Cette recherche se distingue par son approche multilinéaire des distances dans l'espace vectoriel d'entrée, offrant une flexibilité accrue dans la formation des clusters. Ce choix méthodologique permet de dépasser les limites traditionnelles des formes géométriques fixes dans le regroupement, ouvrant ainsi la possibilité de créer des clusters aux formes arbitraires.

Le cœur de l'étude réside dans l'application de cette métrique de distance à un ensemble de données multimodales de neuroimagerie fonctionnelle (IRMf), prises pendant trois tâches cognitives distinctes. L'objectif étant d'évaluer la capacité des SOM à regrouper ces données de manière significative. Les résultats obtenus à partir de diverses configurations de SOM sont évalués à l'aide de matrices de confusion, taux d'erreur topologique, taux de changement d'ensemble d'activation, et variations de distance intra-cluster.

"Multi-scale metrics and self-organizing maps : a computational approach to the structure of sensory maps" [102] introduit l'idée d'une métrique à échelle bi-modale. Cette innovation permet une segmentation plus précise des cartes SOM en régions distinctes, reflétant la structure de clusters attendue dans les données. Cela est particulièrement pertinent dans le contexte des cartes somatotopiques, où les clusters de données peuvent correspondre à des régions spécifiques de la surface corporelle.

L'utilisation de cette métrique bi-échelle aide à résoudre un problème courant des SOM : celui des neurones de la carte qui ne sont pas activés par aucune donnée d'entraînement. Grâce à cette approche, la structure anticipée des données est mieux préservée, et les cartes SOM deviennent plus représentatives de la distribution réelle des données.

De plus, l'étude se penche sur la plasticité des cartes SOM utilisant cette métrique lorsqu'elles sont réentrainées après la perte de groupes de neurones ou suite à des changements dans les données d'entraînement. Ces simulations sont d'une importance particulière pour comprendre comment les cartes SOM peuvent s'adapter à des modifications dans les données, par exemple, dans un contexte neurobiologique où la perte d'une région de la surface corporelle nécessite un réajustement de la cartographie somatotopique.

2.3.3 Relations de voisinage

La façon dont les SOM gèrent les relations de voisinage entre les neurones est fondamentale pour leur capacité à organiser et à représenter fidèlement les données d'entrée. Cette section se penche sur diverses améliorations et innovations dans la fonction de voisinage, un aspect clé du processus d'apprentissage des SOM. Les recherches présentées ici proposent des méthodes novatrices pour ajuster la portée et l'impact des interactions entre neurones, ce qui a des implications significatives pour la qualité de la cartographie, l'apprentissage continu, et la préservation de la structure des données.

Dans "Improved Learning Performance of Hardware Self-Organizing Map Using a Novel Neighborhood Function" [41], une nouvelle fonction de voisinage spécialement conçue pour les SOM matérielles est proposée pour améliorer leur performance de quantification vectorielle. Les SOM matérielles désignent des implémentations de cartes auto-organisatrices adaptées pour une exécution directe sur du matériel dédié, tel que les FPGA

(Field-Programmable Gate Arrays). Ces systèmes sont conçus pour maximiser l'efficacité computationnelle en exploitant le parallélisme matériel, ce qui les rend particulièrement adaptées aux applications nécessitant une rapidité de traitement élevée. Cette fonction, adaptée aux contraintes du matériel et utilisant des puissances négatives de deux, a été testée par simulation et mise en œuvre sur un FPGA (Field-Programmable Gate Array). Les résultats montrent que cette fonction de voisinage améliore la performance de quantification des SOM matérielles sans augmenter les coûts matériels ni ralentir la vitesse de fonctionnement.

La visualisation des données est un aspect crucial dans l'analyse et l'interprétation des informations complexes. Les Self-Organizing Maps (SOM), avec leur capacité unique à réduire la dimensionnalité tout en préservant les structures topologiques des données, offrent un outil visuellement riche et intuitif pour explorer et comprendre des ensembles de données vastes et multidimensionnels. Cette capacité de visualisation fait des SOM un choix privilégié dans de nombreux domaines, allant de la bio-informatique à la finance, où la compréhension rapide et intuitive des tendances, des groupes et des anomalies est essentielle.

"An Improved Adaptive Self-Organizing Map" [76] présente une version adaptative du SOM ajuste dynamiquement la largeur de voisinage des neurones en fonction de la fréquence d'occurrence des motifs d'entrée dans l'espace de données. Cette approche assure que chaque neurone sur la grille du SOM ait sa propre largeur de voisinage adaptée, permettant ainsi une meilleure visualisation des données, surtout lorsque des différences significatives existent dans les fréquences des motifs d'entrée. L'étude expérimentale menée sur trois ensembles de données réels confirme l'efficacité de cette approche adaptative du SOM, soulignant son potentiel dans l'amélioration de la représentation visuelle et l'analyse des données.

Dans "A New Self-Organizing Map with Continuous Learning Capability" [40], l'auteur introduit une fonction de voisinage innovante qui permet un apprentissage continu. Contrairement à la fonction de voisinage traditionnelle, dont l'intensité et le rayon diminuent avec le temps, cette nouvelle fonction se base uniquement sur la distance entre le vecteur de poids du neurone gagnant et le vecteur d'entrée. Cela signifie que la magnitude

et le rayon de la fonction de voisinage sont ajustés en fonction de cette distance, et non des itérations d'apprentissage. Cette approche permet au SOM de poursuivre son apprentissage de manière ininterrompue, même en présence de distributions d'entrée variables, ce qui est particulièrement utile pour l'apprentissage en ligne ou dans des environnements à entrées dynamiques. L'utilisation de la distance vectorielle dans cette méthode offre au SOM une capacité d'apprentissage volontaire, similaire à la curiosité observée dans le cerveau biologique, permettant une adaptation continue à de nouvelles informations.

"A data-scattering-preserving adaptive self-organizing map" [75] présente une version qui s'ajuste en fonction de la dispersion des données d'entrée pour mieux préserver la structure principale des données. Cette méthode commence par un regroupement préliminaire des données d'entrée pour capturer leur dispersion. Ensuite, les variances intra-cluster obtenues sont utilisées pour définir les largeurs de voisinage des unités de correspondance optimale (BMU) du SOM.

Cette approche a été évaluée empiriquement sur trois ensembles de données réels, variés en taille, dimensionnalité et type, représentant différents domaines expérimentaux. La performance de cette méthode a été comparée à sept autres techniques de visualisation de données. Les résultats ont montré que la méthode proposée surpasse les autres techniques en termes d'efficacité, d'utilité et de précision. Cette méthode améliorée du SOM se distingue donc par sa capacité à s'adapter de manière plus précise à la structure des données, offrant ainsi une visualisation améliorée et plus informative.

"Different Aspects of Clustering The Self-Organizing Maps" [24] explore les méthodes de clustering appliquées aux SOM. Il compare les techniques traditionnelles, comme le k-means et le clustering hiérarchique, avec des approches innovantes basées sur la coloration de graphes. Ces dernières utilisent à la fois les dissimilarités et les relations de voisinage pour partitionner la SOM, offrant une analyse plus complète des données. Les résultats expérimentaux démontrent l'efficacité de ces méthodes graphiques dans l'amélioration du clustering des SOM.

"Generalized Self-Organizing Maps for Automatic Determination of the Number of Clusters and Their Multiprototypes in Cluster Analysis" [29] apporte une contribution à la flexibilité et à l'efficacité des SOM dans l'analyse de cluster. La généralisation proposée des SOM avec des voisinages 1-D (chaînes de neurones) permet au réseau de se scinder en sous-chaînes et de réguler dynamiquement le nombre total de neurones. Cette caractéristique confère au réseau la capacité de générer automatiquement des collections de multiprototypes capables de représenter une large gamme de clusters dans des ensembles de données, le tout de manière entièrement non supervisée. L'article comprend également une analyse de sensibilité aux changements de paramètres de contrôle et une analyse comparative avec une approche alternative. Ces expérimentations démontrent l'efficacité et l'adaptabilité de l'approche proposée, offrant ainsi une méthode puissante pour l'analyse de clusters dans divers contextes de données.

"Incremental Self-Organizing Maps for Collaborative Clustering" [70] aborde le concept de clustering collaboratif, une technique visant à révéler les structures communes de données réparties sur différents sites. Dans le contexte actuel où la quantité de données disponibles ne cesse d'augmenter, les méthodes de clustering incrémentiel sont essentielles. L'algorithme présenté dans cet article permet d'effectuer un clustering collaboratif en utilisant des SOM de manière incrémentielle, sans nécessiter de modifications topologiques de la carte. Cette approche s'avère particulièrement pertinente pour gérer efficacement des volumes de données importants et en constante évolution.

Les expérimentations réalisées sur plusieurs ensembles de données valident la méthode proposée. L'article met également en lumière l'influence de la taille du lot sur le processus d'apprentissage, un aspect crucial pour optimiser la performance du clustering dans des environnements dynamiques et collaboratifs. Ce travail représente ainsi une contribution significative à la fois dans le domaine du clustering incrémentiel et dans celui des SOM appliqués au clustering collaboratif.

2.4 Techniques d'apprentissage

2.4.1 Approche générale d'apprentissage

L'efficacité et la précision des SOM dépendent fortement des techniques d'apprentissage utilisées. Cette section explore des approches innovantes visant à améliorer divers aspects de l'apprentissage dans les SOM, notamment la prise en compte des neurones peu sollicités, l'optimisation de la recherche d'unités correspondantes, la maximisation de l'information d'entrée, et l'intégration de l'apprentissage émotionnel. Ces travaux illustrent comment les avancées dans les méthodologies d'apprentissage peuvent élargir les capacités des SOM dans des domaines tels que la classification, la reconnaissance de motifs et la prise de décision.

"Community SOM (CSOM) : An Improved Self-Organizing Map Learning Technique" [16] aborde l'efficacité d'apprentissage des neurones les moins sollicités ou les plus éloignés du neurone gagnant. Dans un SOM standard, ces neurones reçoivent moins d'exposition aux données d'entrée, ce qui réduit leur capacité d'apprentissage. Pour remédier à cela, les auteurs proposent le CSOM, une technique qui améliore l'apprentissage de ces neurones dits « faibles » ou « éloignés ». Cette approche vise à renforcer l'apprentissage au sein de la communauté des neurones gagnants, ce qui pourrait conduire à une meilleure répartition de la connaissance et à une cartographie plus précise des données d'entrée. En augmentant l'efficacité de l'apprentissage des neurones moins actifs, le CSOM pourrait potentiellement améliorer la performance globale du SOM dans diverses applications, notamment la classification, la visualisation de données, et le clustering.

Dans "A Fast Algorithm to Find Best Matching Units in Self-Organizing Maps" [7], les auteurs s'attaquent au défi de l'efficacité computationnelle en particulier lorsque les SOM comportent un grand nombre de neurones. La recherche de l'unité de correspondance optimale (BMU) est un processus clé dans le fonctionnement des SOM, mais il peut devenir de plus en plus onéreux en termes de temps de calcul avec l'augmentation du nombre de neurones. L'algorithme proposé dans cet article vise à accélérer considérablement la recherche du BMU, tout en minimisant la perte de performance. Cette amélioration de la

vitesse pourrait rendre les SOM plus pratiques et efficaces pour des applications nécessitant une grande quantité de neurones, comme la reconnaissance de motifs complexes ou l'analyse de données volumineuses. L'approche proposée ouvre ainsi la voie à des utilisations plus ambitieuses des SOM, où la taille et la complexité des modèles ne sont plus des contraintes majeures.

"Input information maximization for improving self-organizing maps" [48] propose une méthode théorique d'information innovante pour optimiser les performances, baptisée "maximisation de l'information d'entrée", qui concentre sur les neurones d'entrée en considérant principalement les neurones gagnants. La méthode repose sur l'évaluation de l'incertitude des neurones d'entrée, définie par la différence entre les neurones d'entrée et les neurones gagnants correspondants, puis normalisée pour obtenir une mesure claire.

L'idée centrale est que l'augmentation de l'information d'entrée conduit à une activation réduite des neurones d'entrée, avec l'état maximal se caractérisant par l'activation d'un seul neurone et la désactivation de tous les autres. Cette propriété est particulièrement utile pour réduire les erreurs de quantification et topographiques dans les SOM, améliorant ainsi la qualité globale des représentations obtenues.

Les auteurs ont appliqué cette méthode à deux ensembles de données distincts : les données de vote du Sénat et les attitudes de vote. Les résultats expérimentaux ont confirmé que l'augmentation de l'information d'entrée entraîne une diminution des erreurs de quantification et topographiques, tout en permettant d'extraire une structure de classe plus claire.

"Two novel hybrid Self-Organizing Map based emotional learning algorithms" [21] introduit deux algorithmes, EmSOM et Em-SOR-SOM, pour intégrer les aspects émotionnels dans l'apprentissage automatique. Ces algorithmes visent à améliorer la modélisation des réactions humaines en tenant compte des émotions, un aspect essentiel mais souvent négligé dans la prise de décision.

EmSOM corrige les lacunes de l'Emotional Backpropagation (EmBP) en déterminant les valeurs d'entrée émotionnelles à partir des blocs SOM correspondants. Cette approche permet une représentation plus fidèle et nuancée des émotions dans le processus d'apprentissage.

Em-SOR-SOM combine l'algorithme SOR-SOM (Sparse Online SOM) avec l'apprentissage émotionnel pour améliorer la reconnaissance de motifs. Cette hybridation exploite les avantages du SOR-SOM, notamment sa capacité à gérer efficacement les données éparses.

Les performances d'EmSOM et Em-SOR-SOM ont été validées sur des bases de données de reconnaissance faciale et de crédit. Ces algorithmes ont démontré leur supériorité par rapport à l'EmBP et d'autres méthodes récentes, ouvrant ainsi la voie à des applications avancées dans la reconnaissance de motifs et les systèmes de décision automatisés.

2.4.2 Extraction de caractéristiques

L'extraction de caractéristiques est une étape clé pour améliorer les performances des SOM, en particulier dans des environnements où les données de grande dimension nécessitent une représentation plus concise et pertinente. Cette section examine des approches innovantes combinant des techniques d'apprentissage profond, des architectures multi-couches et des méthodes inspirées des neurosciences. Ces travaux mettent en lumière des solutions visant à enrichir les SOM avec des capacités d'extraction de caractéristiques avancées, ouvrant la voie à des applications dans des contextes variés, notamment le Big Data et les systèmes embarqués.

"A Convolutional Deep Self-Organizing Map Feature extraction for machine learning" [87] propose une approche novatrice pour l'extraction de caractéristiques en apprentissage automatique en développant un algorithme, Unsupervised Deep Self-Organizing Map (UDSOM). Cette méthode combine l'architecture des SOM multi-couches avec des principes de l'apprentissage profond, adressant ainsi les défis liés à la gestion de données de grande dimension et à l'extraction de caractéristiques pertinentes dans un environnement de Big Data.

La principale innovation de l'UDSOM réside dans son processus de traitement des données, qui commence par un découpage en sous-régions, suivi d'une série de couches auto-organisatrices et d'une fonction de rectification (RELU). Chaque couche SOM se concentre sur la modélisation d'une sous-région locale des données. Les neurones les plus actifs de chaque SOM sont ensuite regroupés dans une couche d'échantillonnage pour générer une nouvelle carte 2D. En parallèle, une abstraction de l'espace de données est

réalisée via un module de convolution-pooling, suivi de l'application de la fonction ReLU.

L'architecture de l'UDSOM permet de recueillir des informations locales et de les intégrer pour former une représentation plus globale dans les couches supérieures. Cette approche est particulièrement pertinente dans le contexte du Big Data, où la quantité massive de données nécessite des méthodes efficaces pour extraire des caractéristiques significatives. Les expériences menées démontrent la performance de cette architecture, soulignant son potentiel pour diverses tâches d'apprentissage automatique.

"Improving Self-Organizing Maps with Unsupervised Feature Extraction" [50] explore une approche qui s'éloigne de l'utilisation des données brutes et se concentre plutôt sur l'exploitation des caractéristiques extraites. Cette méthode est particulièrement pertinente dans le cadre des applications embarquées, où l'efficacité et la précision de l'apprentissage non supervisé sont primordiales.

Le cœur de cette étude réside dans la comparaison de deux méthodes d'extraction de caractéristiques : l'une basée sur l'apprentissage automatique avec des Auto-Encodeurs Convolutifs Parcimonieux et l'autre inspirée des neurosciences avec des Réseaux de Neurones à Impulsions utilisant l'apprentissage dépendant du timing des impulsions. Ce contraste entre une approche traditionnelle en machine learning et une approche plus expérimentale en neurosciences offre une perspective fascinante sur la manière dont les caractéristiques des données peuvent être efficacement extraites et utilisées pour améliorer la classification des SOM.

2.4.3 Phase d'entraînement

L'entraînement des SOM est une étape qui détermine leur capacité à capturer la structure des données et à générer des représentations significatives. Cette section explore des améliorations et innovations dans les techniques d'entraînement des SOM, visant à accroître leur efficacité, optimiser leur convergence et améliorer la qualité des cartes produites. Ces contributions mettent en lumière des approches stochastiques optimisées et de nouveaux indicateurs de performance, offrant des solutions pour relever les défis liés à l'apprentissage et à l'adaptation des SOM à divers contextes d'application.

"VSOM : Efficient, Stochastic Self-organizing Map Training" [33] présente VSOM, une implémentation efficace pour l'entraînement stochastique des SOM. VSOM améliore l'algorithme stochastique standard de Kohonen en remplaçant les constructions itératives par des opérations de vecteur et de matrice. Cette approche permet des gains de performance significatifs par rapport à l'algorithme itératif de Kohonen et au batchSOM, l'implémentation la plus rapide des SOM sans recours au multi-processing.

La qualité des cartes produites par VSOM est comparable à celle obtenue avec l'algorithme itératif original et surpasse celle du batchSOM. Adapté pour un fonctionnement en mono-thread, VSOM se prête particulièrement bien comme remplacement de l'entraînement stochastique itératif des SOM, spécialement dans des environnements comme R qui ne gèrent pas efficacement le multi-threading.

L'article "Self-Organizing Map Convergence" [96] introduit un nouvel indice de qualité, nommé indice de convergence. Cet indice combine précision de l'incorporation de la carte et précision topographique estimée, fournissant ainsi un nombre unique et statistiquement significatif qui s'avère plus intuitif à utiliser que d'autres mesures de qualité. Cette recherche se concentre sur la manière dont l'indice de convergence capte l'apprentissage par les SOM de la distribution multivariée d'un ensemble de données d'entraînement. Une attention particulière est portée sur la convergence des marginales et l'influence de différents paramètres régissant l'apprentissage des SOM. De manière surprenante, il est constaté que la fonction de voisinage constante surpasse la populaire fonction de voisinage gaussienne dans la production de modèles SOM plus performants. Ce résultat met en évidence l'importance de choisir une fonction de voisinage appropriée pour optimiser l'apprentissage et la convergence des cartes auto-organisatrices.

2.4.4 Clustering

Le clustering est une application clé des SOM, permettant de regrouper des données en clusters selon leurs similarités tout en préservant leur topologie. Cette section explore diverses innovations visant à améliorer l'efficacité, la généralisation et la précision des SOM dans des scénarios de clustering. Des approches incrémentielles et collaboratives aux méthodes basées sur des ensembles de cartes de tailles différentes, ces contributions

mettent en lumière des techniques avancées pour gérer la complexité croissante des données modernes.

"Incremental Self-Organizing Maps for Collaborative Clustering" [70] aborde le concept declustering collaboratif, une technique visant à révéler les structures communes de données réparties sur différents sites. Dans le contexte actuel où la quantité de données disponibles ne cesse d'augmenter, les méthodes de clustering incrémentiel sont essentielles. L'algorithme présenté dans cet article permet d'effectuer un clustering sans nécessiter de modifications topologiques de la carte. Cette approche s'avère particulièrement pertinente pour gérer efficacement des volumes de données importants et en constante évolution.

Les expérimentations réalisées sur plusieurs ensembles de données valident la méthode proposée. L'article met également en lumière l'influence de la taille du lot sur le processus d'apprentissage, un aspect crucial pour optimiser la performance du clustering dans des environnements dynamiques et collaboratifs. Ce travail représente ainsi une contribution significative à la fois dans le domaine du clustering incrémentiel et dans celui des SOM appliqués au clustering collaboratif.

"An ensemble algorithm for Kohonen self-organizing map with different sizes" [78] explore une approche innovante dans le domaine du clustering de données se concentre sur l'amélioration de la généralisation et de la performance des SOM en utilisant une méthode d'ensemble.

Le clustering de données est une méthode analytique visant à identifier des groupes au sein de données en se basant sur des similarités. Cependant, les résultats du clustering peuvent varier en fonction de nombreux facteurs comme les paramètres de l'algorithme, l'initialisation, les critères d'arrêt, ou encore l'utilisation de différents attributs ou sous-ensembles de données. Les SOM sont fréquemment utilisées pour l'analyse de données, notamment pour la visualisation et le clustering.

L'approche proposée dans cet article consiste à utiliser un ensemble de réseaux SOM travaillant de manière indépendante, combinés par un système qui intègre les résultats individuels en une sortie unique. Cette méthode vise à obtenir une meilleure généralisation par rapport à l'utilisation d'un seul réseau de neurones. Le concept clé de cette méthode

est l'utilisation d'indices de validité des clusters pour combiner les poids des neurones issus de cartes de différentes tailles.

"Self-Organizing Maps Fusion : An Approach to Different Size Maps" [79] propose d'améliorer la précision de classification en fusionnant des cartes de tailles différentes. Cette approche, basée sur l'ensemble de réseaux neuronaux, vise à obtenir une meilleure généralisation du modèle. À travers des expériences factorielles et des simulations informatiques utilisant des ensembles de données de l'UCI Machine Learning Repository et de la Fundamental Clustering Problems Suite, les auteurs démontrent une augmentation significative de la précision de classification. Le test de Wilcoxon Signed Rank valide la faisabilité de cette méthode, soulignant son potentiel dans l'amélioration des techniques de classification basées sur les SOM.

Dans "Improving self organizing maps method for data clustering and classification" [105], l'iSOM est présenté comme une amélioration pour résoudre les difficultés de visualisation et de classification. Ce modèle améliore la distinction entre clusters et sous-clusters en ajoutant une troisième dimension basée sur la distance computationnelle aux neurones gagnants. Testé sur le dataset Iris Flowers, l'iSOM a montré une meilleure séparation des classes et une augmentation de la précision de classification par rapport aux SOM traditionnels, offrant ainsi une méthode plus efficace pour l'analyse de données complexes.

2.5 Visualisation

Les SOM se distinguent par leur capacité à réduire la dimensionnalité tout en préservant les structures topologiques des données, offrant ainsi un outil puissant pour la visualisation de données complexes. Cette section explore les avancées récentes dans l'amélioration des capacités de visualisation des SOM, notamment à travers des extensions comme les cartes en 3D, les adaptations probabilistes, et les projections personnalisées influencées par l'utilisateur. Ces innovations visent à rendre les SOM plus flexibles, interprétables et adaptées à des contextes variés, allant de l'optimisation multidimensionnelle à l'analyse de clusters et aux systèmes de recommandation.

"A novel data-driven visualization of n-dimensional feasible region using interpretable self-organizing maps (iSOM)" [73] aborde la complexité de visualiser des problèmes d'optimisation dans des espaces multidimensionnels. Traditionnellement, l'optimisation graphique est limitée aux problèmes à une ou deux dimensions, où la visualisation des contours des fonctions objectifs et des contraintes facilite la compréhension et la prise de décision. Cependant, cette méthode atteint ses limites lorsque les dimensions augmentent.

Pour surmonter cette difficulté, les auteurs introduisent l'usage des iSOM, une variante des réseaux de neurones artificiels, pour la visualisation de données multidimensionnelles. Grâce à l'iSOM, il est possible de représenter graphiquement des régions réalisables en n-dimensions sur des représentations bi-dimensionnelles. Cette méthode est incarnée par la création d'une "B-matrix", une représentation graphique qui permet de visualiser à la fois la gamme réalisable des variables de conception et la fonction objectif.

La B-matrix réduit efficacement la complexité inhérente aux espaces de recherche multidimensionnels, permettant une analyse plus intuitive des données. La flexibilité et l'efficacité de cette méthode sont démontrées à travers divers exemples analytiques et d'ingénierie, couvrant des dimensions de 2 à 30. Cette approche représente un progrès significatif pour les concepteurs et les ingénieurs, offrant une nouvelle manière d'aborder et de résoudre des problèmes d'optimisation complexes dans de nombreux domaines.

L'article "Rating-Aware Self-Organizing Maps" [81] propose quant à lui une modification de l'algorithme pour intégrer des scores de pertinence dans le processus de visualisation de données en 2D. Traditionnellement, les SOM sont efficaces pour préserver les relations topologiques des données multidimensionnelles, mais elles ne tiennent pas compte de la pertinence des résultats, un aspect crucial dans les systèmes de récupération d'informations multimédia. La modification proposée vise à optimiser simultanément la conservation de l'ordre topologique et de la pertinence des résultats. Cette approche permet non seulement d'améliorer la capacité d'exploration des données, mais aussi d'exploiter de manière plus efficace les résultats les plus pertinents, offrant ainsi une visualisation plus riche et contextuellement adaptée.

L'article "Cluster and visualize data using 3D self-organizing maps" [104] présente une extension en trois dimensions, nommée 3DSOM. Cette évolution vise à améliorer

la capacité de clustering et de visualisation des données multidimensionnelles. Le 3D-SOM développe les algorithmes SOM en termes de nombre, de relation et d'arrangement structurel des neurones de sortie, ainsi que des processus de mise à jour du poids de voisinage et des calculs de distance dans un espace tridimensionnel. Testé sur le dataset "Iris Flowers", le 3D-SOM a montré une représentation des données plus riche en trois dimensions et a réduit significativement les erreurs de quantification et topographiques par rapport au SOM classique, démontrant ainsi son efficacité dans la formation de clusters et la visualisation des données.

"DPSOM : Deep Probabilistic Clustering with Self-Organizing Maps" [69] aborde une problématique importante dans le domaine de la visualisation et du traitement des données complexes : la capacité à générer des visualisations interprétables tout en effectuant un clustering efficace. Les auteurs proposent une fusion innovante des techniques de clustering et d'apprentissage de représentation. Leur approche introduit une nouvelle méthode nommée PSOM pour adapter les SOM avec des attributions de clusters probabilistes. Cette avancée permet de combiner les avantages visuels des SOM avec une performance de clustering améliorée, une lacune notable dans les méthodes existantes.

En outre, l'article propose l'architecture DPSOM, une intégration de la SOM avec un auto-encodeur variationnel (VAE), pour le clustering probabiliste. Cette combinaison offre une performance de clustering supérieure par rapport aux méthodes de clustering profond actuelles, tout en préservant les propriétés de visualisation bénéfiques des SOM. L'extension T-DPSOM pour le clustering de séries temporelles est particulièrement remarquable, car elle permet la prévision dans l'espace latent en utilisant des réseaux LSTM, fournissant des visualisations interprétables des trajectoires d'état des patients et une estimation de l'incertitude.

"Landmark map : An extension of the self-organizing map for a user-intended nonlinear projection" [77] répondent à la problématique de la visualisation personnalisée des données. Les SOM traditionnels, bien qu'efficaces pour la réduction de dimensionnalité, manquent souvent de flexibilité pour adapter leurs projections aux besoins spécifiques des utilisateurs. LAMA, en intégrant des landmarks, ou repères, permet aux utilisateurs

d'influencer directement la projection des données, conduisant à des visualisations plus intuitives et centrées sur l'utilisateur.

Cette approche s'avère particulièrement utile dans des contextes où la relation entre les données et leur représentation spatiale est cruciale, comme dans les systèmes de recommandation ou l'interaction homme-machine. En permettant une projection non linéaire conforme aux intentions de l'utilisateur, LAMA ouvre de nouvelles possibilités pour une exploration de données plus ciblée et significative.

2.6 Performances

Les performances des SOM ont fait l'objet de nombreuses améliorations pour répondre aux besoins croissants en termes de vitesse, d'efficacité et de tolérance aux fautes dans des contextes matériels et logiciels variés. Cette section explore des approches innovantes, telles que l'exploitation du parallélisme matériel, l'optimisation algorithmique, et l'utilisation de technologies modernes comme les GPU et les FPGAs, pour maximiser la rapidité de traitement, la résolution et la robustesse des SOM, offrant ainsi des solutions adaptées à des applications complexes et en temps réel.

Dans "New Hardware Architecture for Self-Organizing Map Used for Color Vector Quantization" [51] la Diagonal-SOM (D-SOM) se présente comme une solution innovante pour accélérer le traitement des SOM, en exploitant un parallélisme à deux niveaux entre les neurones et les connexions. Cette approche vise à améliorer l'efficacité du traitement des données, notamment dans le contexte de la quantification vectorielle des couleurs.

La D-SOM, conçue comme un noyau de langage de description matérielle (Hardware-Description-Language), se distingue par sa flexibilité et son adaptabilité à diverses topologies de SOM. La validation de cette architecture a démontré des performances temporelles presque deux fois supérieures à celles obtenues dans les études récentes. Cette avancée représente une étape significative dans le développement d'architectures SOM plus rapides et plus efficaces pour des applications de traitement de données en temps réel.

"AW-SOM, an Algorithm for High-Speed Learning in Hardware Self-Organizing Maps" [11] s'attaque à une limitation cruciale des implementations matérielles des SOM : la réduction de la vitesse de traitement avec l'augmentation du nombre de neurones et de caractéristiques. Les SOM sont largement utilisés pour le clustering, mais leur performance peut être entravée par des contraintes de vitesse de traitement, en particulier dans des architectures matérielles comme les Field Programmable Gate Arrays (FPGAs).

Pour remédier à ce problème, les auteurs introduisent l'All Winner-SOM (AW-SOM), une version modifiée de l'algorithme SOM traditionnel. Le principal atout de l'AW-SOM est sa capacité à maintenir une vitesse de traitement presque indépendante du nombre de neurones, atteignant une fréquence d'horloge de 145 MHz pour des configurations allant de 16 à 256 neurones. Cette caractéristique représente une avancée pour les applications de SOM en hardware, permettant une augmentation notable de la performance du système en termes de taux de calcul.

"High Resolution Self-organizing Maps" [74] exploite la puissance de calcul parallèle des unités de traitement graphique (GPU) pour augmenter de manière substantielle la résolution de l'espace d'affichage du SOM. Cette amélioration permet de visualiser les données de haute dimensionnalité sur un espace d'affichage de résolution élevée en un temps de calcul raisonnable.

Le principal avantage de cette approche est la capacité de la SOM à haute résolution à révéler des détails complexes sur les relations entre les vecteurs de caractéristiques d'entrée, qui seraient autrement perdus avec une SOM de faible résolution. Cette méthode est démontrée par une application à un ensemble de données artificiellement généré, le dataset "policeman", conçu pour illustrer des relations complexes entre les vecteurs de caractéristiques. L'utilisation de la puissance des GPU pour augmenter la résolution du SOM ouvre de nouvelles possibilités pour l'exploration et la compréhension détaillée des données dans divers domaines d'application.

"Randomized Self Organizing Map" [27] explore la tolérance aux fautes en se concentrant sur des implémentations ciblées sur des Field Programmable Gate Arrays (FPGA).

L'étude utilise un modèle de faute par inversion de bits pour injecter des erreurs dans les registres, évaluant l'impact sur la performance des SOM à travers des mesures de quantification et de distorsion sur des ensembles de données synthétiques. L'article examine trois techniques passives visant à améliorer la tolérance aux fautes des SOM pendant l'apprentissage. Les résultats expérimentaux révèlent une propriété importante des SOM : une dégradation en douceur face aux fautes. En outre, la tolérance aux fautes des SOM peut être considérablement améliorée en fonction de certains choix technologiques, tels que l'implémentation séquentielle ou parallèle et les politiques de stockage des poids. Les auteurs démontrent que les SOM peuvent devenir très tolérants aux fautes, notamment lorsque les poids sont stockés de manière individualisée dans l'implémentation.

2.7 Hyperparamétrage

Pour obtenir de meilleurs résultats, l'ajustement des hyperparamètres pour les algorithmes SOM est depuis longtemps reconnu comme crucial [97], mais il reste peu exploré. Plus largement, les techniques de configuration automatique, encapsulées dans le concept d'AutoML [45], sont désormais omniprésentes dans le domaine de l'apprentissage automatique [83]. Ce défi fondamental trouve ses racines dans le problème initial de la sélection d'algorithmes [92], qui a été largement étudié pour optimiser les algorithmes [42]. En conséquence, des systèmes de configuration automatique des paramètres très efficaces sont maintenant disponibles. Contrairement aux paramètres du modèle, qui sont appris automatiquement lors de l'entraînement, les hyperparamètres sont définis avant le processus d'apprentissage et restent constants pendant celui-ci.

Dans le contexte des SOM, les hyperparamètres incluent la sélection de la taille de la grille, le choix de la fonction de voisinage, le taux d'apprentissage, le nombre d'itérations et d'autres aspects cruciaux qui déterminent la qualité de la représentation et la vitesse de convergence du réseau. Un réglage approprié de ces hyperparamètres est essentiel pour obtenir des résultats optimaux, car un choix inapproprié peut entraîner une convergence lente, une représentation imprécise des données ou même l'échec de l'entraînement.

L'hyperparamétrisation est souvent un défi, car elle nécessite une compréhension approfondie de l'impact de chaque hyperparamètre sur le modèle et une exploration minutieuse de l'espace des hyperparamètres. Des méthodes telles que la validation croisée, la recherche sur grille et des techniques plus sophistiquées comme l'optimisation bayésienne sont utilisées pour trouver un ensemble optimal d'hyperparamètres. L'objectif est d'amé-

liorer les performances du modèle tout en maintenant une complexité et un temps de calcul raisonnables.

"Self-Organizing Map Improved for Classification of Partial Discharge using Desirability Function" [47] aborde une approche analytique innovante pour optimiser les paramètres via la méthodologie de surface de réponse (RSM) et la fonction de désirabilité, une combinaison qui permet de déterminer les conditions optimales pour améliorer les performances du SOM.

L'analyse se concentre sur la relation entre diverses variables explicatives, telles que l'algorithme compétitif et le taux d'apprentissage, et des variables de réponse comme le temps d'entraînement et les métriques de qualité pour le SOM. En utilisant des graphiques de surface de réponse, les auteurs examinent les effets d'interaction des principaux facteurs et déterminent les conditions optimales pour la performance en matière de classification des décharges partielles.

Cette approche méthodique révèle des insights précieux sur la manière dont différents paramètres influencent la performance globale du SOM, offrant ainsi une orientation stratégique pour le réglage fin des paramètres dans des applications spécifiques. L'utilisation de la fonction de désirabilité, en particulier, apporte une dimension nouvelle et intéressante, en aidant à équilibrer et à optimiser simultanément plusieurs objectifs de performance, ce qui est crucial dans des contextes complexes comme la classification des décharges partielles.

2.7.1 Convergence

L'amélioration de la convergence des SOM est un domaine dans la recherche sur les réseaux de neurones artificiels. Les SOM, connues pour leur capacité à cartographier et à visualiser des données complexes de haute dimension dans un espace de dimensions inférieures, sont largement appliquées dans divers domaines tels que la reconnaissance de motifs, l'analyse de données et la classification. Toutefois, l'un des défis majeurs liés à l'utilisation des SOM est la convergence efficace et fiable de l'algorithme d'apprentissage.

"Self-Organizing Map Learning with Momentum" [64] introduit le concept de momentum dans le processus d'apprentissage. Cette modification vise à pallier un des problèmes fondamentaux des SOM, à savoir leur tendance à se retrouver piégées dans des optima locaux. En intégrant le momentum, similaire à ce qui est utilisé dans l'algorithme de rétropropagation pour les réseaux de neurones, l'auteur cherche à donner aux neurones la capacité de « sauter » au-delà de ces optima locaux, offrant ainsi un potentiel d'amélioration de la convergence du réseau.

L'étude aborde également la question de la gestion du momentum, en tenant compte du mouvement précédent des neurones. Cette considération est essentielle car elle influe directement sur la dynamique d'apprentissage du réseau, en particulier pour les neurones qui ne sont pas immédiatement gagnants mais qui se trouvent dans le voisinage du neurone gagnant. Cette finesse dans la gestion du momentum est une réponse à la nature spécifique des SOM, où le mouvement d'un neurone dépend fortement de son rôle actuel dans le réseau (gagnant ou non-gagnant).

"Convergence Analysis of a New Self Organizing Map Based Optimization (SOMO) Algorithm" [52] apporte une contribution dans le domaine de l'optimisation sans dérivée. L'algorithme SOMO, une variante innovante de la SOM classique, est conçu pour résoudre des problèmes d'optimisation continue, un domaine où les SOM n'ont traditionnellement pas été largement utilisées.

Une des contributions majeures de ce travail est l'établissement de preuves formelles de convergence pour l'algorithme SOMO. Ces preuves, basées sur une mesure de distance spécifique, garantissent que la distance entre les neurones diminue à chaque itération, convergeant finalement vers zéro. De plus, il est démontré que la valeur de la fonction du neurone gagnant diminue après chaque itération, un point important pour l'efficacité de l'optimisation.

2.7.2 Paramètres

L'optimisation des paramètres des cartes auto-organisatrices SOM joue un rôle clé dans l'amélioration de leurs performances, que ce soit pour le clustering, la classification, ou l'extraction de caractéristiques. Différentes approches ont été proposées pour affiner

les hyperparamètres des SOM, notamment la taille de la carte, le taux d'apprentissage et le nombre d'itérations, afin de mieux répondre aux besoins spécifiques des applications. Cette section explore des méthodes variées, allant de l'utilisation d'algorithmes génétiques à des stratégies de fine-tuning pour des scénarios spécifiques, démontrant l'importance d'un paramétrage précis pour maximiser l'efficacité et la précision des SOM.

"Optimizing Self-Organizing Maps Parameters Using Genetic Algorithm : A Simple Case Study" [2] traite de l'optimisation des paramètres d'une Self-Organizing Map (SOM), clustering, et réduction de dimensionnalité. Les auteurs soulignent que la qualité et l'exactitude des résultats d'une SOM dépendent fortement du choix des hyperparamètres tels que la taille de la carte, le nombre d'itérations, et le taux d'apprentissage initial.

Le papier se concentre sur l'utilisation d'un algorithme génétique pour optimiser ces paramètres. Un algorithme génétique est une méthode de recherche inspirée par la biologie, qui simule le processus de sélection naturelle pour identifier les meilleures solutions. Les auteurs proposent une nouvelle approche pour sélectionner et optimiser les hyperparamètres des SOM en utilisant cette technique. Ils présentent leurs premiers résultats en appliquant cette méthode sur le problème classique de regroupement de couleurs. Cette application sert d'étude de cas pour démontrer l'efficacité de leur proposition. Le choix du problème de clustering de couleurs offre une base solide pour évaluer la performance de l'algorithme génétique dans l'optimisation des paramètres des SOM, car c'est un problème bien connu et relativement simple.

"Automatic Parameter Selection of Granual Self-organizing Map for Microblog Summarization" [86] présente une technique de classification non supervisée basée sur un réseau de neurones pour la synthèse de tweets, en sélectionnant des tweets informatifs en fonction de leur importance. La méthode se déroule en deux étapes : dans un premier temps, une carte auto-organisatrice (SOM) est utilisée pour réduire le nombre de tweets. Dans la seconde étape, une carte auto-organisatrice granulaire (GSOM), qui est un réseau neuronal à deux couches utilisant la théorie des ensembles flous et rugueux pour son entraînement, est utilisée pour regrouper l'ensemble réduit de tweets. Ensuite, un résumé de longueur fixe est généré en sélectionnant des tweets des clusters obtenus. La GSOM possède un ensemble de paramètres dont la sélection adéquate influence la

performance. Par conséquent, une technique d'optimisation évolutive est utilisée pour la sélection des meilleures combinaisons de paramètres. L'efficacité de l'approche proposée a été évaluée sur quatre ensembles de données de microblogs liés à des catastrophes. Les résultats montrent clairement que la méthode proposée surpasse les méthodes actuelles les plus avancées.

"A model to estimate the Self-Organizing Maps grid dimension for Prototype Generation Article type : Research Article" [90] traite de l'amélioration du classificateur K plus proches voisins (KNN) en utilisant une méthode de génération de prototypes (PG) pour réduire le temps de classification, particulièrement pour les grands ensembles de données. KNN est précis mais lent, et la PG vise à représenter les exemples de données par des prototypes, rendant la classification plus rapide. Cependant, la détermination du nombre idéal de prototypes reste un défi. Cette étude propose un modèle pour estimer la meilleure dimension de grille pour les cartes auto-organisatrices (SOM) et ainsi déterminer le nombre optimal de prototypes basé sur le nombre d'exemples dans l'ensemble de données. Comparée à d'autres méthodes de PG utilisant l'intelligence artificielle, la méthode proposée présente l'avantage de maintenir un bon équilibre entre un nombre réduit de prototypes et la précision, sans dégrader la performance de classification de KNN.

"A constant learning rate self-organizing map (CLRSOM) learning algorithm" [15] aborde une innovation dans l'entraînement, l'utilisation d'un taux d'apprentissage constant. Contrairement aux SOM traditionnels qui nécessitent un taux d'apprentissage décroissant et précisément calibré, le CLRSOM maintient un taux d'apprentissage constant tout au long du processus. Cette méthode simplifie le choix du taux d'apprentissage, un défi commun avec les SOM classiques où un taux inadapté peut conduire à une convergence insuffisante ou à un temps de traitement excessif.

Le CLRSOM sélectionne intelligemment à la fois le neurone le plus proche et le plus éloigné de l'Unité de Meilleure Correspondance (BMU), offrant ainsi une performance améliorée malgré un taux d'apprentissage constant. Appliqué à divers ensembles de données standards, le CLRSOM a démontré une amélioration substantielle de la performance d'apprentissage, préservant la topologie des données d'entrée sans sacrifier l'erreur de

quantification et les niveaux d'utilisation des neurones, comparativement aux SOM traditionnels.

"Fine-Tuning Self-Organizing Maps for Sentinel-2 Imagery : Separating Clouds from Bright Surfaces" [62] se concentre sur l'utilisation des SOM pour le masquage des nuages dans les images satellites Sentinel-2, une méthode prometteuse grâce à sa capacité à préserver les relations topologiques et à son temps d'entraînement plus rapide par rapport à d'autres réseaux de neurones. Les auteurs proposent une méthodologie de fine-tuning pour améliorer la performance du SOM dans la séparation des nuages des surfaces terrestres brillantes. Cette méthode repose sur l'ajustement des étiquettes incorrectes des neurones, identifiées par le réseau initial, sans nécessiter de formation supplémentaire.

L'étude est remarquable par son approche méthodique : le réseau SOM a été entraîné sur la plus grande base de données spectrale publique pour le masquage des nuages de Sentinel-2 et testé sur une base de données indépendante. L'évaluation du SOM fine-tuné a été réalisée tant qualitativement que quantitativement, en utilisant plusieurs techniques de visualisation pour interpréter son comportement.

2.8 Conclusion

La gestion des données dans le contexte des SOM a montré une évolution significative pour répondre aux besoins complexes et diversifiés des applications modernes. Ce chapitre a exploré des avancées majeures dans l'adaptation des SOM pour traiter divers types de données, notamment semi-supervisées, bruitées, catégorielles, volumineuses, ou temporelles, en mettant en lumière des solutions spécifiques et novatrices.

Les contributions récentes ont permis d'améliorer les performances des SOM dans des scénarios où les données sont complexes, rares ou mal structurées. Qu'il s'agisse d'intégrer des données catégorielles directement sans conversion préalable, de gérer efficacement les données manquantes, ou d'exploiter des métriques adaptées pour des distributions complexes, ces innovations élargissent le champ d'application des SOM. De plus, les approches visant à traiter de grandes quantités de données ou à capturer des dynamiques temporelles démontrent une capacité d'adaptation accrue des SOM aux exigences des données modernes.

Gestion des données	Datasets semi-supervisés	[9, 3]
	Datasets perturbés	[94, 30]
	Données catégoriques	[43, 18, 53]
	Distributions spécifiques des données	[1, 20]
	Datasets volumineux	[46, 4]
	Données temporelles et séquences	[89, 44]
Topologies et Métriques	Topologies	[68, 5, 93, 85, 99]
	Distances	[101, 13, 12, 23, 102]
	Relations de voisinage	[41, 76, 40, 75, 24, 29]
Techniques d'apprentissage	Approche générale d'apprentissage	[16, 7, 48, 21]
	Extraction de caractéristiques	[87, 50]
	Phase d'entraînement	[33, 96]
	Clustering	[70, 78, 79, 105]
Visualisation		[73, 81, 104, 69, 77]
Performances		[51, 11, 74, 27]
Hyperparamétrisation	Convergence	[64, 52]
	Paramètres	[2, 86, 90, 62]

TABLE 2.1 – Résumé biobliographique des travaux cités

Ces avancées soulignent non seulement l'adaptabilité des SOM mais aussi leur potentiel à devenir des outils incontournables dans l'exploration et l'analyse de données multidimensionnelles et complexes. Les perspectives futures incluent des efforts continus pour accroître leur scalabilité, leur précision et leur interprétabilité, ouvrant ainsi la voie à de nouvelles applications dans des domaines tels que l'intelligence artificielle, la bioinformatique, et l'analyse des systèmes complexes.

Table 2.1 fournit un aperçu des travaux précédemment cités selon notre schéma d'analyse illustré dans la Figure 2.2.

2.9 Application aux produits clients

L'application des SOM dans des domaines spécifiques, notamment le marketing et les produits orientés client, présente un potentiel considérable, bien qu'elle soit encore relativement peu exploitée comparativement à d'autres secteurs tels que le médical. Les SOM peuvent offrir des perspectives uniques dans l'analyse de comportements de consommation, la segmentation de clientèle, et le développement de recommandations personnalisées.

"Online recommendation based on incremental-input self-organizing map" [103] traite d'un problème crucial dans le domaine du commerce électronique : l'amélioration des modèles de recommandation en ligne. Dans les plateformes de e-commerce, les algorithmes de recommandation jouent un rôle essentiel en aidant les consommateurs à découvrir des produits susceptibles de les intéresser. Cependant, la plupart des modèles traditionnels de recommandation, basés sur des données statiques, se heurtent à des difficultés lorsqu'il s'agit de gérer les nouvelles évaluations des consommateurs ou des produits récemment ajoutés.

L'inconvénient principal des modèles statiques est la nécessité de les réentraîner hors ligne avec de nouvelles évaluations, un processus souvent inefficace et chronophage. Pour résoudre ce problème, cet article propose un modèle de recommandation en ligne innovant utilisant une carte auto-organisatrice à entrées incrémentales. Cette approche permet d'ajouter de nouveaux consommateurs au modèle en augmentant les unités d'entrée et en mettant à jour les poids correspondants entre la couche d'entrée et la couche de sortie. Les nouveaux produits sont traités comme de nouveaux échantillons arrivant à la SOM, et une stratégie d'apprentissage incrémental spécialement conçue pour la SOM est mise en œuvre.

