Thèse
de doctorat
de l'UTT

Guillaume LEBONVALLET

Intelligence artificielle embarquée pour la surveillance des infrastructures ferroviaires

Champ disciplinaire :
Sciences pour l'Ingénieur

2025TROY0037 **Année 2025**



utt
UNIVERSITÉ DE TECHNOLOGIE
TROYES

THESE

pour l'obtention du grade de

DOCTEUR
de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

en SCIENCES POUR L'INGENIEUR

Spécialité : OPTIMISATION ET SURETE DES SYSTEMES

présentée et soutenue par

Guillaume LEBONVALLET

le 11 décembre 2025

Intelligence artificielle embarquée pour la surveillance des infrastructures ferroviaires

JURY

M. Abdel-Ouahab BOUDRAA	PROFESSEUR DES UNIVERSITES	Président
M. Jaafar GABER	MAITRE DE CONFERENCES	Rapporteur
M. Kim-Phuc TRAN	MAITRE DE CONFERENCES	Rapporteur
Mme Farah MOURAD-CHEHADE	MAITRE DE CONFERENCES	Examinatrice
M. Faicel HNAIEN	PROFESSEUR DES UNIVERSITES	Directeur de thèse
M. Hichem SNOUSSI	PROFESSEUR DES UNIVERSITES	Directeur de thèse

Remerciements

Tout d'abord, je souhaite adresser mes remerciements les plus sincères à M. Abdel-Ouahab Boudraa, Professeur des universités à l'École Navale, et Mme Farah Mourad-Chehade, Maître de conférences HDR à l'Université de technologie de Troyes, pour avoir accepté leur participation au jury de cette thèse en tant qu'examinateur.

Je souhaite ensuite exprimer toute ma gratitude à M. Jaafar Gaber, Maître de conférences HDR à l'Université de technologie de Belfort-Montbéliard, et M. Kim-Phuc Tran, Maître de conférences HDR à l'Université de Lille (ENSAIT), pour avoir accepté d'être les rapporteurs de cette thèse.

Je remercie également mes directeurs de thèse, Hichem Snoussi et Faicel Hnaien, pour leur encadrement tout au long de la thèse.

Je remercie tous les membres de l'équipe de VOSSLOH impliquée dans le projet RailMon : Anaïs Ménétrier, Victoria Simonetta, Brice Nélain, Sébastien Berge, Célestin Métral, Pierre Francolin, et ceux dont j'oublie le nom.

Je remercie tout particulièrement Malak Saiem et Luis Salazar Zendeja pour leur soutien et leur aide précieuse dans l'aboutissement de mon travail.

Enfin, je remercie tout les membres de l'UR LIST3N et de l'École Doctorale de l'UTT pour leur accueil, leur soutien et leur accompagnement tout au long de la thèse.

Résumé

Cette thèse s'inscrit dans le cadre du projet RailMon, porté par l'entreprise Vossloh et financé par Bpifrance dans le cadre du Plan d'Investissement d'Avenir (PIA4). L'objectif du projet est de développer un système de télésurveillance intelligente permettant le suivi des infrastructures ferroviaires, afin d'anticiper les besoins en maintenance et d'optimiser les opérations de maintenance. Pour ce faire, des accéléromètres connectés et autonomes en énergie ont été déployés sur les aiguillages ferroviaires, éléments particulièrement sensibles du réseau afin d'enregistrer les signaux vibratoires générés lors du passage des trains. Nous proposons un modèle léger et efficace basé sur Perceiver IO, un autoencodeur modulaire exploitant les mécanismes d'attention, afin de compresser les signaux sur ces capteurs aux ressources limitées, puis de les reconstruire après transmission. Nous intégrons également en son sein une capacité de détection précoce des signes de dégradation liés à l'usure des infrastructures, tout en préservant la légèreté du traitement embarqué. Enfin, nous introduisons un indicateur de vitesse de dégradation des aiguillages, fondé sur un apprentissage non supervisé sous contraintes, permettant d'évaluer l'évolution de leur état à partir des signaux mesurés.

Abstract

This thesis is part of the RailMon project, led by the company Vossloh and funded by Bpifrance as part of the Future Investment Plan (PIA4). The goal of the project is to develop an intelligent remote monitoring system for railway infrastructure, aimed at anticipating maintenance needs and optimizing maintenance operations. To achieve this, energy-autonomous connected accelerometers have been deployed on railway switches, which are particularly sensitive elements of the network, to record the vibratory signals generated during train passages. We propose a lightweight and efficient model based on Perceiver IO, a modular autoencoder leveraging attention mechanisms, to compress signals from these resource-constrained sensors and then reconstruct them after transmission. The system also includes an early detection capability for signs of degradation associated with infrastructure wear, while maintaining lightweight on-board processing. Finally, we introduce a deterioration speed indicator for railway switches, based on constrained unsupervised learning, which allows for the assessment of their condition evolution from the measured signals.

Table des matières

Table des figures ix

Table des tableaux xi

1 Introduction générale 1
- 1.1 Contexte . . . 1
- 1.2 Le projet RailMon . . . 3
- 1.3 Problématique et objectifs de la thèse . . . 5
- 1.4 Organisation de la thèse . . . 5

2 État de l'art : Les modèles d'apprentissage profond 7
- 2.1 Introduction . . . 7
- 2.2 L'apprentissage profond . . . 8
  - 2.2.1 Définition . . . 8
  - 2.2.2 Apprentissage automatique et profond . . . 9
- 2.3 Méthodes classiques d'apprentissage profond . . . 10
  - 2.3.1 Réseaux de Neurones Conventionnels . . . 10
  - 2.3.2 Réseaux de Neurones Convolutifs (CNN) . . . 14
  - 2.3.3 Réseaux de Neurones Récurrents (RNN) et Long Short-Term Memory (LSTM) . . . 17
- 2.4 Auto-encodeurs . . . 20
- 2.5 Méthodes d'apprentissage profond basées sur l'attention . . . 21
  - 2.5.1 Transformers . . . 21
  - 2.5.2 Attention multi-tête . . . 23
  - 2.5.3 Perceiver et Perceiver IO . . . 25
- 2.6 Autres avancées et architectures récentes . . . 29
- 2.7 Conclusion . . . 30

3 Première contribution : Compression de signaux temporels pour l'embarqué 33
- 3.1 Étude exploratoire . . . 33
  - 3.1.1 Données . . . 33
  - 3.1.2 Transformée de Fourier à Court Terme (STFT) . . . 34
  - 3.1.3 Problématique . . . 35
  - 3.1.4 Choix du modèle . . . 36
  - 3.1.5 Synthèse . . . 44

3.2 Optimisation de la taille du modèle 49
3.2.1 Tête d'attention 49
3.2.2 Optimisations mono-tête 50
3.2.3 Conclusion 53
3.3 Embarquement du modèle 54
3.3.1 Présentation de la cible embarquée 54
3.3.2 Implémentation du modèle en embarqué 54
3.3.3 Résultats 56
3.4 Conclusion 56

4 Seconde contribution : Compression et estimation de la dégradation 59
4.1 Introduction 59
4.1.1 Travaux associés 61
4.1.2 Techniques Conventionnelles de Détection d'Anomalie . . . 62
4.1.3 Détection d'Anomalie par Auto-Encodage 65
4.1.4 Compression et détection d'anomalie 67
4.2 Méthodologie proposée 70
4.2.1 Architecture du Perceiver IO proposé 70
4.2.2 Apprentissage contrastif 72
4.3 Configuration Expérimentale 72
4.3.1 Étude de Cas sur la Surveillance Ferroviaire 72
4.3.2 Description du Jeu de Données 73
4.3.3 Entraînement du Modèle 74
4.4 Résultats et Discussion 75
4.4.1 Performance des Modèles de Compression de Données et d'Apprentissage Profond 76
4.4.2 Résultats sur l'Apprentissage par Contraste 79
4.4.3 Environnements embarqués 85
4.5 Estimation de la dégradation 87
4.5.1 Mesure de la Dégradation dans le Contexte Ferroviaire . . . 88
4.5.2 Compression et estimation d'indicateur de dégradation . . . 89
4.5.3 Méthodologie 91
4.5.4 Résultats 92
4.6 Conclusion 95

5 Conclusion générale 97
5.1 Synthèse 97
5.2 Perspectives 98
5.2.1 Quantification 98
5.2.2 Apprentissage en ligne 98
5.2.3 Réseaux de capteurs 99

Bibliographie 103

Table des figures

1 L'accident de Bretigny-sur-Orge (Source : BEA-TT). 2
2 Partenaires impliqués dans le projet RailMon 5
3 Le perceptron 11
4 Architecture d'un réseau de neurones conventionnel avec une couche cachée 13
5 Architecture d'un réseau de neurones convolutif 15
6 Architecture d'un RNN 18
7 Architecture d'un Transformer 22
8 Architecture de la tête d'attention 23
9 Architecture du modèle Perceiver 25
10 Architecture du modèle Perceiver IO 28
11 Architecture de l'apprentissage d'un GAN 29
12 Double représentation des signaux. 35
13 Architecture du modèle de réseaux de neurones conventionnels évalué 37
14 Courbe d'apprentissage du modèle de réseaux de neurones conventionnels 38
15 Exemple de reconstructions en spectrogramme par le réseau de neurones conventionnel (gauche : original, centre : reconstruit, droite : erreur) 39
16 Architecture du modèle de réseaux de neurones convolutifs évalué . 40
17 Courbe d'apprentissage du modèle de réseaux de neurones convolutif 41
18 Exemple de reconstructions en spectrogramme par le réseau de neurones convolutif (gauche : original, centre : reconstruit, droite : erreur) 43
19 Courbe d'apprentissage du modèle LSTM 44
20 Exemple de reconstructions en spectrogramme par le modèle LSTM (gauche : original, centre : reconstruit, droite : erreur) 45
21 Architecture du modèle Perceiver IO étudié 46
22 Courbe d'apprentissage du modèle Perceiver IO 46
23 Exemple de reconstructions en spectrogramme par le modèle Perceiver IO (gauche : original, centre : reconstruit, droite : erreur) 47
24 Architecture du bloc de traitement du modèle Perceiver IO 49
25 Architecture du calcul des scores d'attention 49
26 Architecture optimisée du calcul des scores d'attention 50
27 Architecture optimisée du calcul des scores d'attention avec l'entrée du bloc de traitement 52
28 Architecture de la sortie de la tête d'attention 53
29 Architecture optimisée de la sortie de la tête d'attention 53

30 Architecture optimisée du bloc de traitement Perceiver IO 54
31 STM32H755ZI-Q : carte d'évaluation utilisée pour les tests d'embarquement 55
32 Architecture Perceiver IO pour la compression et la détection d'anomalies. 71
33 Reconstructions de signaux d'exemple. (a), (c) et (e) : en utilisant l'algorithme K-SVD. (b), (d) et (f) : en utilisant l'algorithme Perceiver IO. 77
34 Projection des deux premières composantes PCA : (a) sans apprentissage contrastif, (b) avec apprentissage contrastif et (c) avec apprentissage contrastif partiel. Histogramme de la première composante PCA : (d) sans apprentissage contrastif, (e) avec apprentissage contrastif et (f) avec apprentissage contrastif partiel. 83
35 Reconstructions en spectrogramme des trois modèles présentés pour deux exemples de signaux. 84
36 Comparaison des indicateurs de dégradation 93
37 Évolution de la température sur la plage de temps correspondant au jeu témoin 94

Table des tableaux

2.1 Comparaison de l'apprentissage automatique avec l'apprentissage profond 10
3.1 Synthèse des performances des différents modèles testés 48
3.2 Synthèse des tailles des différents modèles testés en nombre de paramètres 48
4.1 Métriques d'entraînement pour K-SVD et Perceiver IO 77
4.2 Résumé des pertes et scores F1 suivant le poids alloué à l'entraînement contrastif 79
4.3 Métriques de classification et d'entraînement 81
4.4 Comparaison des tailles de modèles suivant ses paramètres 86

Chapitre 1

Introduction générale

1.1 Contexte

La maintenance ferroviaire est un pilier essentiel de la sécurité et de l'efficacité des réseaux de transport à travers le monde. Chaque jour, des millions de voyageurs et des tonnes de marchandises circulent sur des milliers de kilomètres de voies ferrées, de ponts, de tunnels et d'équipements électriques. Pourtant, derrière cette apparente fluidité se cache un travail minutieux et constant : l'entretien des infrastructures. Sans une maintenance rigoureuse, les risques d'accidents, de retards et de coûts exorbitants s'accumulent, menaçant non seulement la sécurité des usagers, mais aussi la viabilité économique des opérateurs ferroviaires.

L'entretien des infrastructures ferroviaires ne se limite pas à la simple réparation des rails ou au remplacement des traverses usées. Il englobe une multitude d'actions préventives et correctives visant à garantir la fiabilité de l'ensemble du système. Les voies ferrées, par exemple, subissent des contraintes mécaniques et environnementales considérables : variations de température, pluies, inondations, vibrations répétées des trains, ou encore la corrosion due à l'humidité. Ces facteurs, combinés à l'usure naturelle des matériaux, peuvent entraîner des déformations, des fissures ou des ruptures, compromettant la stabilité des voies. Les équipements électriques, tels que les caténaires ou les systèmes de signalisation, sont tout aussi vulnérables. Une panne de signalisation peut provoquer des collisions ou des blocages en cascade, paralysant des portions entières du réseau.

Les conséquences d'une maintenance insuffisante sont souvent dramatiques et coûteuses. L'histoire ferroviaire regorge d'exemples tragiques illustrant les risques encourus. En 2013, le déraillement d'un train à Brétigny-sur-Orge (Figure 1), en France, a causé la mort de sept personnes et blessé des dizaines d'autres. L'enquête a révélé que l'accident était dû à une pièce métallique appelée éclisse, joignant les rails entre eux. En raison d'un défaut de maintenance, cette pièce s'est détachée de son support et a obstrué le passage de la roue. Plus récemment, en 2020, un

déraillement en Écosse a été attribué à des glissements de terrain non détectés, mettant en lumière l'importance de la surveillance des infrastructures environnantes. Ces accidents, outre leur coût humain, entraînent des pertes financières colossales : indemnisation des victimes, réparations d'urgence, perturbations du trafic, et perte de confiance des usagers.



(a) Vue de l'accident (b) Vue de la pièce incriminée

FIGURE 1 – L'accident de Bretigny-sur-Orge (Source : BEA-TT).

Au-delà des accidents, une maintenance défaillante se traduit par une dégradation progressive des performances du réseau. Les retards s'accumulent, les vitesses moyennes des trains diminuent, et les coûts d'exploitation explosent. En Allemagne, des études ont montré que les retards sur le réseau ferroviaire étaient en partie imputables à des problèmes d'infrastructure non résolus à temps. Aux États-Unis, des portions entières du réseau sont aujourd'hui considérées comme obsolètes, nécessitant des investissements massifs pour éviter un effondrement partiel. Ces situations illustrent comment une maintenance insuffisante peut, à terme, asphyxier un système ferroviaire entier, rendant les déplacements moins fiables et moins attractifs.

Face à ces enjeux, la télésurveillance des équipements ferroviaires s'impose comme une solution innovante et efficace. Grâce à des capteurs intelligents, des caméras thermiques, des drones et des systèmes de traitement des données en temps réel, il est désormais possible de surveiller en continu l'état des infrastructures. Ces technologies permettent de détecter précocement les anomalies : fissures dans les rails, surchauffe des équipements électriques, mouvements de terrain suspects, ou encore usure anormale des roues des trains. En analysant ces données, les opérateurs peuvent anticiper les pannes et planifier les interventions de maintenance avant que les problèmes ne deviennent critiques.

La télésurveillance offre plusieurs avantages majeurs. D'abord, elle réduit les coûts en évitant les réparations d'urgence, souvent plus onéreuses que les interventions planifiées. Ensuite, elle améliore la sécurité en minimisant les risques d'acci-

dents liés à des défauts non détectés. Enfin, elle optimise la disponibilité du réseau en limitant les interruptions de trafic pour maintenance. En France, la SNCF a déployé des systèmes de télésurveillance sur plusieurs lignes à grande vitesse, permettant de réduire de 30 % les temps d'immobilisation des voies pour entretien. En Asie, des pays comme le Japon ou la Chine utilisent des réseaux de capteurs pour surveiller en permanence l'état de leurs infrastructures, garantissant ainsi une fiabilité exceptionnelle de leurs services ferroviaires.

Cependant, la mise en place de la télésurveillance ne se fait pas sans défis. Elle nécessite des investissements initiaux importants en équipements et en formation du personnel. De plus, la gestion des données collectées pose des questions de cybersécurité et de protection des informations. Malgré ces obstacles, les bénéfices sur le long terme sont indéniables. La télésurveillance transforme la maintenance ferroviaire, passant d'une approche réactive, où l'on répare après la panne, à une approche proactive, où l'on anticipe et prévient les défaillances.