Grâce à cette méthode, le modèle proposé n'a pas besoin d'être réentraîné et peut être mis à jour en ligne avec très peu d'évaluations de nouveaux consommateurs ou produits. Les résultats expérimentaux sur des ensembles de données réels montrent que l'algorithme proposé surpasse plusieurs algorithmes de recommandation en ligne existants en termes de précision de recommandation, et offre des performances similaires à celles de l'algorithme de recommandation statique qui nécessite un réentraînement avec l'ensemble des données.

Cette approche représente une avancée significative dans le domaine des systèmes de recommandation, offrant une solution plus dynamique et réactive aux besoins changeants des plateformes de e-commerce et de leurs utilisateurs. En intégrant des données en temps réel et en adaptant continuellement le modèle aux nouveaux utilisateurs et produits, cet algorithme peut améliorer considérablement l'expérience d'achat en ligne, en rendant les recommandations plus pertinentes et personnalisées.

"Using Self Organizing Maps and K Means Clustering Based on RFM Model for Customer Segmentation in the Online Retail Business" [100] explore une approche innovante pour la segmentation des clients dans le commerce de détail en ligne. Le modèle RFM (Récence, Fréquence, Montant) est une méthode analytique utilisée pour segmenter les clients en fonction de leur historique d'achat. Il évalue les clients sur la base de trois critères : la récence (quand un client a effectué son dernier achat), la fréquence (à quelle fréquence un client effectue des achats) et le montant monétaire (combien un client dépense).

Dans cet article, les performances des algorithmes de clustering k-means et des SOM sont comparées pour l'ensemble de données ciblé filtré. Les SOM sont utilisées pour fournir un cadre de calcul de réseaux neuronaux, qui peut être comparé au simple algorithme k-means utilisé par Chen et al. Cette approche hybride, combinant les SOM et k-means, offre une nouvelle perspective dans l'analyse des comportements d'achat des clients et la segmentation dans le secteur de la vente en ligne.

L'utilisation des SOM, en particulier, permet une visualisation plus intuitive des groupes de clients et peut révéler des modèles complexes dans les données qui ne sont pas immédiatement évidents avec des méthodes traditionnelles comme le k-means. Cette méthode offre donc une opportunité d'améliorer la personnalisation des stratégies de marketing et de vente, en ciblant plus efficacement les différents segments de clients en fonction de leurs comportements d'achat spécifiques.

"Conceptual design of a user-centric smart product-service system using self-organizing map" [17] s'intéresse à un paradigme commercial émergent : le système de produit-service intelligent (Smart PSS). Ce concept combine des produits intelligents et connectés (SCP) avec des services associés, offrant ainsi un ensemble de solutions intégrées. L'étude reconnaît l'importance croissante des Smart PSS dans la recherche, notamment en ce qui concerne la conception conceptuelle qui tient compte simultanément des SCP et des services.

Un défi majeur identifié dans la conception des Smart PSS est l'adéquation des solutions proposées avec les exigences émotionnelles des utilisateurs. Pour relever ce défi, l'article propose une méthode de conception conceptuelle axée sur l'analyse des émotions générées par les utilisateurs lors de leurs interactions avec les produits. Cette méthode

comprend l'identification de produits traditionnels pertinents, l'analyse des émotions des utilisateurs à partir de données d'avis publics, et l'utilisation d'un tableau d'émotions interactif comme outil de conception pour organiser ces émotions et les éléments de conception associés.

La carte SOM joue un rôle clé dans cette méthode, car elle est utilisée pour le regroupement des échantillons de produits et des mots Kansei (sensations ou émotions en japonais), facilitant ainsi l'analyse et l'organisation des données émotionnelles. De plus, le processus d'évaluation de la solution améliorée est effectué en utilisant l'analyse hiérarchique des processus (AHP), une méthode structurée pour organiser et analyser des décisions complexes.

"A Proposal of a Model Using Kansei Evaluation Integrated with Fuzzy Rules and Self-Organizing Map for Evaluation of Bio-Food Products" [82] introduit une méthode innovante pour évaluer la qualité des produits bio-alimentaires dans le contexte de marchés mondiaux. Cette approche se concentre particulièrement sur les comportements des consommateurs interculturels et leur inclination vers des aliments sains.

La méthode proposée utilise l'évaluation Kansei, qui est une technique japonaise visant à capturer et à analyser les réactions émotionnelles des consommateurs face à un produit ou un service. Cette évaluation est intégrée avec des règles floues et un modèle de carte SOM pour mieux comprendre et répondre aux préférences et sensibilités des clients et des experts. L'objectif est de sélectionner des alternatives appropriées en matière de produits bio-alimentaires et de technologies alimentaires qui correspondent aux comportements des groupes de consommateurs et d'experts de masse.

Le modèle vise à améliorer la qualité des produits alimentaires dans les marchés bio-alimentaires mondiaux en tenant compte des diverses préférences culturelles et des tendances de consommation saine. Pour valider l'efficacité de cette approche, le modèle a été testé expérimentalement et appliqué dans plusieurs domaines dans les pays asiatiques.

La série d'élément examiné dans cette section met en évidence l'utilité des SOM dans le domaine du marketing et du développement de produits. Chaque article apporte une perspective unique sur la manière dont les SOM peuvent être utilisées pour mieux comprendre et répondre aux besoins et préférences des clients. Ensemble, ces études illustrent le potentiel des SOM pour apporter des solutions innovantes dans le domaine du marketing et du développement de produits, en particulier dans des secteurs où la compréhension

des préférences et des comportements des clients est cruciale. Elles soulignent également l'importance de l'intégration de techniques avancées d'intelligence artificielle et d'analyse de données pour répondre aux besoins en constante évolution des consommateurs dans l'économie mondiale actuelle.

Chapitre 3

Gestion des données

La gestion des données est une étape clé dans tout processus d'analyse, en particulier lorsque l'on travaille avec des modèles comme les cartes auto-organisatrices (SOM). Les données réelles sont souvent imparfaites, contenant des valeurs manquantes, des variables qualitatives ou encore des informations temporelles. Ce chapitre explore les différentes stratégies pour gérer ces aspects, en se concentrant sur l'intégration des données manquantes, qualitatives et temporelles dans l'algorithme SOM.

3.1 Données manquantes

Les données manquantes représentent un défi courant dans le traitement et l'analyse de données, affectant potentiellement la fiabilité et la précision des modèles d'apprentissage automatique, y compris les cartes auto-organisatrices (SOM). L'approche traditionnelle des SOM requiert un ensemble de données complet pour fonctionner efficacement, car chaque dimension des données contribue à la formation de la carte. Cependant, dans la pratique, il est fréquent de se retrouver avec des jeux de données incomplets pour diverses raisons, telles que des erreurs de saisie, des réponses manquantes dans les enquêtes, ou des capteurs défectueux dans les contextes de collecte de données automatisée. La gestion des données manquantes devient donc une étape cruciale dans le processus d'analyse des SOM.

Les données manquantes peuvent conduire à une distorsion de la carte auto-organisatrice, affectant la capacité du modèle à apprendre correctement la structure inhérente des données. Cela peut se traduire par une représentation inexacte des clusters ou par des difficultés dans l'interprétation des résultats obtenus.

Plusieurs stratégies peuvent être adoptées pour gérer les données manquantes dans le contexte des SOM, chacune avec ses avantages et inconvénients :

Plusieurs stratégies peuvent être adoptées pour gérer les données manquantes dans le contexte des SOM, chacune présentant ses avantages et limites. Nous en distinguons trois

principales, que nous formalisons ci-dessous.

Imputation des données manquantes Soit $X = (x_{ij}) \in \mathbb{R}^{N \times p}$ un ensemble de données comportant des valeurs manquantes. L'imputation consiste à remplacer toute valeur manquante $x_{ij} = \text{NA}$ par une estimation $\hat{x}_{ij}$ issue d'une fonction d'estimation f, telle que :

$$\hat{x}_{ij} = f(X_{-ij}),$$

où X_{-ij} désigne l'ensemble des valeurs observées dans X (hors x_{ij}). Les fonctions f les plus couramment utilisées sont la moyenne, la médiane ou des modèles de prédiction supervisée.

Suppression des entrées incomplètes Cette approche consiste à retirer toute observation x_i contenant au moins une valeur manquante :

$$X' = \{x_i \in X \mid \forall j,\, x_{ij} \neq \text{NA}\}.$$

Elle garantit l'absence de valeurs manquantes dans X', mais réduit la taille effective de l'échantillon, ce qui peut introduire un biais si les données manquantes ne sont pas aléatoires.

Adaptation de la mesure de distance dans les SOM Plutôt que de modifier ou supprimer les données, on adapte la mesure de distance utilisée pour la comparaison entre un vecteur d'entrée X^* et les vecteurs de poids W_j de la carte. Soit $J^* = \{j \mid X_j^* \text{ observé}\}$ l'ensemble des dimensions connues. La distance est alors redéfinie comme :

$$d(X^*, W_j) = \sqrt{\sum_{k \in J^*} (X_k^* - W_{jk})^2}.$$

Cette adaptation permet d'exploiter directement les données partielles sans recourir à des imputations et sans exclure d'observations.

Le choix de la stratégie de gestion des données manquantes dépend de plusieurs facteurs, notamment la proportion de données manquantes, la nature des données (complètement aléatoire, aléatoire, ou non aléatoire), et les objectifs spécifiques de l'analyse. Il est essentiel d'évaluer soigneusement l'impact de chaque approche sur la qualité et l'interprétation des cartes SOM générées.

L'approche traditionnelle dans les SOM, qui repose sur le calcul de la distance complète entre un vecteur d'entrée et les vecteurs de poids des neurones, se heurte à une difficulté majeure en présence de données manquantes. Ces données manquantes empêchent un calcul direct de la distance, puisque certaines dimensions du vecteur d'entrée X^* peuvent être absentes.

Pour pallier ce problème, nous proposons d'ajuster le calcul de la distance d de manière à exclure les dimensions manquantes du vecteur d'entrée X^*. Cette adaptation permet de calculer une distance modifiée qui ne tient compte que des dimensions présentes, offrant ainsi une solution viable pour intégrer des données partielles dans le processus d'apprentissage des SOM. En pratique, cela implique de redéfinir la mesure de distance utilisée pour l'attribution des vecteurs d'entrée aux neurones de la carte, en fonction des dimensions effectivement disponibles.

Cette méthode modifiée pour le calcul de la distance est particulièrement pertinente dans le cadre des SOM, où la flexibilité dans le traitement des données est souvent nécessaire pour traiter avec des jeux de données réels, fréquemment incomplets ou imparfaits. En adaptant l'algorithme pour ignorer les dimensions manquantes dans X^*, nous maintenons l'intégrité de l'apprentissage sans sacrifier la qualité de la modélisation.

En adoptant cette approche modifiée, l'algorithme SOM devient plus robuste face aux données manquantes, élargissant ainsi son applicabilité à une gamme plus large de scénarios d'analyse de données. Cette adaptation représente une étape significative vers la création de modèles SOM plus flexibles et plus résilients, capables de gérer efficacement les incertitudes et les imperfections inhérentes aux ensembles de données réels.

Cette approche, bien qu'elle existe déjà dans la littérature [58, 39], est rarement employée en pratique pour gérer les données manquantes dans les SOM. Par simplicité, de nombreuses implémentations privilégient des stratégies telles que l'imputation des données manquantes ou l'exclusion des observations incomplètes, malgré les biais et les limitations qu'elles peuvent introduire. Pourtant, ajuster le calcul de la distance pour ignorer les dimensions manquantes offre une solution plus pertinente dans de nombreux cas, en préservant l'intégrité des données originales tout en permettant une modélisation plus fidèle et robuste. Cette méthode constitue un choix particulièrement adapté dans les contextes où les jeux de données sont fréquemment incomplets, démontrant ainsi une flexibilité accrue face aux imperfections des données réelles.

Rappelons que dans l'algorithme SOM (voir chapitre 1), la distance d est utilisée pour identifier le neurone dont le vecteur de poids W_j est le plus proche d'un vecteur d'entrée

X^*. En présence de données manquantes dans X^*, le calcul de cette distance devient problématique car certaines composantes du vecteur sont absentes. Nous proposons donc une adaptation formelle du calcul de d.

Distance adaptée en présence de données manquantes Soient :

- $X^* = (x_1, \ldots, x_p)$ un vecteur d'entrée partiellement observé,
- $W_j = (w_{j1}, \ldots, w_{jp})$ le vecteur de poids associé au neurone j,
- $J^* = \{k \mid x_k \text{ observé}\}$ l'ensemble des indices correspondant aux dimensions présentes dans X^*.

La distance adaptée est alors définie par :

$$d(X^*, W_j) = \sqrt{\sum_{k \in J^*} (x_k - w_{jk})^2}.$$

Cette formulation garantit que seules les dimensions effectivement observées contribuent au calcul de la distance, sans qu'aucune imputation ne soit nécessaire.

Extension du calcul de distance Dans certains cas, il est utile d'étendre la définition de d à des sous-ensembles de l'espace des entrées $E = [0, 1]^n$. Soit $A, B \subset E$ deux sous-ensembles non vides. On définit :

$$d(x, A) = \inf_{a \in A} d(x, a), \quad \text{avec } x \in E,$$

$$d(A, B) = \inf_{a \in A,\, b \in B} d(a, b).$$

Cette extension permet d'interpréter les cas de données partielles comme des comparaisons entre un point (vecteur observé partiellement) et un sous-espace (ensemble des vecteurs complets compatibles avec les valeurs observées). Ainsi, tant que chaque vecteur X^* contient au moins une composante observée ($|J^*| \geq 1$), l'algorithme SOM reste applicable sans prétraitement particulier. Le seul cas bloquant est celui où un vecteur d'entrée est entièrement vide ($|J^*| = 0$), empêchant tout calcul de distance.

En résumé, les données manquantes constituent un défi majeur pour les cartes auto-organisatrices (SOM), dont le calcul repose sur la comparaison complète entre les vecteurs d'entrée et les vecteurs de poids. Trois approches principales peuvent être envisagées :

- **Imputation** : Remplacer les valeurs manquantes par des estimations (moyenne, médiane, prédictions).
- **Suppression** : Exclure les observations incomplètes, au risque de perdre une partie significative des données.
- **Adaptation de l'algorithme** : Modifier la mesure de distance pour ignorer les dimensions manquantes, sans altérer les données originales.

Dans cette thèse, nous retenons cette dernière approche, en adaptant la distance d de sorte qu'elle ne prenne en compte que les dimensions présentes dans chaque vecteur d'entrée X^*. Cette méthode permet à l'algorithme SOM de fonctionner efficacement sur des données incomplètes sans recourir à des imputations potentiellement biaisées. Elle rend le modèle plus robuste, plus flexible et mieux adapté aux ensembles de données réels, souvent partiellement observés.

3.2 Données qualitatives

Les données qualitatives contiennent souvent des connaissances précieux sur les phénomènes étudiés, offrant une richesse d'informations qui ne peut être capturée par des mesures purement quantitatives.

La principale difficulté réside dans la conversion des attributs qualitatifs en une forme numérique qui préserve leur signification originale, sans introduire de distorsions ou de biais. Cette conversion nécessite une attention particulière pour s'assurer que les distances calculées entre les vecteurs reflètent fidèlement les similarités et les différences dans l'espace des attributs qualitatifs.

Plusieurs stratégies peuvent être employées pour encoder les données qualitatives, telles que l'encodage one-hot [37, 80], l'encodage par fréquence ou par probabilité conditionnelle [8], ou encore des techniques plus sophistiquées basées sur l'embedding de mots [71, 22]. Chaque méthode a ses avantages et ses limites, et le choix parmi celles-ci dépend du contexte spécifique et des objectifs de l'analyse.

Pour les attributs avec un nombre limité d'occurrences, tels que les catégories ou les étiquettes, l'encodage binaire, souvent appelé encodage one-hot, est fréquemment utilisé. Cette technique consiste à générer une colonne supplémentaire pour chaque catégorie

unique dans le jeu de données. Dans ces colonnes, un "1" représente la présence de la catégorie dans l'observation, tandis qu'un "0" indique son absence.

Toutefois, bien que l'encodage binaire soit efficace pour traiter les données qualitatives, il peut considérablement augmenter la dimensionnalité des données, surtout quand le nombre de catégories distinctes est important. Une dimensionnalité accrue peut non seulement rendre l'analyse plus complexe mais aussi diminuer la performance des SOM en diluant l'information significative parmi un grand nombre de caractéristiques binaires.

Pour contrer ce problème de dimensionnalité élevée, il est courant de recourir à des techniques de réduction de dimension telles que l'Analyse en Composantes Principales (ACP). L'ACP permet de résumer efficacement l'information contenue dans les données encodées en un nombre réduit de composantes, qui capturent l'essentiel de la variance des données. En appliquant l'ACP ou des méthodes similaires, il est possible de conserver les informations cruciales pour l'analyse tout en réduisant la complexité des données traitées par le SOM.

En somme, l'encodage binaire, combiné à des techniques de réduction de dimension, offre une stratégie viable pour intégrer des données qualitatives dans les analyses SOM, en surmontant les défis posés par la dimensionnalité accrue et en enrichissant l'analyse avec une diversité de types de données.

Les données textuelles, souvent riches et variées, ne se prêtent pas à une simple transformation numérique comme l'encodage binaire. Elles contiennent une multitude d'informations qui ne se répètent pas nécessairement de manière régulière, rendant leur vectorisation et leur analyse par les méthodes classiques non triviales. Pour traiter ces données d'une manière qui préserve leur richesse et leur nuance, il est essentiel d'adopter des techniques avancées de traitement du langage naturel (NLP).

Des méthodes telles que **TF-IDF** (Term Frequency–Inverse Document Frequency) [88], **Word2Vec** [72], ou encore des modèles de langage pré-entraînés basés sur des réseaux de neurones profonds, tels que **BERT** (Bidirectional Encoder Representations from Transformers) [22], offrent des moyens sophistiqués de convertir le texte en vecteurs numériques. Ces représentations permettent de capturer des relations sémantiques et contextuelles entre les mots ou les documents, qui peuvent ensuite être analysées par les SOM pour identifier des regroupements et des motifs dans les données textuelles.

La spécificité et la complexité de la vectorisation des données textuelles exigent une attention particulière dans le choix des techniques de NLP, ainsi qu'une compréhension approfondie des données elles-mêmes. Le processus de conversion du texte en vecteurs

doit être soigneusement conçu pour s'assurer que les nuances et les subtilités des données textuelles sont préservées et correctement représentées dans l'espace vectoriel.

En résumé, le prétraitement des données qualitatives, et plus particulièrement des données textuelles uniques, est une étape cruciale pour leur intégration réussie dans les analyses SOM. Cette étape nécessite une combinaison de méthodes de NLP et de techniques de vectorisation avancées pour garantir que l'analyse SOM puisse bénéficier pleinement de la richesse des informations qualitatives. Il est possible d'enrichir significativement les modèles SOM avec des analyses et classifications robustes et précises, ouvrant ainsi la voie à des insights plus profonds et nuancés sur les données étudiées.

3.3 Données textuelles

Word2Vec [72] est un modèle de traitement de langage naturel qui transforme les mots en vecteurs numériques, capturant ainsi leur signification et les relations contextuelles. Ce modèle s'appuie sur des réseaux de neurones et utilise deux architectures principales : le modèle « Continuous Bag of Words » (CBOW) et le modèle « Skip-Gram ».

CBOW : Le modèle CBOW prédit un mot cible à partir d'un ensemble de mots contextuels. Dans cette architecture, les mots contextuels sont utilisés comme entrée du réseau de neurones, et le modèle est entraîné pour prédire le mot cible. Le réseau ajuste ses poids de manière à maximiser la probabilité du mot cible étant donné le contexte. Cette approche est particulièrement efficace pour traiter de grands corpus de texte et pour apprendre des représentations de mots de haute qualité en termes de sémantique.

Skip-Gram : À l'inverse, le modèle Skip-Gram fonctionne en prédisant les mots contextuels à partir d'un mot cible. Dans ce cas, le mot cible est fourni en entrée, et le réseau est entraîné pour prédire les mots contextuels. Ce modèle est particulièrement utile pour capturer des représentations précises de mots rares ou de mots avec plusieurs significations.

Dans les deux cas, le résultat est un ensemble de vecteurs, où chaque vecteur représente un mot du vocabulaire. Ces vecteurs sont positionnés dans un espace multidimensionnel de telle manière que les mots ayant des contextes similaires sont proches l'un de l'autre dans cet espace. Cela permet non seulement de capturer la signification des mots individuels, mais aussi de détecter des similarités et des relations entre différents mots.

L'un des principaux avantages de Word2Vec est sa capacité à transformer le texte en

vecteurs denses de faible dimension, qui encapsulent les relations sémantiques complexes. Cela permet au SOM de traiter efficacement les données textuelles, offrant une possibilité d'analyser et de classer des textes basés sur leur contenu sémantique.

Cependant, l'utilisation de Word2Vec présente également certaines limites. Le modèle nécessite un large corpus de texte pour un entraînement efficace, ce qui peut être un défi pour des domaines spécifiques avec un vocabulaire limité ou spécialisé. De plus, Word2Vec peut parfois omettre des nuances de sens, en particulier avec des mots polysémiques (ayant plusieurs significations) ou des contextes très spécifiques. En outre, la conversion de textes entiers en vecteurs moyens peut conduire à une perte d'informations contextuelles et temporelles.

En résumé, les données textuelles, riches en informations sémantiques, nécessitent une vectorisation adaptée pour être exploitées dans les cartes auto-organisatrices (SOM). Le modèle **Word2Vec** transforme les mots en vecteurs denses continus, en apprenant leurs relations contextuelles à partir de grands corpus textuels. Deux architectures principales sont utilisées :

- **CBOW (Continuous Bag of Words)** : prédit un mot à partir de son contexte.
- **Skip-Gram** : prédit le contexte à partir d'un mot cible, utile pour les mots rares.

Ces représentations vectorielles permettent au SOM de regrouper des textes selon leur similarité sémantique, facilitant l'identification de thématiques ou de styles. Cependant, Word2Vec requiert un corpus volumineux et peut perdre des nuances contextuelles, notamment dans le cas de mots polysémiques ou de domaines spécialisés. Malgré ces limites, il constitue une approche efficace pour intégrer des données textuelles dans les SOM, en conservant la richesse sémantique du langage sous une forme exploitable pour l'apprentissage non supervisé.

3.4 Données temporelles

Dans de nombreuses applications commerciales et industrielles, les données temporelles jouent un rôle clé dans la compréhension des comportements et des tendances. Par exemple, dans le cadre de la gestion omnicanale et de l'analyse des comportements clients, il est crucial de suivre l'évolution des achats au fil du temps pour identifier des schémas saisonniers, des changements de préférences, ou des impacts liés à des événements spéci-

fiques. L'intégration efficace des données temporelles dans l'algorithme SOM permet de mieux exploiter cette dimension temporelle et de fournir des connaissances stratégiques à forte valeur ajoutée.

Pour traiter les données temporelles dans les SOM, une approche commune consiste à représenter chaque instant ou période temporelle comme une dimension distincte dans les vecteurs d'entrée [58, 14]. Cette représentation permet de capturer les variations temporelles, mais elle peut aussi augmenter significativement la dimensionnalité des données. Il est donc important de considérer des méthodes pour gérer cette augmentation de dimension, telles que la sélection de caractéristiques ou les techniques de réduction de dimension [66].

En outre, lors de l'analyse des cartes SOM obtenues, il est crucial de prendre en compte l'aspect temporel des données. Les regroupements ou motifs découverts dans la carte doivent être interprétés à la lumière de leur séquence temporelle, afin d'identifier des tendances, transitions ou comportements dynamiques [98, 14].

Chapitre 4

Hyperparamétrage

Dans cette section, nous allons introduire une méthodologie qui vise à affiner les critères d'évaluation traditionnels des SOM, en proposant des métriques adaptées pour mieux capturer la qualité et la pertinence des résultats obtenus. Une synthèse de ces travaux a fait l'objet d'une publication à la conférence AIxSET 2024 [32], présentant en détail la méthodologie d'optimisation des hyperparamètres des cartes auto-organisatrices et l'évaluation expérimentale de ses performances.

4.1 Score

L'évaluation de la qualité d'un algorithme de classification non supervisée, comme les cartes auto-organisatrices (SOM), repose généralement sur des mesures de cohésion et de séparation entre les groupes formés. Ces mesures peuvent être divisées en deux grandes catégories : les mesures classiques de distances inter et intra-clusters, et les mesures spécifiques à l'architecture et au fonctionnement des SOM.

4.1.1 Mesures classiques de distances

Les mesures de distances intra et inter-clusters constituent les indicateurs fondamentaux de la qualité d'un partitionnement.

Distance intra-cluster : Cette mesure évalue la cohésion interne des clusters formés par l'algorithme. Une faible distance intra-cluster signifie que les éléments d'un même cluster sont proches les uns des autres dans l'espace des caractéristiques, indiquant une bonne homogénéité au sein des clusters. Dans le contexte des SOM, cela correspond à la proximité des neurones activés pour des entrées similaires, traduisant une organisation spatiale cohérente de la carte.

$$\text{Distance intra-cluster} = \frac{1}{N_c} \sum_{X_i \in C} d(X_i, \mu_C)$$

— N_c : Nombre d'éléments dans le cluster C ;
— $X_i \in C$: Points appartenant au cluster C ;
— $d(X_i, \mu_C)$: Distance entre un point X_i et le centroïde μ_C ;
— $\mu_C = \frac{1}{N_c} \sum_{X_i \in C} X_i$: Centroïde du cluster.

Distance inter-cluster : Cette mesure évalue la séparation entre les différents clusters identifiés. Une grande distance inter-cluster traduit une bonne différenciation entre les groupes, donc une segmentation claire de l'espace des données.

$$\text{Distance inter-cluster} = d(\mu_{C_1}, \mu_{C_2})$$

— μ_{C_1} et μ_{C_2} : Centroïdes de deux clusters C_1 et C_2 ;
— $d(\mu_{C_1}, \mu_{C_2})$: Distance entre ces deux centroïdes.

Ces deux mesures constituent des bases d'évaluation générales pour tout algorithme de clustering, mais elles ne capturent pas les spécificités topologiques propres aux cartes SOM.

4.1.2 Mesures spécifiques aux cartes SOM

Au-delà de ces indicateurs génériques, d'autres métriques sont spécifiquement conçues pour évaluer les SOM. Parmi elles, l'erreur de quantification [26] occupe une place centrale : elle mesure l'écart moyen entre chaque vecteur d'entrée et le neurone vainqueur (Best Matching Unit, BMU) sur la carte. Une faible erreur de quantification indique que la SOM représente fidèlement la distribution des données d'origine.

Comme le souligne l'étude "A Survey and Implementation of Performance Metrics for Self-Organized Maps" [26], l'analyse de telles métriques permet d'affiner la configuration des SOM pour des applications spécifiques, en recherchant un équilibre entre la qualité de représentation et l'efficacité computationnelle.

Pour définir formellement les différents scores employés dans cette étude, nous introduisons les notations suivantes :

— $X = \{X_1, \ldots, X_N\}$: Ensemble des données initiales ;
— E : Ensemble des poids de la carte SOM ;
— N : Nombre total d'observations dans X ;
— $d(W, X_i)$: Distance entre un neurone $W \in E$ et une donnée $X_i \in X$ (voir Chapitre 1) ;
— $W_i^* = \arg\min_{W \in E} d(W, X_i)$: Neurone vainqueur (BMU) associé à X_i ;

— $d^*(W_a, W_b)$: Distance topologique entre deux neurones sur la carte ;

— σ_0 : Rayon initial de voisinage sur la carte.

Ces définitions serviront de base à la formulation des différents scores de performance détaillés dans les sections suivantes.

4.1.3 Erreur de quantification

L'erreur de quantification [26] est une métrique utilisée pour évaluer la précision de la projection des données sur une carte SOM. Elle indique combien la représentation sur la carte s'écarte des données originales.

On définit le score S_1 :

$$S_1(X, E) = \frac{1}{N} \sum_{i=1}^{N} d(W_i^*, X_i)$$

Avec,

$$W_i^* = argmin(d(W, X_i), W \in E), \quad i \in [\![1, N]\!]$$

L'erreur de quantification est calculée en prenant la distance euclidienne entre chaque échantillon de données et son neurone correspondant le plus proche sur la carte. Après avoir calculé ces distances pour tous les échantillons de données, on en fait la moyenne pour obtenir l'erreur de quantification. Une faible erreur de quantification indique que la SOM représente fidèlement les données d'origine, tandis qu'une erreur élevée suggère une représentation moins précise. Toutefois, il est important de noter qu'une erreur de quantification extrêmement faible n'est pas nécessairement un indicateur d'une bonne carte, car cela peut aussi signifier que la carte est sur-ajustée aux données. Elle doit être interprétée dans le contexte d'autres métriques et de l'objectif global de la SOM pour donner une image complète de sa performance.

4.1.4 Distortion

La distorsion [26] est une métrique utilisée pour évaluer l'efficacité de l'algorithme d'apprentissage SOM. Son rôle est similaire à celui du K-moyennes, mais elle prend en compte une contrainte topologique supplémentaire.

On définit le score S_2 :

$$S_2(X, E) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} \Theta_{i,j} d(X_i, W_j)$$

Avec,

$$\Theta_{i,j} = e^{-d^*(W_i^*, W_j)^2 / 4\sigma_2^0}$$

$$K = 4\sigma_0^2$$

La distorsion est calculée en sommant les distances euclidiennes au carré entre chaque échantillon de données et son neurone de carte associée. Ces distances sont ensuite pondérées par une fonction de voisinage, qui donne plus de poids aux neurones qui sont proches les uns des autres sur la carte. La fonction de voisinage garantit que la mise à jour des poids pendant l'apprentissage est faite de manière à respecter la topologie de la carte. Elle donne un aperçu de la qualité avec laquelle la carte représente les données d'origine, tout en prenant en compte la structure topologique des données. Minimiser cette distorsion assure que la carte offre une représentation fidèle et significative des données d'origine.

4.1.5 Erreur topographique

L'essence des SOM est de préserver, autant que possible, les relations de voisinage du jeu de données original. L'erreur topographique [54] mesure dans quelle mesure ce voisinage est respecté. Si deux échantillons sont proches dans l'espace d'entrée, leurs neurones correspondants sur la carte devraient également être proches ou voisins. Si ce n'est pas le cas, cela augmente l'erreur topographique.

On définit le score S_3 :

$$S_3(X, E) = \frac{1}{N} \sum_{i=1}^{N} \begin{cases} 1 & \text{si } d^*(W_i^*, W_i^{**}) > 1 \\ 0 & \text{sinon} \end{cases}$$

L'erreur topographique est calculée en analysant la proportion d'échantillons dont les deux meilleures neurones correspondants (le plus proche et le deuxième plus proche) ne sont pas adjacents sur la carte. Si, par exemple, 10% des échantillons ont leurs deux meilleurs neurones non adjacents, alors l'erreur topographique est de 10%. Le terme "lissage" fait référence à l'harmonie ou à la continuité de la projection. Une carte avec une

faible erreur topographique aurait une transition en douceur d'un neurone à l'autre, reflétant fidèlement les relations de proximité des données d'entrée. Une erreur topographique élevée peut indiquer que la carte ne capture pas correctement la structure sous-jacente des données. Cela pourrait être dû à plusieurs facteurs, tels qu'un mauvais choix de la taille ou de la topologie de la carte, un taux d'apprentissage inapproprié, ou peut-être que les données elles-mêmes ne sont pas bien adaptées à une projection bi-dimensionnelle.

4.1.6 Erreur combinée

L'erreur combinée [49] est une métrique qui cherche à intégrer deux aspects clés de la qualité d'une carte SOM, l'exactitude de la représentation des données (erreur de quantification) et la préservation de la topologie des données (erreur topographique). Elle offre ainsi une vision plus holistique de la performance des SOM. L'idée derrière cette erreur est de mesurer à quel point chaque échantillon de données est bien représenté non seulement par son neurone correspondant le plus proche, mais aussi par le suivant.

On définit le score S_4 :

$$S_4(X, E) = \frac{1}{N} \sum_{i=1}^{N} \left(d(X_i, W_i^*)^2 + \min_p \sum_{k=0}^{P-1} d(W_{p(k)}, W_{p(k+1)})^2 \right)$$

Avec p le chemin le plus court sur la carte de taille P du neurone $p(0) = W_i^*$ au neurone $p(P-1) = W_i^{**}$.

Pour chaque échantillon X_i, les deux meilleurs neurones correspondants, W_i^* et W_i^{**}, sont identifiées. Un chemin est tracé de W_i^* et W_i^{**} à travers la carte, ne passant que par des unités voisines. La distance le long de ce chemin est ensuite calculée comme une somme des distances entre X_i et chaque unité le long du chemin. Le principe derrière cette approche est de tenir compte non seulement de la "distance directe" entre l'échantillon et ses meilleurs neurones, mais également du "chemin" que l'on doit parcourir sur la carte pour passer de l'un à l'autre. Cette métrique est donc une combinaison d'erreurs de quantification et topographiques. Si l'erreur combinée est faible, cela signifie que la SOM a bien capturé à la fois la densité des données et la structure topologique des données.

4.1.7 Produit topographique

Le produit topographique [6] est une métrique destinée à évaluer la qualité d'une carte SOM. Son objectif est de mesurer la conservation des relations de voisinage entre l'espace

d'entrée X , où se trouvent les données d'origine et la carte elle-même.

On définit le score S_5 :

$$S_5(X,E) = \frac{1}{K(K-1)} \sum_{j=1}^{K} \sum_{k=1}^{K-1} log \left(\prod_{l=1}^{k} Q_1(j,l) Q_2(j,l) \right)^{\frac{1}{2k}}$$

Avec,

$$Q_1(j,l) = \frac{d(W_j, W_j^{d_l^*})}{d(W_j, W_j^{d_l})}$$

$$Q_2(j,l) = \frac{d^*(W_j, W_j^{d_l^*})}{d^*(W_j, W_j^{d_l})}$$

$$K = 4\sigma_0^2$$

Où $W_j^{d_l}$ est l-ième neurone le plus proche de W_j en fonction de la distance d et $W_j^{d_l^*}$ est l-ième neurone le plus proche de W_j en fonction de la distance d^*.

Le calcul du produit topographique commence par définir deux rapports basés sur la distance entre un neurone j et son k-ème voisin le plus proche, que ce soit sur la carte ou dans l'espace d'entrée. Ces rapports sont fusionnés pour créer une mesure équilibrée, visant à atténuer l'impact des variations locales. La valeur finale du produit topographique est obtenue en prenant le logarithme de cette mesure combinée et en effectuant une moyenne sur toutes les unités de la carte. Un produit topographique négatif suggère que la dimension de la carte est trop faible pour représenter correctement l'ensemble de données. Un produit topographique nul indique que la dimension de la carte est adéquate. Un produit topographique positif signifie que la dimension est trop élevée, entraînant des violations des relations de voisinage.

4.1.8 Erreur de Kruskal-Shepard

L'erreur de Kruskal-Shepard [63], introduite pour la mise à l'échelle multidimensionnelle, est une métrique conçue pour évaluer la conservation des distances par paire entre deux espaces différents. Cette erreur a été adoptée pour évaluer les cartes SOM dans certains travaux.

On définit le score S_6 :

$$S_6(X, E) = \frac{1}{N}||D^X - D^{SOM}||_F^2$$

Avec,

$$D_{i,j}^X = \frac{d(X_i, X_j)^2}{\max_{i',j'}(d(X_{i'}, X_{j'})^2)}$$

$$D_{i,j}^{SOM} = \frac{d^*(W_i^*, W_j^*)^2}{\max_{k,l}(d^*(k, l)^2)}$$

Où F est la norme de Frobenius.

$$||D||_F^2 = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} D_{ij}^2}$$

Pour calculer l'erreur de Kruskal-Shepard, on prend d'abord la norme de Frobenius carrée entre deux matrices de distance : celle des données d'origine et celle des unités sur la carte SOM. Ces matrices de distance quantifient les relations entre chaque paire de points, soit dans l'espace d'origine, soit sur la carte SOM. Pour obtenir une erreur normalisée, qui nous donne une meilleure idée de l'ampleur de l'erreur par rapport à la taille de l'ensemble de données, on divise l'erreur par N, le nombre de données. Une faible erreur de Kruskal-Shepard indique que la SOM a bien préservé les distances par paire entre les données lors de la projection. À l'inverse, une erreur élevée signifie que la structure des distances n'a pas été bien conservée, ce qui peut avoir des implications pour l'interprétabilité et la fiabilité de la carte SOM. L'erreur de Kruskal-Shepard offre un moyen précieux d'évaluer la qualité d'une projection, que ce soit pour la mise à l'échelle multidimensionnelle ou les SOM. Elle donne une indication quantitative de la fidélité avec laquelle les relations entre les données ont été conservées, ce qui est essentiel pour de nombreuses applications où la structure des données est primordiale.

4.1.9 C mesure

La C mesure [28] est une métrique conçue pour évaluer la conservation des relations de voisinage entre deux espaces, à l'instar de l'erreur de Kruskal-Shepard. Cependant, sa méthodologie diffère en ce qu'elle se concentre sur les produits élément par élément entre les distances par paire dans l'espace d'entrée et l'espace de sortie.

On définit le score S_7 :

$$S_7(X, E) = \sum_{i=1}^{N} \sum_{j<i} d(X_i, X_j) d^*(W_i^*, W_j^*)$$

La C mesure est calculée en prenant les produits élément par élément des distances par paire dans l'espace d'entrée et l'espace de sortie. Cette quantité est conçue pour être maximisée. Cela signifie qu'il est préférable que les distances, tant dans l'espace d'entrée que de sortie, prennent simultanément des valeurs élevées. Ainsi, si deux points sont éloignés dans l'espace d'entrée, ils devraient également être éloignés dans l'espace de sortie. Une valeur élevée pour la C mesure indique une bonne conservation des relations de voisinage entre les deux espaces. Cela signifie que la projection respecte largement les distances originales entre les points de données. Si la C mesure est faible, cela suggère que les relations de voisinage n'ont pas été correctement préservées lors de la projection. La C mesure fournit une perspective unique sur la qualité d'une projection en se concentrant sur les produits des distances. Elle offre une compréhension complémentaire de la manière dont les relations entre les données sont maintenues lorsqu'elles sont transférées d'un espace à l'autre. En combinant la mesure C avec d'autres métriques, comme l'erreur de Kruskal-Shepard, on peut obtenir une image plus complète de la performance d'une projection.

4.1.10 Fonction topographique

La fonction topographique [55] a été conçue pour surmonter certaines des limites associées au produit topographique, en particulier sa difficulté à distinguer les repliements naturels d'un ensemble de données non linéaire des repliements incorrects causés par des violations de voisinage. Contrairement au produit topographique qui utilise uniquement les vecteurs neurones, la fonction topographique prend également en compte les champs réceptifs de chaque unité. Un champ réceptif d'une unité est défini comme l'ensemble des points de données pour lesquels cette unité est l'unité correspondante la plus proche.

$$S_8 = \sum_{c=1}^{K} f_c(k)$$

Pour chaque unité c, son champ réceptif est défini par R_c, qui représente l'ensemble des points X_i tels que c est l'unité la plus proche de X_i. Pour chaque unité c et chaque entier k, on compte combien d'unités ont des champs réceptifs adjacents et une distance à c sur la carte supérieure à k. La fonction topographique est définie en sommant ces distances sur toutes les unités de la carte. Une étape de normalisation est effectuée pour permettre

des comparaisons entre des cartes de tailles différentes. On remplace k par $\frac{k}{\max\limits_{E}(d^*)}$ et on divise par $K(K-3^p)$, où p est généralement égal à 2 car la carte SOM est typiquement une carte bi-dimensionnelle. La fonction topographique offre une perspective plus raffinée et précise de la qualité topographique d'une SOM par rapport au produit topographique. En prenant en compte les champs réceptifs des unités et en incorporant une mécanique de normalisation, elle fournit une évaluation robuste de la manière dont la SOM préserve les relations topologiques intrinsèques des données d'origine.

4.1.11 Normalisation

Afin de synthétiser les différentes mesures employées, le tableau 4.1 ci-dessous présente l'ensemble des scores utilisés pour évaluer les cartes auto-organisatrices.

Score	**Formule**
S_1	$S_1(X,E) = \frac{1}{N}\sum_{i=1}^{N} d(W_i^*, X_i)$ [26]
S_2	$S_2(X,E) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} \Theta_{i,j} d(X_i, W_j)$ [26]
S_3	$S_3(X,E) = \frac{1}{N}\sum_{i=1}^{N} \begin{cases} 1 & \text{si } d^*(W_i^*, W_i^{**}) > 1 \\ 0 & \text{sinon} \end{cases}$ [54]
S_4	$S_4(X,E) = \frac{1}{N}\sum_{i=1}^{N} \left(d(X_i, W_i^*)^2 + \min_p \sum_{k=0}^{P-1} d(W_{p(k)}, W_{p(k+1)})^2 \right)$ [49]
S_5	$S_5(X,E) = \frac{1}{K(K-1)}\sum_{j=1}^{K}\sum_{k=1}^{K-1} \log\left(\prod_{l=1}^{k} Q_1(j,l)Q_2(j,l)\right)^{\frac{1}{2k}}$ [6]
S_6	$S_6(X,E) = \frac{1}{N}\|\|D^X - D^{SOM}\|\|_F^2$ [63]
S_7	$S_7(X,E) = \sum_{i=1}^{N}\sum_{j<i} d(X_i, X_j) d^*(W_i^*, W_j^*)$ [28]
S_8	$S_8(X,E) = \sum_{c=1}^{K} f_c(k)$ [55]

TABLE 4.1 – Récapitulatif des scores utilisés pour évaluer les SOM.

Par la suite, tous les scores obtenus à partir des différentes métriques évaluant la qualité des projections dans le contexte des SOM sont normalisés afin de se situer dans l'intervalle $[0, 1]$. Cette normalisation garantit une comparabilité directe entre les scores, facilitant ainsi leur interprétation et leur analyse. Un score de 1 représente le résultat optimal, indiquant que la projection respecte parfaitement les critères évalués par la métrique correspondante, tandis qu'un score de 0 traduit une performance minimale ou une absence totale de conservation des propriétés considérées.

4.1.12 Autre approche : analyse du rendu visuel

Une autre façon d'appprocher cette notion de score pour l'algorithme SOM est d'utiliser un critère visuel. En effet, il est possible de savoir empiriquement si le visuel d'une carte de Kohonen est pertinent ou non. Ainsi, on peut imaginer un model que l'on pourrait utiliser pour déterminer un score à partir des cartes de Kohonen obtenues. Ceci impose la création manuelle d'une base de données de différentes cartes auxquelles on a associé un score.

Une dernière approche peut être de cumuler plusieurs des critères vus ci-dessus au sein d'une même fonction de score.

4.2 Qualité de l'apprentissage

Dans le processus d'implémentation et d'optimisation des cartes auto-organisatrices (SOM), deux paramètres jouent un rôle essentiel dans la détermination de la qualité et de l'efficacité de l'apprentissage : le nombre maximal d'itérations (T_{max}) et le taux d'apprentissage initial (ϵ_0). Le paramètre T_{max} définit la durée du processus d'apprentissage, influençant directement la capacité du modèle à capturer et à organiser les structures inhérentes aux données. D'autre part, ϵ_0, qui détermine l'intensité de l'ajustement des poids des neurones lors des premières itérations, est crucial pour établir l'équilibre entre une convergence rapide et une adaptation précise au jeu de données.

La combinaison appropriée de T_{max} et ϵ_0 est donc primordiale pour atteindre une performance optimale des SOM. Un nombre insuffisant d'itérations peut empêcher le modèle d'atteindre sa convergence, tandis qu'un nombre excessif peut conduire à un surajustement ou à une perte de temps de calcul inutile. De même, un taux d'apprentissage initial trop élevé peut causer une convergence prématurée vers des solutions sous-optimales, tandis qu'un taux trop faible peut ralentir inutilement le processus d'apprentissage.

Nous explorerons l'impact de ces deux paramètres sur les performances des SOM, en examinant comment leur ajustement peut améliorer la représentation des données et l'efficacité de l'algorithme. Cette analyse vise à fournir des directives pour l'optimisation de T_{max} et ϵ_0, facilitant ainsi la mise en œuvre des SOM dans divers contextes d'analyse de données.

4.2.1 Analyse expérimentale

Dans cette section, nous présentons une série de graphiques visant à analyser l'interaction entre deux paramètres essentiels des cartes auto-organisatrices (SOM) : le **taux d'apprentissage initial** (ϵ_0) et le **nombre maximal d'itérations** (T_{max}). L'objectif est de comprendre comment ces paramètres, pris isolément et conjointement, influencent la performance globale et la stabilité des SOM selon différentes configurations.

Paramètres étudiés Les expérimentations portent sur l'analyse combinée de :

- le **nombre maximal d'itérations** T_{max}, qui contrôle la durée de l'apprentissage ;
- le **taux d'apprentissage initial** ϵ_0, qui influence la rapidité et l'amplitude des ajustements de poids en début d'entraînement ;
- la **topologie de la carte**, avec trois formes testées : carrée, hexagonale et triangulaire ;
- la **taille de la carte**, variant de 5×5 à 20×20 neurones.

Conditions expérimentales Les données utilisées proviennent des bases internes de l'environnement commercial d'Octave. Elles incluent :

- des **jeux de données synthétiques**, conçus pour tester la robustesse du modèle face à des structures connues ;
- des **jeux de données réels**, issus d'applications commerciales et présentant une forte hétérogénéité de dimensions et de distributions.

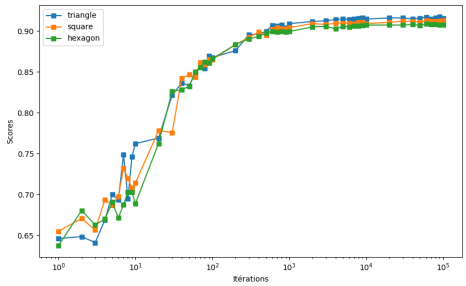
Chaque configuration de SOM (combinaison de topologie, taille, T_{max} et ϵ_0) a été évaluée sur plusieurs instances indépendantes afin d'assurer la reproductibilité et la robustesse des résultats obtenus.

Métriques d'évaluation La performance de chaque modèle est mesurée à l'aide de trois indicateurs :

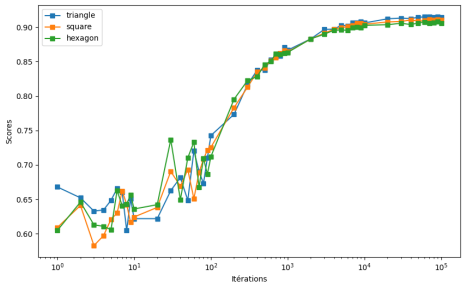
- **l'erreur de quantification moyenne**, mesurant la fidélité de la représentation des données ;
- **l'erreur topographique**, évaluant la capacité de la carte à préserver les relations de voisinage ;
- **la capacité de préservation des relations structurelles globales**, analysant la cohérence spatiale entre clusters voisins.