En conclusion, la maintenance ferroviaire est un enjeu de sécurité, d'efficacité et de durabilité. Les exemples d'accidents et de dysfonctionnements passés rappellent cruellement les conséquences d'une négligence dans ce domaine. La télésurveillance, en offrant une vision en temps réel de l'état des infrastructures, répond directement à ces problématiques. Elle permet non seulement de réduire les risques et les coûts, mais aussi d'assurer la pérennité des réseaux ferroviaires, essentiels à la mobilité des personnes et des biens. À l'ère de la révolution numérique, intégrer ces technologies n'est plus une option, mais une nécessité pour garantir un transport ferroviaire sûr, performant et résilient.

1.2 Le projet RailMon

Les travaux de cette thèse s'inscrivent dans le cadre de RailMon : un grand projet partenarial financé par le PIA4 (Plan d'investissement d'avenir) et porté par l'entreprise Vossloh avec un consortium de 7 partenaires (3 partenaires académiques et 4 entreprises). L'objectif principal du projet est d'aboutir à une solution de télésurveillance nouvelle génération légère, sécurisée et facile à déployer. Ce système doit être conçu pour renforcer la confiance dans la gestion de la maintenance ferroviaire en se concentrant sur l'anticipation et la détection des pannes. Le projet se compose de 3 branches fonctionnelles couvrant le spectre complet de la maintenance assistée par des outils numériques :

— capter l'information : mise au point de capteurs innovants, autonomes en énergie et légers à installer.

— traiter la donnée pour en extraire des indicateurs pertinents : développement d'un nouveau concentrateur de données flexible et de méthodes de traitement de données basées sur des algorithmes d'intelligence artificielle, tout en assurant la cyber-sécurité sur les canaux de communication et les traitements effectués.
— aider la maintenance dans sa tâche sur le terrain : développement et au déploiement d'outils d'aide à la maintenance sur le terrain, basés sur la réalité augmentée.

Le projet est structuré en 5 lots principaux, chacun traitant d'un aspect spécifique du développement du système :

— **Lot 1 : Capteurs** :
  — **Tache 1.1 : Capteurs IoT autonomes en énergie** : conception d'un dispositif de récupération d'énergie vibratoire destiné à des capteurs autonomes fonctionnant en environnement ferroviaire.
  — **Tache 1.2 : Traitement local des données** : développement de logiciels embarqués conçus pour le traitement local de données à l'aide de réseaux neuronaux légers et intégration de ces algorithmes sur le matériel cible.
— **Lot 2 : Concentrateur de données (*Vossloh Open Gateway*)** : développement d'un concentrateur de données modulaire et cybersécurisé, adapté aux infrastructures ferroviaires.
— **Lot 3 : Anticipation et détection des défaillances** : développement d'un logiciel basé sur l'IA pour la détection d'anomalies en temps réel et la planification de la maintenance.
— **Lot 4 : Outils digitaux pour l'assistance à la maintenance** : développement de technologies pour l'assistance aux opérateurs dans les tâches de maintenance standard, le tutorat à distance pour les situations nécessitant l'intervention d'experts et l'assistance à la gestion de crise en cas de déclenchement d'alarmes.
— **Lot 5 : Cyber-sécurité** : co-conception d'une nouvelle génération d'architecture de télémesure résiliente aux cyberattaques afin de garantir la confiance dans la maintenance des systèmes ferroviaires, par la sécurisation de trois couches critiques : l'acquisition de données de capteurs, le concentrateur VOG et la télésurveillance globale.

Le projet rassemble un groupe diversifié de partenaires issus de l'industrie et du monde universitaire, chacun apportant son expertise à différents aspects du projet (Figure 2). Parmi les partenaires industriels figurent Siema Applications, Aquilae, Cedrat Technologies et Yumain. Sur le plan académique, le Laboratoire AMPERE (INSA Lyon) est responsable du Lot 5, tandis que le Laboratoire SYMME est

impliqué dans les tâches du Lot 1. L'Université de Technologie de Troyes (UTT) joue un rôle important, étant responsable du Lot 1.2, du Lot 3, et participant également au Lot 4.



FIGURE 2 – Partenaires impliqués dans le projet RailMon

1.3 Problématique et objectifs de la thèse

Les travaux de cette thèse concernent principalement le développement d'algorithmes IA pour la compression, la détection d'anomalie et l'extraction d'indicateurs. Les capteurs étant autonomes en énergie, il convient de minimiser la transmission de données brutes au profit d'un traitement local de données afin d'extraire des métadonnées utiles. On souhaite également estimer la dégradation du système à l'aide des signaux vibratoires. Dans le cadre de ce projet, on propose une nouvelle génération de réseaux de neurones avec des architectures allégées pour être embarqués sur des capteurs avec des contraintes en termes de consommation et de capacité de calcul.

1.4 Organisation de la thèse

La thèse est divisé en une présentation des architectures d'apprentissage profond considérées pour répondre à la problématique donnée à la section précédente, ainsi que trois contributions, chacune présentée dans un chapitre de ce manuscrit.

Le chapitre 2 présente les fondamentaux de l'apprentissage profond, ainsi que différentes architectures de modèles qui seront testés sur la première contribution.

La première contribution consiste à trouver une architecture de modèle permettant la compression des signaux vibratoires mesurés par le capteur et la reconstruction des données sur le concentrateur. Ce modèle devra valider les contraintes inhérentes à la cible embarquée présentée à la section précédente.

La seconde contribution consiste à intégrer dans ce modèle de compression un module capable de détecter des anomalies selon différents cas d'usages.

La troisième contribution consiste en la définition d'un indicateur de dégradation du système surveillé à partir des signaux mesurés. Cet indicateur est extrait des données compressées via un autre modèle d'apprentissage profond.

Enfin, la thèse se concluera par des possibilités d'amélioration et de travaux futurs sur les contributions présentées.

Chapitre 2

État de l'art : Les modèles d'apprentissage profond

2.1 Introduction

L'apprentissage profond (deep learning) s'impose aujourd'hui comme l'un des piliers de la révolution de l'intelligence artificielle, transformant radicalement des domaines aussi variés que la vision par ordinateur, le traitement automatique du langage, la robotique ou encore la bioinformatique. Son émergence a permis des avancées majeures, rendant possibles des tâches autrefois considérées comme hors de portée des machines, telles que la reconnaissance d'images avec une précision surpassant celle de l'humain, la traduction automatique en temps réel, ou encore la génération de texte et d'images réalistes. Ces progrès, souvent qualifiés de disruptifs, reposent sur la capacité des modèles profonds à extraire automatiquement des représentations hiérarchiques et complexes à partir de données brutes, sans nécessiter une ingénierie manuelle des caractéristiques.

L'importance de l'apprentissage profond ne se limite pas à ses performances techniques : il redéfinit les frontières de l'interaction homme-machine, ouvre de nouvelles perspectives pour la recherche scientifique et soulève des questions éthiques et sociétales inédites. Dans ce contexte, comprendre les fondements, les architectures et les enjeux de cette discipline devient essentiel, tant pour les chercheurs que pour les praticiens ou les décideurs.

Cet état de l'art vise à fournir une synthèse claire et structurée des concepts clés, des architectures emblématiques et des avancées récentes en apprentissage profond. Les modèles présentés seront utilisés dans le chapitre suivant pour définir l'algorithme répondant à la problématique de compression.

Nous commencerons par définir l'apprentissage profond et le situer par rapport à l'apprentissage automatique classique, avant d'explorer ses fondements, depuis le perceptron jusqu'aux architectures les plus sophistiquées comme les réseaux de

neurones convolutifs (CNN), récurrents (RNN), les LSTM, les auto-encodeurs et les Transformers. Une attention particulière sera portée aux mécanismes qui ont permis de surmonter les défis historiques, ainsi qu'aux innovations récentes qui repoussent toujours plus loin les limites de l'IA. Enfin, nous aborderons les perspectives futures, en soulignant les défis restants et les pistes de recherche prometteuses.

2.2 L'apprentissage profond

2.2.1 Définition

L'apprentissage profond, ou deep learning, est une sous-catégorie de l'apprentissage automatique (machine learning) qui se distingue par l'utilisation de réseaux de neurones artificiels profonds, c'est-à-dire composés de multiples couches de traitement successives. Ces architectures permettent aux modèles d'apprendre des représentations hiérarchiques des données : les premières couches capturent des caractéristiques simples (comme des bords ou des textures dans une image), tandis que les couches profondes combinent ces informations pour extraire des motifs plus abstraits et complexes (comme des formes, des objets, ou des concepts sémantiques).

Les principales caractéristiques de l'apprentissage profond incluent :

— L'automatisation de l'extraction de caractéristiques : contrairement aux méthodes classiques nécessitant une ingénierie manuelle des caractéristiques, les modèles profonds apprennent directement à partir des données brutes.

— La capacité à modéliser des relations non linéaires complexes : grâce à l'empilement de couches non linéaires, les réseaux profonds peuvent approximer des fonctions arbitrairement complexes.

L'apprentissage profond s'inscrit dans une longue histoire, marquée par des périodes d'enthousiasme et de désillusion. Le perceptron, proposé par (Rosenblatt 1958), est considéré comme le précurseur des réseaux de neurones modernes, mais ses limitations (notamment son incapacité à résoudre des problèmes non linéaires comme la fonction logique du OU exclusif) ont conduit à un ralentissement des recherches dans les années 1970-1980. La renaissance de l'apprentissage profond au début des années 2010 est attribuée à plusieurs facteurs :

— L'augmentation exponentielle des données disponibles (images, textes, vidéos, etc.), rendue possible par le développement du web et des capteurs numériques.

— Les progrès matériels, notamment l'utilisation des GPU pour accélérer les calculs parallèles, réduisant ainsi le temps d'entraînement des grands modèles.

— Les avancées algorithmiques, comme la rétropropagation du gradient, les fonctions d'activation ReLU, ou les techniques de régularisation (dropout, batch normalization), qui ont permis de surmonter les problèmes historiques (vanishing/exploding gradients, surapprentissage).

2.2.2 Apprentissage automatique et profond

Contrairement à l'apprentissage automatique classique, qui repose souvent sur des modèles statistiques ou des algorithmes d'apprentissage peu profonds, l'apprentissage profond mise sur la profondeur des architectures pour capturer des motifs complexes, au prix d'une plus grande opacité et d'un coût computationnel élevé. La principale différence réside dans la représentation des données et la manière dont les modèles apprennent à extraire des caractéristiques. En apprentissage automatique classique, l'extraction de caractéristiques est souvent manuelle ou semi-automatique : il est nécessaire de définir des descripteurs pertinents pour la tâche à résoudre. Ces caractéristiques sont ensuite utilisées par des algorithmes tels que les machines à vecteurs de support, les forêts aléatoires ou les méthodes bayésiennes pour effectuer la classification, la régression ou d'autres tâches. En revanche, l'apprentissage profond automatise cette étape en utilisant des réseaux de neurones profonds capables de découvrir et d'apprendrent eux-mêmes des représentations hiérarchiques à partir des données brutes. Cela permet de traiter des données complexes et non structurées, comme des images ou du texte, avec une précision souvent supérieure à celle des méthodes classiques.

Une autre différence essentielle concerne la quantité de données et la puissance de calcul requises. L'apprentissage profond nécessite généralement des volumes de données plus importants et des ressources de calcul plus conséquentes pour atteindre des performances optimales, alors que l'apprentissage automatique traditionnel peut souvent fonctionner efficacement avec moins de données et des algorithmes moins gourmands en ressources.

En résumé, si l'apprentissage automatique classique repose sur des caractéristiques explicites et des modèles plus simples, l'apprentissage profond exploite des architectures complexes pour apprendre directement à partir des données, offrant ainsi une capacité accrue à traiter des problèmes plus complexes et variés (Table 2.1).

TABLE 2.1 – Comparaison de l'apprentissage automatique avec l'apprentissage profond

Critère	Apprentissage automatique	Apprentissage profond
Extraction des caractéristiques	Manuelle	Automatique / Intégrée
Interprétabilité	Élevée	Faible
Type et quantité de données	Structurées en volume faible à moyen	Non structurées en volume moyen à important
Complexité du problème	Faible à modérée	Élevée
Ressources computationnelles	Faibles à modérées	Modérées à Élevées

2.3 Méthodes classiques d'apprentissage profond

2.3.1 Réseaux de Neurones Conventionnels

Les réseaux de neurones conventionnels, également connus sous le nom de réseaux entièrement connectés ou FC (Fully Connected Networks), occupent une place fondamentale dans l'histoire de l'intelligence artificielle et de l'apprentissage automatique. Leur développement s'étend sur plusieurs décennies, marquées par des avancées significatives qui ont permis de mieux comprendre et modéliser le fonctionnement du cerveau biologique ainsi que la capacité des machines à apprendre à partir de données.

Les origines de ces réseaux remontent aux années 1940 et 1950, lorsque Warren McCulloch et Walter Pitts ont proposé un modèle mathématique du neurone biologique. En 1943, ils ont introduit ce qu'on appelle aujourd'hui le neurone McCulloch-Pitts (McCulloch et Pitts 1943), une abstraction simple mais puissante du fonctionnement neuronal. Leur modèle consistait en une unité qui recevait plusieurs entrées, effectuait une somme pondérée, appliquait une fonction de seuil, et produisait une sortie binaire. Ce concept a permis aux chercheurs de formaliser le fonctionnement des circuits neuronaux et de commencer à envisager leur utilisation dans des systèmes artificiels.

Ce n'est qu'au cours des années 1950 que la recherche a véritablement pris de l'ampleur avec l'introduction du perceptron par (Rosenblatt 1958). Le perceptron représentait une avancée notable à l'époque, car il s'agissait d'un modèle capable

d'apprendre à classifier des entrées en ajustant ses poids de manière supervisée. La simplicité du perceptron lui permettait d'être facilement entraîné à l'aide d'algorithmes de mise à jour basés sur l'erreur. Cependant, cette approche montrait rapidement ses limites, notamment face à des problèmes non linéairement séparables, comme le XOR (ou exclusif), qui échappait à la capacité de classification du perceptron simple. Ces limitations ont temporairement freiné l'intérêt pour les réseaux de neurones, jusqu'à ce qu'une nouvelle étape majeure survienne dans les années 1980.

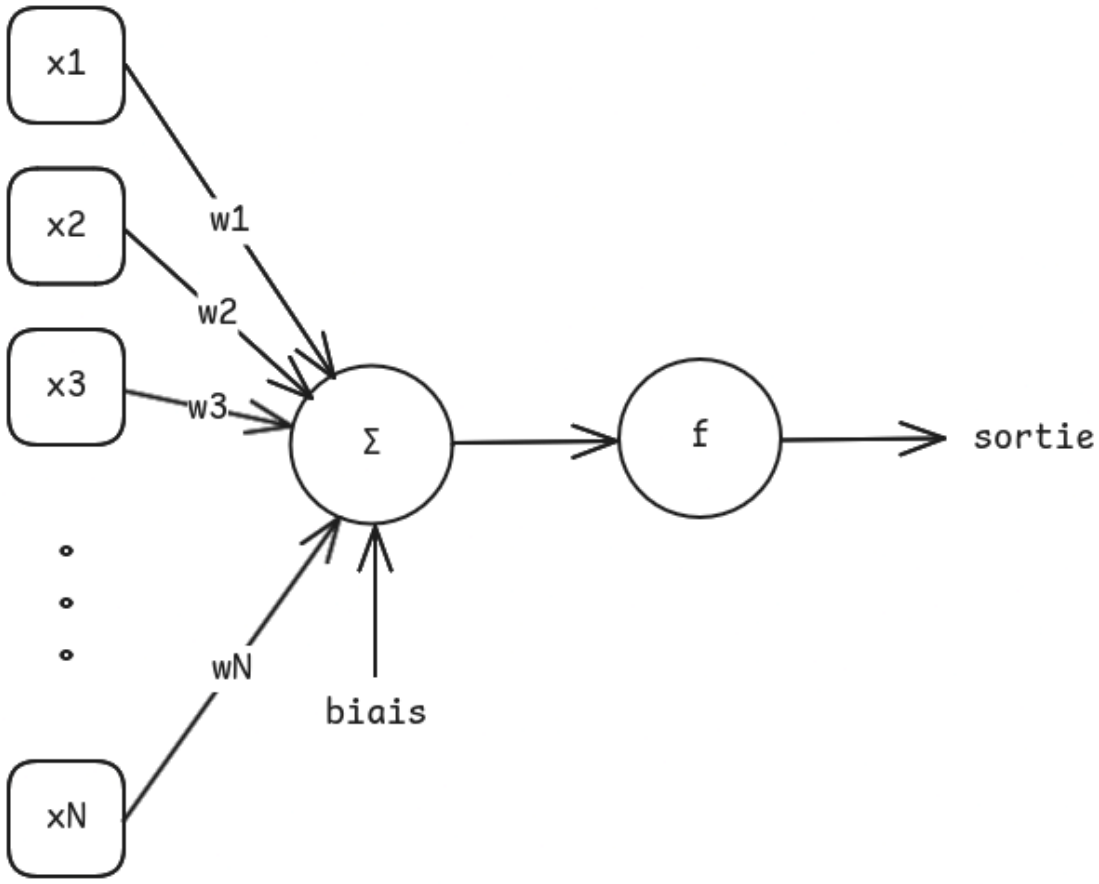


FIGURE 3 – Le perceptron

Dans cette période, la renaissance des réseaux de neurones a été catalysée par l'introduction de l'algorithme de rétropropagation (Rumelhart et al. 1986). La rétropropagation a permis l'entraînement efficace de réseaux multicouches, autrement dit, l'apprentissage d'architectures plus complexes avec plusieurs couches cachées. Avec cette méthode, le réseau pouvait ajuster ses poids en minimisant une fonction de perte grâce à un processus itératif de propagation de l'erreur en arrière, ce qui a ouvert la voie à une multitude d'applications plus sophistiquées. La rétropropa-