L'objectif principal de cette étude est d'identifier les combinaisons de T_{max} et ϵ_0 qui permettent aux SOM d'atteindre une convergence rapide tout en maintenant une organisation optimale des données sur la carte, pour chaque topologie et taille. Les résultats obtenus fournissent des directives utiles pour paramétrer les SOM dans des contextes variés.

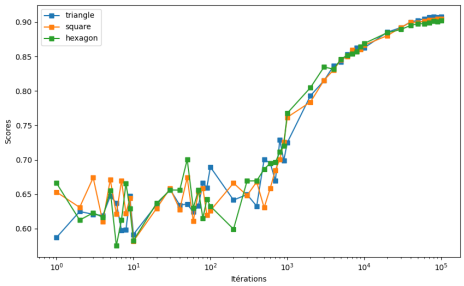
En examinant ces graphiques, nous chercherons à identifier des tendances claires et à déduire des recommandations pratiques pour le réglage de T_max et ϵ_0 en fonction de la taille et de la topologie de la carte, dans le but d'atteindre une performance optimale.



(a) $\epsilon_0 = 0.1$



(b) $\epsilon_0 = 0.01$



(c) $\epsilon_0 = 0.001$

FIGURE 4.1 – Analyse de l'impact de $T_{\max}$ et ϵ_0 sur des cartes SOM de tailles 5x5 et différentes topologies avec le score S'_1.

Pour alléger la lecture, seules les courbes les plus représentatives sont montrées ici ; l'ensemble des résultats complémentaires (tailles 10×10 et 20×20, et déclinaisons par topologie) est reporté en annexe A.2.1.

Analyse : Les courbes d'apprentissage obtenues pour les différentes configurations des SOM révèlent des tendances intéressantes concernant l'impact du taux d'apprentissage initial (ϵ_0) sur le processus de convergence du modèle. Plus précisément, nous observons que lorsque ϵ_0 est réglé à une valeur plus basse, le modèle nécessite un nombre plus élevé d'itérations (T_{max}) pour atteindre la convergence. Ce phénomène est conforme à l'attente théorique : un taux d'apprentissage plus faible signifie que les ajustements apportés aux poids des neurones à chaque itération sont plus modérés, conduisant ainsi à une progression plus graduelle vers la convergence.

Cependant, un aspect crucial mis en évidence par nos résultats est que cette convergence plus lente n'entraîne pas nécessairement une amélioration proportionnelle des scores, tels que l'erreur de quantification. En d'autres termes, le bénéfice en termes de performance obtenu en prolongeant le processus d'apprentissage avec un taux d'apprentissage plus faible n'est pas aussi significatif qu'on pourrait le supposer. Cela soulève une question importante sur l'efficacité de l'utilisation de faibles valeurs d'ϵ_0, en particulier dans des contextes où les ressources de calcul sont limitées ou lorsque le temps de convergence est un facteur critique.

Ces observations suggèrent qu'il est essentiel de trouver un équilibre entre le taux d'apprentissage initial et le nombre d'itérations pour optimiser les performances des SOM. Un taux d'apprentissage trop bas, bien que potentiellement bénéfique pour une exploration minutieuse de l'espace des données, peut ne pas justifier le temps et les ressources computationnelles supplémentaires requis pour une amélioration marginale des scores. À l'inverse, un taux d'apprentissage plus élevé peut conduire à une convergence plus rapide, mais avec le risque d'une adaptation moins précise aux caractéristiques des données.

En conclusion, ces résultats mettent en lumière l'importance d'une approche équilibrée dans le réglage des paramètres des SOM, où les avantages d'une convergence plus lente doivent être soigneusement pesés contre les gains potentiels en termes de performance et d'efficacité.

4.2.2 Extension aux SOM en 3 dimensions

Après avoir analysé en détail les courbes d'apprentissage des cartes auto-organisatrices bi-dimensionnelles (SOM 2D), nous étendons notre étude aux configurations tri-dimensionnelles (SOM 3D). Cette extension vers une dimension supplémentaire offre une perspective enrichie sur la dynamique d'apprentissage et les performances des SOM, tout en présentant des défis uniques en termes de réglage des paramètres et d'interprétation des résultats.

Dans cette section, nous présentons une série de courbes d'apprentissage pour les SOM 3D, qui mettent en lumière l'évolution des performances en fonction de paramètres clés tels que le taux d'apprentissage initial (ϵ_0) et le nombre maximal d'itérations (T_{max}). Ces courbes sont comparées à travers différentes tailles de cartes et topologies, offrant ainsi un aperçu complet de l'impact de ces paramètres dans un cadre tridimensionnel.

L'objectif est de comprendre comment les spécificités des cartes 3D, notamment leur capacité à modéliser des relations plus complexes, influencent la dynamique d'apprentissage et les critères de performance. En observant ces courbes, nous cherchons à déterminer si les tendances observées dans les SOM 2D persistent dans les configurations 3D et comment les spécificités des cartes tri-dimensionnelles modifient l'interaction entre les paramètres d'apprentissage et les résultats finaux.

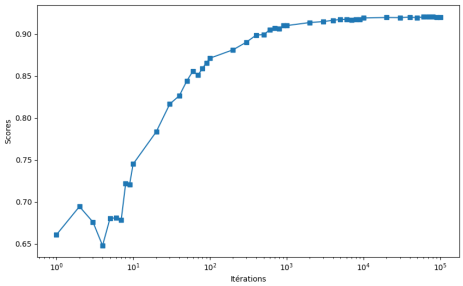
Ces analyses fourniront des informations précieuses pour optimiser les SOM 3D, en aidant à identifier les paramètres les plus efficaces pour diverses applications et en soulignant les considérations uniques à prendre en compte lors de l'utilisation de cartes tridimensionnelles.

Pour alléger la lecture, seules les courbes les plus représentatives sont montrées ici ; l'ensemble des résultats complémentaires (tailles 5x5x5 et 8x8x8, et déclinaisons par topologie) est reporté en annexe A.2.1.

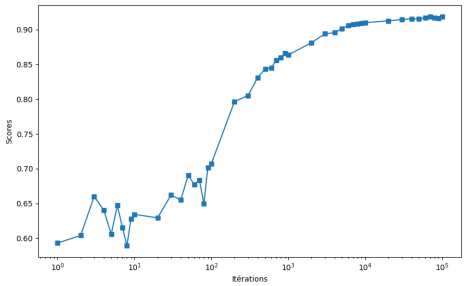
Nos observations indiquent que les courbes d'apprentissage des cartes auto-organisatrices tri-dimensionnelles (SOM 3D) présentent des tendances similaires à celles observées dans les configurations bi-dimensionnelles (SOM 2D). Cette constatation est particulièrement intéressante car elle suggère que, malgré l'augmentation de la complexité spatiale dans les SOM 3D, les dynamiques fondamentales régissant le processus d'apprentissage et l'optimisation des paramètres tels que le taux d'apprentissage initial (ϵ_0) et le nombre maximal d'itérations (T_{max}) restent cohérentes.

Cette similarité implique que les principes généraux d'optimisation des paramètres des SOM sont largement applicables, indépendamment de la dimensionnalité de la carte. Elle suggère également que les techniques d'ajustement de paramètres développées pour les SOM 2D peuvent être efficacement transposées aux configurations 3D. Comme pour les SOM 2D, nous constatons que des taux d'apprentissage plus faibles conduisent à une convergence plus lente dans les SOM 3D, sans une amélioration significative des scores. De plus, cette convergence plus lente s'accompagne d'une augmentation du temps de calcul, un aspect d'autant plus critique dans les SOM 3D en raison de leur complexité computationnelle accrue. Bien que les SOM 3D offrent une capacité améliorée à modéliser

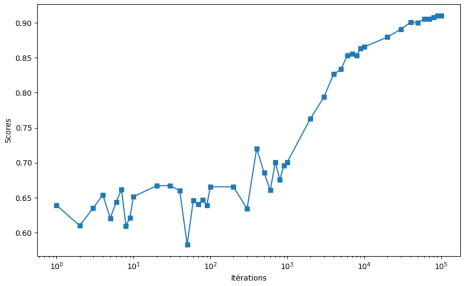
des relations complexes, la décision d'utiliser des configurations tri-dimensionnelles devrait être guidée par des considérations spécifiques au problème, telles que la nécessité de capturer des relations spatiales ou des structures inhérentes qui ne sont pas clairement discernables dans des configurations bi-dimensionnelles.



(a) $\epsilon_0 = 0.1$



(b) $\epsilon_0 = 0.01$



(c) $\epsilon_0 = 0.001$

FIGURE 4.2 – Analyse de l'impact de $T_{\max}$ et ϵ_0 sur des cartes SOM de tailles 3x3x3 et différentes topologies avec le score S'_1.

En conclusion, les résultats des SOM 3D, bien que présentant des similarités avec ceux des SOM 2D, doivent être interprétés en tenant compte de la spécificité de leur application et du compromis entre la complexité spatiale et la performance computationnelle. L'analyse comparative entre les SOM 2D et 3D souligne l'importance d'une évaluation approfondie des besoins et des contraintes du problème avant de choisir la configuration optimale pour une application donnée.

4.2.3 Mise en oeuvre d'un score combiné

Pour évaluer de manière plus holistique la performance des cartes auto-organisatrices (SOM), nous avons introduit un score combiné qui prend en compte à la fois l'erreur de quantification (représentée par S_1') et le temps de calcul. Ce score vise à fournir un indicateur global de l'efficacité, reflétant non seulement la précision du modèle, mais aussi son efficacité computationnelle.

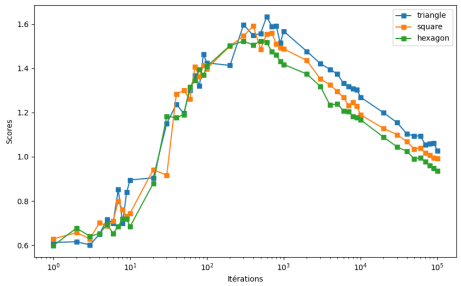
La nécessité de ce score combiné découle de la compréhension que l'optimisation des SOM ne doit pas se concentrer exclusivement sur la minimisation de l'erreur de quantification, mais aussi sur la considération du coût computationnel. En utilisant ce score combiné pour évaluer les performances des SOM, nous pouvons examiner comment différents réglages de paramètres (comme (ϵ_0) et T_{max}) influencent non seulement la précision du modèle, mais aussi son efficacité. Cela nous permet de trouver des configurations qui offrent un bon équilibre entre précision et rapidité.

On définit le score $S_{\Delta t}(S_1')$:

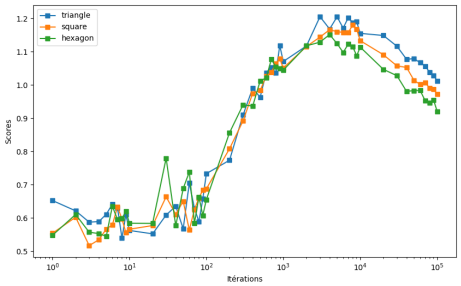
$$S_{\Delta t}(S_1') = \frac{1}{(1 - S_1')(ln(\Delta t))}$$

Avec Δt le temps de calcul, comprenant à la fois le temps d'initialisation de la carte et le temps d'apprentissage.

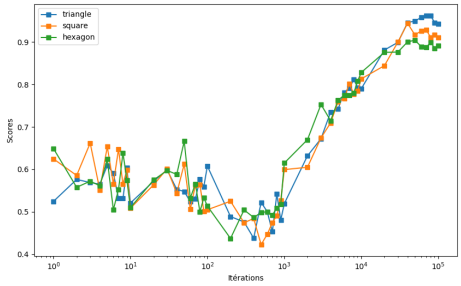
Dans l'exemple suivant, nous reprenons l'expérimentation précédente en utilisant le score $S_{\Delta t}(S_1')$.



(a) $\epsilon_0 = 0.1$



(b) $\epsilon_0 = 0.01$



(c) $\epsilon_0 = 0.001$

FIGURE 4.3 – Analyse de l'impact de $S_{\Delta t}(S'_1)$ sur des cartes SOM de taille 5x5 et différentes topologies avec le score $S_{\Delta t}(S'_1)$.

Pour alléger la lecture, seules les courbes les plus représentatives sont montrées ici ; l'ensemble des résultats complémentaires (tailles 10×10 et 20×20, et déclinaisons par topologie) est reporté en annexe A.2.2.

Analyse : Les courbes d'apprentissage obtenues pour le score combiné, intégrant à la fois l'erreur de quantification (S'_1) et le temps de calcul, révèlent un point de maximisation coïncide avec le moment où l'augmentation du score S'_1 n'est plus significative, ce qui indique un équilibre optimal entre précision et efficacité computationnelle.

Cette observation suggère que l'objectif ne devrait pas être de maximiser à tout prix le score S'_1, mais plutôt de chercher le point où un équilibre est atteint entre la qualité de la carte et l'efficacité du temps de calcul. Cela aide à éviter un usage inutilement intensif des ressources computationnelles tout en conservant une qualité satisfaisante de la carte SOM.

En pratique, cette analyse peut guider le réglage des paramètres tels que le taux d'apprentissage initial (ϵ_0) et le nombre maximal d'itérations (T_{max}). Les paramètres devraient être ajustés pour atteindre ce point de maximisation, assurant ainsi une utilisation optimale des ressources tout en maintenant des performances adéquates.

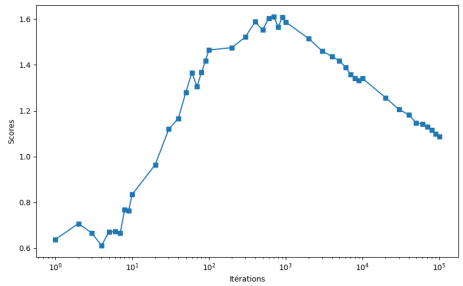
La corrélation entre le point de maximisation du score combiné et la stabilisation de l'augmentation de (S_1') valide l'approche de l'utilisation d'un score combiné. Ce résultat démontre l'efficacité de cette méthode pour trouver un compromis entre l'exactitude des résultats et la viabilité computationnelle, un équilibre essentiel dans de nombreuses applications pratiques des SOM.

Sur la base de nos analyses, le score combiné $S_{\Delta t}(S_1')$, intégrant l'erreur de quantification (S_1') et le temps de calcul, se révèle être un excellent candidat pour un critère d'arrêt dans l'entraînement des SOM. Cette approche offre une méthode dynamique et efficace pour déterminer le moment le plus opportun pour conclure l'apprentissage, en optimisant l'équilibre entre la qualité de la représentation des données et l'efficience computationnelle.

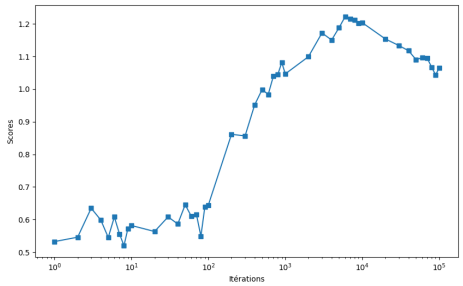
Après avoir développé et discuté le concept d'un score combiné pour évaluer les performances des cartes auto-organisatrices (SOM), nous avons procédé à l'application de ce score aux configurations tridimensionnelles des SOM. Ces tests représentent une étape importante pour comprendre comment le score combiné, intégrant à la fois l'erreur de quantification (S_1') et le temps de calcul, fonctionne dans un contexte plus complexe et spatial.

Nous examinerons comment le score évolue en fonction de variations des paramètres clés tels que le taux d'apprentissage initial ((ϵ_0)) et le nombre maximal d'itérations (T_{max}), ainsi que l'impact de la taille de la carte sur les performances globales. Ces résultats fourniront des connaissances précieux sur l'applicabilité du score combiné dans des environnements de modélisation plus complexes et sur son potentiel en tant qu'outil d'optimisation pour les SOM 3D.

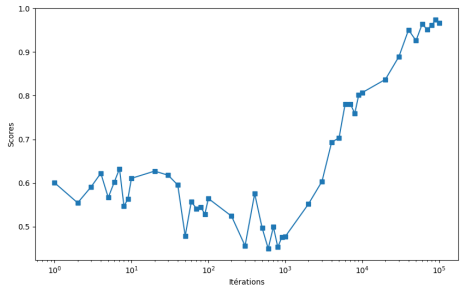
Dans l'exemple suivant, nous reprenons l'expérimentation précédente en y appliquant le score $S_{\Delta t}(S_1')$.



(a) $\epsilon_0 = 0.1$



(b) $\epsilon_0 = 0.01$



(c) $\epsilon_0 = 0.001$

FIGURE 4.4 – Analyse de l'impact de $S_{\Delta t}(S'_1)$ sur des cartes SOM de taille 3x3x3 et différentes topologies avec le score $S_{\Delta t}(S'_1)$.

Pour alléger la lecture, seules les courbes les plus représentatives sont montrées ici ; l'ensemble des résultats complémentaires (tailles 5x5x5 et 8x8x8, et déclinaisons par topologie) est reporté en annexe A.2.1.

Les résultats obtenus en appliquant le score combiné aux cartes auto-organisatrices tridimensionnelles (SOM 3D) s'avèrent remarquablement similaires à ceux observés pour les configurations bidimensionnelles (SOM 2D). Cette similarité indique que les principes fondamentaux régissant l'efficacité et la précision des SOM, capturés par notre score combiné, restent cohérents malgré l'augmentation de la complexité spatiale. Cette constatation renforce l'utilité du score combiné comme outil d'évaluation et d'optimisation, applicable de manière transversale à différentes dimensions des SOM.

4.2.4 Étude du critère d'arrêt

Le score combiné $S_{\Delta t}(S'_1)$ permet d'évaluer simultanément la précision et l'efficacité computationnelle d'une carte SOM. Il constitue donc un bon indicateur pour définir un critère d'arrêt automatique. Cependant, son calcul complet à chaque itération est coûteux, car il nécessite d'évaluer l'erreur de quantification sur l'ensemble des données, tout en prenant en compte le temps de calcul cumulé.

Dans les configurations complexes — notamment pour les SOM tridimensionnelles — ce coût devient rapidement prohibitif : chaque évaluation du score implique un balayage complet de la carte et de l'ensemble des points d'entrée, multiplié par plusieurs milliers d'itérations. Ainsi, un calcul trop fréquent du score combiné risquerait d'annuler les gains de performance qu'il est censé mesurer.

Pour rendre ce critère d'arrêt exploitable en pratique, il est possible d'adopter une approche **échantillonnée et périodique**. Plutôt que de recalculer le score à chaque itération, on ne l'évalue qu'à intervalles réguliers (par exemple toutes les 100 ou 500 itérations), en utilisant un sous-ensemble des données d'entrée. Cette approximation permet de suivre la tendance générale de l'apprentissage sans en supporter le coût complet.

Le critère d'arrêt peut alors être formulé ainsi : l'apprentissage est interrompu dès que la variation relative du score combiné entre deux évaluations successives devient inférieure à un seuil η pendant un certain nombre de cycles consécutifs (patience P). Cette approche permet d'identifier automatiquement le moment où la carte cesse de s'améliorer de manière significative, tout en limitant la charge de calcul.

En résumé, le score combiné $S_{\Delta t}(S'_1)$ peut servir de fondement à un critère d'arrêt efficace et robuste, à condition d'en adapter le mode d'évaluation. L'utilisation d'une estimation parcimonieuse, à fréquence contrôlée, permet d'équilibrer la précision de la mesure et le coût de calcul, assurant ainsi une convergence rapide sans dégradation notable de la qualité de la carte.

4.3 Paramétrage automatique de SOM via SMAC

SMAC (Configuration d'Algorithme basée sur des Modèles Séquentiels) [65] a été conçu pour permettre aux utilisateurs d'ajuster et d'affiner leurs algorithmes via des configurations d'hyperparamètres efficaces. Basé sur l'optimisation bayésienne et les techniques de Racing, l'outil SMAC peut rapidement converger vers des solutions optimales, ou du moins très performantes, dans un espace d'hyperparamètres souvent vaste et complexe. SMAC

permet de combiner exploration et exploitation efficace de l'espace des hyperparamètres.

Algorithm 2 Sequential Model-based Algorithm Configuration (SMAC)

Initialisation
$\mathcal{X} \leftarrow$ espace des configurations
$f(x) \leftarrow$ fonction d'évaluation
$M \leftarrow$ modèle prédictif (par exemple, forêt aléatoire)
$a(x) \leftarrow$ fonction d'acquisition (par exemple, Expected Improvement)
$X_0 \leftarrow$ ensemble initial de configurations, généré aléatoirement
$Y_0 \leftarrow \{f(x) : x \in X_0\}$
Initialiser le modèle M avec X_0 et Y_0
while critère d'arrêt non satisfait (par exemple, nombre d'itérations ou limite de temps) **do**
    Mettre à jour le modèle M avec les observations X et Y
    Trouver la prochaine configuration $x_{\text{next}} \leftarrow \arg\max_{x \in \mathcal{X}} a(x)$
    Évaluer $f(x_{\text{next}})$
    Ajouter x_{next} et $f(x_{\text{next}})$ à X et Y
end while
$x^* \leftarrow \arg\min_{x \in X} f(x)$
Retourner x^* comme meilleure configuration

L'utilisation de SMAC pour optimiser les paramètres d'un SOM est une démarche pragmatique. Les cartes auto-organisatrices sont des réseaux de neurones non supervisés qui peuvent avoir une grande variabilité en termes de performances en fonction de la configuration de leurs paramètres. Il est donc essentiel d'avoir une stratégie robuste pour trouver le vecteur de paramètres optimal.

La première étape consiste à définir le vecteur de paramètres. Il s'agit d'identifier tous les hyperparamètres pertinents pour les cartes SOM spécifique que l'on souhaite optimiser avec SMAC. Cela peut inclure, par exemple, le taux d'apprentissage, la taille de la carte, le type de voisinage ou toute autre caractéristique spécifique à l'implémentation.

La seconde est l'élaboration d'une fonction d'évaluation. C'est sans doute l'étape la plus cruciale. Pour que SMAC puisse fonctionner efficacement, il faut disposer d'une méthode qui évalue rapidement et précisément la performance du SOM pour un vecteur de paramètres donné. Cette fonction d'évaluation sera basée sur les scores vus précédemment.

Il est crucial que cette fonction d'évaluation soit à la fois représentative de la performance réelle de SOM et suffisamment rapide pour permettre de nombreuses évaluations en un temps raisonnable. Une fois que l'on a cette fonction, SMAC peut la prendre comme entrée et commencer le processus d'optimisation, en évaluant de manière itérative différentes configurations d'hyperparamètres jusqu'à ce que le vecteur de paramètres optimal soit identifié.

4.3.1 Expérimentation

Dans les exemples qui vont suivre, nous allons nous limiter de façon arbitraire à 10 itérations pour un ensemble défini de paramètres. Ce nombre est suffisamment grand pour pouvoir mesurer l'écart-type au sein de chaque expérimentation. Initialement, nous nous concentrerons uniquement sur le score S'_1, représentant l'erreur de quantification, car il s'agit de la métrique la plus intuitive en rapport avec l'algorithme SOM. Notre but est de mesurer et de comparer les performances obtenues en utilisant un vecteur de paramètres optimisé par SMAC face à des paramètres choisis aléatoirement ou uniformément. Nous examinerons également comment ces performances varient en fonction de différents facteurs, tels que la topologie des cartes, leur dimension, le nombre d'itérations et le taux d'apprentissage. Ainsi, nous identifierons trois scénarios distincts :

— **Cas SMAC** : utilisation du vecteur de paramètres optimisé automatiquement par *SMAC* (Sequential Model-based Algorithm Configuration).
— **Cas Uniform** : utilisation d'un vecteur où tous les poids et hyperparamètres sont fixés à une valeur constante (ici 1), représentant une configuration neutre.
— **Cas Random** : utilisation d'un vecteur de paramètres généré aléatoirement dans les bornes admissibles, afin de simuler une configuration non optimisée.

Les résultats obtenus à partir des expérimentations avec les cartes auto-organisées bidimensionnelles révèlent plusieurs tendances importantes en fonction de la topologie, de la taille de la carte, du nombre maximum d'itérations (T_{max}), du taux d'apprentissage initial (ϵ_0), et des différents cas de paramétrisation (SMAC, Uniform, Random).

Score

Topologie	Taille	T_{max}	ϵ_0	μ_{SMAC}	$\mu_{Uniform}$	μ_{Random}	σ_{SMAC}	$\sigma_{Uniform}$	σ_{Random}
triangle	5x5	1000	0.1	0.9731	0.9376	0.9414	0.0018	0.0018	0.0063
triangle	5x5	50000	0.01	0.9794	0.9431	0.9463	0.0007	0.0006	0.0065
triangle	10x10	1000	0.1	0.9777	0.9416	0.9453	0.0008	0.0014	0.0066
triangle	10x10	50000	0.01	0.9835	0.9496	0.9556	0.0006	0.0007	0.0068
triangle	20x20	1000	0.1	0.9782	0.9438	0.9486	0.0009	0.0014	0.0064
triangle	20x20	50000	0.01	0.9849	0.9517	0.9560	0.0004	0.0006	0.0049
carré	5x5	1000	0.1	0.9659	0.9368	0.9402	0.0014	0.0014	0.0046
carré	5x5	50000	0.01	0.9719	0.9407	0.9446	0.0003	0.0004	0.0048
carré	10x10	1000	0.1	0.9694	0.9397	0.9429	0.0022	0.0011	0.0068
carré	10x10	50000	0.01	0.9754	0.9458	0.9489	0.0012	0.0006	0.0053
carré	20x20	1000	0.1	0.9715	0.9427	0.9468	0.0019	0.0014	0.0078
carré	20x20	50000	0.01	0.9768	0.9476	0.9556	0.0009	0.0006	0.0050
hexagone	5x5	1000	0.1	0.9643	0.9347	0.9389	0.0007	0.0014	0.0056
hexagone	5x5	50000	0.01	0.9690	0.9391	0.9460	0.0010	0.0003	0.0063
hexagone	10x10	1000	0.1	0.9671	0.9375	0.9401	0.0018	0.0011	0.0070
hexagone	10x10	50000	0.01	0.9716	0.9424	0.9472	0.0014	0.0007	0.0064
hexagone	20x20	1000	0.1	0.9679	0.9390	0.9437	0.0018	0.0013	0.0043
hexagone	20x20	50000	0.01	0.9732	0.9437	0.9506	0.0008	0.0005	0.0066

Analyse : Premièrement, il est observable que l'augmentation de la taille des cartes (de 5x5 à 20x20) améliore généralement les scores d'erreur de quantification (S_1') dans tous les cas testés. Cela suggère que des cartes plus grandes permettent une meilleure représentation et distinction des données, bien que cela puisse également entraîner une augmentation des ressources computationnelles requises.

Deuxièmement, l'augmentation du nombre d'itérations (de 1000 à 50000) semble également améliorer les performances, indiquant que permettre à l'algorithme de converger davantage peut conduire à des cartes mieux organisées et à une erreur de quantification réduite.

Enfin, en comparant les différentes topologies (triangle, carré, hexagone), nous pouvons constater que les différences de performance ne sont pas significativement élevées, bien que des variations mineures existent. Ceci suggère que le choix de la topologie peut être guidé par des considérations spécifiques à l'application plutôt que par des différences marquées en termes de performance.

Ces observations mettent en lumière l'importance de l'hyperparamétrage dans l'optimisation des SOM 2D, en montrant comment les variations de taille, d'itérations et de topologie peuvent influencer les résultats obtenus.

Les écarts types révèlent la consistance des résultats à travers différentes exécutions. Des valeurs faibles indiquent une forte stabilité, c'est-à-dire que les performances sont relativement insensibles aux variations dans l'initialisation ou aux fluctuations mineures durant le processus d'apprentissage. En revanche, des valeurs plus élevées suggèrent une plus grande variabilité, ce qui peut indiquer une sensibilité aux conditions initiales ou une dépendance plus marquée aux spécificités du jeu de données.

Dans nos résultats, nous observons généralement des écarts types faibles pour les Cas SMAC et Uniform, indiquant une stabilité raisonnable dans ces configurations. Cependant, pour le Cas Random, l'écart type est systématiquement plus élevé, ce qui suggère une variabilité accrue dans les performances. Ceci est dû à la nature des paramètres utilisés dans ce cas, qui sont aléatoires.

Les résultats obtenus dans le Cas SMAC démontrent l'efficacité de l'optimisation des paramètres. Nous observons que dans plusieurs configurations, en particulier celles avec un nombre élevé d'itérations (T_{max}) et des tailles de cartes plus grandes, le score d'erreur de quantification (S_1') est nettement amélioré par rapport aux autres cas. Cela indique que l'optimisation des paramètres peut conduire à une meilleure organisation de la carte et à une représentation plus précise des données.

En outre, les faibles écarts types associés à ces résultats optimisés témoignent de leur stabilité et de leur fiabilité. Cette cohérence est particulièrement importante, car elle suggère que le vecteur de paramètres optimisé est non seulement performant en moyenne, mais également robuste face aux variations dans les conditions initiales ou les spécificités des jeux de données.

Ces résultats soulignent l'importance de l'hyperparamétrage dans la mise en œuvre des SOM et démontrent que l'utilisation de techniques d'optimisation avancées peut significativement améliorer les performances de l'algorithme.

Bien que l'optimisation des paramètres des cartes auto-organisées bi-dimensionnelles basée sur le score (S_1') ait montré des améliorations notables en termes de performance, il est important de reconnaître les limites d'une telle approche. L'hyperparamétrage axé uniquement sur l'erreur de quantification, représentée par l'erreur de quantification (S_1'), peut entraîner des résultats potentiellement trompeurs.

L'une des préoccupations majeures est que la maximisation de (S_1') pourrait se faire

au détriment d'autres critères d'évaluation importants. Par exemple, la topologie des données, la conservation des relations entre les caractéristiques, et d'autres aspects qualitatifs pourraient être négligés. Cela signifie qu'en poursuivant aveuglément l'amélioration de (S_1'), on pourrait obtenir une carte qui, bien que présentant une très faible erreur de quantification, ne représente pas fidèlement les structures et les relations intrinsèques des données.

En outre, cette approche pourrait conduire à un surajustement, où la carte SOM devient excessivement adaptée aux spécificités du jeu de données utilisé pour l'optimisation, en particulier lorsque la taille de la carte augmente, réduisant ainsi sa généralisabilité et son applicabilité à de nouveaux ensembles de données. Il est donc crucial d'équilibrer l'optimisation de (S_1') avec la considération d'autres mesures de performance pour assurer une représentation globale et équilibrée des données.

Cette prise de conscience souligne l'importance d'une approche holistique dans l'hyperparamétrage des SOM, où plusieurs critères de performance doivent être pris en compte pour obtenir des résultats véritablement significatifs et fiables.

4.4 SOM 3D

Dans cette section, nous étendons notre exploration de l'hyperparamétrage aux cartes auto-organisées tridimensionnelles (SOM-3D). Tout comme pour les cartes bi-dimensionnelles, notre objectif est d'évaluer et de comparer les performances en utilisant différentes configurations de paramètres. Cependant, étant donné la complexité accrue des cartes tridimensionnelles, nous accorderons une attention particulière aux défis uniques qu'elles présentent.

Nous limiterons nos expérimentations à 10 itérations pour un ensemble défini de paramètres, tout en mettant l'accent sur le score S_1', un indicateur adapté pour évaluer l'erreur de quantification dans le contexte tridimensionnel. Ce score nous aidera à comprendre comment l'augmentation de la dimensionnalité affecte la performance de l'algorithme SOM. Les paramètres examinés incluront, mais ne seront pas limités à, la topologie des cartes 3D, leur dimension, le nombre d'itérations et le taux d'apprentissage.

Nous envisageons trois scénarios pour nos tests sur les cartes 3D-SOM : le premier implique l'utilisation de paramètres optimisés par SMAC (cas SMAC), le deuxième utilise des paramètres uniformes (cas Uniform), et le troisième se base sur des paramètres choisis aléatoirement (cas Random). L'analyse de ces scénarios nous permettra de dégager des

insights sur les spécificités de l'hyperparamétrage dans les cartes tridimensionnelles et sur la manière dont ces paramètres influencent les performances globales du modèle.

Dans le cadre de nos expérimentations sur les cartes auto-organisées tridimensionnelles, une attention particulière est accordée aux dimensions des cartes. Alors que pour les cartes bidimensionnelles, les tailles variaient de 5x5, 10x10 à 20x20, dans le cas des cartes tridimensionnelles, nous adaptons les dimensions pour refléter la complexité additionnelle. Ainsi, les tailles des cartes SOM-3D examinées dans nos tests sont de 3x3x3, 5x5x5, et 8x8x8.

Ce choix de dimensions pour les cartes SOM-3D est motivé par la nécessité de gérer la complexité croissante tout en préservant la capacité d'analyse et d'interprétation. Les tailles plus petites pour les cartes 3D, comparativement aux cartes 2D, tiennent compte de l'augmentation exponentielle du nombre de neurones et des défis computationnels qui en découlent. Cette adaptation dimensionnelle est cruciale pour assurer une exploration efficace de l'espace des paramètres tout en maintenant des temps de calcul raisonnables.

Après avoir exploré en détail les performances des cartes auto-organisées bi-dimensionnelles (SOM 2D), nous étendons notre analyse aux cartes tri-dimensionnelles (SOM 3D). Ces cartes présentent une complexité supplémentaire due à leur dimension spatiale additionnelle, et nous examinons comment cela affecte les scores et les écarts types dans différentes configurations.

				Score					
Topologie	Taille	T_{max}	ϵ_0	μ_{SMAC}	$\mu_{Uniform}$	μ_{Random}	σ_{SMAC}	$\sigma_{Uniform}$	σ_{Random}
cube	3x3x3	1000	0.1	0.9683	0.9266	0.9094	0.0012	0.0011	0.0013
cube	3x3x3	50000	0.01	0.9733	0.9392	0.9159	0.0016	0.0006	0.0006
cube	5x5x5	1000	0.1	0.9722	0.9468	0.9180	0.0008	0.0014	0.0010
cube	5x5x5	50000	0.01	0.9786	0.9533	0.9271	0.0011	0.0011	0.0008
cube	8x8x8	1000	0.1	0.9745	0.9550	0.9224	0.0009	0.0012	0.0006
cube	8x8x8	50000	0.01	0.9812	0.9538	0.9326	0.0007	0.0009	0.0007

Analyse : Les résultats pour les SOM 3D montrent des tendances intéressantes en termes de scores obtenus dans les différents cas, ainsi que les écarts types associés. Comme pour les SOM 2D, l'analyse des scores nous permet d'évaluer la précision et l'efficacité des cartes tri-dimensionnelles dans la représentation des données.

Un aspect crucial dans l'analyse des SOM 3D est de comprendre comment l'augmentation de la dimension spatiale affecte la performance et la stabilité du modèle. Les écarts

types, en particulier, nous informent sur la consistance des résultats à travers différentes exécutions, fournissant un aperçu de la fiabilité des configurations testées.

En comparant les performances entre les SOM 2D et 3D, nous pouvons également tirer des conclusions sur l'impact de la dimensionnalité sur l'efficacité de l'algorithme SOM. Cette comparaison nous aide à déterminer dans quels contextes il est préférable d'utiliser des cartes bi-dimensionnelles ou tri-dimensionnelles, en fonction des objectifs spécifiques de l'analyse.

De la même manière que pour les cartes bi-dimensionnelles, l'hyperparamétrage des cartes auto-organisées tridimensionnelles (SOM 3D) basé principalement sur un critère unique présente certaines limitations. Bien que l'optimisation des paramètres en fonction de ce critère spécifique puisse améliorer les scores de performance, il est important de rester conscient des potentiels compromis.

L'accent mis sur un seul aspect de la performance peut occulter d'autres caractéristiques importantes des données, comme la topologie ou les relations intrinsèques entre les caractéristiques. Cette approche pourrait conduire à des résultats qui, bien que performants selon le critère choisi, ne capturent pas pleinement la complexité et la richesse des données en trois dimensions.

Ainsi, tout comme avec les SOM 2D, il est judicieux d'adopter une stratégie d'hyperparamétrage plus globale pour les SOM 3D, en prenant en compte divers critères de performance pour garantir une représentation complète et fidèle des données.

Chapitre 5

SOM alternative

5.1 Introduction

Malgré leur succès, les SOM souffrent de limites de passage à l'échelle, d'une forte dépendance aux hyperparamètres, d'une topologie de grille rigide et d'une incompatibilité avec les méthodes d'optimisation différentiables modernes.

5.1.1 Problématique

Face aux limites structurelles des SOM classiques, une piste de recherche consiste à repenser la construction de la carte non plus comme un processus d'adaptation itératif, mais comme la génération explicite d'un champ continu de poids définis sur la grille. Autrement dit, il s'agit d'apprendre une fonction générative $f_\theta : [0,1]^2 \to \mathbb{R}^d$ qui, à partir des coordonnées normalisées (i, j) d'un neurone, produit directement son vecteur de poids dans l'espace des données.

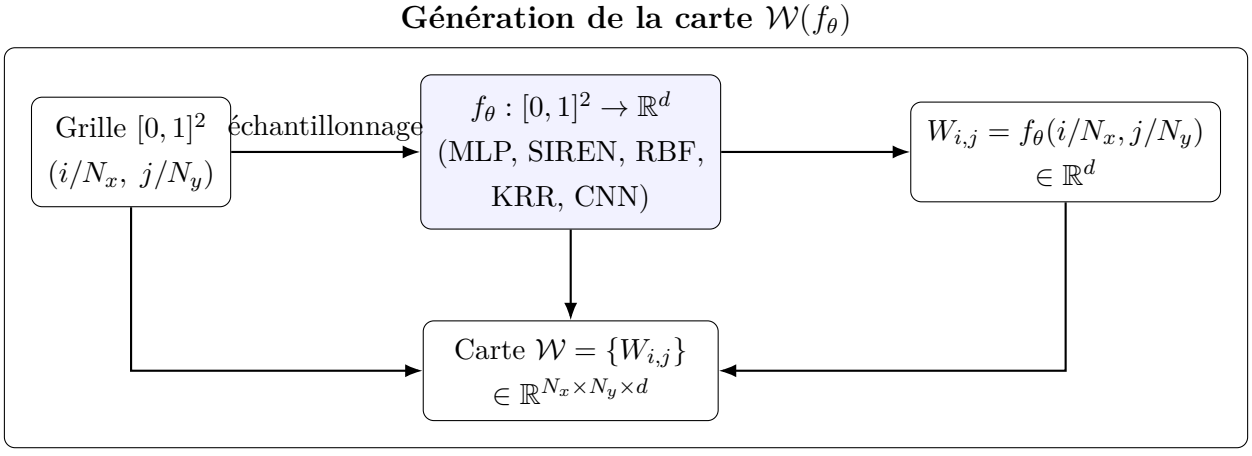


Figure 5.1 – Génération de la carte par un générateur f_θ.

Cette reformulation présente plusieurs motivations :

— **Continuité et expressivité** : en définissant la carte comme l'échantillonnage d'une fonction continue, il devient possible d'obtenir des cartes lisses, interpolables à résolution arbitraire, et plus adaptées à la modélisation de structures complexes.
— **Compatibilité avec les méthodes modernes** : la fonction générative peut être implémentée par divers modèles (réseaux de neurones, noyaux, modèles convolutionnels), dont les paramètres peuvent être optimisés par des techniques contemporaines, qu'il s'agisse d'optimisation par gradient, de méthodes évolutionnaires ou de régularisations explicites.
— **Compacité du modèle** : au lieu de stocker un vecteur de poids pour chaque neurone, l'ensemble de la carte est implicitement représenté par les paramètres de la fonction générative, ce qui permet de manipuler des cartes de grande taille sans accroître proportionnellement le nombre de paramètres libres.
— **Flexibilité topologique** : la grille (i, j) n'est qu'un support d'échantillonnage, et peut donc être généralisée (par exemple en coordonnées continues, sphériques ou irrégulières), ouvrant la voie à des cartes mieux adaptées aux structures de données spécifiques.

L'objectif de ce chapitre est d'évaluer si des fonctions génératives f_θ peuvent constituer une alternative viable aux SOM classiques pour construire des cartes auto-organisées. Nous considérons trois grandes familles de générateurs : (i) des réseaux fully-connected (MLP, MLP enrichis par Fourier features ou activations sinusoïdales SIREN), (ii) des modèles à noyaux (RBF, KRR), et (iii) des réseaux convolutionnels adaptés aux coordonnées (CoordConv). Nous analyserons leurs performances en termes de quantification, de préservation topologique et de couverture de l'espace de données, afin d'identifier leurs avantages, limites et potentiel face aux SOM standards.

5.1.2 Approche générale

Concrètement, plutôt que de mettre à jour les poids $W_{i,j}$ neurone par neurone au fil d'un processus itératif, nous cherchons à apprendre une fonction générative f_θ qui associe à chaque position (i, j) de la grille un vecteur dans l'espace des données. L'ensemble de la carte est alors obtenu en une seule passe, comme un échantillonnage de f_θ sur la grille considérée.

L'optimisation des paramètres θ du générateur peut ensuite être réalisée selon différents paradigmes :

— **par optimisation locale** (par exemple rétropropagation du gradient) lorsque la

fonction objectif est différentiable ou peut être approchée de manière continue,

— **par optimisation globale sans gradient** (telle que CMA-ES [34] ou d'autres algorithmes évolutionnaires) lorsque la fonction objectif repose sur des métriques globales non différentiables, comme la préservation de la topologie ou la couverture des données.

Cette approche offre un cadre unifié permettant de comparer différents générateurs (MLP, SIREN, modèles à noyaux, CNN coord-conv, etc.) ainsi que plusieurs stratégies d'optimisation. Elle ouvre ainsi la voie à une conception plus flexible et généralisable des SOM, détachée des contraintes historiques de l'apprentissage compétitif classique.

5.2 Générateurs continus de SOM

5.2.1 Cadre général de génération et optimisation

Nous considérons la construction d'une carte auto-organisée comme l'échantillonnage d'une fonction générative continue

$$f_\theta : [0,1]^2 \longrightarrow \mathbb{R}^d,$$

où (i, j) désigne la position normalisée d'un neurone dans la grille et $W_{i,j} = f_\theta(i, j)$ est son vecteur de poids dans l'espace des données.

La carte complète est alors donnée par

$$\mathcal{W}(f_\theta) = \Big\{ f_\theta\Big(\tfrac{i}{N_x}, \tfrac{j}{N_y}\Big) \;\Big|\; 0 \le i < N_x,\ 0 \le j < N_y \Big\},$$

où (N_x, N_y) est la taille de la grille.

Le problème de construction d'une SOM peut ainsi être reformulé comme une tâche d'optimisation :

$$\theta^\star = \arg\max_\theta \; S\Big(\mathcal{W}(f_\theta)\Big),$$

où $S(\cdot)$ désigne une fonction de score globale mesurant la qualité de la carte (par exemple via l'erreur de quantification, l'erreur topographique, ou une combinaison pondérée de plusieurs métriques).

Le score S n'étant pas différentiable (il repose sur des opérations discrètes comme l'appariement BMU-données ou des mesures de voisinage topologique), il n'est pas adapté aux méthodes classiques d'optimisation par gradient. Nous faisons donc appel à CMA-ES

(Covariance Matrix Adaptation Evolution Strategy), une méthode évolutionnaire d'optimisation sans gradient.

Avant de détailler son application à la génération de cartes auto-organisées, rappelons le principe générique de CMA-ES (Covariance Matrix Adaptation Evolution Strategy) [36]. Elle fait partie des algorithmes à estimation de distribution, et correspond à une stratégie d'évolution pour l'optimisation continue sans gradient, qui adapte dynamiquement une distribution gaussienne multivariée sur l'espace des paramètres.

Le principe est le suivant : à chaque itération, une population de candidats est échantillonnée, évaluée, puis utilisée pour mettre à jour successivement :

— le centre m de la distribution (direction de recherche),
— la matrice de covariance Σ (formes et corrélations entre variables),
— le pas d'exploration σ (taille globale des mutations).

Cette adaptation progressive confère à CMA-ES une grande robustesse, en particulier dans les contextes non convexes, bruités ou non différentiables. L'algorithme générique est résumé ci-dessous.

Algorithm 3 CMA-ES – Covariance Matrix Adaptation Evolution Strategy

Require: Fonction objectif $f : \mathbb{R}^n \to \mathbb{R}$, initialisation $m \in \mathbb{R}^n$, $\sigma > 0$, $\Sigma = I_n$, nombre d'itérations T, taille de population λ

Définir $\mu = \lfloor \lambda/2 \rfloor$, et des poids w_i tels que $\sum w_i = 1$
Initialiser $p_c \leftarrow 0$, $p_\sigma \leftarrow 0$
for $t = 1$ **to** T **do**
▷ — Échantillonnage —
    for $k = 1$ **to** λ **do**
        $z_k \sim \mathcal{N}(0, I)$
        $x_k \leftarrow m + \sigma \cdot A \cdot z_k, \quad$ où A telle que $\Sigma = AA^\top$
        $f_k \leftarrow f(x_k)$
    end for
▷ — Sélection —
    Trier les x_k selon f_k croissant
    $m \leftarrow \sum_{i=1}^{\mu} w_i \cdot x_{i:\lambda}$ ▷ Mise à jour du centre
▷ — Mise à jour des chemins de cumul —
    $y = m - m_{\text{old}}, \quad z = A^{-1} \cdot y/\sigma$
    $p_\sigma \leftarrow (1 - c_\sigma)p_\sigma + \sqrt{c_\sigma(2 - c_\sigma)\mu_{\text{eff}}} \cdot z$
    $p_c \leftarrow (1 - c_c)p_c + h_\sigma \sqrt{c_c(2 - c_c)\mu_{\text{eff}}} \cdot y/\sigma$
▷ — Mise à jour de la covariance —
    $\Sigma \leftarrow (1 - c_1 - c_\mu)\Sigma + c_1 p_c p_c^\top + c_\mu \sum_{i=1}^{\mu} w_i (x_i - m)(x_i - m)^\top$
▷ — Mise à jour du step-size —
    $\sigma \leftarrow \sigma \cdot \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|p_\sigma\|}{E[\|\mathcal{N}(0,I)\|]} - 1\right)\right)$
    $m_{\text{old}} \leftarrow m$
end for

CMA-ES maintient une distribution gaussienne multivariée $\mathcal{N}(m, \sigma^2\Sigma)$ sur l'espace des paramètres θ, et l'adapte au fil des générations en sélectionnant les meilleurs candidats selon S. À chaque itération, une population de paramètres est échantillonnée, transformée en cartes SOM via f_θ, évaluée par S, puis la distribution est mise à jour pour concentrer la recherche dans les régions les plus prometteuses.