gation a marqué un tournant décisif, transformant la recherche en apprentissage automatique et permettant de traiter des tâches plus complexes que jamais auparavant.

Le fonctionnement d'un réseau de neurones conventionnel repose sur une architecture structurée en plusieurs couches. La première, appelée couche d'entrée, est responsable de recevoir les données brutes. Chaque neurone de cette couche correspond généralement à une caractéristique spécifique des données, comme une variable dans un ensemble de données tabulaires ou un pixel dans une image. Ensuite, viennent une ou plusieurs couches cachées qui effectuent des transformations non linéaires sur ces entrées. Chaque neurone dans ces couches calcule une somme pondérée de ses entrées, à laquelle il ajoute un biais, puis applique une fonction d'activation non linéaire comme la sigmoïde, ReLU (Rectified Linear Unit), ou tanh. Ces fonctions permettent au réseau de modéliser des relations complexes et non linéaires entre les variables d'entrée et de sortie, rendant possible l'apprentissage de tâches variées.

$$Sigmoide(x) = \frac{1}{1 + e^{-x}}$$

$$ReLU(x) = \begin{cases} x \text{ si } x > 0 \\ 0 \text{ sinon} \end{cases}$$

$$tanh(x) = \frac{sinh(x)}{cosh(x)} = \frac{e^{2x} - 1}{e^{2x} + 1}$$

Enfin, la couche de sortie produit la prédiction finale du réseau. La nature de cette sortie dépend de la tâche : pour une classification, il peut s'agir d'un vecteur de probabilités ou d'étiquettes, tandis que pour une régression, il s'agit d'une valeur continue.

L'ensemble du processus d'apprentissage repose sur une méthode appelée rétropropagation du gradient (Rumelhart et al. 1986), couplée à un algorithme d'optimisation comme la descente de gradient. Concrètement, dans un premier temps, les poids du réseau sont initialisés aléatoirement. Ensuite, lors d'une étape connue sous le nom de propagation avant, les données d'entrée sont passées à travers le réseau, couche par couche, pour générer une sortie. La différence entre cette sortie et la valeur attendue (ou étiquette) est calculée à l'aide d'une fonction de coût ou d'erreur, comme l'erreur quadratique moyenne pour la régression ou l'entropie croisée pour la classification.

Une fois cette erreur calculée, la rétropropagation entre en jeu : l'erreur est propagée en arrière à travers le réseau, en ajustant chaque poids selon le gradient de la fonction d'erreur par rapport à ce poids. La mise à jour de ces poids est effec-

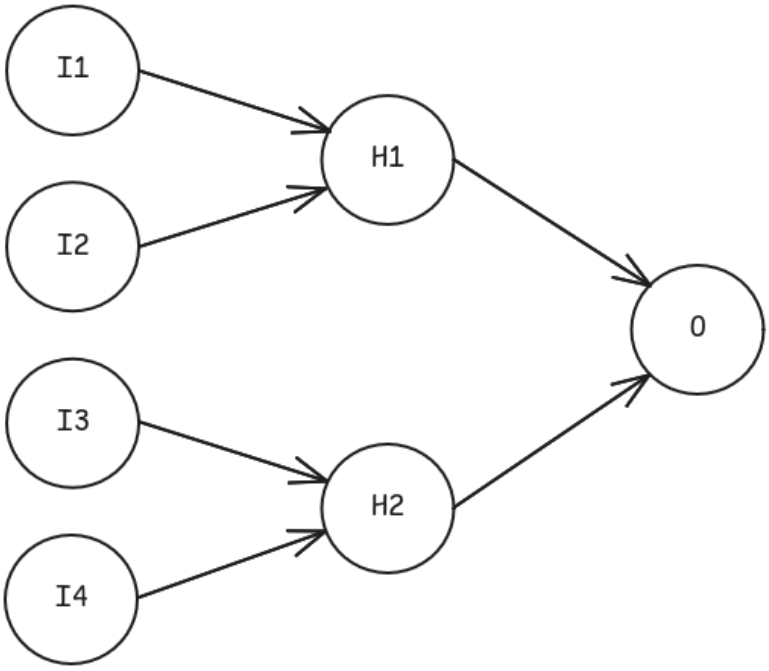


FIGURE 4 – Architecture d'un réseau de neurones conventionnel avec une couche cachée

tuée pour réduire l'erreur globale, généralement via la descente de gradient ou ses variantes. Ce processus est répété sur de nombreuses itérations, ou *epochs*, jusqu'à ce que le réseau atteigne une performance satisfaisante ou que l'erreur se stabilise. À travers cette procédure, le réseau apprend à modéliser des relations complexes et à généraliser sur de nouvelles données.

Les applications des réseaux de neurones conventionnels sont variées, bien que leur usage ait été en grande partie supplanté par des architectures plus avancées comme les réseaux convolutifs ou récurrents. Néanmoins, ils restent un outil précieux dans plusieurs domaines. En classification, par exemple, ils ont été utilisés pour distinguer différentes espèces de fleurs ou pour reconnaître des chiffres manuscrits dans des jeux de données comme MNIST. La simplicité de leur architecture en fait un point de départ idéal pour l'apprentissage, notamment pour les chercheurs débutants ou pour des tâches où la complexité n'est pas excessive.

Dans le domaine de la régression, ces réseaux peuvent modéliser des relations continues complexes, comme la prédiction des prix immobiliers en fonction de caractéristiques telles que la surface, le nombre de pièces ou la localisation géographique. Ils ont également été employés dans la reconnaissance de motifs simples, bien que pour des tâches plus complexes, des architectures spécialisées soient préférées. Dans l'industrie, leur utilisation s'est étendue à la prédiction de la demande, à la détection

d'anomalies dans des données de capteurs, ou encore à l'optimisation des processus de production. Ces applications illustrent leur potentiel dans l'analyse de données structurées ou semi-structurées et dans la prise de décision automatisée.

Malgré leur âge relatif, les réseaux de neurones conventionnels ont joué un rôle crucial dans le développement de l'apprentissage profond. Leur simplicité et leur flexibilité en font encore aujourd'hui un outil de référence pour comprendre les principes fondamentaux des réseaux de neurones. En effet, leur étude permet d'appréhender la manière dont les systèmes peuvent apprendre à partir de données, ajuster leurs paramètres, et effectuer des tâches variées. Bien que de nouvelles architectures plus avancées aient éclipsé leur usage dans certains domaines, leur importance historique et leur contribution à la compréhension des mécanismes d'apprentissage restent indéniables. Ils constituent ainsi une étape essentielle dans l'évolution de l'intelligence artificielle, tout en restant pertinents pour des applications simples ou comme base pédagogique pour initier les débutants à la discipline.

2.3.2 Réseaux de Neurones Convolutifs (CNN)

Les réseaux de neurones convolutifs, communément appelés CNN (Convolutional Neural Networks), ont constitué une révolution dans le domaine de la vision par ordinateur et de l'analyse d'images. Leur origine remonte aux travaux pionniers de Yann LeCun et de ses collègues dans les années 1980 et 1990 (Le Cun et al. 1989), qui se sont inspirés des découvertes en neurobiologie concernant le cortex visuel des mammifères. Ces recherches, notamment celles de David Hubel et Torsten Wiesel dans les années 1960, ont montré que certaines cellules du cortex répondent à des motifs spécifiques dans des régions locales du champ visuel (Hubel et Wiesel 1959). Ces observations ont conduit à l'idée que le traitement de l'information visuelle dans le cerveau repose sur des mécanismes locaux, hiérarchiques et filtrants. Le concept de convolution, tel qu'il est utilisé dans ces réseaux, cherche à imiter cette organisation en permettant à un modèle d'apprendre automatiquement des caractéristiques locales à différentes échelles.

Le premier réseau de neurones convolutifs à avoir véritablement marqué l'histoire est LeNet-5 (LeCun et al. 1999). Ce modèle avait pour objectif la reconnaissance de chiffres manuscrits, notamment dans le contexte de la lecture automatique de chèques bancaires. La performance de LeNet-5 sur la base de données MNIST, qui contient des images de chiffres écrits à la main, a été particulièrement remarquable à l'époque, prouvant que cette architecture pouvait traiter efficacement des données visuelles structurées (Lecun et al. 2000). Depuis cette étape initiale, les CNN ont connu une croissance exponentielle en termes de complexité, de performance

et d'applications. Leur adoption s'est généralisée dans tous les domaines liés à la vision, devenant la référence incontournable pour tout problème impliquant la reconnaissance et l'analyse d'images.

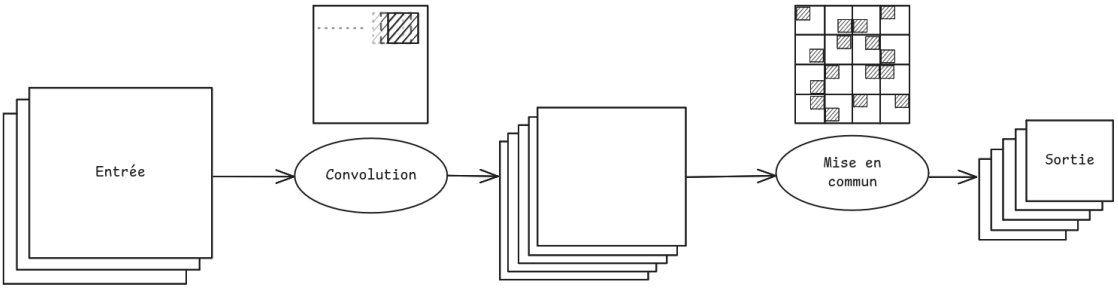


FIGURE 5 – Architecture d'un réseau de neurones convolutif

Le fonctionnement d'un CNN repose sur une série de couches qui exploitent la structure spatiale inhérente aux images. Contrairement aux réseaux entièrement connectés, où chaque neurone est connecté à tous les neurones de la couche précédente, les CNN utilisent des filtres ou noyaux de convolution qui glissent sur l'image pour extraire des caractéristiques. La couche de convolution applique une série de ces filtres à l'entrée, produisant des cartes de caractéristiques (feature maps) qui mettent en évidence des motifs spécifiques tels que des bords, des textures ou des formes géométriques. La convolution consiste à faire glisser le filtre sur l'image tout en calculant la somme pondérée des pixels qu'il couvre, ce qui permet au modèle de détecter localement des motifs invariants à la translation.

Après la convolution, une fonction d'activation non linéaire, généralement ReLU (Rectified Linear Unit), est appliquée pour introduire de la non-linéarité, essentielle à l'apprentissage de relations complexes. Ensuite, intervient souvent une étape de pooling, ou sous-échantillonnage, qui réduit la dimension des cartes de caractéristiques tout en conservant les informations essentielles. La technique la plus courante est le max pooling, qui consiste à prendre la valeur maximale dans une fenêtre glissante, ce qui permet de rendre le modèle plus robuste aux déformations et aux petites variations. Ces opérations sont répétées à travers plusieurs couches, créant une hiérarchie de caractéristiques allant des éléments simples comme les bords aux motifs complexes.

Après plusieurs couches de convolution et de pooling, les cartes de caractéristiques sont aplaties en un vecteur unidimensionnel, qui est ensuite passé à des couches entièrement connectées. Ces couches, similaires à celles des réseaux classiques, effectuent des transformations non linéaires supplémentaires et produisent la sortie finale du modèle. Pour une tâche de classification, par exemple, la dernière couche utilise souvent une fonction softmax pour convertir les scores en probabilités

sur différentes catégories. La capacité d'un CNN à apprendre automatiquement ces caractéristiques hiérarchiques, sans nécessiter de pré-traitement manuel, constitue sa principale force.

L'apprentissage dans un CNN repose sur la rétropropagation, une méthode qui ajuste les poids de tous les filtres et des couches complètement connectées en fonction de l'erreur de prédiction. Comme pour les autres réseaux, les poids sont initialisés aléatoirement, puis le modèle passe par une phase de propagation avant où les données sont traitées de l'entrée à la sortie. La sortie du réseau est comparée à la vérité terrain, et une erreur est calculée à l'aide d'une fonction de coût appropriée, comme l'entropie croisée pour la classification. Ensuite, la rétropropagation calcule les gradients de cette erreur par rapport à chaque paramètre du réseau, en utilisant la règle de la chaîne. Ces gradients servent à mettre à jour les poids via une méthode d'optimisation, souvent la descente de gradient ou ses variantes. Ce processus est répété sur plusieurs epochs jusqu'à ce que le modèle converge vers une solution satisfaisante.

Les applications des CNN sont vastes et ont profondément transformé le domaine de la vision par ordinateur. La reconnaissance d'images est sans doute l'une des plus célèbres, avec des modèles qui surpassent largement les méthodes traditionnelles basées sur la détection manuelle de caractéristiques. Par exemple, dans la compétition ImageNet, les CNN ont permis de classer des millions d'images avec une précision sans précédent. La détection d'objets constitue une autre avancée majeure ; des architectures comme YOLO (You Only Look Once) ou Faster R-CNN permettent de localiser et d'identifier rapidement plusieurs objets dans une image ou une vidéo en temps réel. La segmentation d'images, où chaque pixel doit être classé, bénéficie également de l'architecture CNN, notamment dans des domaines sensibles tels que la médecine, avec des outils comme U-Net ou SegNet qui segmentent les organes ou les lésions dans des images médicales.

Les CNN sont également très utilisés dans la reconnaissance faciale, où ils permettent d'identifier ou de vérifier l'identité d'une personne à partir d'une image ou d'une vidéo. Des architectures comme FaceNet ou DeepFace exploitent la capacité des CNN à apprendre des représentations faciales discriminantes, ce qui a des applications dans la sécurité, la surveillance ou la personnalisation des services. Dans le secteur médical, ces réseaux ont permis d'automatiser l'analyse d'images radiologiques, d'IRM ou de scanners, facilitant le diagnostic, la détection d'anomalies ou la segmentation de structures anatomiques. Leur capacité à traiter de grandes quantités de données visuelles avec une précision en constante amélioration en fait un outil incontournable pour la recherche et la pratique clinique.

Dans l'industrie, les CNN sont employés pour l'inspection de la qualité, la détection de défauts ou la surveillance automatique des processus de fabrication. Par exemple, ils peuvent analyser des images de pièces ou de produits pour repérer des anomalies, des fissures ou des imperfections invisibles à l'œil humain, permettant d'améliorer la fiabilité et la contrôle qualité. Leur capacité à apprendre des motifs complexes dans des données visuelles en fait également un atout pour la robotique, la conduite autonome, la surveillance de sécurité ou l'analyse de vidéos.

En somme, les réseaux de neurones convolutifs ont non seulement permis d'atteindre des performances exceptionnelles dans le traitement d'images, mais ils ont aussi ouvert de nouvelles perspectives dans de nombreux secteurs industriels, médicaux et de recherche. Leur architecture, inspirée par le fonctionnement du cortex visuel, leur capacité à apprendre automatiquement des hiérarchies de caractéristiques et leur aptitude à traiter efficacement de grandes quantités de données visuelles en font aujourd'hui un pilier central de l'intelligence artificielle moderne. La recherche continue à faire évoluer ces modèles, avec de nouvelles architectures, techniques d'entraînement et applications innovantes qui repoussent sans cesse les limites de ce que l'on peut réaliser en vision par ordinateur.

2.3.3 Réseaux de Neurones Récurrents (RNN) et Long Short-Term Memory (LSTM)

Les réseaux de neurones récurrents (RNN) et leur extension, les Long Short-Term Memory (LSTM), ont constitué des avancées majeures dans le traitement des données séquentielles. Leur développement remonte aux années 1980, période durant laquelle la communauté de la recherche en intelligence artificielle a cherché à dépasser les limitations des réseaux feedforward pour traiter efficacement des informations dépendantes du contexte temporel ou séquentiel. Contrairement aux perceptrons multicouches classiques, qui traitent chaque exemple de manière indépendante, les RNN introduisent une architecture avec des boucles ou connexions récurrentes permettant de maintenir une mémoire des états précédents. Cela leur confère la capacité de modéliser des dépendances à long terme, essentielles pour traiter des séquences telles que le langage naturel, les séries temporelles, ou les signaux audio.