Le processus complet est résumé dans l'algorithme 4, qui constitue le socle commun à tous les générateurs étudiés dans ce chapitre. Seule la forme paramétrique de f_θ varie d'un modèle à l'autre (MLP, SIREN, RBF, etc.), tandis que la boucle d'optimisation reste

inchangée.

Algorithm 4 Optimisation d'un générateur de SOM par CMA-ES

Require: Générateur f_θ, taille de carte (N_x, N_y), fonction de score S, distribution initiale $\mathcal{N}(m_0, \sigma_0^2 I)$

1: **while** critère d'arrêt non atteint **do**
2: Échantillonner une population $\{\theta_k\}_{k=1}^{\lambda} \sim \mathcal{N}(m, \sigma^2\Sigma)$
3: **for** chaque θ_k **do**
4: Générer la carte $\mathcal{W}_k = \{f_{\theta_k}(i/N_x, j/N_y)\}$
5: Évaluer $s_k = S(\mathcal{W}_k)$
6: **end for**
7: Mettre à jour (m, Σ, σ) selon la règle CMA-ES en fonction des meilleurs s_k
8: **end while**
9: **return** $\theta^\star = \arg\max_{\theta_k} s_k$

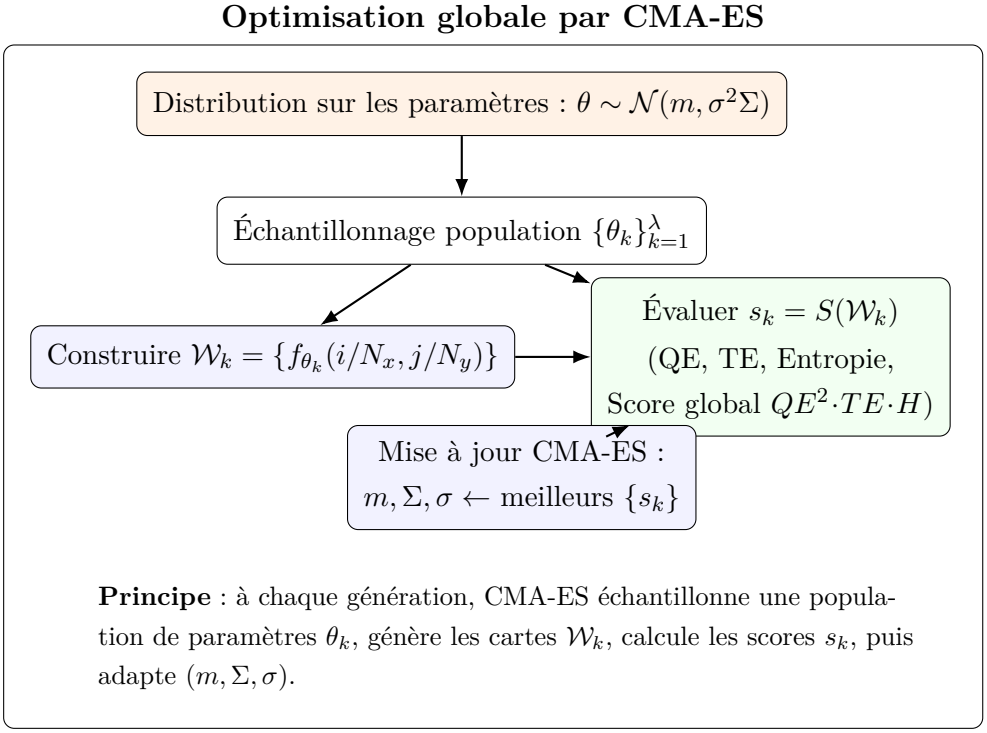


FIGURE 5.2 – Optimisation globale des paramètres par CMA-ES en maximisant le score $S(\mathcal{W})$.

5.2.2 MLP

La première famille de générateurs que nous considérons repose sur un perceptron multicouche (MLP), utilisé comme fonction de régression continue. L'idée est simple : associer à chaque position normalisée $(i, j) \in [0, 1]^2$ de la grille SOM un vecteur de poids $W_{i,j} \in \mathbb{R}^d$ via une fonction paramétrée f_θ implémentée par un MLP. (cf. Annexe 5)

Formulation : Un MLP standard est défini par une succession de couches linéaires et de fonctions d'activation non linéaires. Soit une architecture à L couches cachées de dimensions $d_1, \ldots, d_L$, le calcul s'écrit :

$$\begin{aligned} a^{(0)} &= (i, j) \in [0, 1]^2, \\ z^{(\ell)} &= W^{(\ell)} a^{(\ell-1)} + b^{(\ell)}, \quad \ell = 1, \ldots, L, \\ a^{(\ell)} &= \sigma(z^{(\ell)}), \\ f_\theta(i, j) &= W^{(L+1)} a^{(L)} + b^{(L+1)} \in \mathbb{R}^d, \end{aligned}$$

où σ est typiquement une fonction d'activation ReLU, et $\theta = \{W^{(\ell)}, b^{(\ell)}\}$ l'ensemble des paramètres.

La carte SOM générée correspond alors à l'échantillonnage

$$W_{i,j} = f_\theta\Big(\tfrac{i}{N_x}, \tfrac{j}{N_y}\Big), \quad 0 \le i < N_x,\ 0 \le j < N_y.$$

Optimisation : Les paramètres θ sont optimisés via CMA-ES en maximisant la fonction de score $S(\mathcal{W}(f_\theta))$ (cf. Section précédente). Le rôle du MLP est donc uniquement de fournir une paramétrisation compacte et continue de la carte, tandis que CMA-ES guide l'adaptation des poids pour que la carte respecte les critères de qualité (quantification, préservation topologique, couverture).

Avantages :

- **Simplicité** : le MLP est une architecture élémentaire, facile à implémenter et à manipuler.
- **Universalité** : un MLP est un approximateur universel de fonctions continues, donc capable en théorie de représenter n'importe quel champ $(i, j) \mapsto W_{i,j}$.
- **Compacité** : quelques centaines ou milliers de paramètres suffisent à représenter des cartes de grande taille.

Limites :

- **Rigidité des activations classiques** : avec des activations ReLU ou sigmoïdes, le MLP peine à représenter des variations rapides ou des structures oscillantes dans la carte.
- **Lissage excessif** : par défaut, le MLP tend à produire des champs trop lisses, ce qui peut limiter la finesse des frontières de clusters.
- **Sensibilité à l'initialisation** : selon les poids initiaux, le champ peut présenter des discontinuités locales ou des régions mortes.

Cette configuration constitue néanmoins une **baseline essentielle**, à partir de laquelle les variantes étudiées dans les sections suivantes (encodages de Fourier, SIREN, modèles à noyaux, CNN coord-conv) cherchent à pallier ces limitations et à améliorer l'expressivité de la fonction générative.

5.2.3 MLP avec Fourier features

Afin d'augmenter l'expressivité du MLP de base, nous considérons l'ajout d'un encodage harmonique des coordonnées (i, j) avant leur passage dans le réseau. Cette idée, introduite notamment dans le contexte des champs de NeRF [95], permet au modèle de mieux représenter des variations de haute fréquence dans les données générées. (cf. Annexe 6)

Encodage harmonique : Au lieu de fournir directement les coordonnées normalisées $(i, j) \in [0, 1]^2$ en entrée du MLP, on les transforme à l'aide d'un encodage sinusoïdal $\phi : [0, 1]^2 \to \mathbb{R}^{4L}$ défini par :

$$\phi(i, j) = \Big[\sin(2^k \pi i), \cos(2^k \pi i),\ \sin(2^k \pi j), \cos(2^k \pi j) \Big]_{k=0}^{L-1},$$

où L est le nombre de fréquences retenues.

L'entrée effective du MLP est donc le vecteur $\phi(i, j)$, qui enrichit les coordonnées par une base harmonique multi-échelle. En pratique, nous utilisons des valeurs de L comprises entre 4 et 8, offrant un compromis entre expressivité et dimension d'entrée.

Formulation : La fonction générative devient alors :

$$W_{i,j} = f_\theta\Big(\phi(i/N_x, j/N_y)\Big),$$

où f_θ est un MLP classique, identique à celui présenté dans la section précédente.

Impact attendu : L'introduction des Fourier features modifie profondément la capacité du MLP :

- **Représentation des motifs oscillants** : grâce aux harmoniques sinusoïdales, le MLP peut représenter des variations rapides dans la carte (frontières abruptes, structures périodiques).
- **Interpolation plus riche** : les cartes générées peuvent capturer des structures non triviales qui échappent à un MLP simple, en particulier lorsque les données présentent des régularités périodiques.
- **Contrôle de la fréquence** : le choix de L agit comme un hyperparamètre de régularisation, limitant ou favorisant l'apparition de détails fins.

Limites :

- **Risque d'aliasing** : un encodage trop riche (grand L) peut conduire à des oscillations parasites dans la carte.
- **Dimension accrue** : l'entrée du MLP passe de 2 à $4L$ dimensions, ce qui augmente la complexité et le coût d'optimisation.
- **Sensibilité à l'échelle** : le choix des fréquences doit être adapté aux structures présentes dans les données, sous peine de sous-exploitation ou de sur-ajustement.

Ainsi, l'encodage harmonique constitue une extension naturelle du MLP baseline, visant à surmonter sa tendance au lissage excessif et à accroître la finesse des structures topologiques apprises.

5.2.4 SIREN : réseaux à activations sinusoïdales

Une autre alternative au MLP standard consiste à remplacer les fonctions d'activation classiques (ReLU, sigmoid) par des activations sinusoïdales. Les SIREN (Sinusoidal Representation Networks) [91] ont été proposés initialement pour l'approximation de champs continus, et se distinguent par leur capacité à représenter des fonctions régulières avec des variations rapides ou lisses. (cf. Annexe 7)

Formulation : Un SIREN implémente une fonction générative $f_\theta : [0,1]^2 \to \mathbb{R}^d$ définie par :

$$a^{(l)} = \sin\Big(W^{(l)} a^{(l-1)} + b^{(l)}\Big),$$

où $a^{(0)} = (i/N_x, j/N_y)$ représente les coordonnées normalisées, et $a^{(L)}$ constitue la sortie finale, c'est-à-dire le vecteur de poids $W_{i,j}$ du neurone correspondant.

La sortie est donc directement obtenue par une composition linéaire-sinusoïdale de profondeur L, ce qui confère au modèle un comportement oscillatoire naturel.

Paramètres clés : Le bon fonctionnement des SIREN repose sur une initialisation adaptée des poids, contrôlée par deux hyperparamètres :

— w_0 : facteur de fréquence appliqué à la première couche, qui règle la densité des oscillations initiales.

— $w_{0,\text{hidden}}$: facteur similaire appliqué aux couches cachées, influençant la propagation des hautes fréquences au sein du réseau.

Des valeurs typiques sont $w_0 = 30$ et $w_{0,\text{hidden}} = 1$, comme recommandé dans [91].

Impact attendu : L'utilisation d'activations sinus offre plusieurs avantages :

— **Représentation de champs lisses** : la sinusoïde agit comme un inducteur naturel de régularité spatiale.

— **Capture de hautes fréquences** : contrairement aux ReLU qui tendent à lisser, les SIREN peuvent représenter fidèlement des motifs complexes ou oscillants.

— **Continuité différentiable** : la fonction générative est partout dérivable, ce qui peut faciliter certaines formes d'optimisation.

Limites :

— **Risque de sur-oscillation** : des valeurs trop élevées de w_0 ou $w_{0,\text{hidden}}$ entraînent des cartes bruitées ou instables.

— **Optimisation sensible** : la sinusoïde peut produire des gradients oscillants, rendant certaines méthodes d'optimisation instables (notamment par descente de gradient).

— **Interprétation plus délicate** : contrairement aux MLP standards, l'effet des hyperparamètres sur la topologie finale est moins intuitif.

En résumé, les SIREN constituent une alternative particulièrement adaptée lorsque la carte cible présente des structures continues et régulières, tout en conservant la capacité à modéliser des variations fines grâce à la richesse fréquentielle de la sinusoïde.

5.2.5 Réseaux RBF

Une autre famille de générateurs considérés dans ce travail repose sur les *Radial Basis Function Networks* (RBF). Contrairement aux MLP ou aux SIREN, qui génèrent la carte via une composition de couches linéaires et d'activations, les réseaux RBF reposent sur une interpolation explicite à partir de centres et de noyaux radiaux. Ils constituent une alternative naturelle lorsque l'on souhaite imposer une régularité spatiale forte et une interpolation locale des poids de la carte. (cf. Annexe 8)

Formulation : Un RBF network définit la fonction générative $f_\theta : [0,1]^2 \to \mathbb{R}^d$ sous la forme :

$$W_{i,j} = f_\theta(i/N_x, j/N_y) = \sum_{k=1}^{K} \alpha_k \, \varphi\left(\frac{\|(i/N_x, j/N_y) - c_k\|}{\sigma_k} \right),$$

où :

- $c_k \in [0,1]^2$ sont les centres des noyaux,
- $\sigma_k > 0$ leur largeur,
- $\alpha_k \in \mathbb{R}^d$ les poids associés,
- $\varphi(r) = \exp(-r^2)$ est un noyau gaussien radial (cas le plus courant).

Ainsi, la carte est obtenue comme une combinaison linéaire de contributions locales, chaque noyau RBF influençant son voisinage spatial.

Régularité spatiale et interpolation : Cette formulation présente plusieurs propriétés intéressantes :

- **Régularité spatiale** : les noyaux gaussiens imposent une continuité naturelle de la carte, garantissant une transition douce entre neurones voisins.
- **Interpolation locale** : chaque centre c_k agit comme une base locale, capable de modéliser finement des structures présentes dans certaines régions spécifiques de la carte.
- **Compacité** : le nombre de noyaux K peut être choisi inférieur au nombre de neurones, ce qui réduit le nombre de paramètres tout en maintenant la capacité d'interpolation.

Impact attendu : Les RBF networks devraient favoriser :

- une couverture homogène de l'espace des données,
- une continuité accrue dans les cartes générées,

— une meilleure stabilité face aux méthodes d'optimisation globales comme CMA-ES, grâce au lissage intrinsèque des noyaux.

Limites :

— **Choix des centres** : la qualité dépend fortement de la répartition des c_k (aléatoire, grille régulière, ou appris).

— **Largeur des noyaux** : des valeurs trop petites mènent à une carte bruitée, trop grandes conduisent à une carte sur-lissée.

— **Scalabilité** : le calcul direct devient coûteux pour un grand nombre de noyaux.

En résumé, les réseaux RBF représentent une approche intermédiaire entre un générateur global (MLP, SIREN) et une interpolation locale directe. Ils offrent un contrôle explicite sur la régularité spatiale des cartes et se prêtent bien à une optimisation par CMA-ES.

5.2.6 Régression ridge à noyau (KRR)

La régression ridge à noyau (Kernel Ridge Regression, KRR) constitue une extension naturelle des réseaux RBF, en introduisant une régularisation explicite des poids. Elle permet de définir une fonction générative lisse et contrôlée, tout en bénéficiant du formalisme bien établi des méthodes à noyaux. (cf. Annexe 9)

Formulation : On considère un ensemble de M points d'ancrage $\{c_k\}_{k=1}^{M} \subset [0,1]^2$, choisis par exemple sur une grille régulière. La fonction générative est définie comme :

$$f_\theta(x) = \sum_{k=1}^{M} \alpha_k \, K(x, c_k),$$

où K est un noyau symétrique positif (typiquement gaussien : $K(x,y) = \exp(-\|x - y\|^2/2\sigma^2)$), et $\alpha_k \in \mathbb{R}^d$ sont les coefficients associés. Pour une coordonnée (i,j), le poids généré est donc :

$$W_{i,j} = f_\theta\Big(\tfrac{i}{N_x}, \tfrac{j}{N_y}\Big)\,.$$

Régularisation : Les coefficients $\alpha = (\alpha_1, \ldots, \alpha_M)$ sont obtenus par une résolution fermée de type ridge :

$$\alpha = (K + \lambda I)^{-1} Y,$$

où K est la matrice noyau $K_{kl} = K(c_k, c_l)$, Y la matrice des vecteurs cibles associés aux centres, et $\lambda > 0$ le paramètre de régularisation. La présence du terme λI contrôle la complexité du modèle :

— λ petit : interpolation quasi-parfaite, risque de sur-ajustement.
— λ grand : lissage fort, carte plus régulière mais moins fidèle.

Impact attendu : L'utilisation de KRR comme générateur de SOM présente plusieurs avantages :

— **Contrôle explicite de la régularité** via le paramètre λ et la largeur du noyau σ.
— **Solution analytique** (hors optimisation par CMA-ES), garantissant une construction déterministe et reproductible.
— **Lien direct avec les méthodes RBF** : la KRR généralise l'interpolation par noyaux en y ajoutant une pénalisation de la norme des coefficients.

Limites :

— **Complexité computationnelle** : l'inversion de la matrice noyau $K \in \mathbb{R}^{M \times M}$ est coûteuse lorsque M est grand ($\mathcal{O}(M^3)$).
— **Choix des hyperparamètres** : la performance dépend fortement du réglage conjoint de σ et λ, ce qui peut nécessiter une validation croisée ou une optimisation globale.
— **Moins flexible que les réseaux profonds** : contrairement aux MLP ou CNN, la KRR ne bénéficie pas d'une hiérarchie de représentations.

En résumé, la KRR fournit un cadre théorique robuste et régularisé pour générer des cartes SOM continues. Elle se situe à l'interface entre l'interpolation locale (RBF) et les modèles paramétriques complexes (MLP, CNN), et offre un contrôle direct de la complexité et du lissage des cartes via ses hyperparamètres λ et σ.

5.2.7 CNN CoordConv

Enfin, nous considérons une approche convolutionnelle adaptée aux coordonnées, inspirée des travaux de [67]. Les *CoordConv CNN* consistent à enrichir l'entrée du réseau convolutionnel par les coordonnées spatiales explicites (x, y), ce qui permet de lever une limitation des CNN classiques : leur difficulté à apprendre directement des relations basées sur la position absolue dans la grille. (cf. Annexe 10)

Formulation : Dans ce cadre, les coordonnées normalisées $(i/N_x, j/N_y)$ sont transformées en deux canaux supplémentaires concaténés aux cartes de caractéristiques internes. Ainsi, l'entrée du réseau devient un tenseur de dimension $(N_x, N_y, d+2)$, où d est la dimension des variables de données, et les deux canaux supplémentaires codent les coordonnées (x, y).

Le générateur convolutionnel f_θ est alors une composition de convolutions 1×1 et 3×3, suivies d'activations non linéaires (ReLU ou tanh), appliquées sur cet espace enrichi. La sortie est un tenseur (N_x, N_y, d) représentant directement les poids de la SOM.

Motivation : Les CoordConv CNN combinent les atouts des approches convolutionnelles et paramétriques :

- **Capture de régularités locales** : grâce aux filtres convolutionnels, la génération des poids intègre des motifs spatiaux partagés dans toute la grille.
- **Sensibilité aux positions absolues** : l'ajout explicite des coordonnées permet au réseau de différencier les neurones par leur emplacement, ce qui est crucial pour la topologie des SOM.
- **Scalabilité** : contrairement aux RBF ou KRR, la complexité est linéaire en fonction de la taille de la grille, ce qui rend possible l'utilisation de cartes de grande dimension.

Impact attendu : Cette approche devrait favoriser :

- une meilleure organisation spatiale des cartes,
- une généralisation efficace aux grandes grilles,
- une capacité accrue à reproduire des régularités locales présentes dans les données.

Limites :

- **Moins d'expressivité globale** : les CNN sont principalement adaptés à des motifs locaux ; ils peuvent avoir du mal à modéliser des dépendances globales de la carte.
- **Choix architectural non trivial** : la profondeur, la largeur et la taille des filtres influencent fortement la qualité de la carte générée.
- **Optimisation plus coûteuse** : l'augmentation du nombre de paramètres peut alourdir l'optimisation par CMA-ES.

En résumé, le CoordConv CNN introduit une structure inductive explicite qui tire parti à la fois des coordonnées et des régularités spatiales locales. Il constitue une alternative

prometteuse pour générer des cartes SOM, particulièrement dans le cas de grilles de grande taille ou de données structurées spatialement.

5.2.8 Comparaison des modèles

Modèle	Avantages	Inconvénients
SOM classique	Méthode robuste, bien établie dans la littérature ; ne nécessite pas d'architecture paramétrique ; sert de référence historique.	Scalabilité limitée ; sensible aux hyperparamètres ; rigidité de la grille fixée a priori.
MLP (baseline)	Simplicité de mise en œuvre ; capacité d'approximation universelle ; bonne robustesse avec CMA-ES.	Peut manquer de finesse pour modéliser des variations rapides ; performances dépendantes de l'architecture (profondeur/largeur).
MLP + Fourier features	Encodage harmonique permettant de capturer des motifs de haute fréquence ; expressivité accrue sans augmenter fortement la taille du réseau.	Peut introduire des instabilités à haut budget ; peine à s'imposer si les données ne présentent pas de structures oscillantes.
SIREN	Activations sinusoïdales adaptées aux champs continus et lisses ; capacité à modéliser des transitions douces et régulières.	Paramètres sensibles (notamment w_0) ; risque d'instabilité en optimisation prolongée.
RBF networks	Interpolation locale simple et intuitive ; régularité spatiale naturelle grâce aux noyaux gaussiens.	Peu expressif pour des structures complexes ; dépend fortement du placement des centres.
KRR (Kernel Ridge Regression)	Formulation analytique claire ; contrôle explicite par λ (régularisation) et la largeur du noyau ; robuste à faible budget.	Peut être coûteux en calcul pour des cartes de grande taille ; risque de sur-lissage limitant la finesse des structures.
CNN CoordConv	Exploite la structure spatiale locale ; peut capturer des motifs réguliers ou répétitifs dans la grille.	Expressivité accrue mais sensible à l'optimisation ; risque de surajustement ; moins performant à faible budget.

TABLE 5.1 – Avantages et inconvénients des modèles génératifs testés.

5.2.9 Justification des modèles étudiés

Chaque générateur retenu répond à une motivation précise, que nous résumons ci-dessous :

- **SOM classique** : sert de *baseline historique*, afin de comparer les approches alternatives à la méthode originale de Kohonen.
- **MLP (baseline)** : représente l'option la plus simple de fonction générative continue, capable d'approximer toute fonction $(i, j) \mapsto W_{i,j}$ selon le théorème d'approximation universelle. Il sert de point de départ minimal.
- **MLP + Fourier features** : introduit un encodage harmonique des coordonnées, permettant de mieux capturer des motifs oscillants ou de haute fréquence dans les cartes générées.
- **SIREN** : utilise des activations sinus, adaptées à la modélisation de champs continus et lisses. Il s'agit d'un candidat naturel pour reproduire des cartes régulières avec transitions douces.
- **RBF networks** : basés sur des noyaux gaussiens centrés sur des prototypes, ils favorisent l'interpolation locale et la régularité spatiale des cartes.
- **KRR (Kernel Ridge Regression)** : offre une alternative analytique, où le champ génératif est appris via une régression à noyau régularisée. Ce modèle apporte un contrôle explicite par la pénalisation λ et la largeur du noyau.
- **CNN CoordConv** : enrichit l'entrée par des coordonnées (x, y) injectées comme canaux supplémentaires. Cette approche permet de tirer parti de la structure convolutionnelle pour capturer des régularités locales dans la grille.

Ainsi, la sélection couvre à la fois des approches classiques (SOM, MLP), des variantes à expressivité accrue (Fourier, SIREN), des modèles à noyaux (RBF, KRR), et des modèles convolutionnels (CoordConv), offrant un panorama représentatif des familles de générateurs possibles.

5.3 Protocole expérimental

5.3.1 Optimisation

L'optimisation des paramètres θ des générateurs f_θ constitue un élément central du protocole expérimental. Dans notre approche, nous privilégions par défaut une stratégie sans gradient, robuste aux fonctions objectifs non différentiables, tout en explorant des

variantes complémentaires.

La méthode retenue est **CMA-ES (Covariance Matrix Adaptation Evolution Strategy)**. Cet algorithme évolutionnaire est particulièrement adapté dans notre contexte, où les critères d'évaluation des cartes (erreurs de quantification et de topologie, entropie d'usage, etc.) ne sont pas nécessairement différentiables par rapport aux paramètres du générateur.

CMA-ES procède par itérations successives : une population de candidats est échantillonnée selon une distribution gaussienne multivariée, évaluée sur la fonction objectif, puis utilisée pour mettre à jour le centre, la covariance et l'amplitude de la distribution. Cette dynamique permet d'explorer efficacement des espaces de grande dimension tout en s'adaptant progressivement à la géométrie de la fonction objectif.

Afin de garantir une comparaison équitable entre les générateurs, le **budget d'évaluations** (nombre de générations multiplié par la taille de population) est fixé identiquement pour toutes les méthodes. Chaque modèle dispose donc du même nombre total d'évaluations de la fonction objectif, indépendamment de sa complexité interne.

5.3.2 Métriques

De nombreuses métriques ont été proposées dans la littérature pour évaluer la qualité des cartes auto-organisées, comme détaillé au chapitre 4 (erreur de quantification, distorsion, erreur topographique, erreur combinée, produit topographique, erreur de Kruskal-Shepard, C-mesure, fonction topographique, etc.). Cependant, plusieurs de ces mesures capturent des aspects similaires de la carte, et il est rarement nécessaire de toutes les mobiliser dans un protocole expérimental.

Dans notre cadre, nous retenons trois métriques jugées **centrales et complémentaires** :

— **Erreur de quantification (QE)** : mesure la compacité locale en évaluant la distance moyenne entre les données et leur neurone gagnant (BMU). Cette métrique est au cœur du principe même de l'algorithme SOM, qui vise à représenter les données par un ensemble de prototypes.

$$QE = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \|x - W_{BMU(x)}\|$$

— **Erreur topographique (TE)** : quantifie la préservation du voisinage dans la carte en mesurant la proportion de données dont les deux BMU les plus proches

ne sont pas voisins sur la grille.

$$TE = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \mathbb{1}\Big[BMU_1(x) \not\sim BMU_2(x) \Big]$$

— **Entropie d'usage (H)** : reflète la couverture effective de la carte, en mesurant l'équilibre d'activation des neurones. Soit $p_{i,j}$ la proportion de données dont le BMU est le neurone (i, j) :

$$H = -\sum_{i,j} p_{i,j} \log p_{i,j}$$

Une entropie élevée traduit une utilisation équilibrée des neurones, tandis qu'une entropie faible révèle un effondrement vers une sous-partie de la carte.

Comme contribution méthodologique, nous proposons de combiner ces trois critères en une métrique globale :

$$Score = QE^2 \times TE \times H$$

Le choix de cette formulation repose sur plusieurs considérations :

— la QE, centrale dans le principe de l'algorithme SOM, est volontairement favorisée en apparaissant au carré,

— la TE reflète la préservation de la structure topologique, complémentaire de la compacité,

— l'entropie d'usage assure que la carte soit effectivement exploitée dans son ensemble.

Ainsi définie, cette métrique synthétique permet de couvrir de manière efficace les trois dimensions essentielles de la qualité d'une SOM : fidélité locale, préservation globale de la topologie et équilibre d'utilisation. Elle sera utilisée comme critère principal de comparaison des générateurs dans la suite du chapitre.

5.3.3 Statistiques et protocole

Les expérimentations de ce chapitre reposent sur des données issues de l'environnement commercial d'Octave. Afin de garantir la robustesse et la généralisabilité des conclusions, nous considérons 10 extractions indépendantes de la base, chacune contenant entre 100 et 1000 clients. Ces jeux de données couvrent volontairement une diversité de tailles et de structures, allant de situations simples (clusters compacts, faible dimensionnalité) à des cas plus complexes (variabilité forte, hétérogénéité structurelle). Pour chaque combinaison

(générateur, budget), l'optimisation est répétée sur les 10 extractions, et les résultats présentés correspondent à la moyenne et à l'écart-type des scores obtenus. Ce protocole permet de lisser les fluctuations liées aux caractéristiques spécifiques d'un dataset et de donner une estimation plus fidèle des performances typiques.

L'optimisation des paramètres θ est réalisée via CMA-ES, dont la progression dépend principalement du nombre total d'itérations et de la taille de population λ. Afin de couvrir différents scénarios, trois niveaux de budget sont fixés :

— **Budget faible** : 100 itérations, correspondant à un apprentissage rapide sous contrainte, utile pour évaluer l'efficacité initiale des générateurs ;
— **Budget moyen** : 500 itérations, offrant un compromis entre temps de calcul et qualité attendue des cartes ;
— **Budget élevé** : 2000 itérations, destiné à approcher le potentiel maximal de chaque modèle dans des conditions favorables.

Le nombre total d'évaluations de la fonction objectif correspond à budget $\times$ λ, où λ est la taille de population de CMA-ES (fixée identiquement pour tous les modèles). Ainsi, chaque générateur dispose du même budget en termes d'évaluations, garantissant une **comparaison équitable** indépendamment de la complexité interne du modèle ou du nombre de paramètres.

Les performances sont évaluées selon les métriques définies précédemment (erreur de quantification, erreur topographique, entropie d'usage, score global combiné). Pour chaque modèle et chaque budget, nous reportons :

— les valeurs moyennes de chaque métrique sur les 10 extractions,
— les écarts-types associés, traduisant la variabilité inter-datasets,
— et, lorsque pertinent, les résultats des tests de significativité comparant les modèles entre eux.

En synthèse, le protocole expérimental suit les principes suivants :

1. **Jeux de données** : 10 extractions Octave (100 à 1000 clients chacune).
2. **Budgets** : 100, 500, 2000 itérations CMA-ES (population λ fixée).
3. **Résultats** : moyennes $\pm$ écarts-types, avec tests non paramétriques pour comparer les modèles.
4. **Équité** : même nombre total d'évaluations pour tous les générateurs.

Ce cadre méthodologique garantit à la fois la rigueur statistique et la comparabilité des approches étudiées.

5.3.4 Résultats

Les résultats expérimentaux sont présentés pour trois niveaux de budget d'optimisation : faible, moyen et élevé. Pour chaque configuration, nous reportons les valeurs moyennes des métriques principales (QE, TE, H) ainsi que le score global proposé, défini comme $QE^2 \times TE \times H$.

Budget faible

Modèle	μ_{QE}	μ_{TE}	μ_H	μ_{Score}	σ_{QE}	σ_{TE}	σ_H	σ_{Score}
SOM	0.931	0.785	0.864	**0.588**	0.002	0.089	0.013	**0.071**
MLP	0.889	0.960	0.836	**0.634**	0.009	0.028	0.045	**0.034**
MLPFF	0.870	0.873	0.661	**0.437**	0.017	0.050	0.030	**0.032**
SIREN	0.905	0.931	0.785	**0.598**	0.011	0.035	0.037	**0.036**
RBF	0.839	0.660	0.590	**0.271**	0.020	0.086	0.057	**0.027**
KRR	0.915	0.951	0.727	**0.579**	0.005	0.015	0.018	**0.016**
CNN	0.851	0.870	0.685	**0.430**	0.029	0.067	0.045	**0.037**

Budget moyen

Modèle	μ_{QE}	μ_{TE}	μ_H	μ_{Score}	σ_{QE}	σ_{TE}	σ_H	σ_{Score}
SOM	0.939	0.876	0.905	**0.699**	0.001	0.050	0.006	**0.039**
MLP	0.915	0.977	0.937	**0.767**	0.0152	0.012	0.028	**0.049**
MLPFF	0.886	0.955	0.736	**0.550**	0.014	0.022	0.072	**0.046**
SIREN	0.922	0.963	0.833	**0.682**	0.007	0.022	0.033	**0.032**
RBF	0.819	0.668	0.601	**0.268**	0.031	0.061	0.049	**0.020**
KRR	0.938	0.995	0.782	**0.684**	0.003	0.006	0.030	**0.026**
CNN	0.878	0.924	0.711	**0.507**	0.017	0.042	0.048	**0.047**

Budget élevé

Modèle	μ_{QE}	μ_{TE}	μ_H	μ_{Score}	σ_{QE}	σ_{TE}	σ_H	σ_{Score}
SOM	0.941	0.870	0.903	**0.696**	0.001	0.029	0.005	**0.020**
MLP	0.920	0.989	0.955	**0.799**	0.010	0.010	0.018	**0.028**
MLPFF	0.907	0.999	0.795	**0.653**	0.019	0.002	0.067	**0.061**
SIREN	0.934	0.997	0.864	**0.752**	0.004	0.003	0.036	**0.036**
RBF	0.848	0.700	0.583	**0.292**	0.021	0.067	0.056	**0.022**
KRR	0.941	0.993	0.784	**0.690**	0.003	0.008	0.034	**0.036**
CNN	0.886	0.964	0.757	**0.573**	0.018	0.021	0.049	**0.051**

5.3.5 Analyse des résultats

Les résultats expérimentaux obtenus pour les trois niveaux de budget (faible, moyen, élevé) sont présentés dans les tableaux précédents, plusieurs enseignements se dégagent.

De manière globale, les approches génératives atteignent des performances comparables, voire supérieures, aux SOM classiques dès que le budget d'optimisation devient moyen ou élevé. L'augmentation du budget améliore significativement les scores des modèles neuronaux continus (MLP, MLP+Fourier, SIREN), tandis que les méthodes à base de noyaux (RBF, KRR) montrent une progression plus limitée.

La SOM classique fournit un point de référence robuste, en particulier à faible budget où elle reste compétitive. Cependant, dès le budget moyen (500 itérations), plusieurs générateurs (notamment MLP, SIREN et KRR) surpassent systématiquement la SOM sur le score global, confirmant l'intérêt du cadre génératif optimisé par CMA-ES.

— **MLP** : très régulier, il surpasse la SOM à budget moyen et élevé, grâce à sa flexibilité.

— **MLP+Fourier** : plus instable (scores modestes à budget faible), mais il bénéficie d'une nette amélioration à budget élevé, traduisant sa capacité à capturer des variations complexes avec un temps d'adaptation plus long.

— **SIREN** : montre une progression particulièrement robuste ; ses activations sinusoïdales favorisent la régularité du champ génératif, et à budget élevé il atteint un des meilleurs compromis QE/TE/H.

— **RBF** : reste systématiquement en retrait, avec un score global inférieur aux autres méthodes, traduisant une difficulté à capturer la structure globale au-delà d'une interpolation locale.

— **KRR** : bien plus compétitive, elle se rapproche du MLP et du SIREN à budget moyen et élevé, et constitue une alternative efficace grâce à son contrôle régularisé.

Le CoordConv CNN obtient des résultats modestes à faible et moyen budget.

En résumé :

— **À faible budget**, la SOM conserve un avantage relatif, tandis que seuls MLP et SIREN rivalisent efficacement.

— **À budget moyen**, MLP, SIREN et KRR surpassent systématiquement la SOM, confirmant la pertinence de la génération continue.

— **À budget élevé**, MLP et SIREN offrent les meilleures performances globales, tandis que les méthodes par noyaux plafonnent.

Ainsi, les fonctions génératives neuronales, optimisées par CMA-ES, apparaissent

comme des alternatives viables et compétitives aux SOM classiques, particulièrement lorsque l'on dispose d'un budget d'optimisation suffisant.

Chapitre 6

Application industrielle

6.1 Contexte industriel et besoin spécifique de l'entreprise

Octave, fondée en 1996 par Christophe et Michel Perrinet, est un éditeur de logiciels B2B spécialisé dans la digitalisation du commerce. Son ERP omnicanal relie en temps réel les opérations front et back-office (caisse, gestion commerciale et d'entrepôt, CRM, PIM, marketing, e-commerce, outils vendeurs), offrant une solution intégrée aux commerçants.

L'entreprise comptait au début du projet une centaine de collaborateurs répartis en cinq pôles (Commercial/Communication, Services généraux, Intégrateur, Éditeur, Infrastructure). Les pôles Éditeur et Intégrateur constituent le cœur du développement et du support de l'ERP.

Octave a régulièrement élargi ses solutions : première version en 1996, ajout du web en 2004, intégration gestion+web en 2007, puis offre omnicanale en 2013, qui lui a permis d'attirer des clients d'envergure comme Cash Piscine.

En 2019, l'entreprise crée un pôle IA dirigé par Christophe Courtois pour explorer l'intégration de l'IA dans ses offres. Un premier Proof of Concept avec Cash Piscine a démontré le potentiel de l'exploitation des données clients. Les résultats ont conduit au développement d'un produit dédié, Data Jubilation, dont une première version a été mise en ligne à l'été 2020.

6.1.1 Objectifs

Gestion des données

La première étape a consisté à restructurer la base SQL de Cash Piscine (250 Go, ∼1 000 tables), trop volumineuse et complexe pour une exploitation directe. Nous avons reconstruit une base allégée, conservant uniquement les informations pertinentes, et migré

vers Couchbase Server, choisi pour ses performances et ses capacités de recherche avancées (FTS).

Cette base a ensuite été enrichie par des données externes (open data) afin d'élargir les dimensions d'analyse :

— **Géocodage** des clients à partir de la BAN (géocodeur interne), permettant leur rattachement aux zones IRIS.
— **Météo** (Météo France, relevés 3h/60 stations), associée aux transactions pour étudier l'impact des conditions climatiques sur les ventes.
— **Foncier** (valeur des terrains sur 10 ans), pertinent pour les produits destinés aux propriétaires de piscines.
— **Âge estimé** par analyse statistique des prénoms, compensant l'absence fréquente de dates de naissance.

Cette phase a permis de transformer une base brute et hétérogène en un socle structuré, enrichi et adapté aux analyses ultérieures de notre produit *Data Jubilation*.

Analyse de données

L'objectif du pôle IA est de transformer les données en informations exploitables pour les équipes commerciales, via une interface simple de requêtes. Celle-ci permet d'accéder à différents modules analytiques, sans expertise technique particulière.

Le premier outil développé est un **tableau de bord interactif**, proposant notamment :

— l'évolution des ventes dans le temps,
— l'impact de la météo,
— des heatmaps par jour et heure,
— un arbre de décision et l'importance des variables.

Un module de **personas** génère des profils types (CA, nombre de commandes, préférences produits, âge estimé), offrant une vision synthétique des clients.

Deux méthodes de segmentation classiques sont intégrées :

— **ABC (Pareto)** : classement des clients selon leur contribution au CA ;
— **RFM** : segmentation par récence, fréquence et montant, avec mise en avant des articles dominants.

En complément, une segmentation avancée par **cartes de Kohonen (SOM)** permet d'analyser un grand nombre de variables simultanément et de révéler des comportements plus fins.

Enfin, les coordonnées GPS des clients permettent une **cartographie** des segmentations (ABC, RFM, SOM), visualisable à l'échelle nationale ou locale autour des points de vente.

Ces outils offrent une lecture claire des données clients et facilitent la définition de stratégies commerciales adaptées.

Visualisation

La visualisation a été un axe central du développement, afin de rendre les données accessibles à des utilisateurs non spécialistes. Destiné aux équipes commerciales, le produit devait transformer des données complexes en graphiques clairs et directement exploitables.

Les **tableaux de bord interactifs** offrent une vue d'ensemble des ventes, avec la possibilité de filtrer par période ou conditions météorologiques. La **segmentation SOM**, riche en variables, a nécessité une attention particulière aux échelles et couleurs pour faciliter l'identification des segments et tendances.

Les outils de **cartographie** (ABC, RFM, SOM) permettent de visualiser la répartition géographique des clients à l'échelle nationale ou locale, mettant en évidence les zones à fort potentiel. De leur côté, les **personas** sont présentés sous forme de représentations visuelles des comportements d'achat (CA, fréquence, produits typiques), facilitant leur interprétation. Enfin, la **segmentation RFM** est accompagnée de graphiques synthétiques mettant en avant les clients les plus actifs ou rentables.

Ainsi, la visualisation ne se limite pas à l'esthétique mais constitue un véritable outil d'aide à la décision, en rendant les analyses lisibles et immédiatement utiles pour orienter les actions commerciales.

Déploiement de l'application

La dernière étape consistait à déployer l'application sur un serveur Linux de l'entreprise, afin de la rendre accessible au client et de recueillir des retours d'usage indispensables pour l'amélioration continue du produit.

L'installation a été réalisée manuellement, sans droits d'administration, ce qui a nécessité la mise en place de Couchbase Server et de l'application (front et back-end) avec l'appui du pôle infrastructure. Ce déploiement a soulevé plusieurs défis : choix de la machine en fonction des ressources et coûts, contraintes de sécurité, optimisation des performances et réduction des transferts de données.

Enfin, la mise en production a imposé l'ajout de scripts de mise à jour quotidienne de la base, ainsi que l'intégration progressive des retours clients et des ajustements liés aux règles métiers spécifiques.

Conclusion

La phase de déploiement a marqué la transition de l'application vers un usage réel, validant l'outil auprès du client. Elle a mis en lumière divers défis techniques et pratiques (installation manuelle sans droits administratifs, mises à jour automatisées, sécurité et performance), qui ont enrichi l'expérience de l'équipe.

Cette étape a également permis de recueillir des retours utilisateurs précieux, favorisant l'adaptation des fonctionnalités et l'amélioration de l'ergonomie. Elle a ainsi posé les bases d'un processus d'évolution continue, garantissant une application opérationnelle, sécurisée et alignée sur les besoins du client.

6.1.2 Création et enrichissement des données

Extraction des données

Nous avons constitué une base centrée sur le client à partir de la base SQL d'origine. La première extraction a créé un identifiant unique (`IDFiche`) et filtré les enregistrements non pertinents (magasins, clients comptoir, fiches incomplètes), aboutissant à environ 1 million de clients.

La seconde extraction a relié les clients à leurs paniers via la table `DOSSIER`, en normalisant les dates et en sélectionnant uniquement les achats. Elle a produit environ 7 millions de lignes.

La troisième extraction a détaillé les articles (prix, poids, tags, classification hiérarchique) et les a reliés aux paniers (`IDFiche`, `IDDossier`, `IDLigne`), pour un total de 30 millions de lignes.

Un batch quotidien incrémental a ensuite été mis en place pour automatiser l'actualisation des clients, transactions et articles, garantissant la fraîcheur des données.

En résumé, ces trois requêtes et leur mise à jour automatique fournissent une base consolidée et évolutive, intégrant identité des clients, transactions et articles achetés, servant de socle aux analyses et segmentations ultérieures.

Déductions à partir des règles métiers

Les règles métiers de Cash Piscine ont permis de déduire certaines caractéristiques clients à partir des achats, notamment le type de piscine, de spa ou de filtration (ex. : présence du mot « CHLORES » ⇒ filtration au chlore). Ces inférences ont été appliquées en cohérence avec les filtres retenus lors de l'extraction des données.

En complément, plusieurs indicateurs génériques ont été calculés pour enrichir les profils : ancienneté, récence, intervalle moyen entre commandes, panier moyen, nombre de commandes et d'articles, ainsi que chiffre d'affaires total et moyen.

Ces déductions et calculs transforment les données brutes en informations directement exploitables, facilitant l'identification de tendances et la segmentation des clients pour adapter les stratégies commerciales.

Couchbase Server

Couchbase Server [1] est une base NoSQL orientée documents, stockant les données en JSON dans des *buckets*. Elle propose une interface d'administration (port 8091) ainsi que des outils en ligne de commande (ex. `cbimport`) et des SDKs pour de nombreux langages, dont Java.

Deux types de requêtes sont disponibles :

— **N1QL** : syntaxe proche de SQL, adaptée aux recherches structurées sur JSON.
— **FTS (Full Text Search)** : recherches textuelles avancées (tolérance aux fautes, casse) mais aussi sur nombres, dates ou coordonnées GPS, incluant la recherche géospatiale.

Le choix de Couchbase s'explique par :

— **Performance** : exécution rapide des requêtes N1QL.
— **Flexibilité** : modèle JSON permettant d'adapter la structure des données sans contrainte de schéma rigide.
— **Outils intégrés** : FTS natif, évitant l'ajout d'Elasticsearch, et support direct de la recherche géospatiale.

En résumé, Couchbase a permis de bâtir une base de données puissante, réactive et évolutive, parfaitement adaptée aux besoins du projet.

1. `https://www.couchbase.com/`

Importation

Les données extraites au format CSV ont été importées dans Couchbase via un programme Java (Maven, SDK Couchbase). Celui-ci regroupe les informations par client et crée des documents JSON identifiés par `idOctave` (correspondant à `IDFiche`).

L'import initial, long en raison du volume, a été complété par un processus de mise à jour quotidienne incrémentale, qui n'intègre que les nouveaux enregistrements. Cette stratégie maintient la base synchronisée avec SQL tout en réduisant les temps de traitement.

Géocodage

Les adresses clients ne comportant pas de coordonnées GPS, un géocodeur interne a été développé afin de permettre les analyses cartographiques, l'enrichissement météo et l'association aux zones IRIS.

Plutôt qu'une API externe (coût élevé ou limitations), nous avons choisi d'importer la **Base Adresse Nationale (BAN)** dans Couchbase (bucket `geo`), en format « par voie » pour réduire le volume. Les champs code postal, commune et rue ont été indexés via FTS, permettant des recherches efficaces.

Un module Java (SDK Couchbase) interroge cette base : pour chaque adresse (CP, ville, voie), la meilleure correspondance FTS est retenue, puis les coordonnées GPS extraites et ajoutées au profil client.

Une vérification empirique (comparaison avec Google Maps) a montré une bonne précision, les erreurs restant rares et limitées (quelques km). La base BAN est mise à jour annuellement pour garantir l'actualité des données.

Données foncières

Une seconde source externe exploitée concerne la valeur foncière, pertinente pour Cash Piscine. Les données, disponibles en open data (Data.gouv) et mises à jour annuellement, couvrent les transactions des cinq dernières années.

Elles ont été importées dans Couchbase (bucket `foncier`, $\sim$2,5M documents) avec les principaux champs : localisation (département, code postal, commune, voie, numéro), type de local, surfaces et valeur foncière. Un index FTS hiérarchique (CP $\rightarrow$ commune $\rightarrow$ voie $\rightarrow$ numéro) facilite les recherches.

Ces informations permettent d'ajouter un contexte socio-économique aux profils clients et d'explorer des corrélations avec l'intérêt pour les produits piscines. Toutefois, la couverture reste partielle (non exhaustivité des adresses) et les correspondances doivent être très strictes. Pour limiter les erreurs, un programme de validation stricte a été mis en place : en cas de doute, la valeur foncière n'est pas renseignée.

La précision n'a pas été évaluée de manière statistique mais vérifiée empiriquement sur un échantillon, avec des résultats satisfaisants. Cette source offre donc un enrichissement utile mais demande des mises à jour et contrôles réguliers pour garantir sa fiabilité.