L'idée de base derrière les RNN est d'intégrer une boucle dans leur architecture. Lorsqu'un réseau récurrent reçoit une séquence d'entrées, il traite chaque élément un à un tout en conservant une information sur l'état précédent, qui est actualisé à chaque étape. Cela permet au réseau d'avoir une mémoire interne, lui conférant une capacité contextuelle.

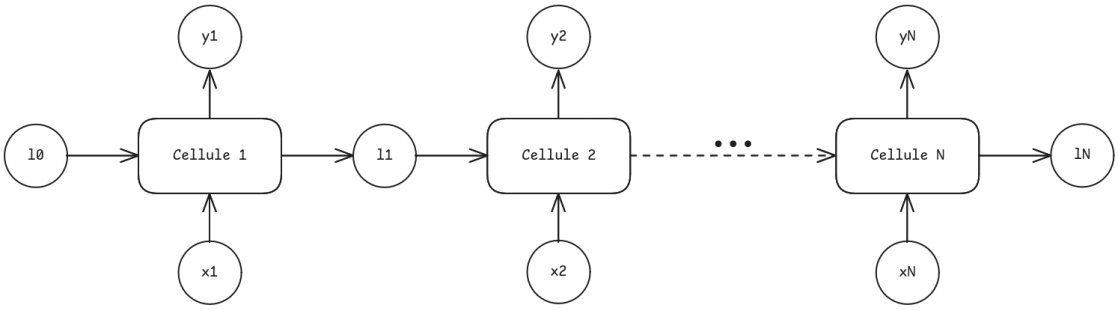


FIGURE 6 – Architecture d'un RNN

Le fonctionnement d'un RNN, en général, repose sur une architecture simple mais efficace. La couche d'entrée reçoit la séquence de données, élément par élément, et chaque étape est traitée à l'aide de la couche récurrente. La sortie de cette couche à chaque étape dépend à la fois de l'entrée courante et de l'état précédent, ce qui permet de transmettre des informations sur toute la séquence. En pratique, l'entraînement d'un RNN s'effectue via la rétropropagation à travers le temps (Backpropagation Through Time, BPTT). Lors de cette étape, l'erreur calculée à la fin de la séquence est propagée en arrière, en ajustant les poids du réseau à chaque étape pour minimiser la fonction de coût. Cependant, en raison de la nature de la rétropropagation sur plusieurs étapes, le gradient peut devenir très petit (disparition) ou très grand (explosion), ce qui pose des défis importants pour apprendre efficacement.

Toutefois, si cette architecture est théoriquement élégante, en pratique, elle souffre de graves problèmes lors de l'entraînement. En particulier, le phénomène de disparition ou d'explosion du gradient rend difficile l'apprentissage de dépendances à long terme. Ces limitations ont conduit à la création de variantes plus sophistiquées, notamment les réseaux LSTM (Hochreiter et Schmidhuber 1997).

Les réseaux LSTM ont été conçus précisément pour pallier ces difficultés d'apprentissage. Leur architecture repose sur une cellule de mémoire, qui peut conserver des informations sur de longues périodes, contrôlée par un ensemble de portes (gates). Ces portes – porte d'entrée, porte d'oubli et porte de sortie – agissent comme des mécanismes de régulation, permettant au réseau de décider quelles informations conserver, lesquelles oublier, et quelles transmettre en sortie. La porte d'entrée contrôle la mise à jour de la cellule de mémoire avec de nouvelles informations ; la porte d'oubli détermine quels éléments doivent être effacés ou conservés ; la porte de sortie décide quelles parties de la cellule seront utilisées pour produire la sortie à chaque étape. Grâce à cette architecture, les LSTM peuvent modéliser

des dépendances à long terme, essentielles pour traiter des séquences longues et complexes.

Les LSTM, avec leur architecture sophistiquée, ont permis de surmonter ces limitations. Leur unité de base, la cellule LSTM, contient plusieurs portes, chacune contrôlant le flux d'informations à travers le réseau. La porte d'entrée décide quelles nouvelles informations doivent être intégrées à la mémoire, la porte d'oubli détermine quelles informations doivent être effacées, et la porte de sortie contrôle ce qui sera produit comme sortie. La cellule de mémoire interne conserve ainsi un état qui peut être mis à jour, conservé ou oublié en fonction des besoins, permettant au réseau de capturer des dépendances à long terme. Lors de l'apprentissage, la rétropropagation se fait également à travers le temps, mais avec un contrôle accru des flux d'informations grâce à ces portes, ce qui atténue considérablement les problèmes de gradients.

Les applications des RNN et des LSTM sont vastes et variées. Dans le traitement du langage naturel (NLP), ils ont été largement adoptés pour des tâches telles que la traduction automatique, la génération de texte, la reconnaissance de la parole, ou l'analyse de sentiments. La capacité des RNN à modéliser des séquences de longueur variable leur confère un avantage évident pour traiter des textes ou des discours, où la dépendance contextuelle est essentielle. Les LSTM, en particulier, ont permis d'améliorer la performance dans ces domaines en capturant des dépendances de longue portée, souvent indispensables dans la compréhension du contexte global d'un texte ou d'un discours.

Dans le domaine des séries temporelles, que ce soit pour la prévision des cours de la bourse, la modélisation des données météorologiques ou la détection d'anomalies dans des systèmes industriels, les RNN et LSTM jouent un rôle central. Leur capacité à apprendre des motifs temporels complexes et à faire des prédictions à partir d'historiques riches en dépendances a révolutionné ces secteurs. En reconnaissance vocale, ces architectures ont permis de transformer des signaux audio en texte avec une précision remarquable, en capturant les dépendances temporelles dans la parole.

Les LSTM, grâce à leur capacité à retenir des informations sur de longues périodes, sont également utilisés dans des tâches variées telles que la génération de musique ou la modélisation de séquences biomédicales. Leur aptitude à apprendre des structures à long terme dans des données séquentielles rend leur usage particulièrement précieux dans des champs comme la médecine, la finance ou la synthèse de contenus. Par ailleurs, de nouvelles architectures dérivées ou améliorées des RNN et LSTM continuent d'émerger, comme les GRU (Gated Recurrent Units), qui com-

binent simplicité et performance, ou encore des modèles hybrides intégrant des mécanismes d'attention.

En conclusion, les réseaux de neurones récurrents et leurs variantes, notamment les LSTM, ont apporté une avancée décisive dans la modélisation des données séquentielles. Leur capacité à capturer des dépendances à la fois proches et lointaines a permis des progrès spectaculaires dans le traitement du langage, la reconnaissance vocale, la prévision de séries temporelles, et bien d'autres domaines. Leur développement a également ouvert la voie à des architectures plus complexes et performantes, telles que les Transformers, qui repoussent encore plus loin les limites de ce qui est réalisable avec des données séquentielles. La recherche dans ce domaine est toujours active, avec de nouvelles innovations qui continuent à améliorer la capacité des modèles à comprendre et générer des séquences complexes.

2.4 Auto-encodeurs

Les auto-encodeurs sont une classe de réseaux de neurones non supervisés conçus pour apprendre une représentation efficace des données en se concentrant sur leur reconstruction. Leur principe fondamental repose sur la capacité à encoder une entrée dans une représentation latente de dimension inférieure, puis à décoder cette représentation pour reconstruire l'entrée initiale. Cette approche permet d'extraire des caractéristiques essentielles tout en réduisant la dimensionnalité des données.

L'architecture typique d'un auto-encodeur comprend deux parties principales : l'encodeur, qui compresse l'information d'entrée en une représentation de plus petite dimension, et le décodeur, qui reconstruit l'entrée à partir de cette représentation comprimée. L'entraînement se fait en minimisant une fonction de perte, généralement l'erreur entre l'entrée originale et sa reconstruction, ce qui oblige le modèle à capturer les aspects saillants des données.

Parmi les variantes d'auto-encodeurs, on trouve notamment :

— les auto-encodeurs convolutifs : adaptés aux données structurées spatialement comme les images, utilisant des couches convolutives pour capturer les caractéristiques locales.