Données météorologiques

La troisième source externe concerne la météorologie. Les données ouvertes de Météo France (60 stations, relevés toutes les 3h) ont été intégrées dans un bucket `meteo` de Couchbase, chaque enregistrement incluant date, station et principales variables (pression, vent, humidité, température, précipitations, visibilité). Un index FTS (date, station) permet des requêtes rapides.

Chaque client est associé à la **station la plus proche**, déterminée par la distance sphérique calculée à partir des coordonnées GPS. Pour chaque achat, les conditions météo sont recherchées dans un intervalle de ±1h30 autour de l'heure de transaction.

Bien que ce module nécessite environ 7 millions de requêtes (tous les achats), des optimisations permettent une exécution en quelques heures. L'intégration des données météo fournit ainsi un contexte supplémentaire pour analyser l'impact des conditions climatiques sur les ventes, particulièrement pertinent pour Cash Piscine.

Estimation de l'âge des clients

La base d'Octave comportant rarement les dates de naissance, nous avons mis en place une estimation de l'âge à partir des prénoms. Pour cela, une base externe indiquant la répartition des naissances par prénom et par année a permis de calculer un âge médian. Par exemple, le prénom « Axel » présente un pic autour de 2000, ce qui suggère un âge probable d'une vingtaine d'années.

Chaque prénom a été associé à un sexe et à un âge médian, puis importé dans Couchbase via `cbimport`. Une indexation FTS a permis des recherches rapides par prénom et sexe, avec tolérance aux fautes d'orthographe.

Ce module enrichit les profils clients par une estimation démographique fiable, utile pour affiner la segmentation et analyser les comportements d'achat selon les groupes d'âge.

6.2 Developpement de l'application

6.2.1 Interface de Sélection des Clients

Accueil et Authentification (Voir annexe A.4.1)

L'accès à l'application Data Jubilation se fait simplement via un navigateur web en se rendant à l'URL fournie. Cette interface est le point d'entrée de l'application, et son accès nécessite une connexion internet. Une fois sur la page d'accueil, l'utilisateur est accueilli par le logo de l'application, qui représente le lien avec la marque, ainsi que des boutons principaux permettant d'accéder aux fonctionnalités d'analyse des clients et des articles achetés. Ces boutons, si l'utilisateur n'est pas encore connecté, ouvrent automatiquement le formulaire d'authentification.

Navigation dans l'interface de sélection (Voir annexe A.4.2)

L'interface de Data Jubilation permet de filtrer les clients selon cinq catégories : **profil**, **achats**, **géographie**, **temps**, et **météo**. Une barre de navigation offre les fonctions principales : retour à l'accueil, réinitialisation des filtres et soumission de la recherche.

Les filtres de **profil** incluent prénom/nom, âge estimé (via prénoms), revenu médian et segments ABC/RFM. Les filtres liés aux **achats** permettent de cibler types de produits, quantités, panier moyen, ou caractéristiques déduites (type de piscine, filtration). La dimension **géographique** autorise des sélections par région, département, ville, zone IRIS, ou par dessin de zones sur carte interactive. Les critères **temporels** portent sur périodes, jours et plages horaires d'achat. Enfin, les filtres **météorologiques** exploitent les données enrichies (température, précipitations, vent, pression).

Cette interface, à la fois intuitive et complète, constitue le cœur de l'application : elle permet une segmentation fine et personnalisée, intégrant à la fois données clients, contexte géographique et conditions environnementales.

Validation de la sélection et accès aux blocs fonctionnels (Voir annexe A.4.3)

Après définition des critères, Data Jubilation affiche un résumé des filtres appliqués afin de permettre leur validation ou ajustement. L'utilisateur accède ensuite aux cinq blocs principaux de l'application :

— **Tableau de bord** : tendances et statistiques globales.

— **SOM** : segmentation avancée via cartes auto-organisées.

— **ABC** : classement des clients selon leur contribution au chiffre d'affaires.
— **RFM** : analyse récence, fréquence, montant pour évaluer la fidélité.
— **Persona** : génération de profils types par segment.

Ces blocs offrent des perspectives complémentaires pour analyser les comportements clients selon différents niveaux de détail.

Tableau de bord (Voir annexe A.4.4)

Premier module de Data Jubilation, le Tableau de Bord fournit une vue d'ensemble des principales métriques commerciales. Il affiche des indicateurs clés (CA, nombre de commandes, articles, clients concernés) ajustés selon les critères de sélection, ainsi que des visualisations interactives.

L'évolution des ventes est analysée par **période** (années, mois, jours, heures via heatmaps), permettant de repérer cycles saisonniers et pics d'activité. Un **Top 10 produits** (quantité et CA) et le suivi de leur évolution dans le temps aident à identifier les articles phares et anticiper la demande.

La dimension **géographique** inclut une heatmap clients et un CA par région/zone IRIS, offrant une vision claire des zones à fort potentiel. Une analyse par **enseignes** complète l'approche, avec répartition du CA et ventes par jour/produit.

L'intégration des **données météo** (température, précipitations, vent, pression) permet d'évaluer l'impact du climat sur les achats, utile pour ajuster stocks et communication.

Enfin, un outil de **tableaux croisés dynamiques** permet de combiner librement produits, segments, périodes ou enseignes, avec export CSV/PNG pour intégration dans des rapports.

En somme, le Tableau de Bord offre une analyse multidimensionnelle (temps, produits, géographie, enseignes, météo) et constitue un outil central pour comprendre les comportements clients et orienter les décisions stratégiques.

Persona (Voir annexe A.4.5)

Le module Persona génère un « client type » à partir d'un groupe sélectionné, offrant une vision synthétique des principales caractéristiques du segment. Cet outil aide les équipes commerciales et marketing à comprendre les comportements et préférences d'un groupe pour adapter communication et offres.

Le Persona repose sur un profil statistique intégrant sexe, âge estimé, chiffre d'affaires moyen, type de piscine ou de filtration, etc. Chaque attribut est affiché visuellement

(barres de progression, valeurs comparées à l'ensemble de la base), permettant d'identifier les spécificités du segment, comme un panier moyen plus élevé ou une fréquence d'achat particulière.

Un **Top 10 produits** complète l'analyse, listant les articles les plus achetés en quantité et en chiffre d'affaires. Ces informations mettent en évidence les préférences dominantes et ouvrent des opportunités de *cross-selling* ou d'*up-selling*.

En combinant données démographiques, comportements d'achat et visualisations claires, le module Persona constitue un outil central pour affiner la segmentation et orienter les stratégies de fidélisation et de communication.

Segmentation ABC (Voir annexe A.4.6)

La segmentation ABC de Data Jubilation repose sur la loi de Pareto : environ 20% des clients (**A**) génèrent 80% du chiffre d'affaires, tandis que les segments **B** et **C** regroupent des clients à contribution décroissante mais essentiels pour la fidélisation et les ventes de masse. Cette classification aide à adapter les efforts : suivi personnalisé pour A, marketing ciblé pour B, actions de masse pour C.

Une cartographie interactive permet de visualiser la répartition des segments par **département** ou par **zones IRIS**, identifiant les zones à fort potentiel commercial. Pour chaque segment, un **Top 10 produits** (quantité et chiffre d'affaires) met en évidence les préférences d'achat, différenciant par exemple l'intérêt des clients A pour des articles premium.

Les résultats sont exportables en **CSV/JSON** pour analyses externes, et un tableau récapitulatif présente la répartition des clients par segment. Ces outils offrent une base solide pour cibler les actions marketing, optimiser les ressources et orienter les stratégies commerciales en fonction de la valeur des clients.

Segmentation RFM (Voir annexe A.4.7)

La segmentation RFM (Récence, Fréquence, Montant) classe les clients selon la date de leur dernier achat, la régularité de leurs transactions et leur contribution financière. Couramment utilisée en marketing, elle permet d'adapter les stratégies de fidélisation et de communication.

L'application distingue plusieurs segments types :

— **Champions** : récents, fréquents et dépensiers, prioritaires pour la fidélisation.

— **Potentiels** : forte dépense mais faible récence ou fréquence, à réactiver.

— **En sommeil** : faible activité et valeur, pouvant être relancés ou écartés.

Les résultats sont présentés via une **carte interactive** (départements et zones IRIS), un **tableau de synthèse** (effectifs, CA cumulé, indicateurs clés) et un **Top 10 produits** par segment. Des exports en CSV/JSON permettent l'intégration dans d'autres outils de reporting ou CRM.

Des **visualisations avancées** (graphiques et heatmaps) montrent la répartition des segments, leur évolution temporelle et leur distribution géographique ou horaire. Ces analyses aident à détecter des tendances (saisonnalité, zones à potentiel, comportements d'achat spécifiques).

Ainsi, la segmentation RFM fournit une vision fine des comportements clients et constitue un levier central pour cibler efficacement les actions marketing et optimiser la gestion de la relation client.

Segmentation SOM (Voir annexe A.4.8)

Le module SOM (Self-Organizing Maps) de Data Jubilation applique l'algorithme de Kohonen pour regrouper les clients en clusters selon leurs caractéristiques communes. Chaque hexagone de la carte représente un cluster, coloré selon une variable (âge, CA, fréquence d'achat, etc.), offrant une visualisation intuitive des comportements.

L'utilisateur peut explorer un cluster en détail : profil statistique, comparaison avec la base globale et **Top 10 produits** (quantité ou chiffre d'affaires), révélant les préférences dominantes et ouvrant des opportunités de ciblage marketing ou de gestion de stock.

Une **cartographie interactive** projette les clusters sur la France (par département ou zone IRIS), identifiant les zones géographiques où se concentrent certains comportements d'achat.

Deux outils complémentaires enrichissent l'analyse :

— **Dendrogramme** : montre les proximités hiérarchiques entre clusters, facilitant l'identification de segments voisins aux comportements similaires.

— **Arbre de décision** : met en évidence les variables discriminantes (âge, CA, produits achetés, conditions météo, etc.) qui distinguent les clusters.

Ce module offre ainsi une segmentation avancée et multidimensionnelle, combinant visualisations et outils analytiques pour affiner la compréhension des profils clients et optimiser les stratégies marketing et commerciales.

Conclusion

Cette thèse s'est attachée à revisiter les cartes auto-organisatrices (SOM) sous trois angles complémentaires : (i) la **gestion de la donnée** en amont de l'apprentissage, (ii) l'**évaluation et l'hyperparamétrage** fondés sur des scores explicites et opérationnels, (iii) la **refonte du paradigme SOM** via des générateurs continus optimisés sans gradient, puis (iv) la **validation industrielle** au travers d'un déploiement réel à grande échelle. Les contributions principales peuvent être résumées comme suit.

- **Gestion de données hétérogènes.** Nous avons formalisé une adaptation simple et efficace du calcul de distance pour les données manquantes, en ignorant les dimensions absentes lors de l'attribution BMU, ce qui permet d'entraîner une SOM sans imputation systématique ni exclusion d'observations. Nous avons également détaillé des schémas d'encodage adaptés aux *données qualitatives* (one-hot, réduction dimensionnelle), *textuelles* (TF-IDF, embeddings Word2Vec/BERT) et *temporelles* (fenêtrage, représentation par tranches, réduction), afin de rendre la SOM opérante sur des jeux réels, riches et imparfaits.
- **Évaluation et hyperparamétrage.** Nous avons proposé et unifié un ensemble de métriques $S_1 \dots S_8$ (erreur de quantification, distorsion, topographie, produit topographique, Kruskal-Shepard, C-mesure, fonction topographique), puis une **normalisation** dans $[0, 1]$ pour faciliter la comparaison inter-configurations. Nous avons introduit un **score combiné** $S_{\Delta t}(S'1)$ intégrant précision et temps de calcul, utile pour piloter le compromis qualité/efficience et suggérer un **critère d'arrêt** pragmatique. Enfin, l'usage de **SMAC** a montré l'intérêt d'une configuration bayésienne des hyperparamètres (taille de carte, $Tmax$, ϵ_0, voisinage), supérieure à des réglages uniformes ou aléatoires et plus stable d'une exécution à l'autre.
- **Analyse systématique de T_{max} et ϵ_0.** Les campagnes expérimentales 2D et 3D confirment des tendances robustes : des ϵ_0 faibles nécessitent davantage d'itérations pour un gain marginal ; au-delà d'un seuil, l'augmentation de T_{max} n'améliore plus significativement S'_1. Le score combiné met en évidence un *point d'équilibre* où la progression de $S'1$ cesse de compenser le coût calculatoire.
- **SOM génératives continues.** Nous avons reformulé la SOM comme l'échantillon-

nage d'un *champ continu* $f\theta : [0,1]^2 \to \mathbb{R}^d$ (MLP, MLP+Fourier, SIREN, RBF, KRR, CNN CoordConv) dont les paramètres sont optimisés par **CMA-ES** sur des scores non différentiables (quantification, topologie, entropie d'usage et score global). À budget moyen/élevé, les **MLP** et **SIREN** surpassent systématiquement la SOM classique selon notre score agrégé, tandis que **KRR** offre une alternative régularisée compétitive. Cette voie ouvre des perspectives de *cartes lisses*, *multi-résolution* et *faiblement paramétrées*.

— **Validation industrielle.** Le projet *Data Jubilation* chez Octave a fourni un terrain d'expérimentation complet : reconstruction d'une base client-centrée (SQL → Couchbase), **géocodage** autonome (BAN), **météo** (relevés 3h, appariement spatio-temporel), **foncier**, **estimation d'âge** par prénoms, intégration dans des **tableaux de bord** et des **segmentations** (ABC, RFM, SOM) cartographiées. Le déploiement a confirmé la *faisabilité opérationnelle* et l'*utilité décisionnelle* des méthodes proposées.

Au-delà des résultats quantitatifs, trois enseignements clés se dégagent.

1. **La donnée prime.** Le traitement frugal des valeurs manquantes par distance partielle, l'alignement d'échelles et les encodages adaptés sont des leviers au moins aussi déterminants que le choix final d'algorithme. Une SOM robuste commence par une *ingénierie des données* robuste.
2. **Mesurer pour piloter.** La diversité des scores est une force si l'on explicite leur rôle (finesse locale, cohésion topologique, usage de la carte) et si l'on propose une *aggrégation parcimonieuse* — ici, le couplage (QE, TE, entropie) et l'intégration du temps de calcul pour guider l'arrêt.
3. **Au-delà de la SOM « classique ».** Les champs génératifs optimisés sans gradient montrent qu'il est possible de *décorréler* la qualité de la carte des contraintes de l'apprentissage compétitif historique, tout en améliorant stabilité, compacité et expressivité.

Nos choix comportent des limites assumées.

— **Scores et pondérations.** Bien que normalisés, les scores restent sensibles aux pondérations implicites (p.,ex. QE au carré). D'autres agrégations pourraient mieux capter des objectifs métier spécifiques (équité inter-segments, stabilité inter-runs, sobriété carbone).

— **Coût calculatoire.** L'estimation répétée de scores non différentiables et l'optimisation évolutionnaire ont un coût. Nous l'avons atténué via le critère $S_{\Delta t}$, mais des

surrogates ou des *évaluations partielles* (mini-batches, sous-échantillonnage de la carte) méritent d'être systématisés.

— **Générateurs et généralisabilité.** Les générateurs continus dépassent la SOM standard à budget suffisant, mais restent sensibles à l'architecture (profondeur, fréquences Fourier, w_0 des SIREN) et à l'initialisation. Des garanties théoriques (préservation topologique, bornes d'erreur) restent à établir.

— **Données externes.** Géocodage, météo, foncier et âge estimé apportent une forte valeur ajoutée, au prix d'incertitudes et de mises à jour régulières ; une *quantification explicite de l'incertitude* n'a pas été intégrée aux scores.

Plusieurs prolongements apparaissent naturels.

— **Optimisation hybride.** Coupler CMA-ES avec des étapes de *descente de gradient* sur des *surrogates différentiables* des scores (p.,ex. relaxations continues de la topologie), ou des approches *multi-fidélité*.

— **Critères d'arrêt auto-adaptatifs.** Apprendre une politique d'arrêt (bandits, détection de plateau robuste) guidée par $S_{\Delta t}$ et des indicateurs de *courbure* de la trajectoire d'apprentissage.

— **Topologies généralisées.** Étendre f_θ à des *variétés* (sphère, tore) et des *grilles irrégulières*, voire à des *graphes* (SOM sur graphe) pour mieux épouser la géométrie des données.

— **SOM en flux.** Développer une variante *online/streaming* des générateurs continus, avec *réapprentissage local* et *oubli contrôlé* pour les cas industriels à données arrivantes.

— **Incertitude et robustesse.** Intégrer des *bornes de confiance* (bootstrap, ensembles), des *perturbations adversariales* légères et des *scores robustes* aux outliers.

— **A/B tests et KPIs métiers.** Relier les segments SOM aux indicateurs d'impact (taux d'ouverture/clic, panier moyen, réachat) via des expérimentations contrôlées.

— **Confidentialité et gouvernance.** Renforcer la *privacy by design* (pseudonymisation, agrégation différenciée, minimisation) et documenter la *traçabilité des enrichissements* (météo, foncier, âge) dans une optique RGPD.

— **Packaging et ouverture.** Verser un *benchmark reproductible* (scripts, configurations, gabarits de données synthétiques) et un *prototype libre* des générateurs SOM-CMA-ES pour la communauté.

Chez Octave, les briques présentées — pipeline client-centré sous Couchbase, enrichissements géo-météo-foncier-âge, tableaux de bord, segments ABC/RFM/SOM, personas,

cartographies — constituent un **socle produit** désormais éprouvé. Elles ont montré leur capacité à :

1. **accélérer l'analyse** (requêtage réactif, vues synthétiques multi-axes),
2. **fournir des segmentations actionnables** (SOM et personas liés aux assortiments et campagnes),
3. **s'industrialiser** (mises à jour incrémentales, déploiement contraint, sécurité).

Au-delà du cas Cash Piscine, l'architecture et les méthodes se généralisent à d'autres verticaux du retail, avec une promesse claire : *des cartes plus justes, plus lisibles, et moins coûteuses à obtenir*.

En replaçant la SOM au cœur d'un triptyque *données soignées — scores explicites — optimisation moderne*, cette thèse propose un itinéraire complet, de la théorie aux usages. Elle montre qu'il est possible de *concilier rigueur et pragmatisme* : traiter la donnée telle qu'elle est, mesurer ce qui compte vraiment, et apprendre des cartes non plus seulement en les ajustant neurone par neurone, mais en *générant des champs* qui les incarnent. À ce titre, les SOM génératives optimisées par CMA-ES constituent une alternative crédible aux SOM historiques, et une base solide pour les prochaines générations de cartes auto-adaptatives, plus *flexibles*, *scalables* et *orientées impact*.

BIBLIOGRAPHIE

[1] Francisco de A. T. DE CARVALHO et al., « Batch Self-Organizing Maps for Distributional Data with an Automatic Weighting of Variables and Components », in : *J. Classif.* 39.*2* (2022), p. 343-375, DOI : 10.1007/S00357-022-09411-1, URL : https://doi.org/10.1007/s00357-022-09411-1.

[2] Reham Fathy M. AHMED, Cherif R. SALAMA et Hani MAHDI, « Optimizing Self-Organizing Maps Parameters Using Genetic Algorithm : A Simple Case Study », in : *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2019, AISI 2019, Cairo, Egypt, 26-28 October 2019*, sous la dir. d'Aboul Ella HASSANIEN, Khaled SHAALAN et Mohamed Fahmy TOLBA, t. 1058, Advances in Intelligent Systems and Computing, Springer, 2019, p. 3-12, DOI : 10.1007/978-3-030-31129-2_1, URL : https://doi.org/10.1007/978-3-030-31129-2%5C_1.

[3] Amin ALLAHYAR, Hadi Sadoghi YAZDI et Ahad HARATI, « Constrained Semi-Supervised Growing Self-Organizing Map », in : *Neurocomputing* 147 (2015), p. 456-471, DOI : 10.1016/J.NEUCOM.2014.06.039, URL : https://doi.org/10.1016/j.neucom.2014.06.039.

[4] Victor Oliveira ANTONINO et Aluizio F. R. ARAÚJO, « Local adaptive receptive field dimension selective self-organizing map for multi-view clustering », in : *2016 International Joint Conference on Neural Networks, IJCNN 2016, Vancouver, BC, Canada, July 24-29, 2016*, IEEE, 2016, p. 698-705, DOI : 10.1109/IJCNN.2016.7727268, URL : https://doi.org/10.1109/IJCNN.2016.7727268.

[5] César A. ASTUDILLO et B. John OOMMEN, « Self Organizing Maps Whose Topologies Can Be Learned With Adaptive Binary Search Trees Using Conditional Rotations », in : *CoRR* abs/1506.02750 (2015), arXiv : 1506.02750, URL : http://arxiv.org/abs/1506.02750.

[6] Hans-Ulrich BAUER, Klaus PAWELZIK et Theo GEISEL, « A Topographic Product for the Optimization of Self-Organizing Feature Maps », in : *Advances in Neural Information Processing Systems*, sous la dir. de J. MOODY, S. HANSON et R.P.

Lippmann, t. 4, Morgan-Kaufmann, 1991, URL : https://proceedings.neurips.cc/paper_files/paper/1991/file/389bc7bb1e1c2a5e7e147703232a88f6-Paper.pdf.

[7] Yann Bernard, Nicolas Hueber et Bernard Girau, « A Fast Algorithm to Find Best Matching Units in Self-Organizing Maps », in : *Artificial Neural Networks and Machine Learning - ICANN 2020 - 29th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 15-18, 2020, Proceedings, Part II*, sous la dir. d'Igor Farkas, Paolo Masulli et Stefan Wermter, t. 12397, Lecture Notes in Computer Science, Springer, 2020, p. 825-837, DOI : 10.1007/978-3-030-61616-8_66, URL : https://doi.org/10.1007/978-3-030-61616-8%5C_66.

[8] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Ouvrage de référence sur les méthodes probabilistes et de codage des variables qualitatives., New York : Springer, 2006, ISBN : 978-0-387-31073-2.

[9] Pedro H. M. Braga et Hansenclever de F. Bassani, « A Semi-Supervised Self-Organizing Map for Clustering and Classification », in : *2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, July 8-13, 2018*, IEEE, 2018, p. 1-8, DOI : 10.1109/IJCNN.2018.8489675, URL : https://doi.org/10.1109/IJCNN.2018.8489675.

[10] Guénaël Cabanes et al., *On the use of Wasserstein metric in topological clustering of distributional data*, 2021, arXiv : 2109.04301 [cs.LG], URL : https://arxiv.org/abs/2109.04301.

[11] Gian Carlo Cardarilli et al., « AW-SOM, an Algorithm for High-Speed Learning in Hardware Self-Organizing Maps », in : *IEEE Trans. Circuits Syst. II Express Briefs* 67-II.*2* (2020), p. 380-384, DOI : 10.1109/TCSII.2019.2909117, URL : https://doi.org/10.1109/TCSII.2019.2909117.

[12] Dorota Celinska-Kopczynska et Eryk Kopczynski, « Non-Euclidean Self-Organizing Maps », in : *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, sous la dir. de Luc De Raedt, ijcai.org, 2022, p. 1938-1944, DOI : 10.24963/IJCAI.2022/269, URL : https://doi.org/10.24963/ijcai.2022/269.

[13] Rakesh Chalasani et José C. Príncipe, « Self-organizing maps with information theoretic learning », in : *Neurocomputing* 147 (2015), p. 3-14, doi : 10.1016/J.NEUCOM.2013.12.059, url : https://doi.org/10.1016/j.neucom.2013.12.059.

[14] Vincent Chappelier et M. Grumbach, « Text Clustering with Self-Organizing Maps : Application to Time-Series Data », in : *Proceedings of the International Workshop on Neural Networks for Signal Processing*, Utilisation des SOM pour le traitement et le regroupement de séquences temporelles., 1999, p. 565-574.

[15] Vikas Chaudhary, R. S. Bhatia et Anil K. Ahlawat, « A Constant Learning Rate Self-Organizing Map (CLRSOM) Learning Algorithm », in : *J. Inf. Sci. Eng.* 31.*2* (2015), p. 387-397, url : http://www.iis.sinica.edu.tw/page/jise/2014/2015/201503%5C_02.html.

[16] Vikas Chaudhary, R. S. Bhatia et Anil K. Ahlawat, « Community SOM (CSOM) : An Improved Self-Organizing Map Learning Technique », in : *Int. J. Fuzzy Syst.* 17.*2* (2015), p. 129-132, doi : 10.1007/S40815-015-0022-7, url : https://doi.org/10.1007/s40815-015-0022-7.

[17] Jingchen Cong et al., « Conceptual design of a user-centric smart product-service system using self-organizing map », in : *Adv. Eng. Informatics* 55 (2023), p. 101857, doi : 10.1016/J.AEI.2022.101857, url : https://doi.org/10.1016/j.aei.2022.101857.

[18] Carmelo del Coso et al., « Mixing numerical and categorical data in a Self-Organizing Map by means of frequency neurons », in : *Appl. Soft Comput.* 36 (2015), p. 246-254, doi : 10.1016/J.ASOC.2015.06.058, url : https://doi.org/10.1016/j.asoc.2015.06.058.

[19] Marie Cottrell et al., « Theoretical and Applied Aspects of the Self-Organizing Maps », in : *Advances in Self-Organizing Maps and Learning Vector Quantization - Proceedings of the 11th International Workshop WSOM 2016, Houston, Texas, USA, January 6-8, 2016*, sous la dir. d'Erzsébet Merényi, Michael J. Mendenhall et Patrick O'Driscoll, t. 428, Advances in Intelligent Systems and Computing, Springer, 2016, p. 3-26, doi : 10.1007/978-3-319-28518-4_1, url : https://doi.org/10.1007/978-3-319-28518-4%5C_1.

[20] Pierpaolo D'Urso, Livia De Giovanni et Riccardo Massari, « Smoothed self-organizing map for robust clustering », in : *Inf. Sci.* 512 (2020), p. 381-401, DOI : 10.1016/J.INS.2019.06.038, URL : https://doi.org/10.1016/j.ins.2019.06.038.

[21] Qun Dai et Lin Guo, « Two novel hybrid Self-Organizing Map based emotional learning algorithms », in : *Neural Comput. Appl.* 31.*7* (2019), p. 2921-2938, DOI : 10.1007/S00521-017-3240-0, URL : https://doi.org/10.1007/s00521-017-3240-0.

[22] Jacob Devlin et al., « BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding », in : *Proceedings of NAACL-HLT*, Modèle de référence pour les représentations contextuelles de texte., 2019, p. 4171-4186.

[23] Georgios Drakopoulos et al., « On Tensor Distances for Self Organizing Maps : Clustering Cognitive Tasks », in : *Database and Expert Systems Applications - 31st International Conference, DEXA 2020, Bratislava, Slovakia, September 14-17, 2020, Proceedings, Part II*, sous la dir. de Sven Hartmann et al., t. 12392, Lecture Notes in Computer Science, Springer, 2020, p. 195-210, DOI : 10.1007/978-3-030-59051-2_13, URL : https://doi.org/10.1007/978-3-030-59051-2%5C_13.

[24] Haytham Elghazel et Khalid Benabdeslem, « Different Aspects of Clustering The Self-Organizing Maps », in : *Neural Process. Lett.* 39.*1* (2014), p. 97-114, DOI : 10.1007/S11063-013-9292-Y, URL : https://doi.org/10.1007/s11063-013-9292-y.

[25] Paulo H. M. Ferreira et Aluizio F. R. Araújo, « Growing Self-Organizing Maps for Nonlinear Time-Varying Function Approximation », in : *Neural Process. Lett.* 51.*2* (2020), p. 1689-1714, DOI : 10.1007/S11063-019-10168-9, URL : https://doi.org/10.1007/s11063-019-10168-9.

[26] Florent Forest et al., « A Survey and Implementation of Performance Metrics for Self-Organized Maps », in : *CoRR* abs/2011.05847 (2020), arXiv : 2011.05847, URL : https://arxiv.org/abs/2011.05847.

[27] Bernard Girau et César Torres-Huitzil, « Fault tolerance of self-organizing maps », in : *Neural Comput. Appl.* 32.*24* (2020), p. 17977-17993, DOI : 10.1007/S00521-018-3769-6, URL : https://doi.org/10.1007/s00521-018-3769-6.

[28] Geoffrey Goodhill et Terrence Sejnowski, « Quantifying Neighbourhood Preservation in Topographic Mappings », in : *Procceeding of 3rd Joint Symposium Neurol Comp* Vol. 6 (oct. 1996).

[29] Marian B. Gorzalczany et Filip Rudzinski, « Generalized Self-Organizing Maps for Automatic Determination of the Number of Clusters and Their Multiprototypes in Cluster Analysis », in : *IEEE Trans. Neural Networks Learn. Syst.* 29.7 (2018), p. 2833-2845, doi : 10.1109/TNNLS.2017.2704779, url : https://doi.org/10.1109/TNNLS.2017.2704779.

[30] Rodrigo César Teixeira de Gouvêa et al., « IntraSOM : A comprehensive Python library for Self-Organizing Maps with hexagonal toroidal maps training and missing data handling », in : *Softw. Impacts* 17 (2023), p. 100570, doi : 10.1016/J.SIMPA.2023.100570, url : https://doi.org/10.1016/j.simpa.2023.100570.

[31] Axel Guérin, Pierre Chauvet et Frédéric Saubion, « A Survey on Recent Advances in Self-Organizing Maps », in : *arXiv preprint arXiv :2501.08416* (2024).

[32] Axel Guérin, Pierre Chauvet et Frédéric Saubion, « Hyperparameter Tuning for Self-Organizing Maps », in : *2024 Conference on AI, Science, Engineering, and Technology (AIxSET)*, 2024, p. 228-235, doi : 10.1109/AIxSET62544.2024.00044.

[33] Lutz Hamel, « VSOM : Efficient, Stochastic Self-organizing Map Training », in : *Intelligent Systems and Applications - Proceedings of the 2018 Intelligent Systems Conference, IntelliSys 2018, London, UK, September 6-7, 2018, Volume 2*, sous la dir. de Kohei Arai, Supriya Kapoor et Rahul Bhatia, t. 869, Advances in Intelligent Systems and Computing, Springer, 2018, p. 805-821, doi : 10.1007/978-3-030-01057-7_60, url : https://doi.org/10.1007/978-3-030-01057-7%5C_60.

[34] Nikolaus Hansen, *The CMA Evolution Strategy : A Tutorial*, 2023, arXiv : 1604.00772 [cs.LG], url : https://arxiv.org/abs/1604.00772.

[35] Nikolaus Hansen et Andreas Ostermeier, « Completely Derandomized Self-Adaptation in Evolution Strategies », in : *Evolutionary Computation* 9.2 (juin 2001), p. 159-195, issn : 1063-6560, doi : 10.1162/106365601750190398, eprint : https://direct.mit.edu/evco/article-pdf/9/2/159/1493523/106365601750190398.pdf, url : https://doi.org/10.1162/106365601750190398.

[36] Nikolaus Hansen et Andreas Ostermeier, « Completely Derandomized Self-Adaptation in Evolution Strategies », in : *Evolutionary Computation* 9 (juin 2001), p. 159-195, doi : 10.1162/106365601750190398.

[37] Zellig S. Harris, « Distributional Structure », in : *Word* 10.*2-3* (1954), Travail fondateur sur la représentation catégorielle et la base du one-hot encoding., p. 146-162, doi : 10.1080/00437956.1954.11659520.

[38] Simon Haykin, *Neural networks : a comprehensive foundation*, Prentice Hall PTR, 1994.

[39] Tom Heskes, « Energy Functions for Self-Organizing Maps », in : *Proceedings of the 5th International Conference on Neural Information Processing (ICONIP'98)*, Tokyo, Japan, 1999, p. 303-308.

[40] Hiroomi Hikawa, Hidetaka Ito et Yutaka Maeda, « A New Self-Organizing Map with Continuous Learning Capability », in : *IEEE Symposium Series on Computational Intelligence, SSCI 2018, Bangalore, India, November 18-21, 2018*, IEEE, 2018, p. 2163-2168, doi : 10.1109/SSCI.2018.8628891, url : https://doi.org/10.1109/SSCI.2018.8628891.

[41] Hiroomi Hikawa et Yutaka Maeda, « Improved Learning Performance of Hardware Self-Organizing Map Using a Novel Neighborhood Function », in : *IEEE Trans. Neural Networks Learn. Syst.* 26.*11* (2015), p. 2861-2873, doi : 10.1109/TNNLS.2015.2398932, url : https://doi.org/10.1109/TNNLS.2015.2398932.

[42] Holger H. Hoos, « Automated Algorithm Configuration and Parameter Tuning », in : *Autonomous Search*, sous la dir. d'Youssef Hamadi, Éric Monfroy et Frédéric Saubion, Springer, 2012, p. 37-71, doi : 10.1007/978-3-642-21434-9_3, url : https://doi.org/10.1007/978-3-642-21434-9%5C_3.

[43] Chung-Chian Hsu et al., « Unsupervised distance learning for extended self-organizing map and visualization of mixed-type data », in : *Intell. Data Anal.* 23.*4* (2019), p. 799-823, doi : 10.3233/IDA-183930, url : https://doi.org/10.3233/IDA-183930.

[44] Di-Wei Huang, Rodolphe J. Gentili et James A. Reggia, « Self-organizing maps based on limit cycle attractors », in : *Neural Networks* 63 (2015), p. 208-222, doi : 10.1016/J.NEUNET.2014.12.003, url : https://doi.org/10.1016/j.neunet.2014.12.003.

[45] Frank Hutter, Lars Kotthoff et Joaquin Vanschoren, éd., *Automated Machine Learning - Methods, Systems, Challenges*, Springer, 2019.

[46] Macario O. Cordel II et Arnulfo P. Azcarraga, « A New Method for Emulating Self-Organizing Maps for Visualization of Datasets », in : *Int. J. Comput. Intell. Appl.* 17.*3* (2018), 1850014 :1-1850014 :15, doi : 10.1142/S1469026818500141, url : https://doi.org/10.1142/S1469026818500141.

[47] Rubén Jaramillo-Vacio, Alberto Ochoa-Zezzatti et Fernando Figueroa-Godoy, « Self-Organizing Map Improved for Classification of Partial Discharge using Desirability Function », in : *Int. J. Comb. Optim. Probl. Informatics* 6.*3* (2015), p. 49-65, url : https://ijcopi.org/index.php/ojs/article/view/51.

[48] Ryotaro Kamimura, « Input information maximization for improving self-organizing maps », in : *Appl. Intell.* 41.*2* (2014), p. 421-438, doi : 10.1007/S10489-014-0525-1, url : https://doi.org/10.1007/s10489-014-0525-1.

[49] Samuel Kaski et Krista Lagus, « Comparing self-organizing maps », in : *Artificial Neural Networks — ICANN 96*, sous la dir. de Christoph von der Malsburg et al., Berlin, Heidelberg : Springer Berlin Heidelberg, 1996, p. 809-814.

[50] Lyes Khacef, Laurent Rodriguez et Benoît Miramond, « Improving Self-Organizing Maps with Unsupervised Feature Extraction », in : *Neural Information Processing - 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 23-27, 2020, Proceedings, Part II*, sous la dir. d'Haiqin Yang et al., t. 12533, Lecture Notes in Computer Science, Springer, 2020, p. 474-486, doi : 10.1007/978-3-030-63833-7_40, url : https://doi.org/10.1007/978-3-030-63833-7%5C_40.

[51] Khaled Ben Khalifa et al., « New Hardware Architecture for Self-Organizing Map Used for Color Vector Quantization », in : *J. Circuits Syst. Comput.* 29.*1* (2020), 2050002 :1-2050002 :35, doi : 10.1142/S0218126620500024, url : https://doi.org/10.1142/S0218126620500024.

[52] Atlas Khan et al., « Convergence Analysis of a New Self Organizing Map Based Optimization (SOMO) Algorithm », in : *Cogn. Comput.* 7.*4* (2015), p. 477-486, doi : 10.1007/S12559-014-9315-7, url : https://doi.org/10.1007/s12559-014-9315-7.

[53] Phillip C. S. R. Kilgore et al., « hSOM : Visualizing Self-Organizing Maps to Accomodate Categorical Data », in : *24th International Conference on Information Visualisation, IV 2020, Melbourne, Australia, September 7-11, 2020*, sous la dir. d'Ebad Banissi et al., IEEE, 2020, p. 644-650, DOI : 10.1109/IV51561.2020.00111, URL : https://doi.org/10.1109/IV51561.2020.00111.

[54] K. Kiviluoto, « Topology preservation in self-organizing maps », in : *Proceedings of International Conference on Neural Networks (ICNN'96)*, t. 1, 1996, 294-299 vol.1, DOI : 10.1109/ICNN.1996.548907.

[55] Kimmo Kiviluoto, « Topology preservation in self-organizing maps », in : *Proceedings of International Conference on Neural Networks (ICNN'96)* 1 (1996), 294-299 vol.1, URL : https://api.semanticscholar.org/CorpusID:122660046.

[56] Teuvo Kohonen, « Essentials of the self-organizing map », in : *Neural Networks* 37 (2013), p. 52-65, DOI : 10.1016/J.NEUNET.2012.09.018, URL : https://doi.org/10.1016/j.neunet.2012.09.018.

[57] Teuvo Kohonen, *Self-Organizing Maps*, t. 30, Springer Series in Information Sciences, Springer, 1995, ISBN : 978-3-642-97612-4, DOI : 10.1007/978-3-642-97610-0, URL : https://doi.org/10.1007/978-3-642-97610-0.

[58] Teuvo Kohonen, *Self-Organizing Maps*, 2nd, Information Sciences, Berlin, Heidelberg : Springer, 1997, ISBN : 978-3-540-67921-9.

[59] Teuvo Kohonen, *Self-Organizing Maps, Second Edition*, t. 30, Springer Series in Information Sciences, Springer, 1997, ISBN : 978-3-540-62017-4, DOI : 10.1007/978-3-642-97966-8, URL : https://doi.org/10.1007/978-3-642-97966-8.

[60] Teuvo Kohonen, *Self-Organizing Maps, Third Edition*, Springer Series in Information Sciences, Springer, 2001, ISBN : 978-3-540-67921-9, DOI : 10.1007/978-3-642-56927-2, URL : https://doi.org/10.1007/978-3-642-56927-2.

[61] Teuvo Kohonen, « The self-organizing map », in : *Proc. IEEE* 78.*9* (1990), p. 1464-1480, DOI : 10.1109/5.58325, URL : https://doi.org/10.1109/5.5832558325.

[62] Viktoria Kristollari et Vassilia Karathanassi, « Fine-Tuning Self-Organizing Maps for Sentinel-2 Imagery : Separating Clouds from Bright Surfaces », in : *Remote. Sens.* 12.*12* (2020), p. 1923, DOI : 10.3390/RS12121923, URL : https://doi.org/10.3390/rs12121923.

[63] J. B. Kruskal, « Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis », in : *Psychometrika* 29.*1* (1964), p. 1-27, doi : 10.1007/BF02289565.

[64] Huang-Cheng Kuo et Shih-Hao Chen, « Self-Organizing Map Learning with Momentum », in : *Comput. Inf. Sci.* 9.*1* (2016), p. 136-146, doi : 10.5539/CIS.V9N1P136, url : https://doi.org/10.5539/cis.v9n1p136.

[65] Marius Lindauer et al., « SMAC3 : A Versatile Bayesian Optimization Package for Hyperparameter Optimization », in : *Journal of Machine Learning Research* 23.*54* (2022), p. 1-9, url : http://jmlr.org/papers/v23/21-0888.html.

[66] Hongbo Liu et Lei Yu, « Feature Selection and Dimensionality Reduction for Time-Series Data », in : *Pattern Recognition Letters* 33.*5* (2012), Méthodes de réduction de dimension pour l'analyse de données temporelles., p. 618-626, doi : 10.1016/j.patrec.2011.12.021.

[67] Rosanne Liu et al., *An Intriguing Failing of Convolutional Neural Networks and the CoordConv Solution*, 2018, arXiv : 1807.03247 [cs.CV], url : https://arxiv.org/abs/1807.03247.

[68] Ezequiel López-Rubio et Antonio Díaz Ramos, « Grid topologies for the self-organizing map », in : *Neural Networks* 56 (2014), p. 35-48, doi : 10.1016/J.NEUNET.2014.05.001, url : https://doi.org/10.1016/j.neunet.2014.05.001.

[69] Laura Manduchi et al., « Variational PSOM : Deep Probabilistic Clustering with Self-Organizing Maps », in : *CoRR* abs/1910.01590 (2019), arXiv : 1910.01590, url : http://arxiv.org/abs/1910.01590.

[70] Denis Maurel, Jérémie Sublime et Sylvain Lefebvre, « Incremental Self-Organizing Maps for Collaborative Clustering », in : *Neural Information Processing - 24th International Conference, ICONIP 2017, Guangzhou, China, November 14-18, 2017, Proceedings, Part I*, sous la dir. de Derong Liu et al., t. 10634, Lecture Notes in Computer Science, Springer, 2017, p. 497-504, doi : 10.1007/978-3-319-70087-8_52, url : https://doi.org/10.1007/978-3-319-70087-8%5C_52.

[71] Tomas Mikolov et al., « Efficient Estimation of Word Representations in Vector Space », in : *Proceedings of the International Conference on Learning Represen-*

tations (ICLR), Introduction du modèle Word2Vec, base des embeddings sémantiques utilisés dans le NLP moderne., 2013.

[72] Tomas Mikolov et al., *Efficient Estimation of Word Representations in Vector Space*, 2013, arXiv : 1301.3781 [cs.CL], URL : https://arxiv.org/abs/1301.3781.

[73] Deepak Nagar, Kiran Pannerselvam et Palaniappan Ramu, « A novel data-driven visualization of n-dimensional feasible region using interpretable self-organizing maps (iSOM) », in : *Neural Networks* 155 (2022), p. 398-412, DOI : 10.1016/J.NEUNET.2022.08.019, URL : https://doi.org/10.1016/j.neunet.2022.08.019.

[74] Van Tuc Nguyen, Markus Hagenbuchner et Ah Chung Tsoi, « High Resolution Self-organizing Maps », in : *AI 2016 : Advances in Artificial Intelligence - 29th Australasian Joint Conference, Hobart, TAS, Australia, December 5-8, 2016, Proceedings*, sous la dir. de Byeong Ho Kang et Quan Bai, t. 9992, Lecture Notes in Computer Science, Springer, 2016, p. 441-454, DOI : 10.1007/978-3-319-50127-7_38, URL : https://doi.org/10.1007/978-3-319-50127-7%5C_38.

[75] Dominik Olszewski, « A data-scattering-preserving adaptive self-organizing map », in : *Eng. Appl. Artif. Intell.* 105 (2021), p. 104420, DOI : 10.1016/J.ENGAPPAI.2021.104420, URL : https://doi.org/10.1016/j.engappai.2021.104420.

[76] Dominik Olszewski, « An Improved Adaptive Self-Organizing Map », in : *Artificial Intelligence and Soft Computing - 13th International Conference, ICAISC 2014, Zakopane, Poland, June 1-5, 2014, Proceedings, Part I*, sous la dir. de Leszek Rutkowski et al., t. 8467, Lecture Notes in Computer Science, Springer, 2014, p. 109-120, DOI : 10.1007/978-3-319-07173-2_11, URL : https://doi.org/10.1007/978-3-319-07173-2%5C_11.

[77] Akinari Onishi, « Landmark map : An extension of the self-organizing map for a user-intended nonlinear projection », in : *Neurocomputing* 388 (2020), p. 228-245, DOI : 10.1016/J.NEUCOM.2019.12.125, URL : https://doi.org/10.1016/j.neucom.2019.12.125.

[78] Leandro Antonio Pasa, José Alfredo F. Costa et Marcial Guerra de Medeiros, « An ensemble algorithm for Kohonen self-organizing map with different sizes », in : *Log. J. IGPL* 25.6 (2017), p. 1020-1033, DOI : 10.1093/JIGPAL/JZX046, URL : https://doi.org/10.1093/jigpal/jzx046.

[79] Leandro Antonio Pasa, José Alfredo F. Costa et Marcial Guerra de Medeiros, « Self-Organizing Maps Fusion : An Approach to Different Size Maps », in : *Hybrid Artificial Intelligent Systems - 10th International Conference, HAIS 2015, Bilbao, Spain, June 22-24, 2015, Proceedings*, sous la dir. d'Enrique Onieva et al., t. 9121, Lecture Notes in Computer Science, Springer, 2015, p. 568-579, doi : 10.1007/978-3-319-19644-2_47, url : https://doi.org/10.1007/978-3-319-19644-2%5C_47.

[80] Fabian Pedregosa et al., « Scikit-learn : Machine Learning in Python », in : *Journal of Machine Learning Research*, t. 12, Implémentation pratique de l'encodage one-hot et des transformations de variables catégorielles., 2011, p. 2825-2830.

[81] Ladislav Peska et Jakub Lokoc, « Rating-Aware Self-Organizing Maps », in : *MultiMedia Modeling - 28th International Conference, MMM 2022, Phu Quoc, Vietnam, June 6-10, 2022, Proceedings, Part I*, sous la dir. de Björn Þór Jónsson et al., t. 13141, Lecture Notes in Computer Science, Springer, 2022, p. 119-130, doi : 10.1007/978-3-030-98358-1_10, url : https://doi.org/10.1007/978-3-030-98358-1%5C_10.

[82] Hai Van Pham et Khang Dinh Tran, « A Proposal of a Model Using Kansei Evaluation Integrated with Fuzzy Rules and Self-Organizing Map for Evaluation of Bio-Food Products », in : *Proceedings of the 1st International Workshop on Semantic Technologies (IWOST), Changchun, China, March 9-12, 2015*, sous la dir. de Steffen Hölldobler et Yanchun Liang, t. 1339, CEUR Workshop Proceedings, CEUR-WS.org, 2015, p. 85-92, url : https://ceur-ws.org/Vol-1339/paper8.pdf.

[83] Yannis Poulakis, Christos Doulkeridis et Dimosthenis Kyriazis, « A Survey on AutoML Methods and Systems for Clustering », in : *ACM Trans. Knowl. Discov. Data* 18.*5* (2024), 120 :1-120 :30, doi : 10.1145/3643564, url : https://doi.org/10.1145/3643564.

[84] Xiaofei Qu et al., « A Survey on the Development of Self-Organizing Maps for Unsupervised Intrusion Detection », in : *Mob. Networks Appl.* 26.*2* (2021), p. 808-829, doi : 10.1007/S11036-019-01353-0, url : https://doi.org/10.1007/s11036-019-01353-0.

[85] Nicolas P. Rougier et Georgios Is. Detorakis, « Randomized Self-Organizing Map », in : *Neural Comput.* 33.*8* (2021), p. 2241-2273, doi : 10.1162/NECO_A_01406, url : https://doi.org/10.1162/neco%5C_a%5C_01406.

[86] Naveen Saini et al., « Automatic Parameter Selection of Granual Self-organizing Map for Microblog Summarization », in : *Neural Information Processing - 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 23-27, 2020, Proceedings, Part I*, sous la dir. d'Haiqin Yang et al., t. 12532, Lecture Notes in Computer Science, Springer, 2020, p. 680-692, doi : 10.1007/978-3-030-63830-6_57, url : https://doi.org/10.1007/978-3-030-63830-6%5C_57.

[87] Mohamed Sakkari et Mourad Zaied, « A Convolutional Deep Self-Organizing Map Feature extraction for machine learning », in : *Multim. Tools Appl.* 79.*27-28* (2020), p. 19451-19470, doi : 10.1007/S11042-020-08822-9, url : https://doi.org/10.1007/s11042-020-08822-9.

[88] Gerard Salton et Christopher Buckley, « Term-Weighting Approaches in Automatic Text Retrieval », in : *Information Processing and Management* 24.*5* (1988), Travail fondateur sur la pondération TF-IDF dans la recherche d'information., p. 513-523, doi : 10.1016/0306-4573(88)90021-0.

[89] Peter Sarlin, « Automated and weighted self-organizing time maps », in : *Knowl. Inf. Syst.* 44.*2* (2015), p. 493-505, doi : 10.1007/S10115-014-0762-Y, url : https://doi.org/10.1007/s10115-014-0762-y.

[90] Leandro Augusto da Silva, Bruno P. de Vasconcelos et Emilio Del-Moral-Hernandez, « A model to estimate the Self-Organizing Maps grid dimension for Prototype Generation », in : *Intell. Data Anal.* 25.*2* (2021), p. 321-338, doi : 10.3233/IDA-205123, url : https://doi.org/10.3233/IDA-205123.

[91] Vincent Sitzmann et al., « Implicit neural representations with periodic activation functions », in : *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Vancouver, BC, Canada : Curran Associates Inc., 2020, isbn : 9781713829546.

[92] Kate Smith-Miles, « Cross-disciplinary perspectives on meta-learning for algorithm selection », in : *ACM Comput. Surv.* 41.*1* (2008), 6 :1-6 :25, doi : 10.1145/1456650.1456656, url : https://doi.org/10.1145/1456650.1456656.

[93] Gerasimos Spanakis et Gerhard Weiss, « AMSOM : Adaptive Moving Self-organizing Map for Clustering and Visualization », in : *CoRR* abs/1605.06047 (2016), arXiv : 1605.06047, URL : http://arxiv.org/abs/1605.06047.

[94] Andrew J. Starkey et Aliyu Usman Ahmad, « Semi-automated data classification with feature weighted self organizing map », in : *13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, ICNC-FSKD 2017, Guilin, China, July 29-31, 2017*, sous la dir. d'Yong Liu et al., IEEE, 2017, p. 136-141, DOI : 10.1109/FSKD.2017.8392964, URL : https://doi.org/10.1109/FSKD.2017.8392964.

[95] Matthew Tancik et al., *Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains*, 2020, arXiv : 2006.10739 [cs.CV], URL : https://arxiv.org/abs/2006.10739.

[96] Robert Tatoian et Lutz Hamel, « Self-Organizing Map Convergence », in : *Int. J. Serv. Sci. Manag. Eng. Technol.* 9.*2* (2018), p. 61-84, DOI : 10.4018/IJSSMET.2018040103, URL : https://doi.org/10.4018/IJSSMET.2018040103.

[97] Akio Utsugi, « Hyperparameter Selection for Self-Organizing Maps », in : *Neural Comput.* 9.*3* (1997), p. 623-635, DOI : 10.1162/NECO.1997.9.3.623, URL : https://doi.org/10.1162/neco.1997.9.3.623.

[98] Mikko Varsta, Juha Heikkonen et Jorma Lampinen, « Time Adaptive Self-Organizing Map », in : *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Introduction du modèle Time Adaptive SOM (TASOM) pour les données séquentielles et temporelles., 1998, p. 1735-1740.

[99] Mahdi Vasighi et Homa Amini, « A directed batch growing approach to enhance the topology preservation of self-organizing map », in : *Appl. Soft Comput.* 55 (2017), p. 424-435, DOI : 10.1016/J.ASOC.2017.02.015, URL : https://doi.org/10.1016/j.asoc.2017.02.015.

[100] Rajan Vohra et al., « Using Self Organizing Maps and K Means Clustering Based on RFM Model for Customer Segmentation in the Online Retail Business », in : *Intelligent Computing Methodologies - 16th International Conference, ICIC 2020, Bari, Italy, October 2-5, 2020, Proceedings, Part III*, sous la dir. de De-Shuang Huang et Prashan Premaratne, t. 12465, Lecture Notes in Computer Science, Springer, 2020, p. 484-497, DOI : 10.1007/978-3-030-60796-8_42, URL : https://doi.org/10.1007/978-3-030-60796-8%5C_42.

[101] William H. Wilson, « Metrics and the Cooperative Process of the Self-organizing Map Algorithm », in : *Advances in Neural Networks - ISNN 2017 - 14th International Symposium, ISNN 2017, Sapporo, Hakodate, and Muroran, Hokkaido, Japan, June 21-26, 2017, Proceedings, Part I*, sous la dir. de Fengyu Cong, Andrew Chi-Sing Leung et Qinglai Wei, t. 10261, Lecture Notes in Computer Science, Springer, 2017, p. 502-510, doi : 10.1007/978-3-319-59072-1_59, url : https://doi.org/10.1007/978-3-319-59072-1%5C_59.

[102] William H. Wilson, « Multi-scale metrics and self-organizing maps : a computational approach to the structure of sensory maps », in : *CoRR* abs/1805.03337 (2018), arXiv : 1805.03337, url : http://arxiv.org/abs/1805.03337.

[103] Na Zhou, Jin Tian et Minqiang Li, « Online recommendation based on incremental-input self-organizing map », in : *Electron. Commer. Res. Appl.* 50 (2021), p. 101096, doi : 10.1016/J.ELERAP.2021.101096, url : https://doi.org/10.1016/j.elerap.2021.101096.

[104] Zalhan Mohd Zin, « Cluster and visualize data using 3D self-organizing maps », in : *11th International Conference on Ubiquitous Robots and Ambient Intelligence, URAI 2014, Kuala Lumpur, Malaysia, November 12-15, 2014*, IEEE, 2014, p. 163-168, doi : 10.1109/URAI.2014.7057523, url : https://doi.org/10.1109/URAI.2014.7057523.

[105] Zalhan Mohd Zin, Rubiyah Yusof et Ehsan Mesbahi, « Improving self organizing maps method for data clustering and classification », in : *10th Asian Control Conference, ASCC 2015, Kota Kinabalu, Malaysia, May 31 - June 3, 2015*, IEEE, 2015, p. 1-7, doi : 10.1109/ASCC.2015.7244588, url : https://doi.org/10.1109/ASCC.2015.7244588.

Annexe A

Annexes

A.1 Exemples de visualisation des cartes SOM

A.1.1 SOM bidimensionnelles



Figure A.1 – Échelle des couleurs utilisée pour représenter les valeurs.

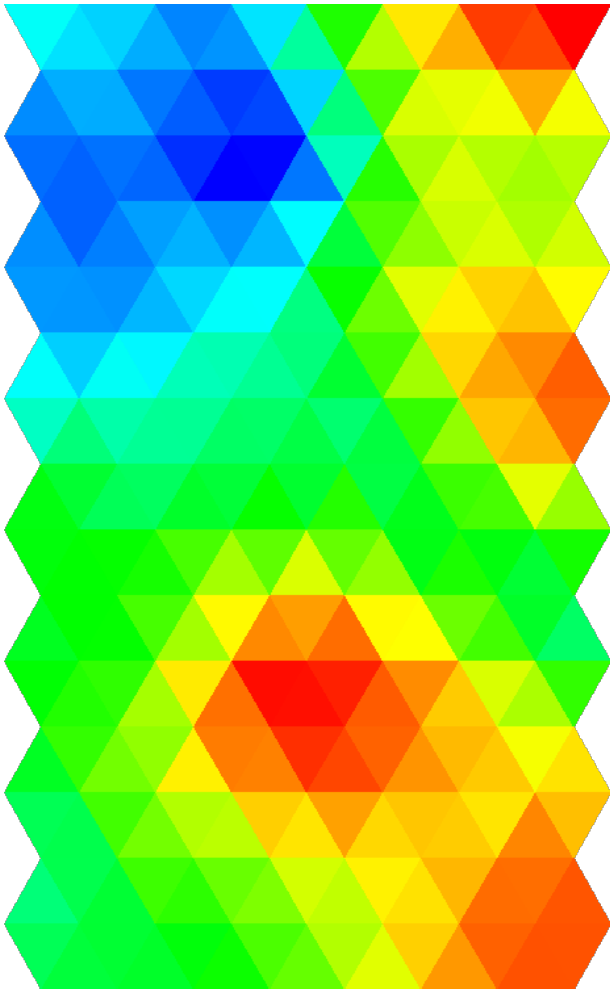


FIGURE A.2 – Carte de triangles représentant une variable d'un jeu de données de 417 entrées.

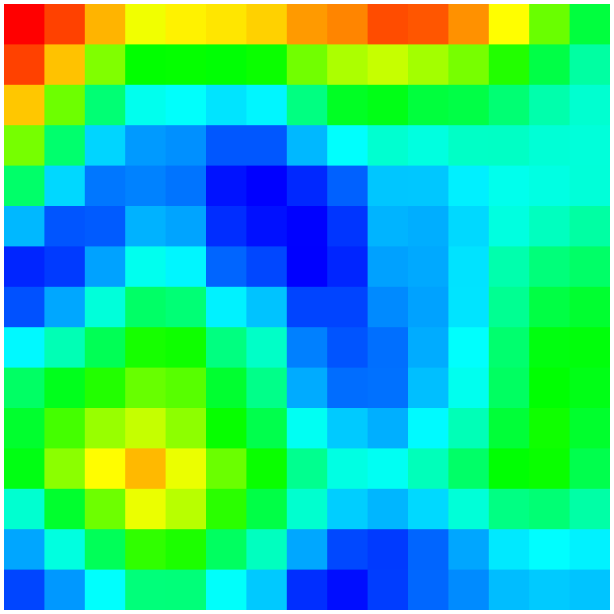


FIGURE A.3 – Carte de carrés représentant une variable d'un jeu de données de 417 entrées.

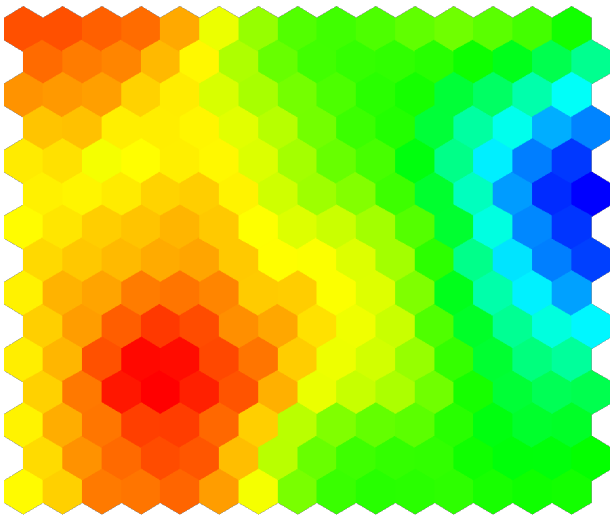


FIGURE A.4 – Carte d'hexagones représentant une variable d'un jeu de données de 417 entrées.

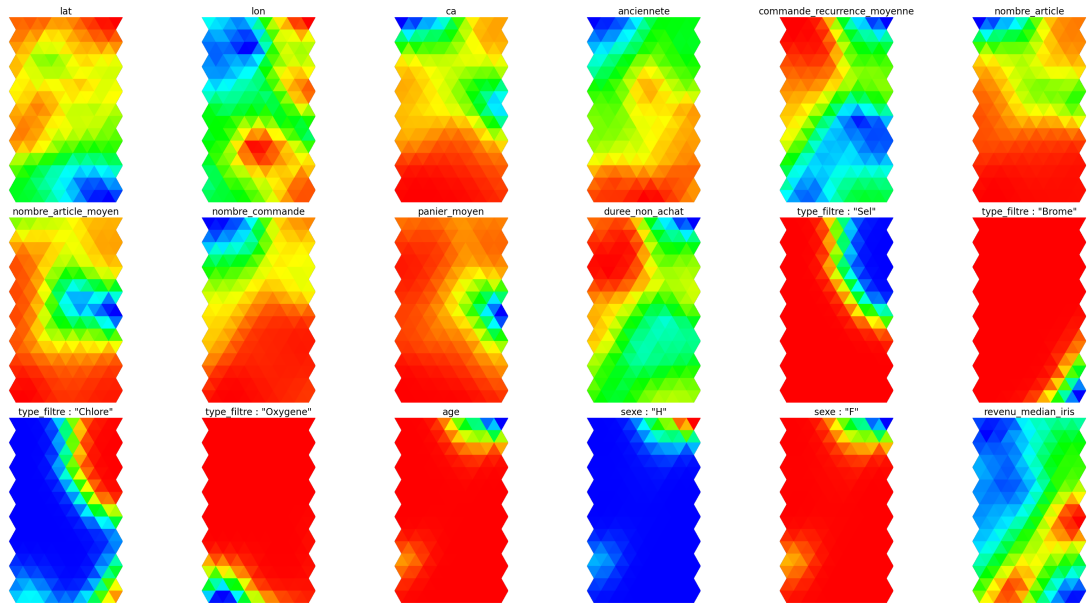


FIGURE A.5 – Panel de cartes de triangles représentant 18 variables d'un jeu de données de 417 entrées.

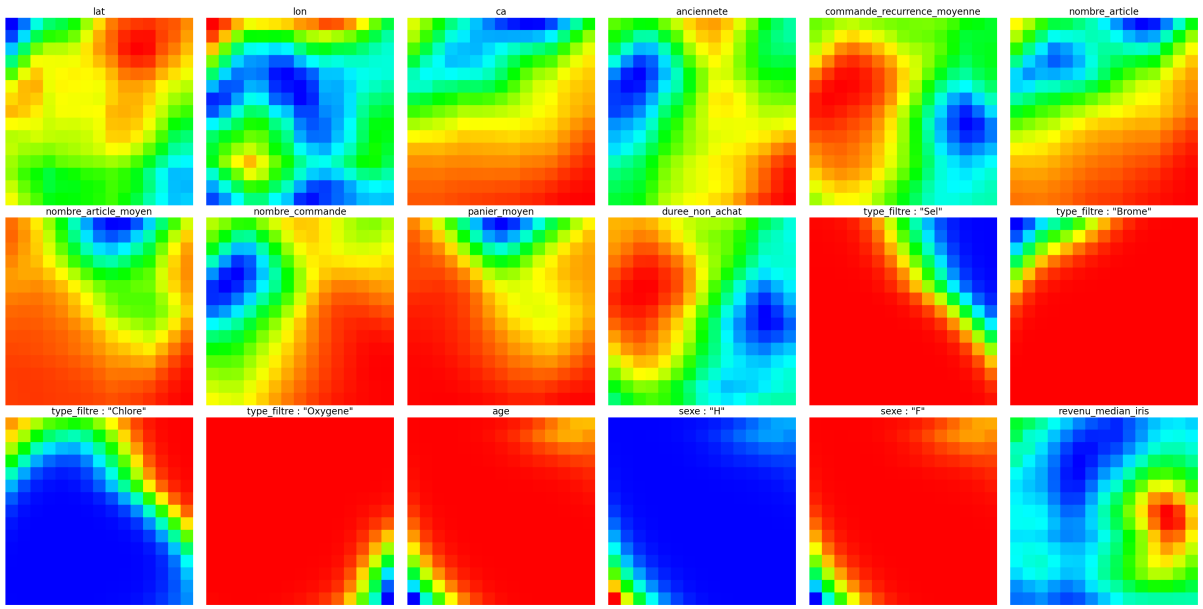


FIGURE A.6 – Panel de cartes de carrés représentant 18 variables d'un jeu de données de 417 entrées.

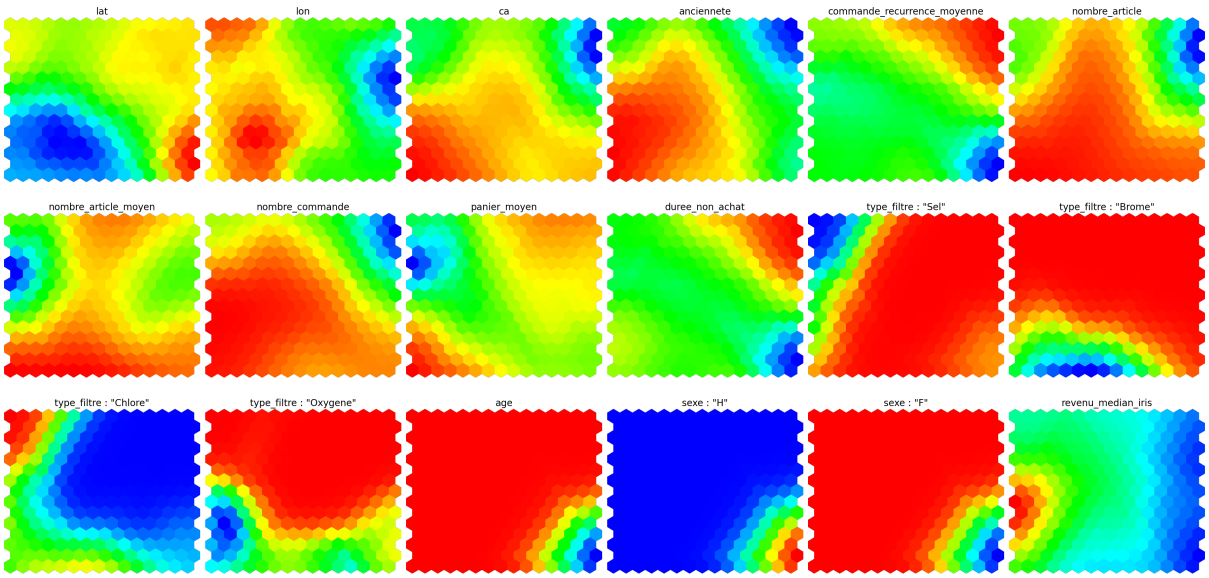


FIGURE A.7 – Panel de cartes d'hexagones représentant 18 variables d'un jeu de données de 417 entrées.

A.1.2 SOM tridimensionnelles

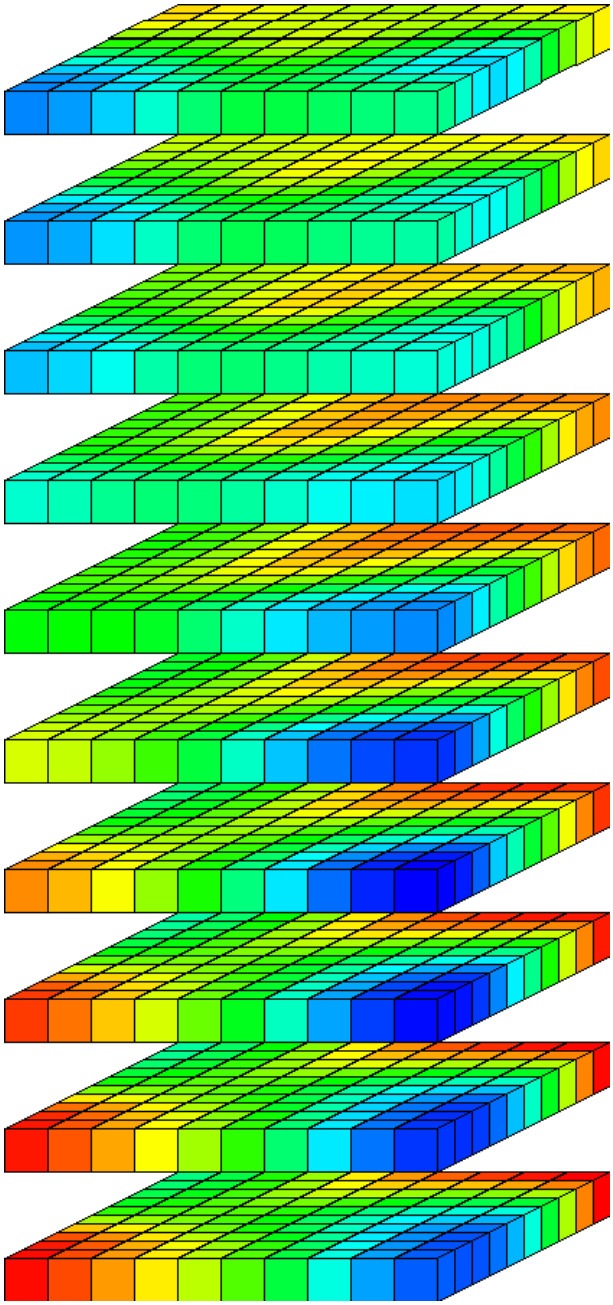


FIGURE A.8 – Carte de cubes représentant une variable d'un jeu de données de 417 entrées.

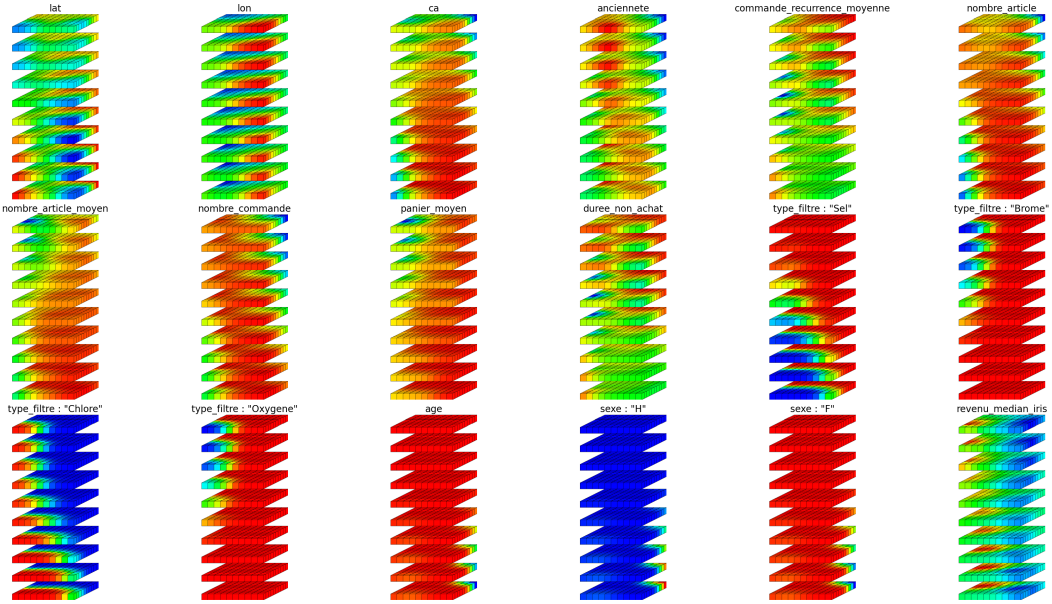
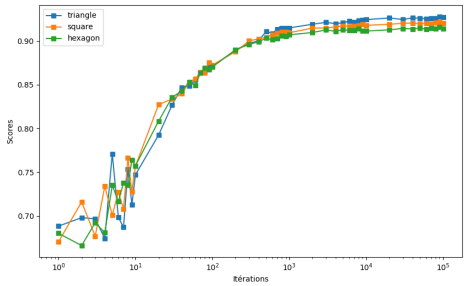


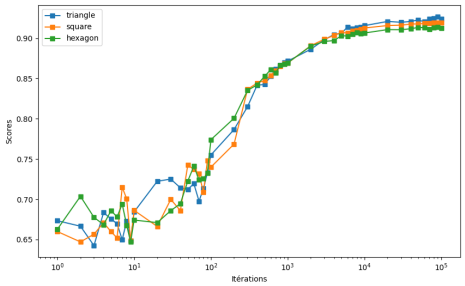
FIGURE A.9 – Panel de cartes de cubes représentant 18 variables d'un jeu de données de 417 entrées.

A.2 Qualité de l'apprentissage

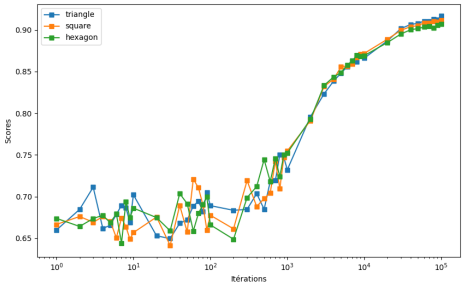
A.2.1 S'_1



(a) $\epsilon_0 = 0.1$

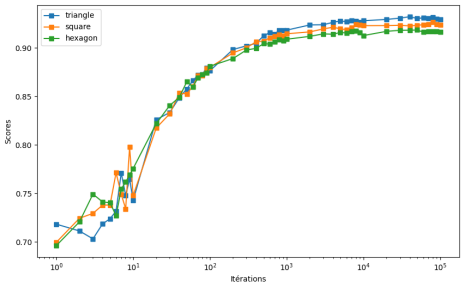


(b) $\epsilon_0 = 0.01$

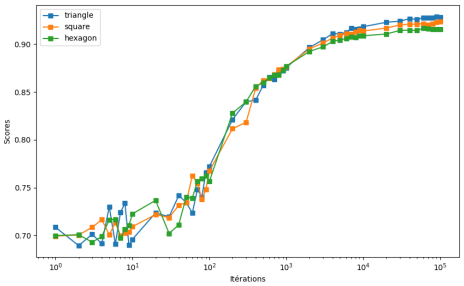


(c) $\epsilon_0 = 0.001$

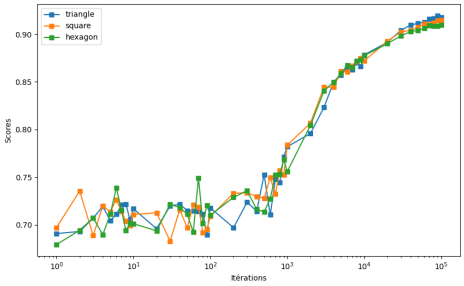
FIGURE A.10 – Analyse de l'impact de $T_{\max}$ et ϵ_0 sur des cartes SOM de tailles 10x10 et différentes topologies avec le score S'_1.



(a) $\epsilon_0 = 0.1$

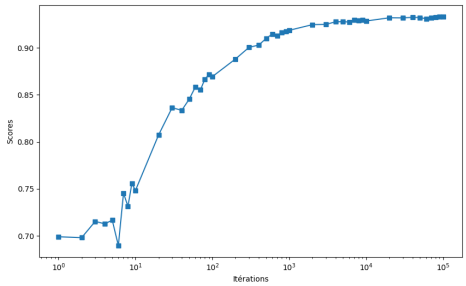


(b) $\epsilon_0 = 0.01$

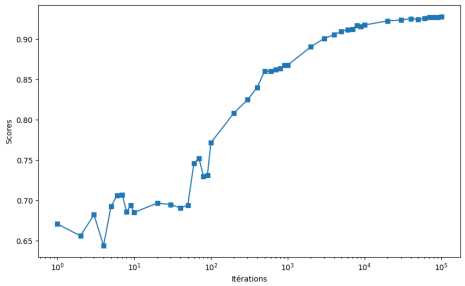


(c) $\epsilon_0 = 0.001$

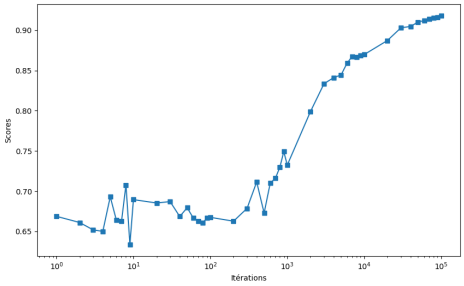
FIGURE A.11 – Analyse de l'impact de $T_{\max}$ et ϵ_0 sur des cartes SOM de tailles 20x20 et différentes topologies avec le score S'_1.



(a) $\epsilon_0 = 0.1$

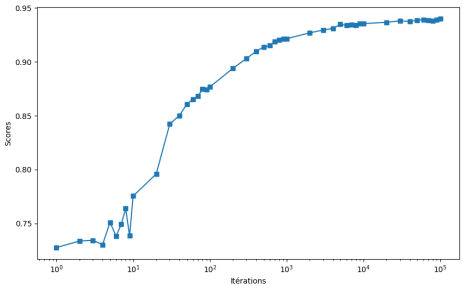


(b) $\epsilon_0 = 0.01$

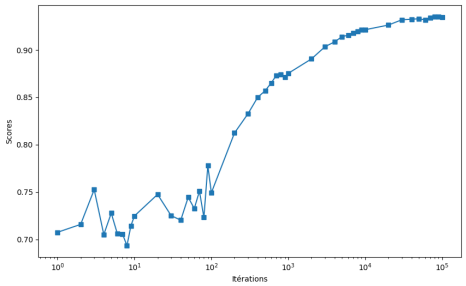


(c) $\epsilon_0 = 0.001$

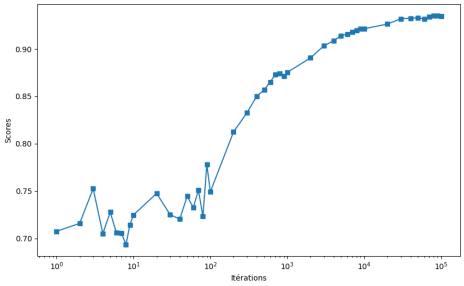
FIGURE A.12 – Analyse de l'impact de $T_{\max}$ et ϵ_0 sur des cartes SOM de tailles 5x5x5 et différentes topologies avec le score S'_1.



(a) $\epsilon_0 = 0.1$



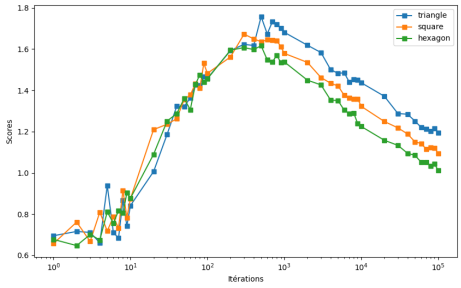
(b) $\epsilon_0 = 0.01$



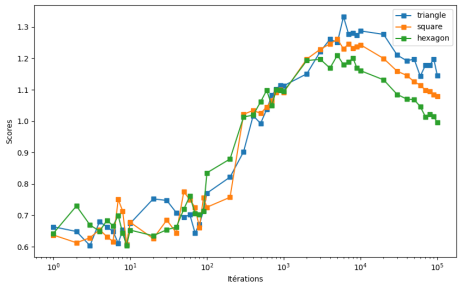
(c) $\epsilon_0 = 0.001$

FIGURE A.13 – Analyse de l'impact de $T_{\max}$ et ϵ_0 sur des cartes SOM de tailles 8x8x8 et différentes topologies avec le score S'_1.

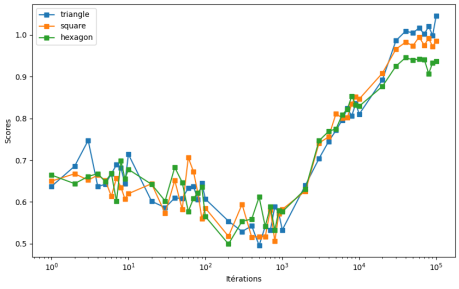
A.2.2 $S_{\Delta t}(S'_1)$



(a) $\epsilon_0 = 0.1$

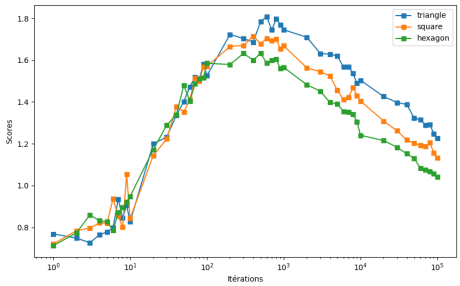


(b) $\epsilon_0 = 0.01$

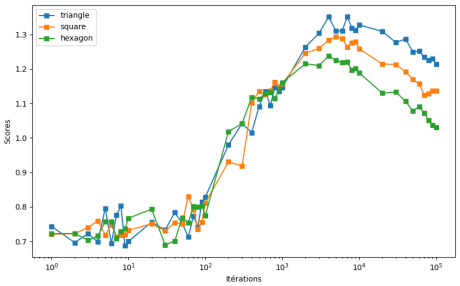


(c) $\epsilon_0 = 0.001$

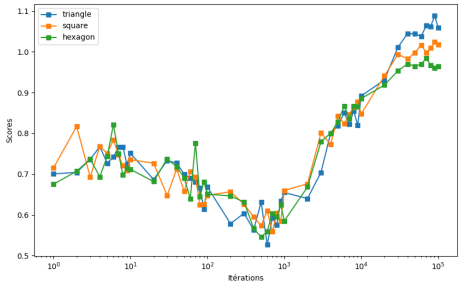
FIGURE A.14 – Analyse de l'impact de $S_{\Delta t}(S'_1)$ sur des cartes SOM de taille 10x10 et différentes topologies avec le score $S_{\Delta t}(S'_1)$.



(a) $\epsilon_0 = 0.1$

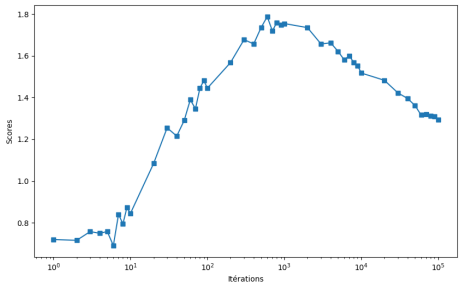


(b) $\epsilon_0 = 0.01$

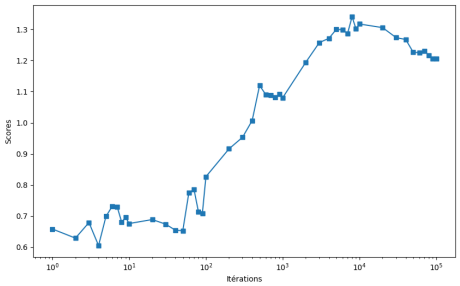


(c) $\epsilon_0 = 0.001$

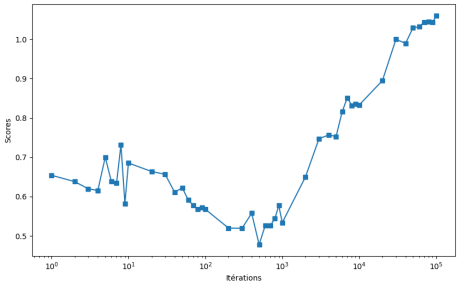
FIGURE A.15 – Analyse de l'impact de $S_{\Delta t}(S'_1)$ sur des cartes SOM de taille 20x20 et différentes topologies avec le score $S_{\Delta t}(S'_1)$.



(a) $\epsilon_0 = 0.1$

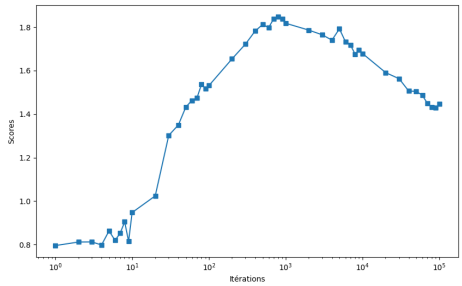


(b) $\epsilon_0 = 0.01$

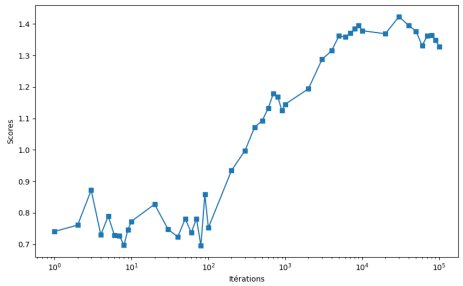


(c) $\epsilon_0 = 0.001$

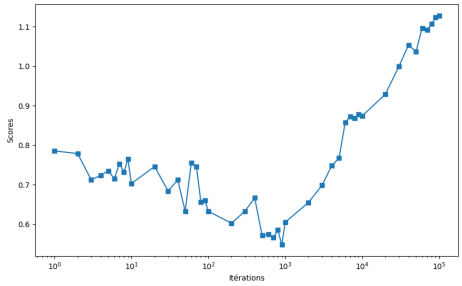
FIGURE A.16 – Analyse de l'impact de $S_{\Delta t}(S'_1)$ sur des cartes SOM de taille 5x5x5 et différentes topologies avec le score $S_{\Delta t}(S'_1)$.



(a) $\epsilon_0 = 0.1$



(b) $\epsilon_0 = 0.01$



(c) $\epsilon_0 = 0.001$

FIGURE A.17 – Analyse de l'impact de $S_{\Delta t}(S'_1)$ sur des cartes SOM de taille 8x8x8 et différentes topologies avec le score $S_{\Delta t}(S'_1)$.

A.3 Algorithmes générateurs

A.3.1 MLP

Algorithm 5 Générateur SOM par MLP (baseline)

Require: Taille de carte (N_x, N_y) ; dimension des données d ; architecture $(2, h_1, \ldots, h_L, d)$

Paramètres $\theta = \{W^{(\ell)} \in \mathbb{R}^{h_{\ell-1} \times h_\ell},\ b^{(\ell)} \in \mathbb{R}^{h_\ell}\}_{\ell=1}^{L+1}$ (avec $h_0 = 2$, $h_{L+1} = d$)

Initialisation He : $W^{(\ell)} \sim \mathcal{N}\Big(0,\ \frac{2}{h_{\ell-1}+h_\ell}\Big)$, $b^{(\ell)} \leftarrow 0$

function ForwardMLP($x \in \mathbb{R}^2$)
  $a \leftarrow x$
  for $\ell = 1$ **to** L **do**
    $z \leftarrow W^{(\ell)} a + b^{(\ell)}$
    $a \leftarrow \max(0, z)$ ▷ ReLU
  end for
  return $W_{out} \leftarrow \sigma\Big(W^{(L+1)} a + b^{(L+1)}\Big)$ ▷ σ : *sigmoid* ou *tanh*
end function
function BuildMap(θ)
  for $i = 0$ **to** $N_x - 1$ **do**
    for $j = 0$ **to** $N_y - 1$ **do**
      $x \leftarrow \Big(\frac{i}{N_x - 1}, \frac{j}{N_y - 1}\Big)$
      $W_{i,j} \leftarrow$ ForwardMLP(x) ▷ $W_{i,j} \in \mathbb{R}^d$
    end for
  end for
  return $\mathcal{W} = \{W_{i,j}\}$
end function
function Score($\mathcal{W}$)
  return $S(\mathcal{W})$ ▷ métrique globale : QE, TE, *trust*, entropie, etc.
end function

A.3.2 MLPFF

Algorithm 6 Générateur SOM par MLP avec Fourier features

Require: (N_x, N_y), d, nombre de fréquences $L \in \{4, 8\}$; architecture $(4L, h_1, \ldots, h_L, d)$

Paramètres θ du MLP comme Alg. **??** ; même init He

function FourierEncode($x = (u, v) \in [0, 1]^2$)
  return $\phi(x) = [\sin(2^k \pi u), \cos(2^k \pi u), \sin(2^k \pi v), \cos(2^k \pi v)]_{k=0}^{L-1} \in \mathbb{R}^{4L}$
end function
function ForwardMLPFF(x)
  $a \leftarrow$ FourierEncode(x)
  for $\ell = 1$ **to** L **do**
    $z \leftarrow W^{(\ell)} a + b^{(\ell)}$; $a \leftarrow \max(0, z)$ ▷ ReLU
  end for
  return $W_{out} \leftarrow \sigma\Big(W^{(L+1)} a + b^{(L+1)}\Big)$
end function
function BuildMap(θ)
  for i, j sur la grille **do**
    $x \leftarrow \Big(\frac{i}{N_x - 1}, \frac{j}{N_y - 1}\Big)$
    $W_{i,j} \leftarrow$ ForwardMLPFF(x)
  end for
  return $\mathcal{W}$
end function
function Score($\mathcal{W}$)
  return $S(\mathcal{W})$
end function

A.3.3 SIREN

Algorithm 7 Générateur SOM par SIREN (activations sinus)

Require: (N_x, N_y), d, architecture $(2, h_1, \ldots, h_L, d)$, facteurs $w_0, w_{0,\text{hidden}}$

Paramètres $\theta = \{W^{(\ell)}, b^{(\ell)}\}$

Init SIREN [91] :

- $W^{(1)} \sim \mathcal{U}\Big(-\frac{1}{h_0}, \frac{1}{h_0}\Big)$ puis $z^{(1)} \leftarrow w_0(W^{(1)}a^{(0)} + b^{(1)})$
- $W^{(\ell)} \sim \mathcal{U}\Big(-\sqrt{\frac{6}{h_{\ell-1}}}/w_{0,\text{hidden}}, \sqrt{\frac{6}{h_{\ell-1}}}/w_{0,\text{hidden}}\Big)$ pour $\ell \geq 2$

function FORWARDSIREN($x \in \mathbb{R}^2$)
  $a \leftarrow x$
  $z \leftarrow w_0(W^{(1)}a + b^{(1)})\,; \quad a \leftarrow \sin(z)$
  for $\ell = 2$ **to** L **do**
    $z \leftarrow w_{0,\text{hidden}}(W^{(\ell)}a + b^{(\ell)})\,; \quad a \leftarrow \sin(z)$
  end for
  return $W_{out} \leftarrow W^{(L+1)}a + b^{(L+1)}$ ▷ sortie linéaire ou $\sigma(\cdot)$ selon les données
end function
function BUILDMAP(θ)
  for i, j sur la grille **do**
    $x \leftarrow \Big(\frac{i}{N_x-1}, \frac{j}{N_y-1}\Big)$
    $W_{i,j} \leftarrow$ FORWARDSIREN(x)
  end for
  return $\mathcal{W}$
end function
function SCORE($\mathcal{W}$)
  return $S(\mathcal{W})$
end function

A.3.4 RBF

Algorithm 8 Générateur SOM par Réseau RBF

Require: (N_x, N_y), d, nombre de noyaux K ; centres $\{c_k\}_{k=1}^{K} \subset [0,1]^2$ (fixés grille ou optimisés) ; largeurs $\{\sigma_k > 0\}$; coefficients $\{\alpha_k \in \mathbb{R}^d\}$

Paramètres $\theta = \{c_k, \sigma_k, \alpha_k\}_{k=1}^{K}$ (on peut fixer $\{c_k\}$ pour réduire la dimension)

function $\mathrm{RBF}(x, c, \sigma)$
  return $\exp\Big(- \|x - c\|^2/(2\sigma^2)\Big)$
end function
function ForwardRBF($x \in [0,1]^2$)
  $W_{out} \leftarrow 0 \in \mathbb{R}^d$
  for $k = 1$ **to** K **do**
    $W_{out} \leftarrow W_{out} + \alpha_k \cdot \mathrm{RBF}(x, c_k, \sigma_k)$
  end for
  return W_{out}
end function
function BuildMap(θ)
  for i, j sur la grille **do**
    $x \leftarrow \Big(\frac{i}{N_x - 1}, \frac{j}{N_y - 1}\Big)$
    $W_{i,j} \leftarrow$ ForwardRBF(x)
  end for
  return $\mathcal{W}$
end function
function Score($\mathcal{W}$)
  return $S(\mathcal{W})$
end function

A.3.5 KRR

Algorithm 9 Générateur SOM par KRR (solution ridge sur ancres)

Require: (N_x, N_y), d, ancres $\{c_k\}_{k=1}^{M} \subset [0,1]^2$ (fixées grille), noyau RBF $K_\ell(x,y) = \exp(-\|x-y\|^2/(2\ell^2))$, régularisation $\lambda > 0$

Paramètres optimisés : $Y \in \mathbb{R}^{M\times d}$ (valeurs cibles aux centres), hyperparams (ℓ, λ) optionnellement

function FitKRR(Y, ℓ, λ)
  Construire $K \in \mathbb{R}^{M\times M}$, $K_{ab} \leftarrow K_\ell(c_a, c_b)$
  Résoudre $(K + \lambda I)A = Y$ pour $A \in \mathbb{R}^{M\times d}$ ▷ décomposition de Cholesky stable
  return A
end function
function ForwardKRR(x, A, ℓ)
  $k(x) \in \mathbb{R}^M$, $k(x)_k \leftarrow K_\ell(x, c_k)$
  return $W_{out} \leftarrow k(x)^\top A \in \mathbb{R}^d$
end function
function BuildMap(Y, ℓ, λ)
  $A \leftarrow$ FitKRR(Y, ℓ, λ)
  for i, j sur la grille **do**
    $x \leftarrow \left(\frac{i}{N_x-1}, \frac{j}{N_y-1}\right)$
    $W_{i,j} \leftarrow$ ForwardKRR(x, A, ℓ)
  end for
  return $\mathcal{W}$
end function
function Score($\mathcal{W}$)
  return $S(\mathcal{W})$
end function

A.3.6 CNN

Algorithm 10 Générateur SOM par CNN CoordConv

Require: (N_x, N_y), d, structure de couches : $[(k_1, C_1), \ldots, (k_L, C_L)]$ avec $k_\ell \in \{1, 3\}$, C_ℓ canaux

Entrée spatiale : tenseur $X \in \mathbb{R}^{N_x \times N_y \times C_0}$ avec $C_0 = 2$ (canaux x, y) ou 3 (si canal constant 1 ajouté)

Paramètres : filtres $W^{(\ell)} \in \mathbb{R}^{k_\ell \times k_\ell \times C_{\ell-1} \times C_\ell}$, biais $b^{(\ell)} \in \mathbb{R}^{C_\ell}$, couche finale 1×1 vers d

Init He sur chaque couche conv

function BuildInput(N_x, N_y)
  for $i = 0$ **to** $N_x - 1$ **do**
    for $j = 0$ **to** $N_y - 1$ **do**
      $X[i, j, 0] \leftarrow \frac{i}{N_x - 1}$, $X[i, j, 1] \leftarrow \frac{j}{N_y - 1}$ ▷ (+ canal constant = 1 si choisi)
    end for
  end for
  return X
end function

function ConvSame(U, W, b) ▷ convolution 2D *same*, stride 1, padding zéro
  return V de même taille spatiale que U
end function

function ForwardCoordConv(θ)
  $U \leftarrow$ BuildInput(N_x, N_y)
  for $\ell = 1$ **to** L **do**
    $U \leftarrow$ ConvSame($U, W^{(\ell)}, b^{(\ell)}$)
    $U \leftarrow \max(0, U)$ ▷ ReLU
  end for
  $Wmap \leftarrow$ ConvSame($U, W_{1\times1}^{(\text{out})}, b^{(\text{out})}$) ▷ $\in \mathbb{R}^{N_x \times N_y \times d}$
  return $\mathcal{W} = \{W_{i,j}\}$ en lisant $Wmap[:, :, :]$
end function

function Score($\mathcal{W}$)
  return $S(\mathcal{W})$
end function

A.4 Application Web

A.4.1 Accueil et Authentification



FIGURE A.18 – Interface d'accueil

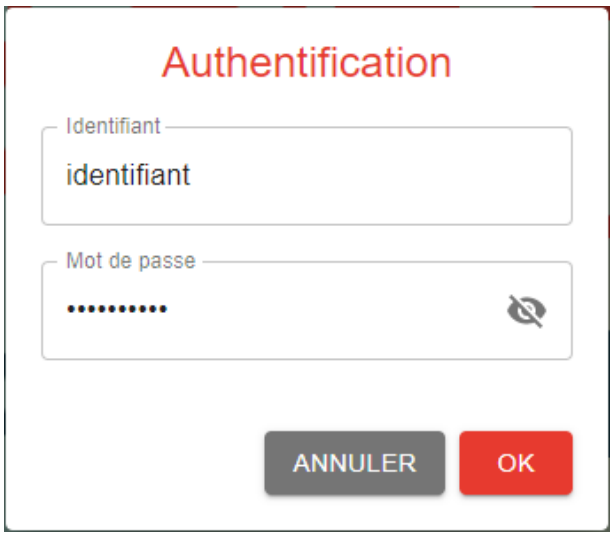


FIGURE A.19 – Interface d'authentification

A.4.2 Sélection

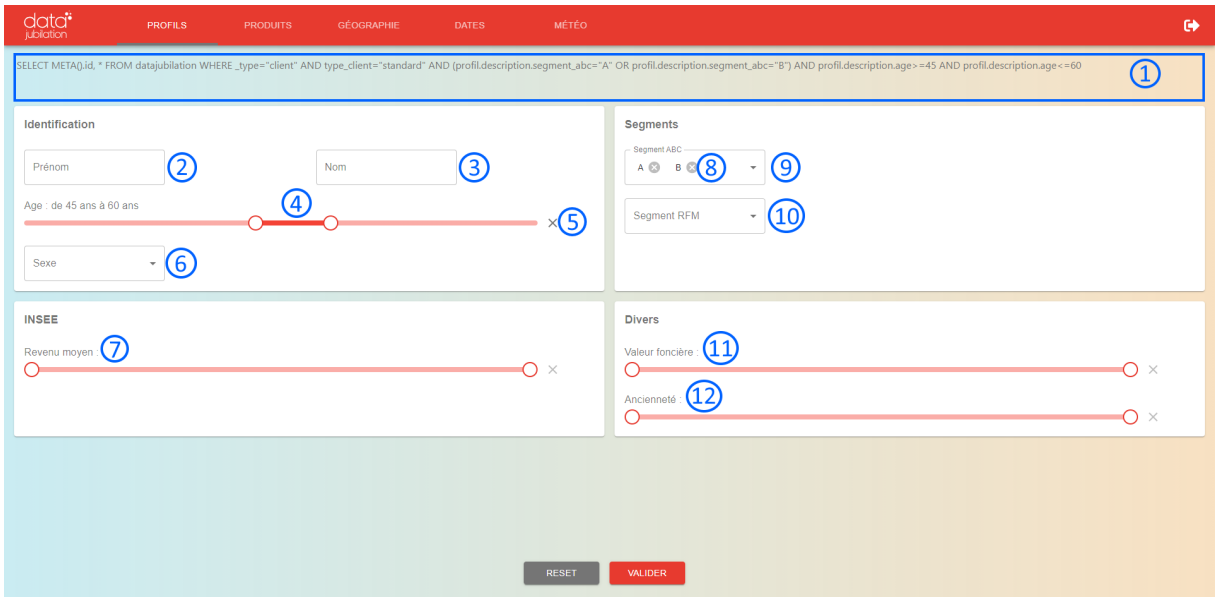


FIGURE A.20 – Interface de sélection d'une extraction de la base de données des clients.

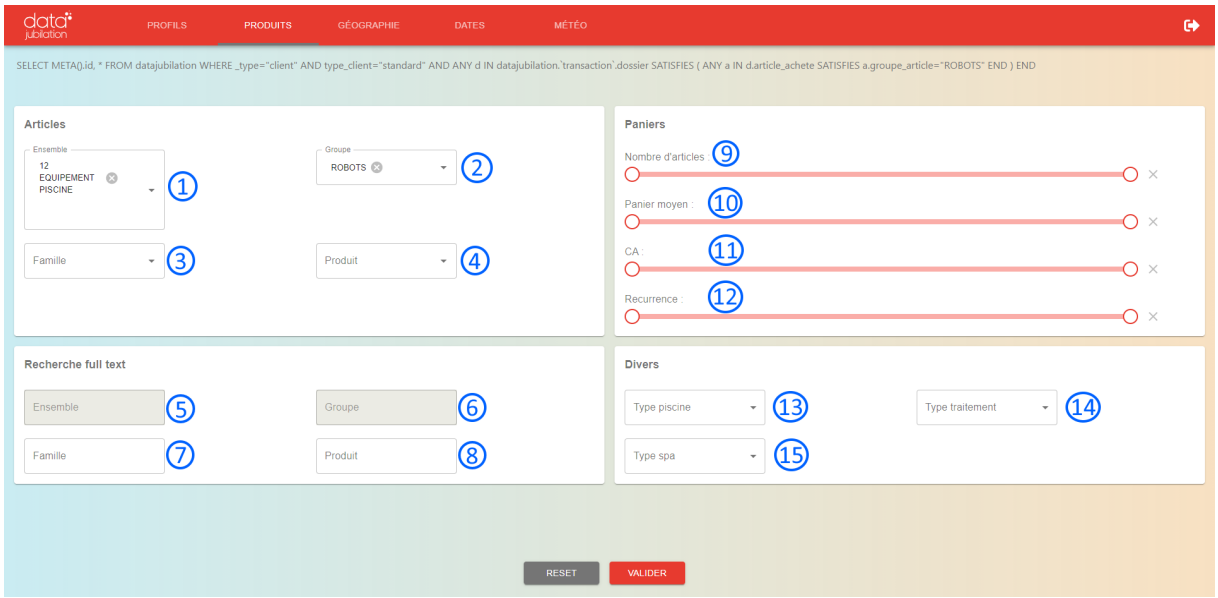


FIGURE A.21 – Interface de sélection d'une extraction de la base de données des clients.

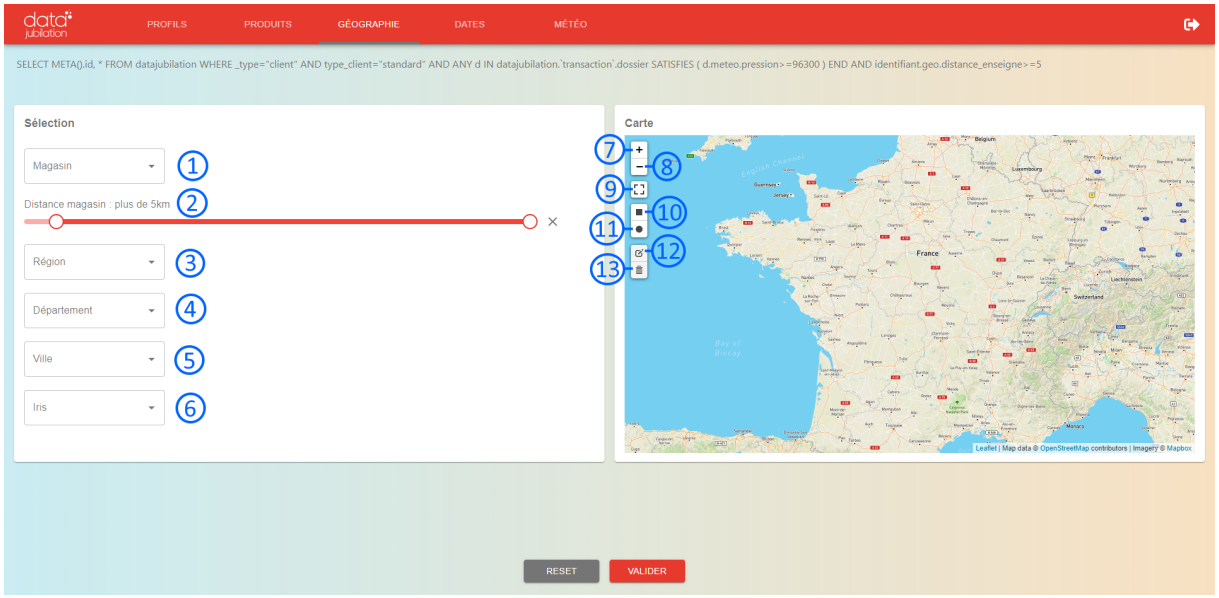


FIGURE A.22 – Interface de sélection d'une extraction de la base de données des clients.

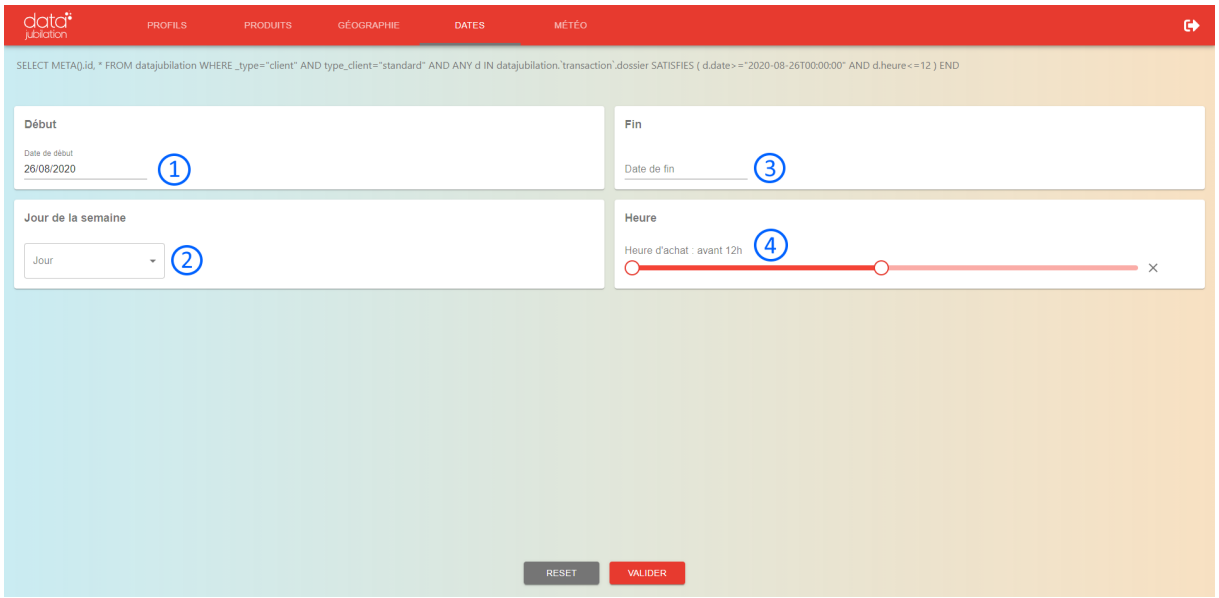


FIGURE A.23 – Interface de sélection d'une extraction de la base de données des clients.

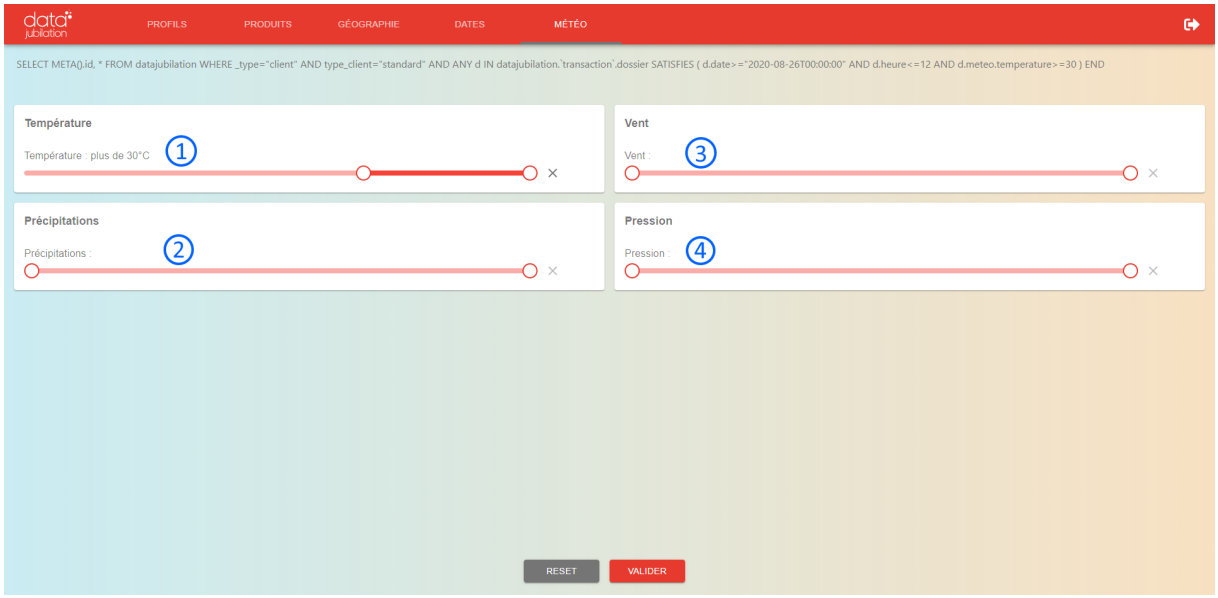


FIGURE A.24 – Interface de sélection d'une extraction de la base de données des clients.

A.4.3 Menu

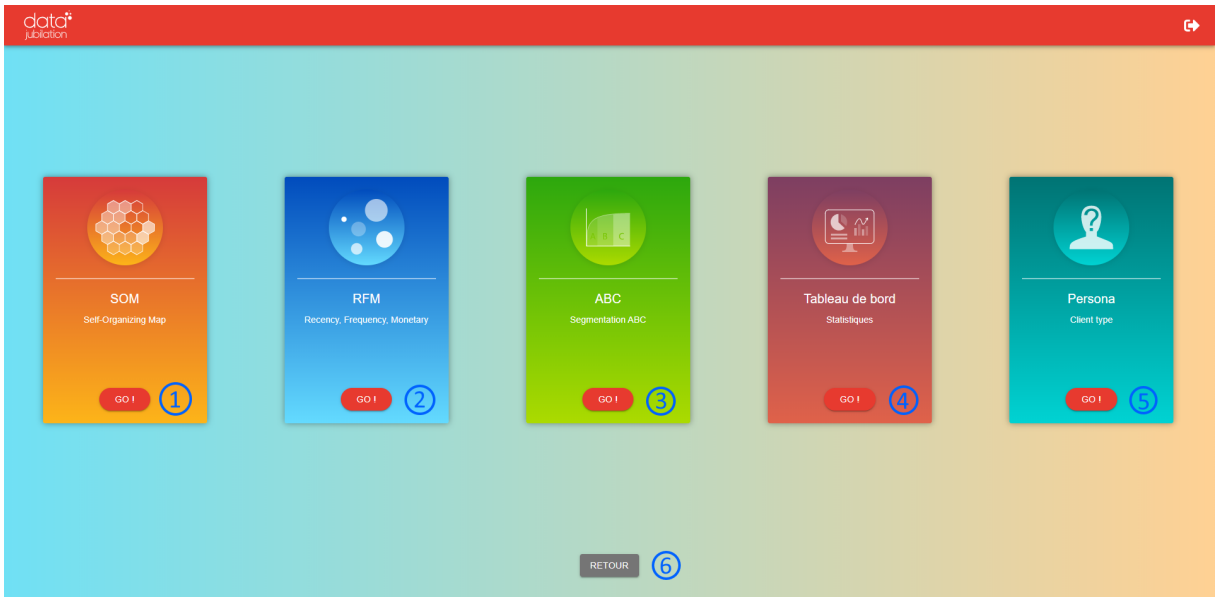


FIGURE A.25 – Interface du menu de sélection

A.4.4 Tableau de bord

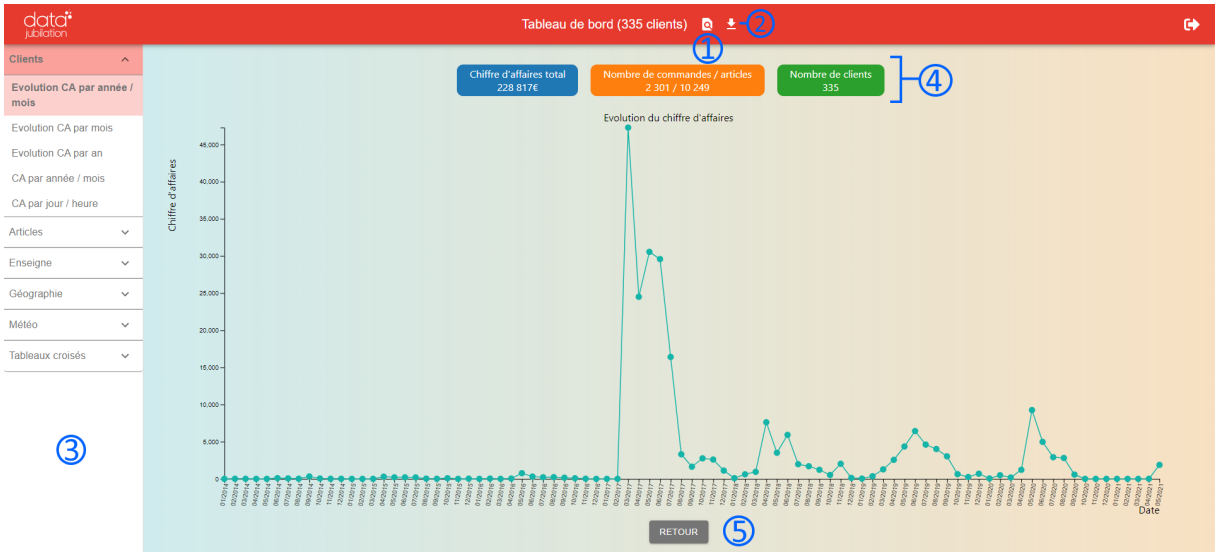


FIGURE A.26 – Interface de visualisation du Tableau de bord

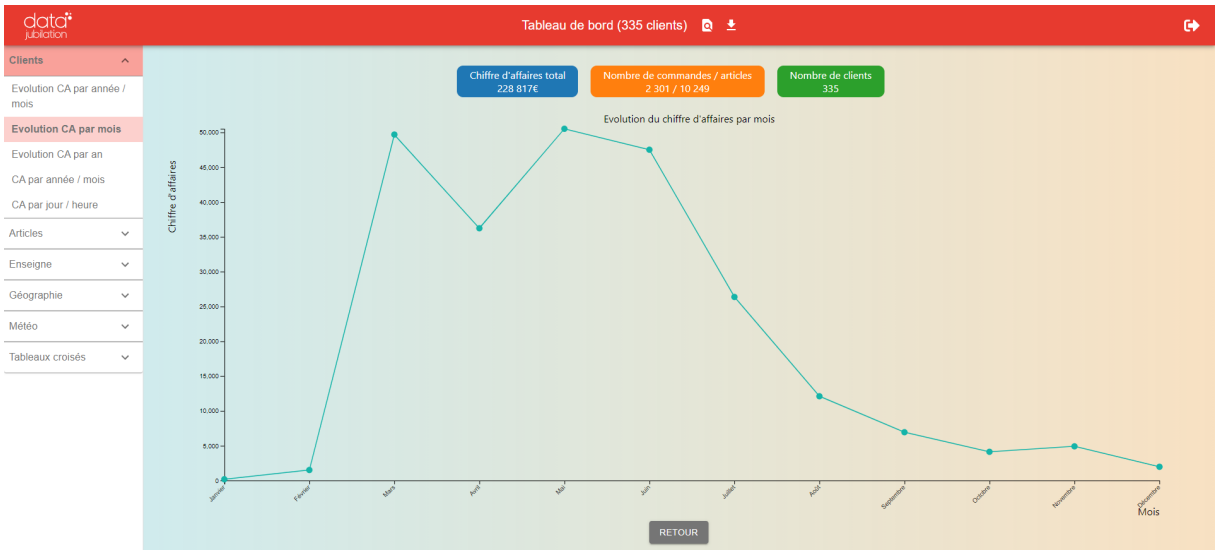


FIGURE A.27 – Interface de visualisation du Tableau de bord



FIGURE A.28 – Interface de visualisation du Tableau de bord

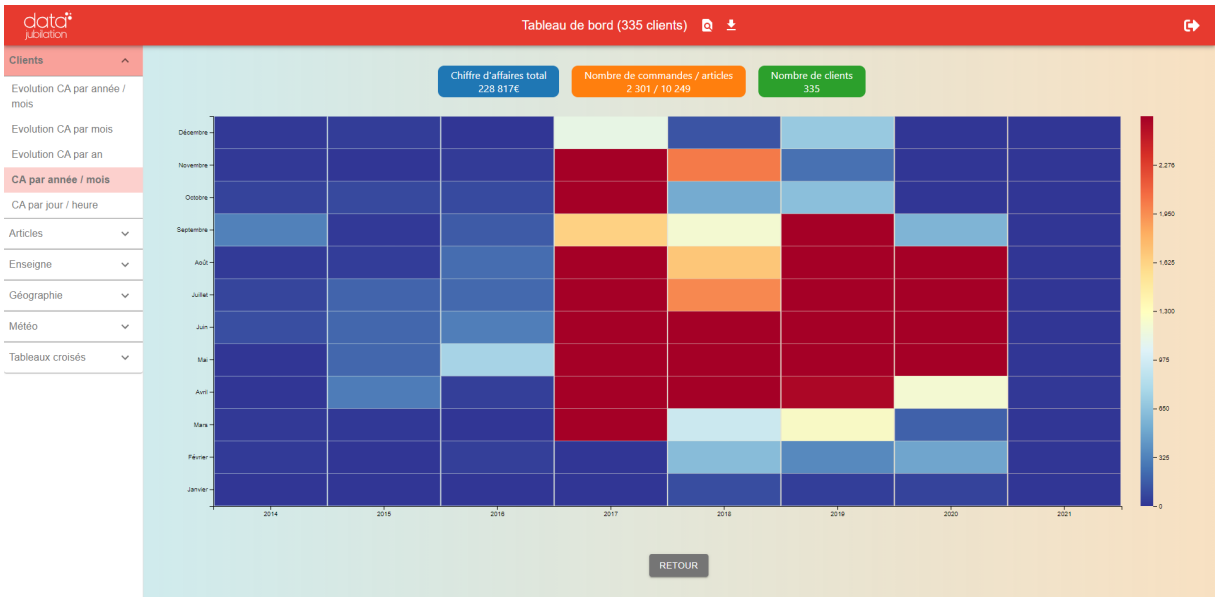


FIGURE A.29 – Interface de visualisation du Tableau de bord

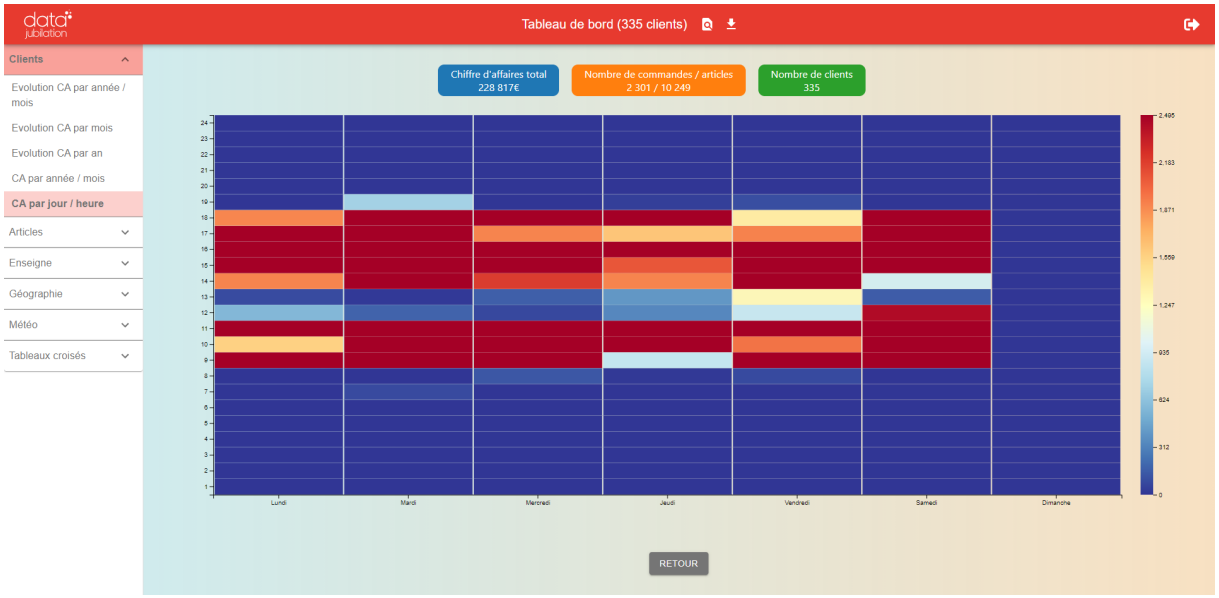


FIGURE A.30 – Interface de visualisation du Tableau de bord

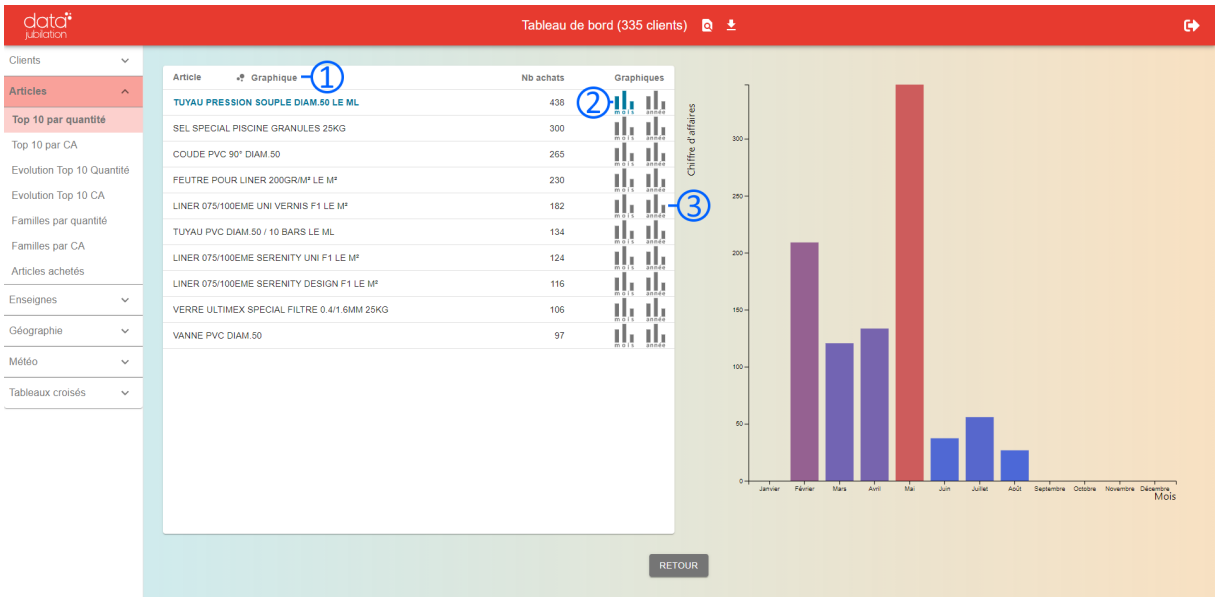


FIGURE A.31 – Interface de visualisation du Tableau de bord

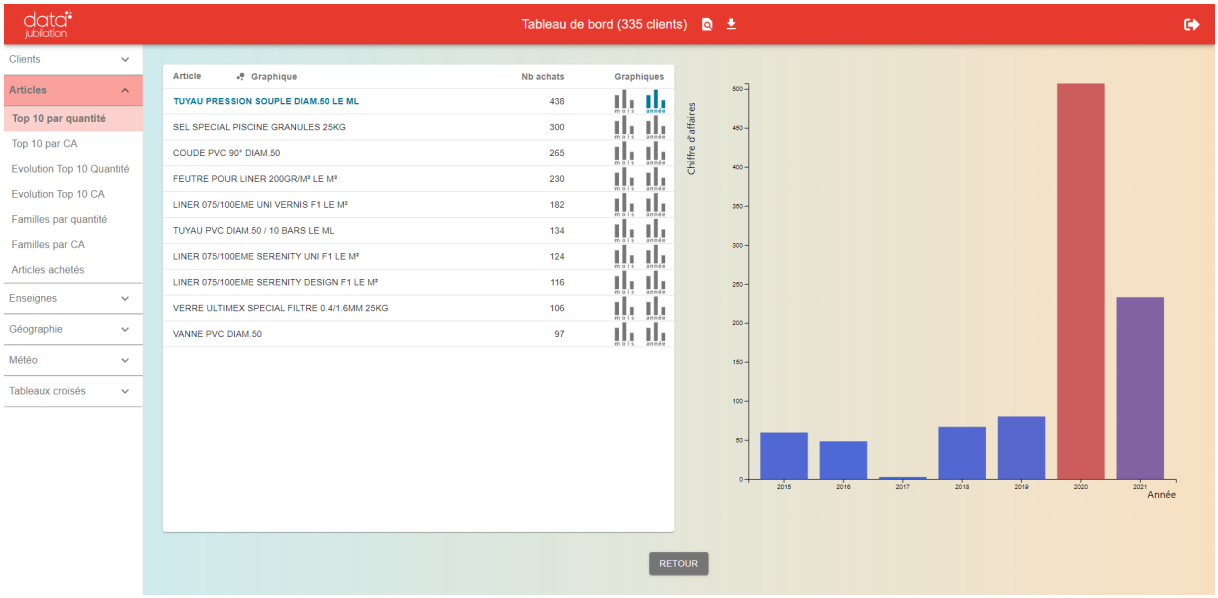


FIGURE A.32 – Interface de visualisation du Tableau de bord

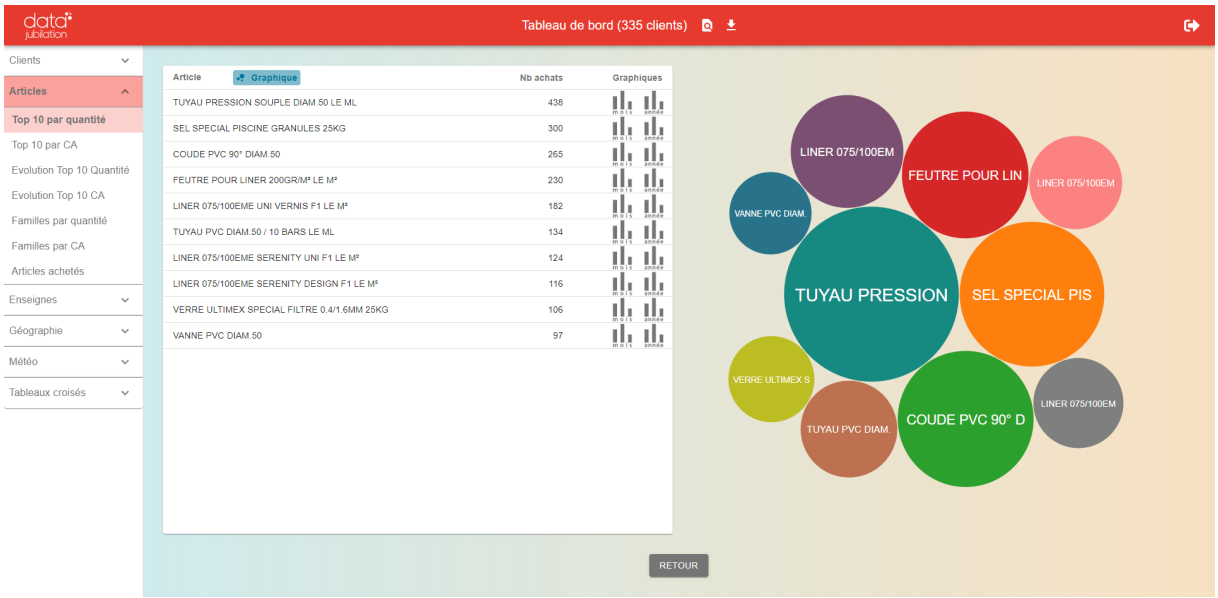


FIGURE A.33 – Interface de visualisation du Tableau de bord

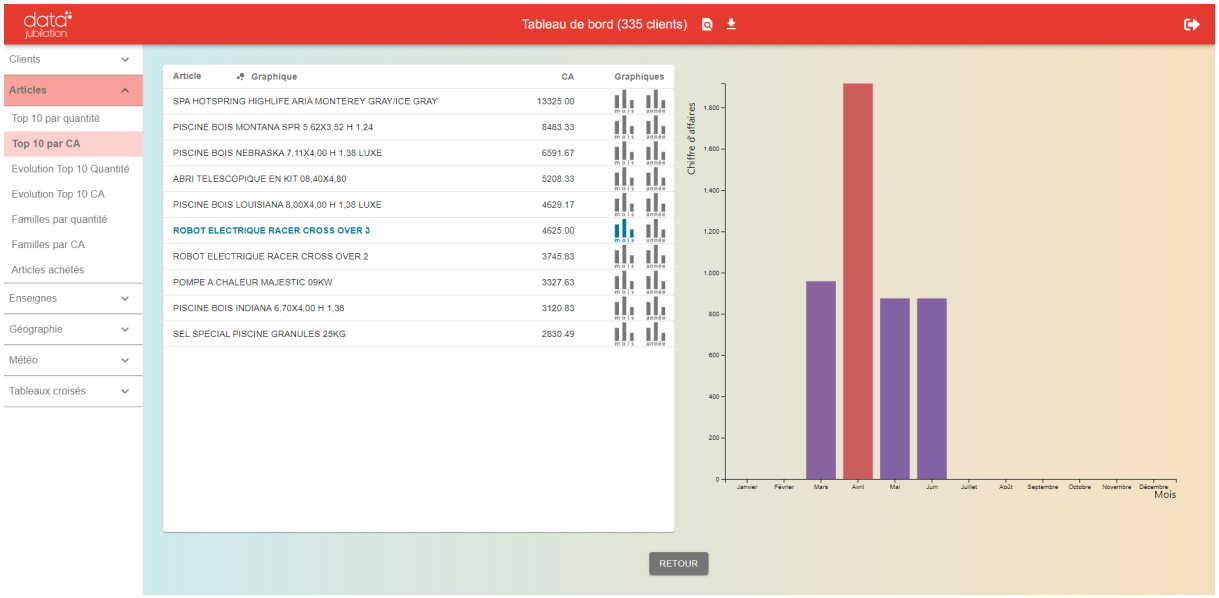


FIGURE A.34 – Interface de visualisation du Tableau de bord

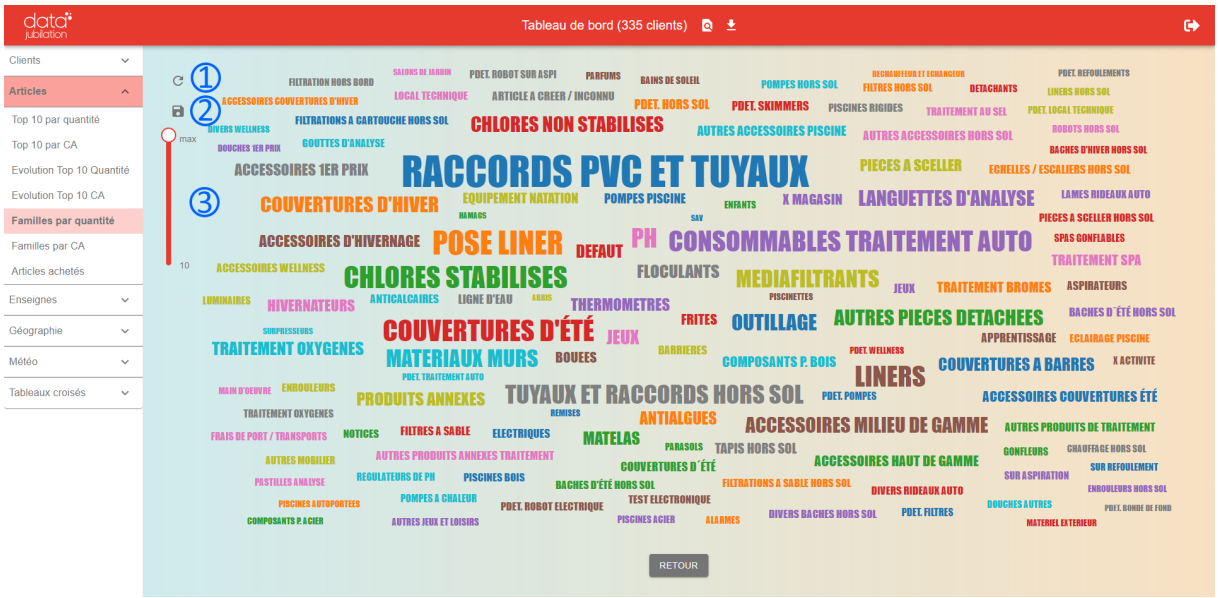


FIGURE A.35 – Interface de visualisation du Tableau de bord

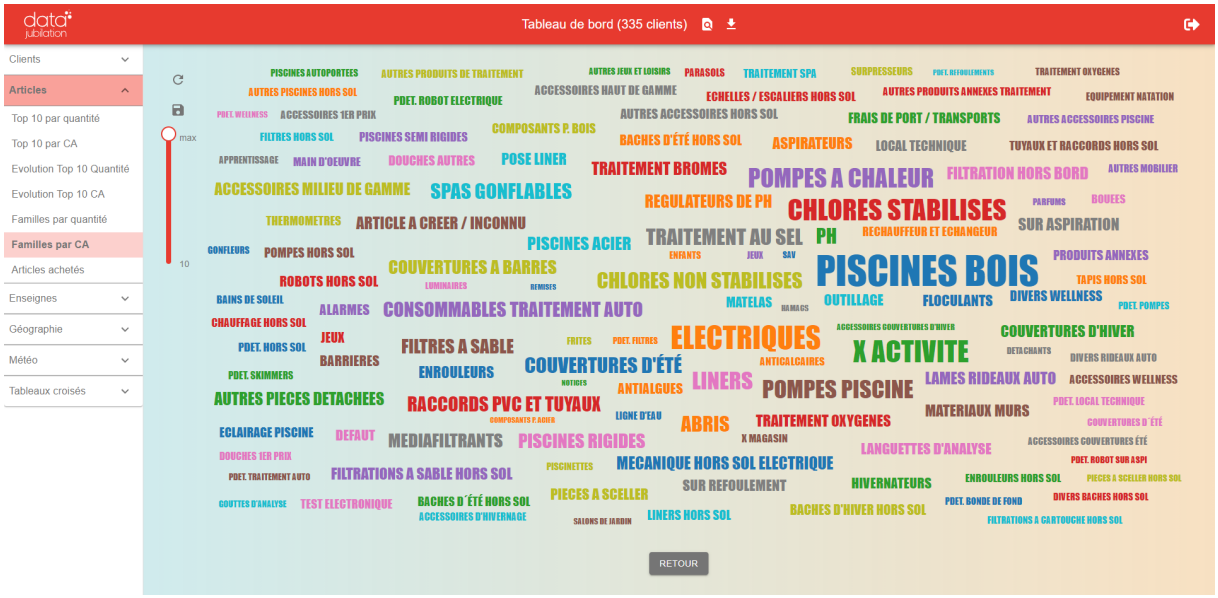


FIGURE A.36 – Interface de visualisation du Tableau de bord

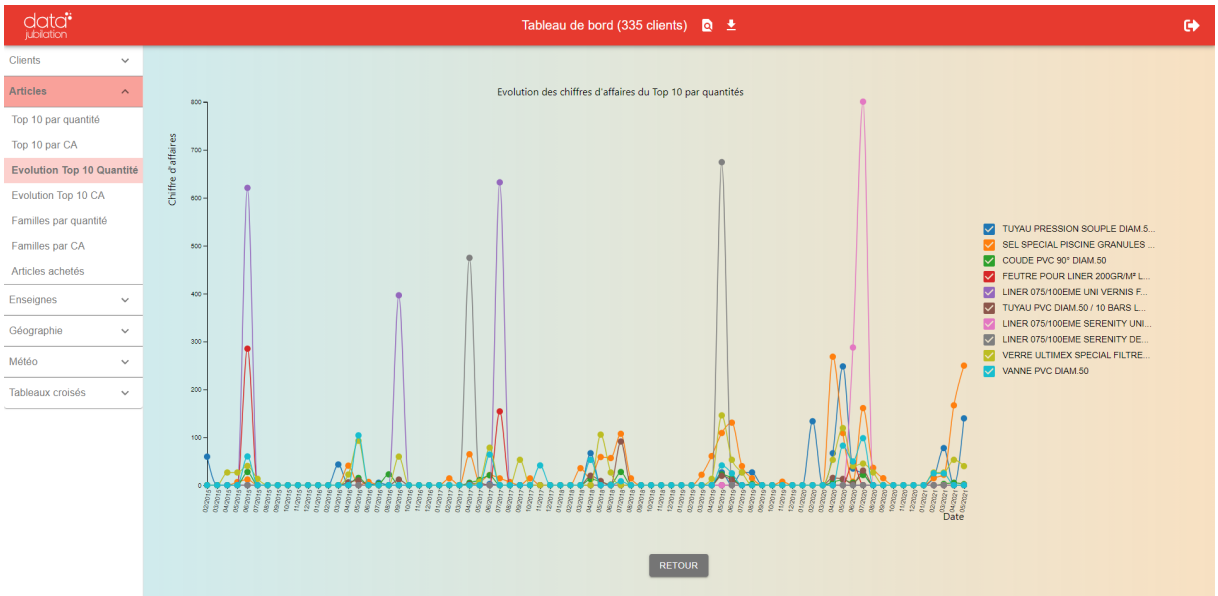


FIGURE A.37 – Interface de visualisation du Tableau de bord

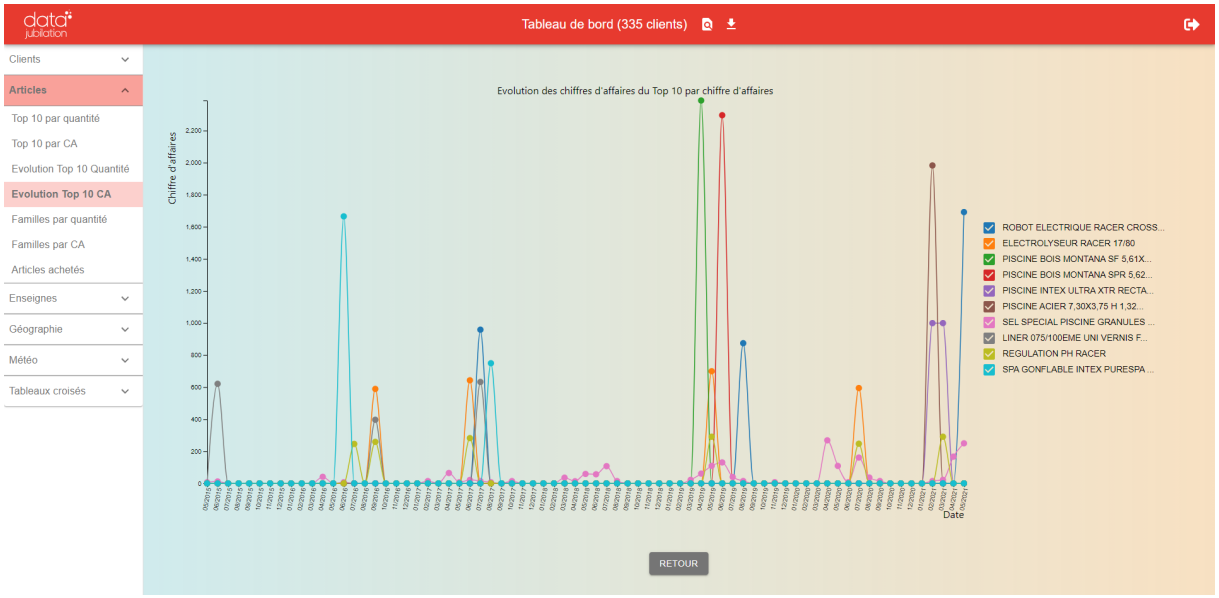


FIGURE A.38 – Interface de visualisation du Tableau de bord



FIGURE A.39 – Interface de visualisation du Tableau de bord



FIGURE A.40 – Interface de visualisation du Tableau de bord

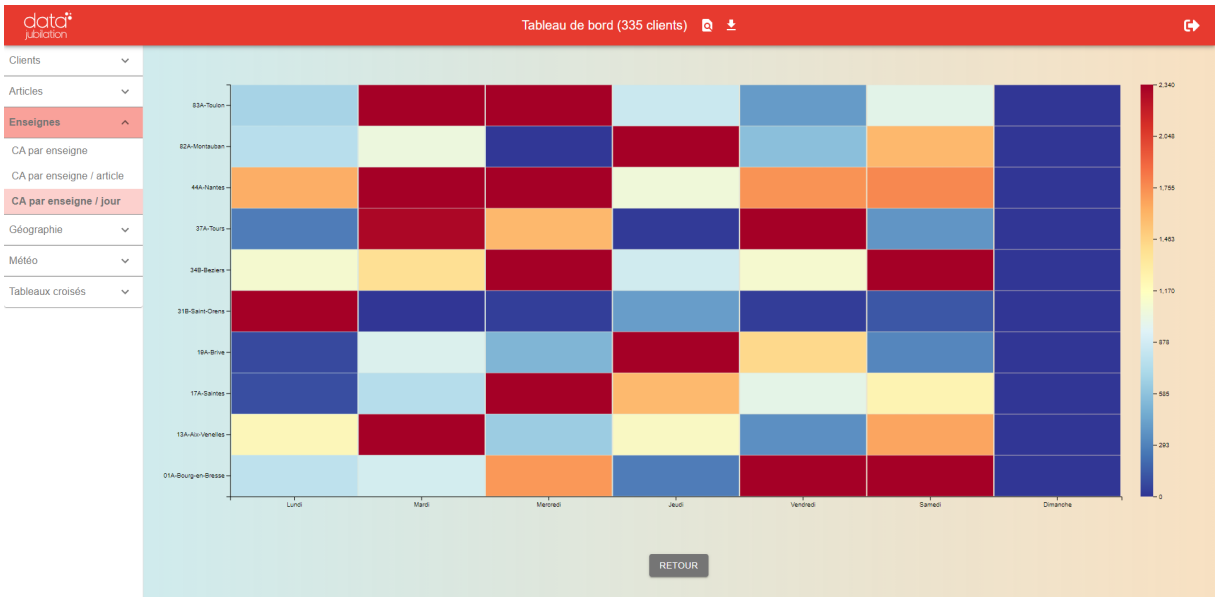


FIGURE A.41 – Interface de visualisation du Tableau de bord

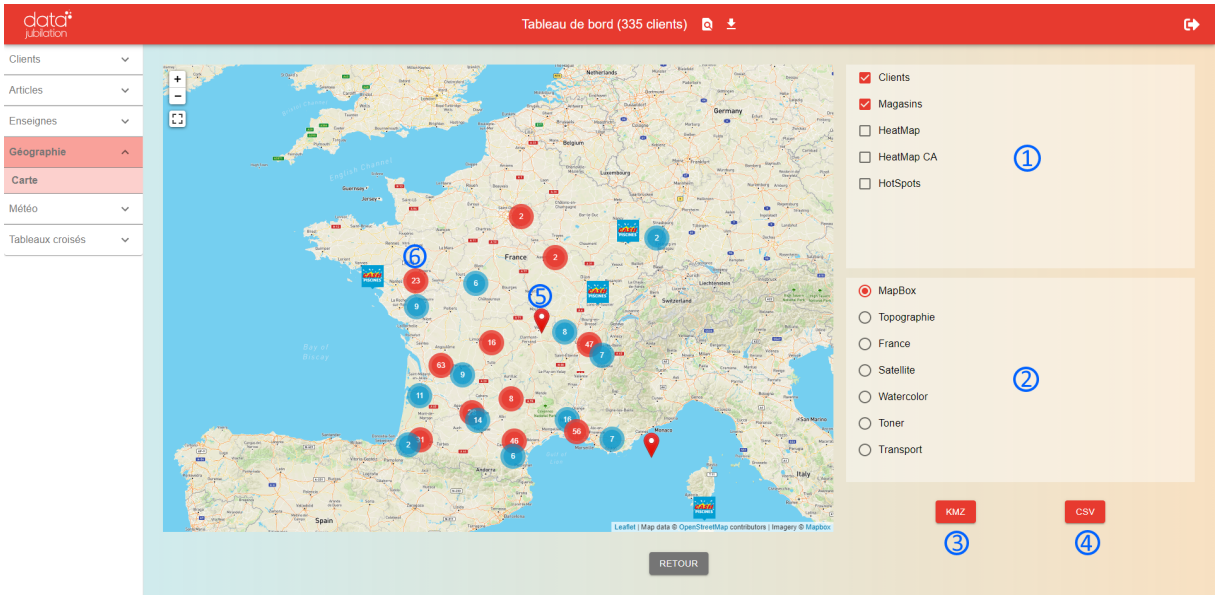


FIGURE A.42 – Interface de visualisation du Tableau de bord

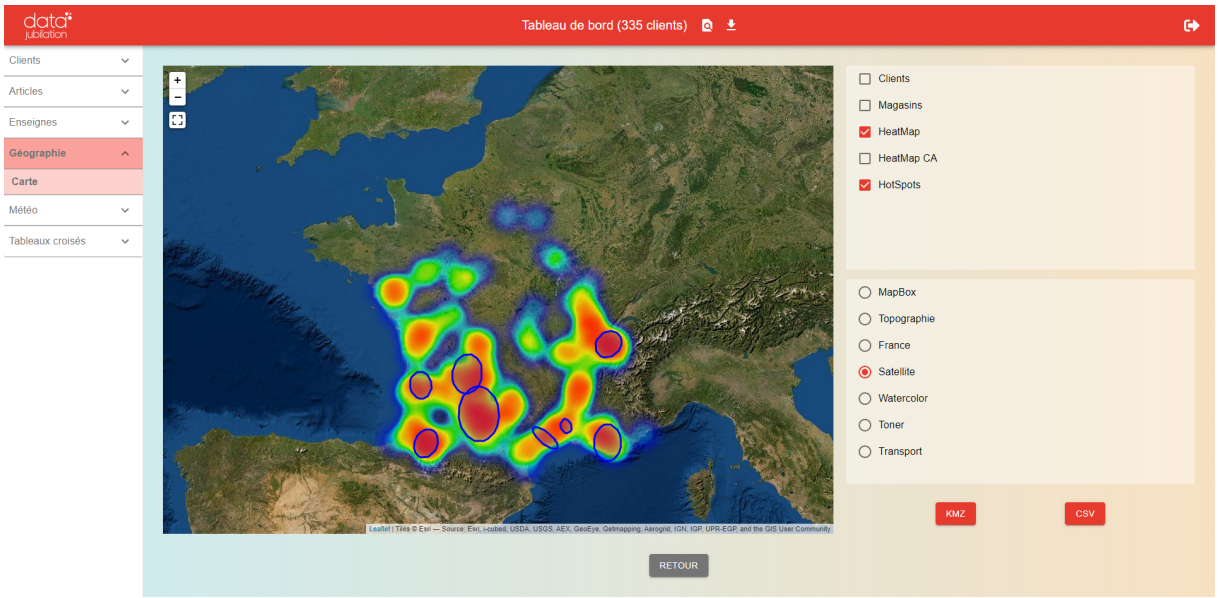


FIGURE A.43 – Interface de visualisation du Tableau de bord

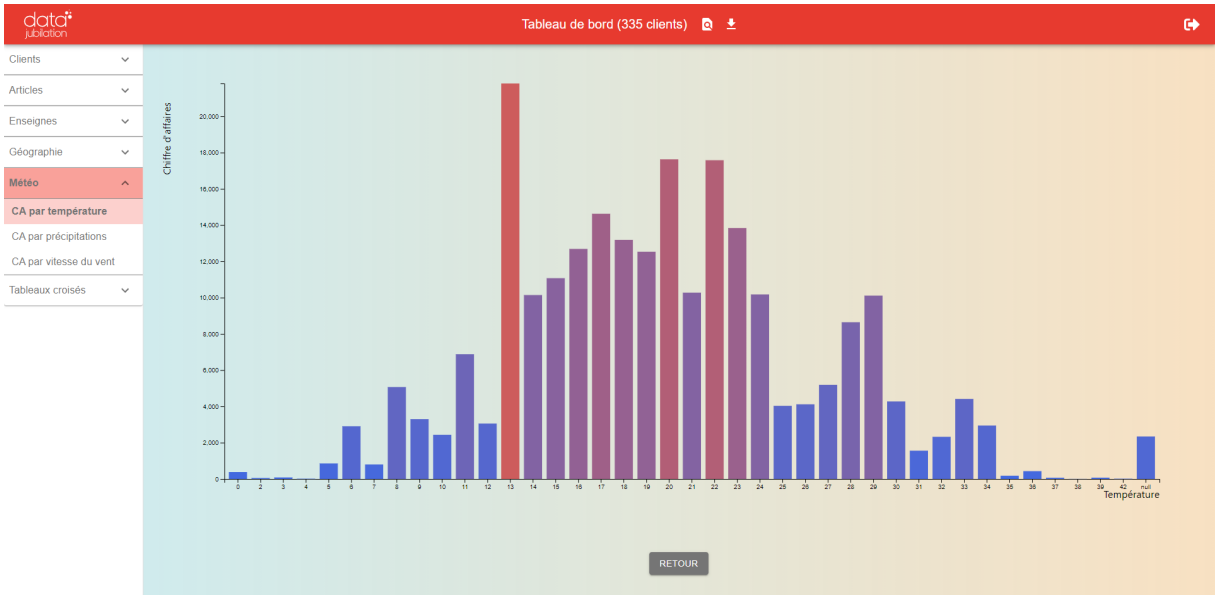


FIGURE A.44 – Interface de visualisation du Tableau de bord

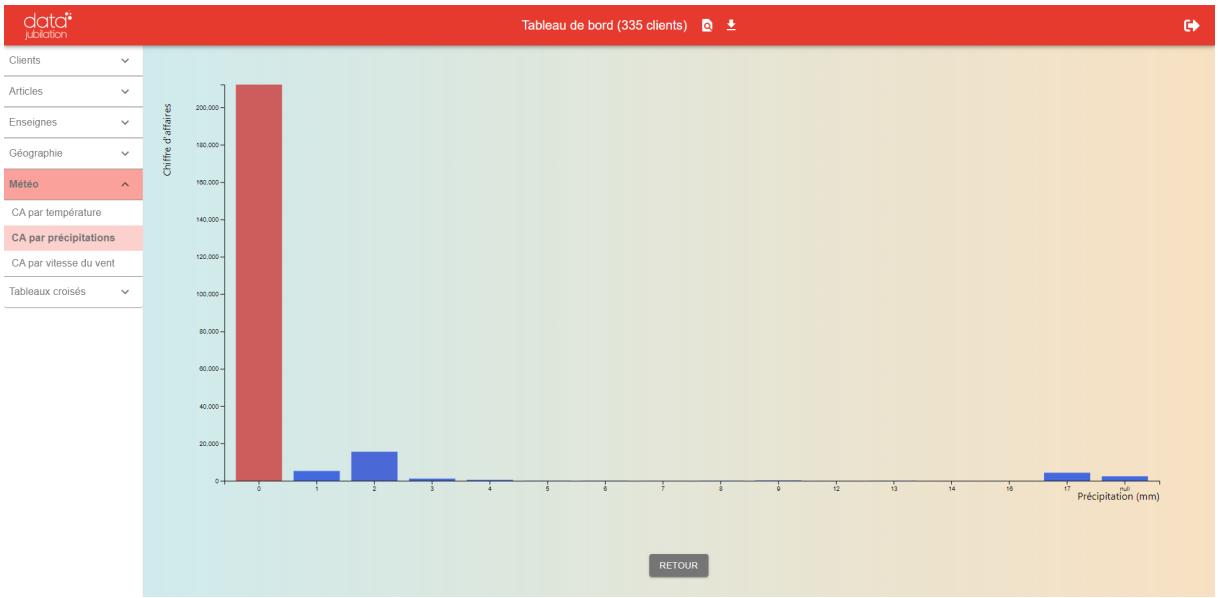


FIGURE A.45 – Interface de visualisation du Tableau de bord



FIGURE A.46 – Interface de visualisation du Tableau de bord

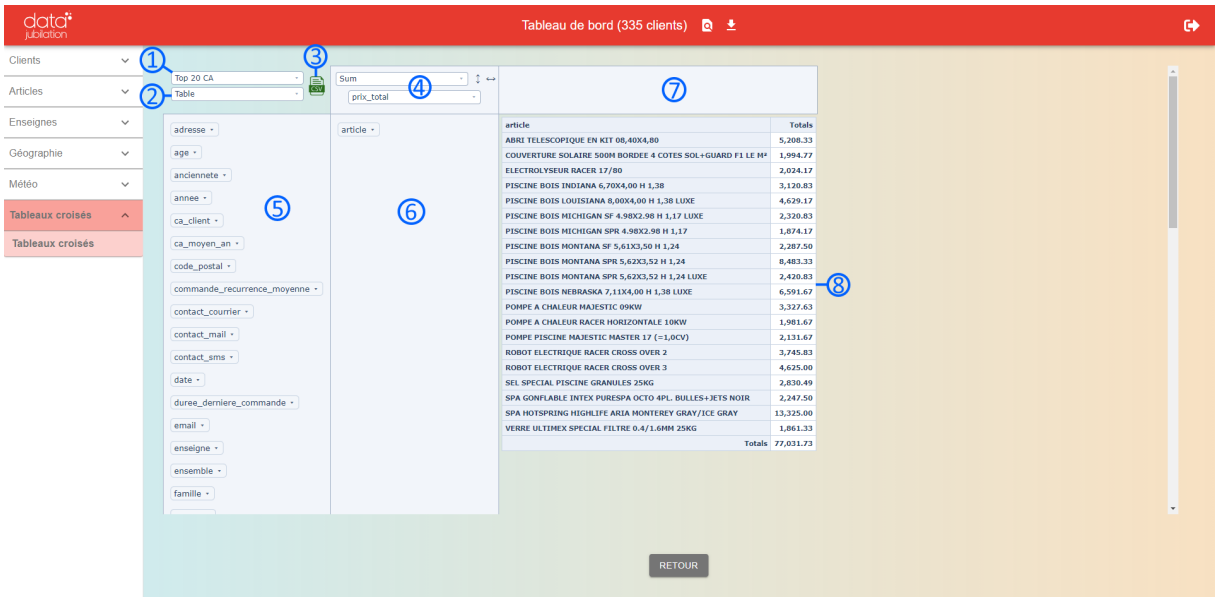


FIGURE A.47 – Interface de visualisation du Tableau de bord

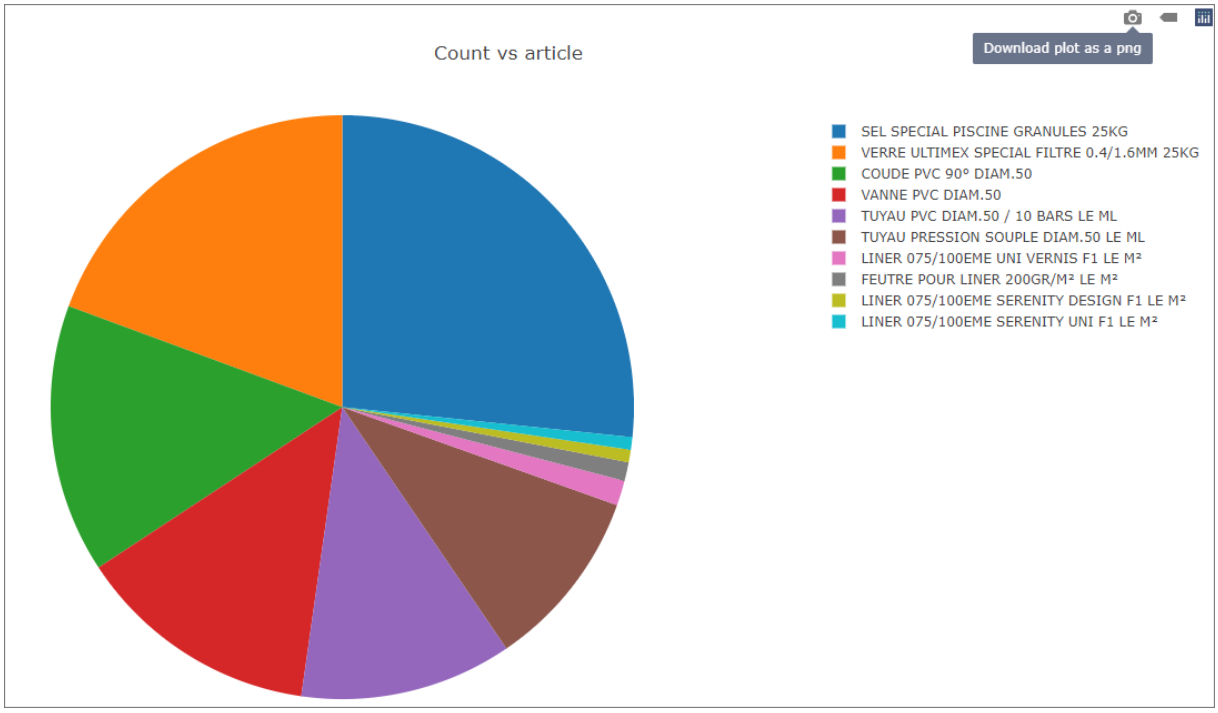


FIGURE A.48 – Interface de visualisation du Tableau de bord

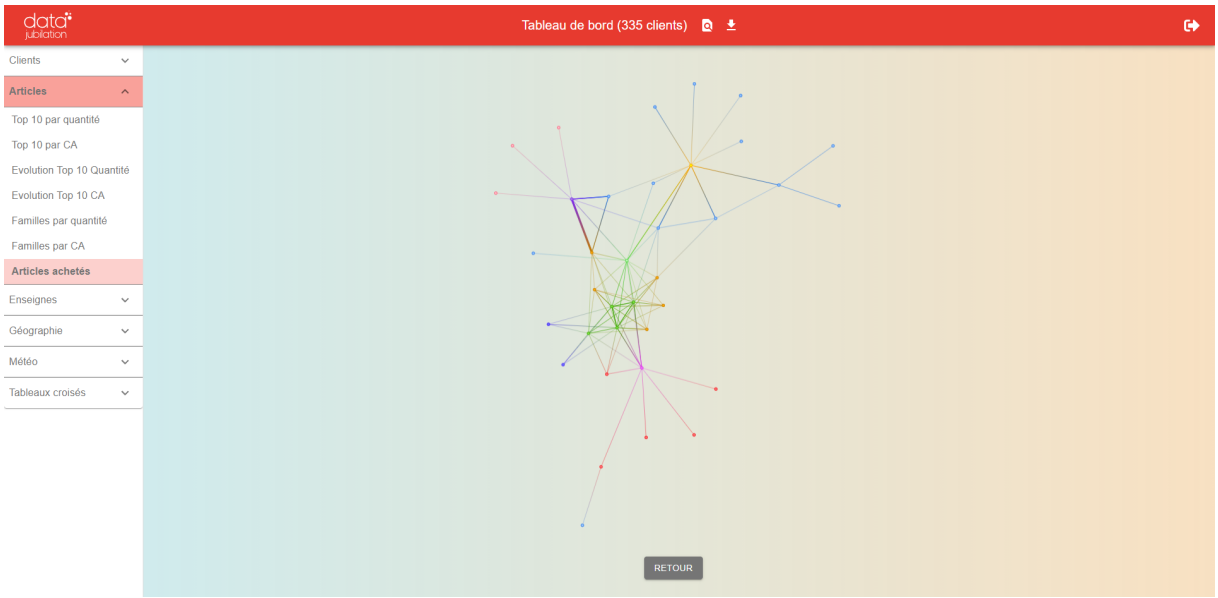


FIGURE A.49 – Interface de visualisation du Tableau de bord

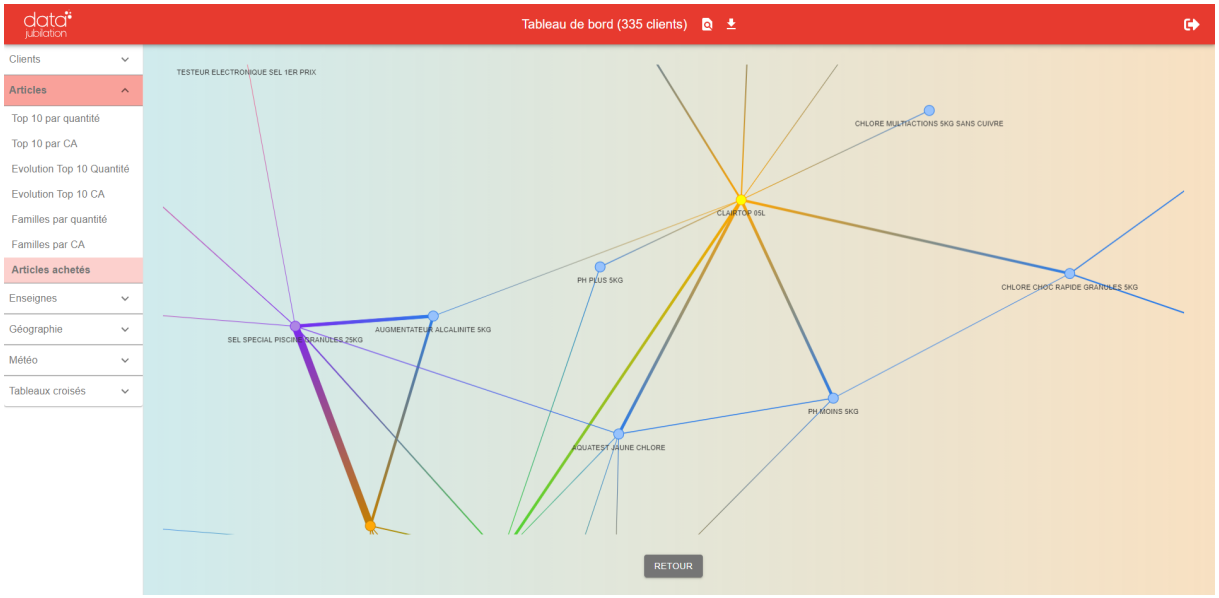


FIGURE A.50 – Interface de visualisation du Tableau de bord

A.4.5 Persona

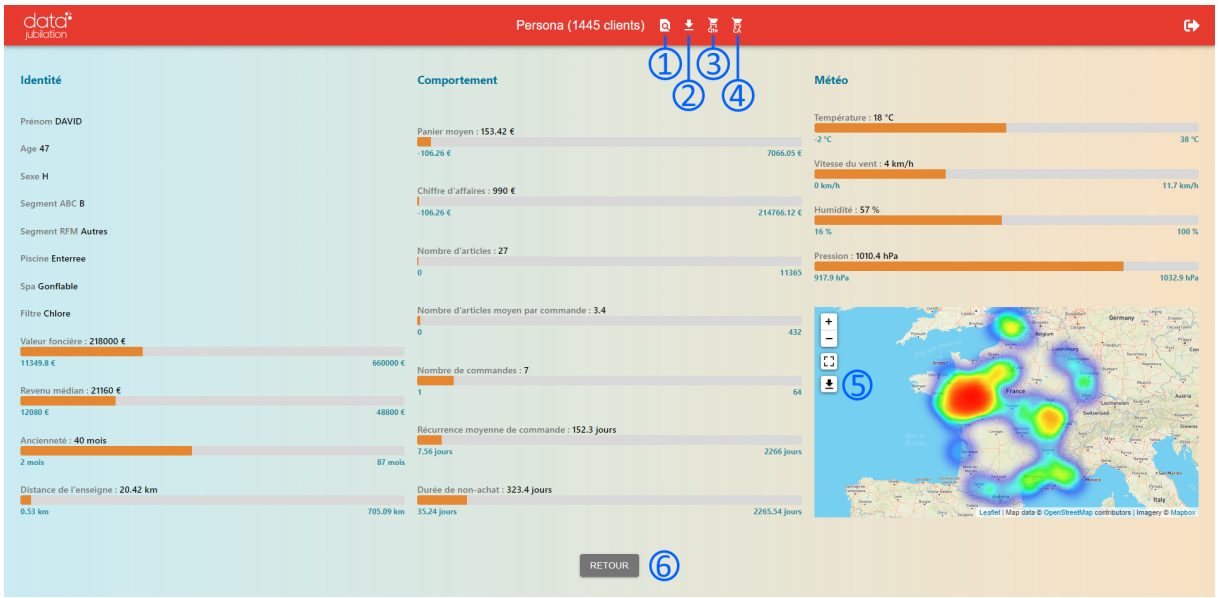


FIGURE A.51 – Interface de visualisation du Persona

A.4.6 ABC

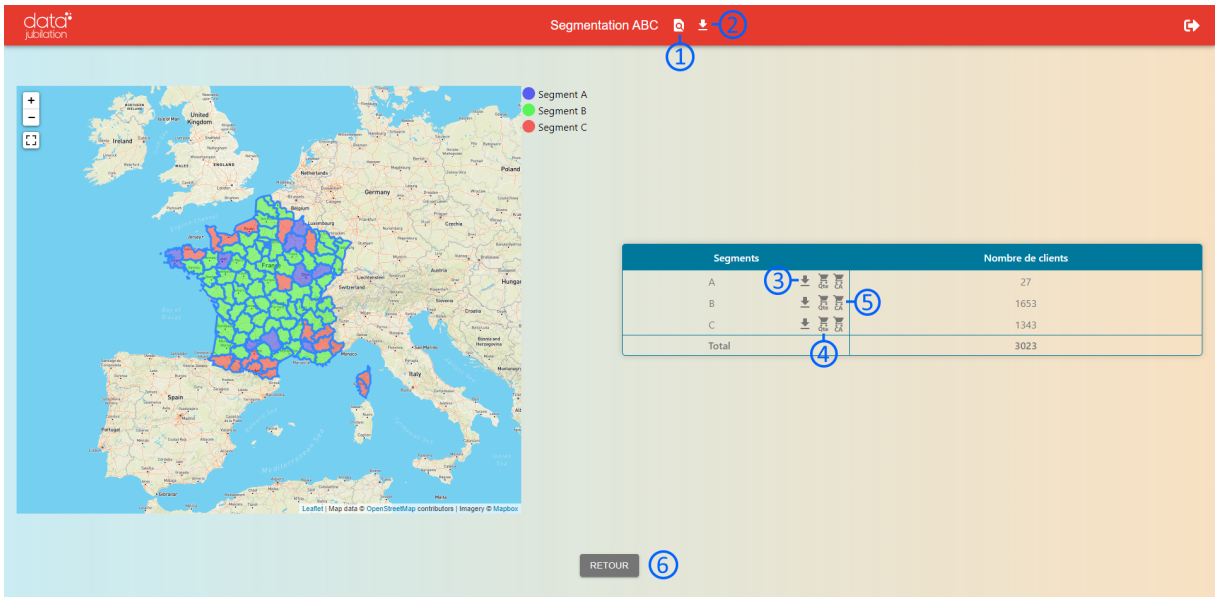


FIGURE A.52 – Interface de visualisation ABC

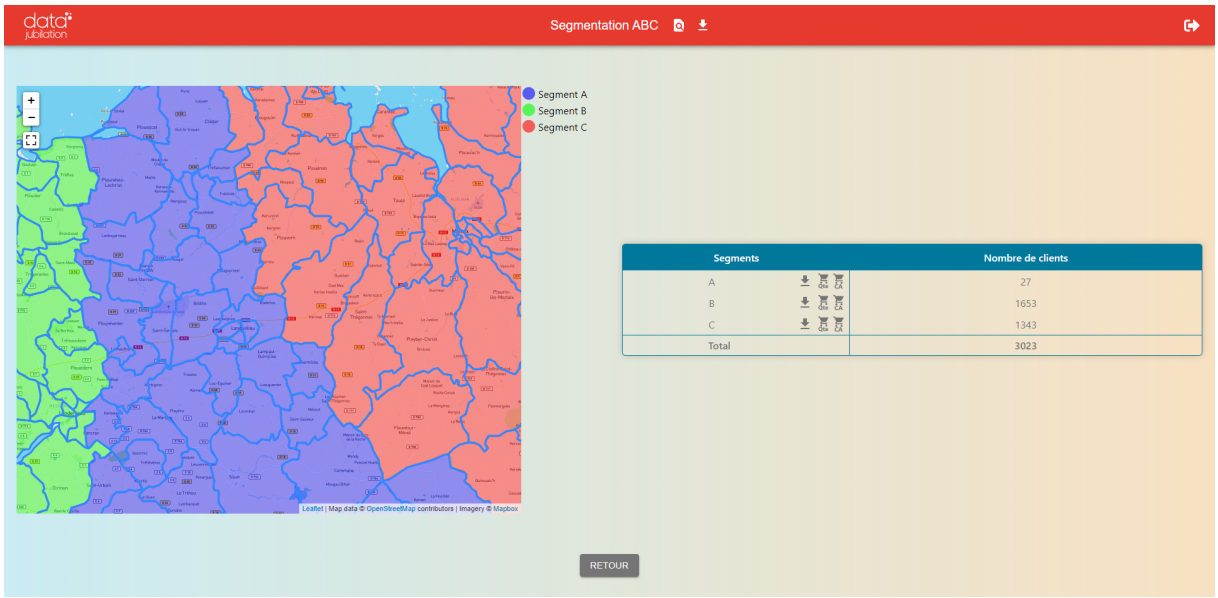


FIGURE A.53 – Interface de visualisation ABC

A.4.7 RFM

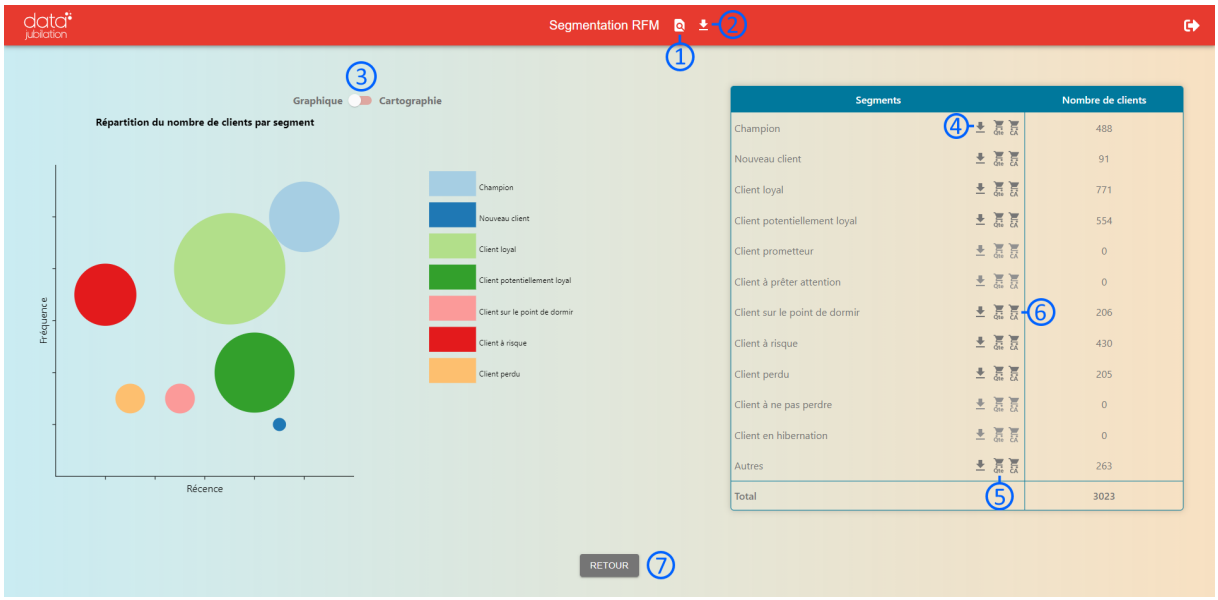


FIGURE A.54 – Interface de visualisation RFM

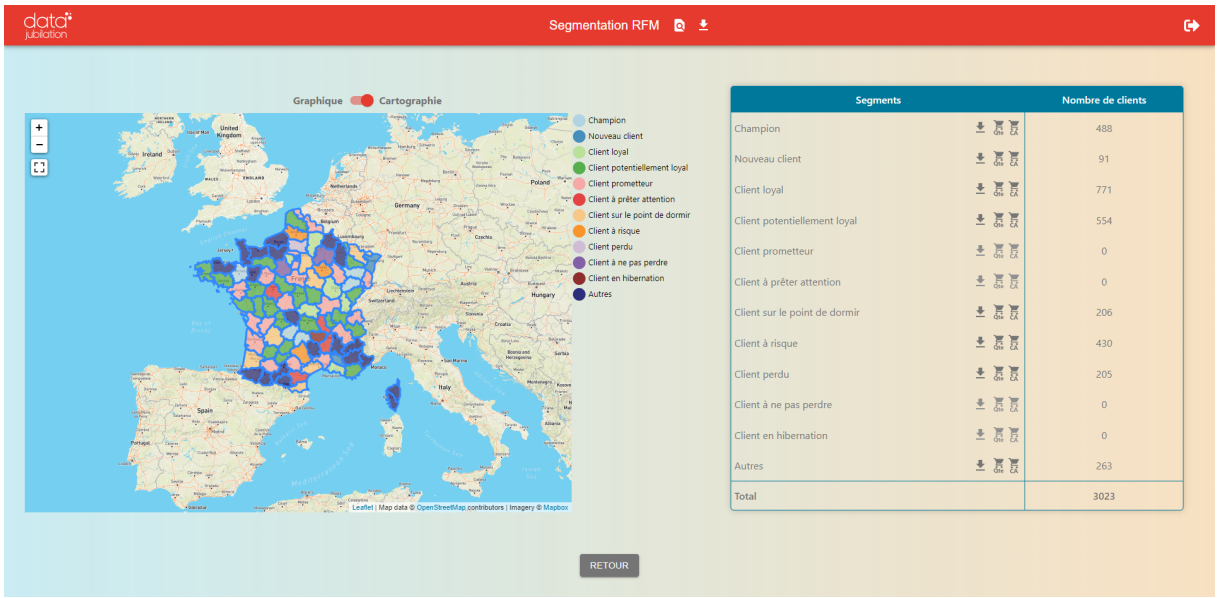


FIGURE A.55 – Interface de visualisation RFM

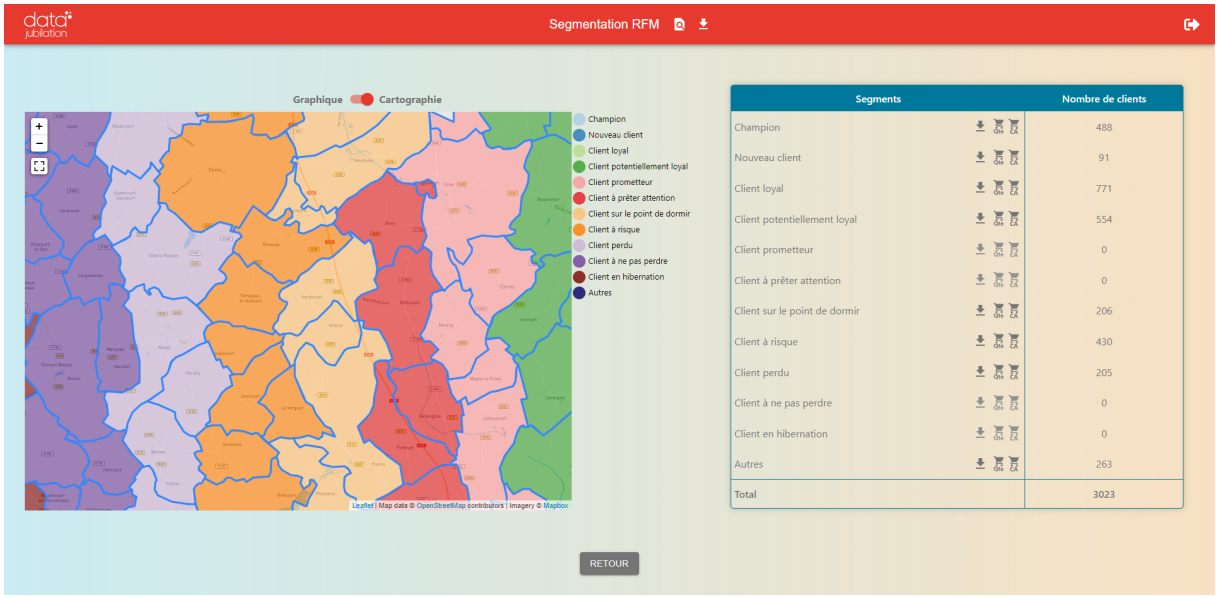


FIGURE A.56 – Interface de visualisation RFM

A.4.8 SOM

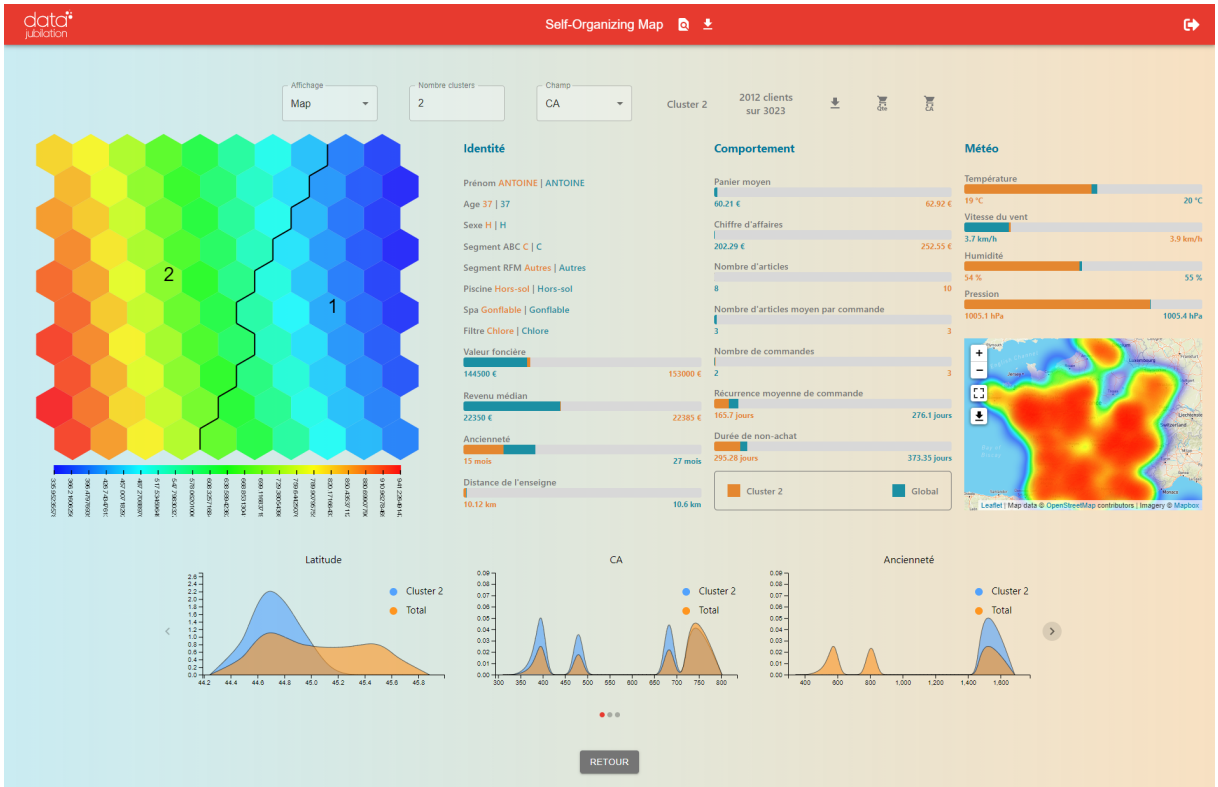


FIGURE A.57 – Interface de visualisation SOM

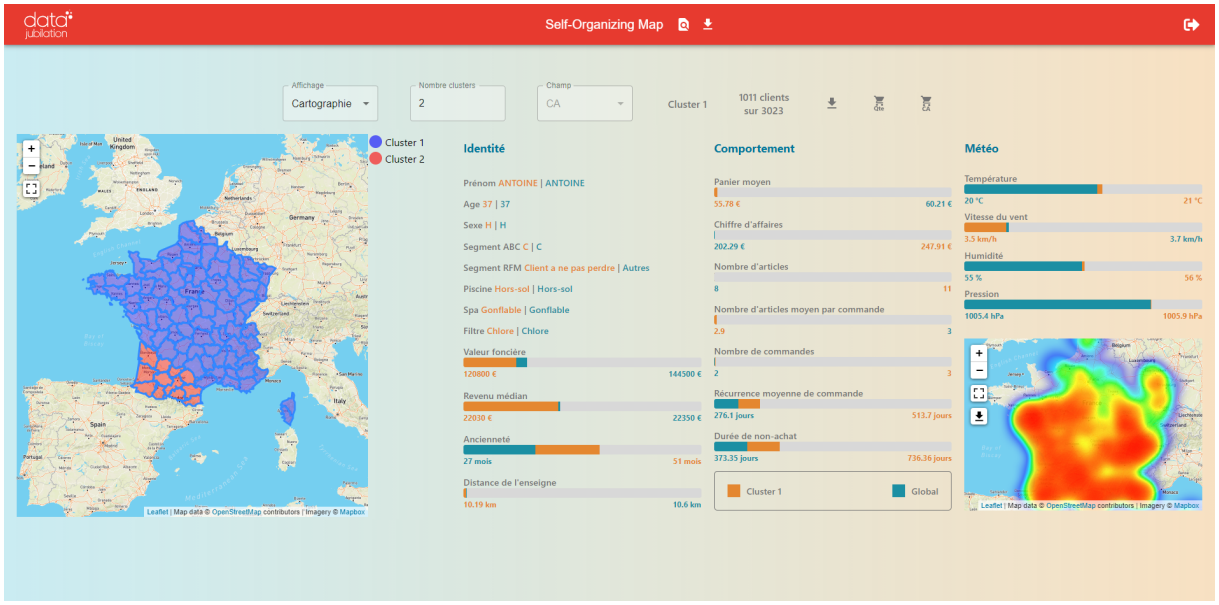


FIGURE A.58 – Interface de visualisation SOM

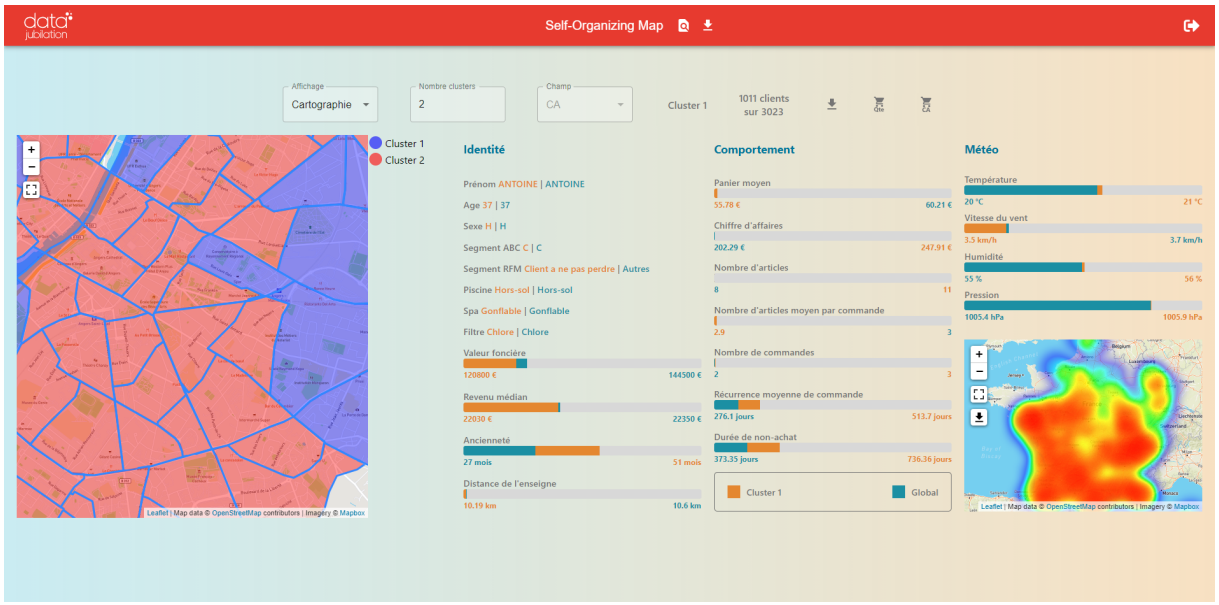


FIGURE A.59 – Interface de visualisation SOM

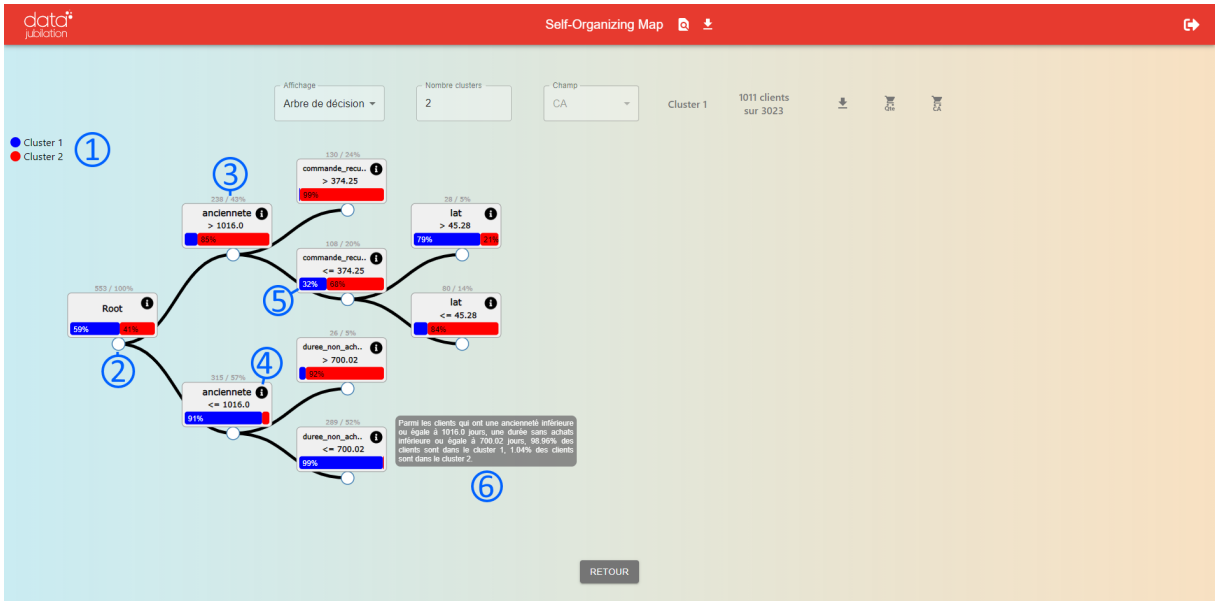


FIGURE A.60 – Interface de visualisation SOM

COLLEGE ED MATHEMATIQUES
DOCTORAL STIC
PAYS DE LA LOIRE

université angers

Titre : Extraction et gestion de la connaissance clients par des méthodes d'apprentissage non supervisées

Mot clés : Self-Organizing Maps, optimisation, CMA-ES, segmentation, intelligence artificielle, données hétérogènes

Résumé : Cette thèse explore de nouvelles approches pour améliorer les cartes auto-organisatrices (Self-Organizing Maps, SOM) dans un contexte industriel de forte hétérogénéité des données. Après une étude approfondie des fondements théoriques et des méthodes d'optimisation, plusieurs contributions sont proposées : une gestion unifiée des données manquantes, qualitatives, textuelles et temporelles ; un ensemble cohérent de métriques normalisées pour évaluer la qualité topologique et quantitative des cartes ; et une approche d'hyperparamétrage automatique via l'algorithme SMAC. La thèse introduit ensuite un paradigme alternatif : les SOM génératives continues, où les poids de la carte sont produits par une fonction neuronale paramétrée, optimisée sans gradient par CMA-ES. Cette reformulation améliore la compacité, la stabilité et la scalabilité du modèle. Enfin, la validation industrielle a permis de développer une application, intégrant des enrichissements de données, ainsi que des modules de segmentation ABC, RFM et SOM. Les résultats démontrent la pertinence scientifique et opérationnelle des approches proposées, conciliant rigueur algorithmique et applicabilité dans des environnements réels de décision.

Title: Customer Knowledge Extraction and Management through Unsupervised Learning Methods

Keywords: Self-Organizing Maps, optimization, CMA-ES, segmentation, artificial intelligence, heterogeneous data

Abstract: This thesis explores novel approaches to improve Self-Organizing Maps (SOM) within an industrial context characterized by highly heterogeneous data. Following an in-depth study of the theoretical foundations and optimization methods, several contributions are proposed: a unified framework for handling missing, qualitative, textual, and temporal data; a consistent set of normalized metrics for assessing both topological and quantitative map quality; and an automatic hyperparameter optimization approach based on the SMAC algorithm. The thesis then introduces an alternative paradigm: continuous generative SOMs, in which the map weights are produced by a parameterized neural function optimized without gradients using the CMA-ES algorithm. This reformulation enhances model compactness, stability, and scalability. Finally, an industrial validation led to the development of an application integrating data enrichments, as well as ABC, RFM, and SOM segmentation modules. The results demonstrate the scientific and operational relevance of the proposed approaches, successfully combining algorithmic rigor with practical applicability in real-world decision-making environments.