— les auto-encodeurs variationnels (VAE) : introduisent une régularisation probabiliste pour apprendre une distribution latente continue, facilitant la génération de nouvelles données similaires à celles d'entraînement.

— les auto-encodeurs débruiteurs : entraînés pour reconstruire des données propres à partir d'entrées bruitées, utiles pour la débruitage et la restauration d'images.

Les auto-encodeurs trouvent de nombreuses applications dans divers domaines, notamment :

— Compression de données : réduction efficace de la taille des données tout en conservant leur contenu essentiel.
— Réduction de dimension : en extrayant des caractéristiques significatives pour des tâches de classification ou de clustering.
— Débruitage et restauration d'images : élimination du bruit ou des artefacts dans les images.
— Génération de nouvelles données : notamment avec les auto-encodeurs variationnels, qui permettent de générer de nouvelles instances en échantillonnant dans l'espace latent.

En résumé, les auto-encodeurs constituent une technique puissante pour l'apprentissage non supervisé, permettant d'apprendre des représentations compactes et pertinentes des données, avec un large éventail d'applications en vision, traitement du signal, et au-delà.

2.5 Méthodes d'apprentissage profond basées sur l'attention

2.5.1 Transformers

Les Transformers, introduits en 2017 par Vaswani et ses collaborateurs dans l'article "Attention is All You Need", ont marqué une étape fondamentale dans l'évolution des modèles de traitement du langage naturel (NLP). Leur architecture repose principalement sur un mécanisme d'attention, qui permet de modéliser efficacement les dépendances entre tous les éléments d'une séquence, tout en permettant un traitement parallèle. Contrairement aux réseaux récurrents (RNN) et aux LSTM, qui traitent les données séquentiellement, les Transformers peuvent traiter toute la séquence simultanément, ce qui accélère considérablement l'entraînement et améliore la capacité de capturer des dépendances à longue portée.

L'architecture d'un Transformer se compose principalement de deux parties : l'encodeur et le décodeur. L'encodeur est constitué de plusieurs couches d'attention multi-tête (Multi-Head Attention) avec des couches de normalisation et de neurones conventionnels. Chaque couche d'attention multi-tête calcule plusieurs représentations différentes en parallèle, ce qui permet au modèle de se concentrer sur différentes parties de la séquence simultanément. Après chaque couche d'attention, une étape de normalisation et un réseau feedforward permettent de transformer

et stabiliser les représentations. Le décodeur possède une structure similaire, mais intègre également des mécanismes d'attention masquée (masked attention) pour assurer que lors de la génération de texte, chaque position ne peut s'appuyer que sur les positions précédentes, garantissant la cohérence temporelle.

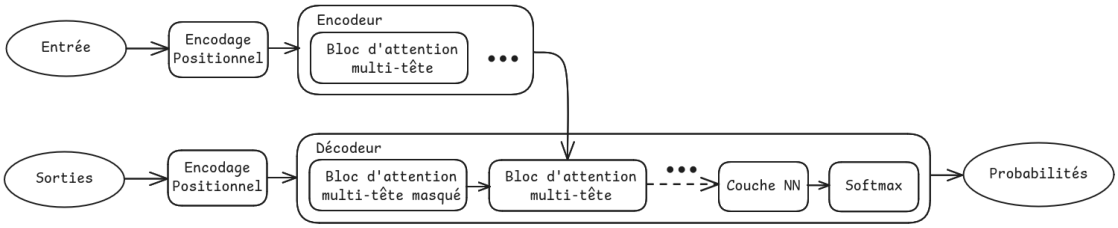


FIGURE 7 – Architecture d'un Transformer

Le processus d'apprentissage d'un Transformer se déroule par rétropropagation classique, mais en intégrant les mécanismes d'attention. Lors de la propagation avant, la séquence d'entrée est traitée par l'encodeur pour produire des représentations contextuelles, qui sont ensuite exploitées par le décodeur pour générer la sortie. La perte, souvent une entropie croisée pour les tâches de classification ou de génération, est calculée en comparant la sortie du modèle à la réponse attendue. La rétropropagation ajuste alors tous les paramètres, y compris ceux des têtes d'attention, des couches de normalisation et des réseaux feedforward, pour minimiser cette erreur. Ce processus est répété sur de nombreux epochs jusqu'à convergence.

Les Transformers ont rapidement trouvé des applications variées et impactantes, notamment dans le domaine du NLP. Parmi les plus célèbres, on trouve BERT (Bidirectional Encoder Representations from Transformers), qui permet des tâches de compréhension du texte telles que la réponse à des questions ou l'analyse de sentiments, et GPT (Generative Pre-trained Transformer), notamment GPT-3, qui excelle dans la génération de textes cohérents, fluides et contextuellement pertinents. Leur capacité à capturer des dépendances complexes dans le texte a permis d'améliorer considérablement la performance dans la traduction automatique, la synthèse de texte, la classification, et la réponse aux questions.

Au-delà du traitement du langage, les Transformers ont aussi été adaptés à la vision par ordinateur. Des modèles comme ViT (Vision Transformer) ont montré qu'en utilisant des mécanismes d'attention, il est possible d'obtenir des performances compétitives voire supérieures aux réseaux convolutionnels (CNN) dans des tâches telles que la classification d'images ou la détection d'objets, notamment avec des architectures comme DETR (Detection Transformer). Leur capacité à modéliser globalement les relations dans une image ouvre de nouvelles perspectives pour la reconnaissance visuelle.

Dans le domaine industriel, les Transformers sont utilisés pour améliorer la compréhension automatique des données textuelles, par exemple dans l'analyse de rapports financiers ou de documents juridiques, ou encore pour renforcer les assistants virtuels comme Siri, Alexa ou Google Assistant. Leur aptitude à traiter de manière efficace de grandes quantités de données textuelles et à produire des réponses pertinentes en fait un outil clé pour les applications de compréhension et de génération automatique de langage.

En conclusion, les Transformers ont transformé radicalement le traitement des données séquentielles en offrant une architecture basée exclusivement sur l'attention, permettant de modéliser efficacement des dépendances longues et complexes. Leur succès dans le NLP, leur adaptation à la vision, et leur potentiel dans de nombreux autres domaines en font une pierre angulaire de la recherche en intelligence artificielle moderne. La poursuite de l'innovation dans ce domaine, avec des architectures améliorées et des techniques hybrides, continue d'étendre leur champ d'application et leur impact.

2.5.2 Attention multi-tête

L'attention dans les Transformers fonctionne en plusieurs étapes, garantissant que la taille de l'entrée n'impacte pas la taille de la sortie.

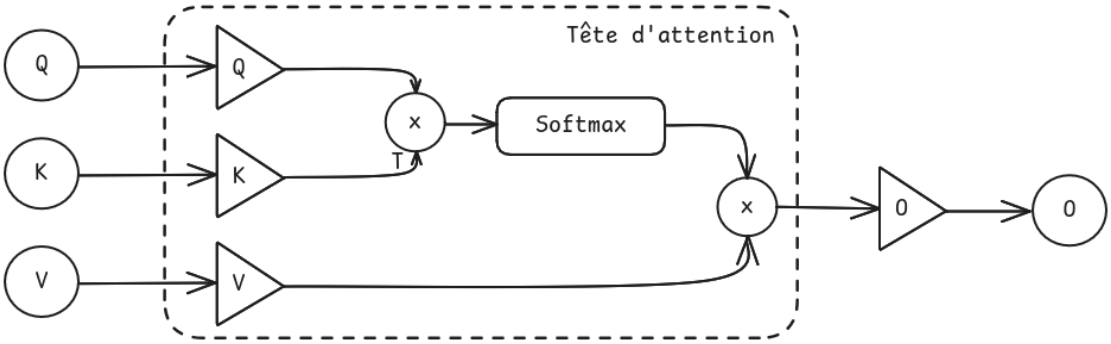


FIGURE 8 – Architecture de la tête d'attention

L'encodeur commence par projeter la représentation latente $Z \in \mathbb{R}^{M \times C}$ et l'entrée $X \in \mathbb{R}^{N \times D}$ en matrices Q, K, V via des transformations linéaires :

$$Z \in \mathbb{R}^{M \times C},\ X \in \mathbb{R}^{N \times D},$$

$$Q = ZW_Q,\ W_Q \in \mathbb{R}^{C \times F} \text{ and } Q \in \mathbb{R}^{M \times F},$$

$$K = XW_K,\ W_K \in \mathbb{R}^{D \times F} \text{ and } K \in \mathbb{R}^{N \times F},$$

$$V = XW_V,\ W_V \in \mathbb{R}^{D\times F} \text{ and } V \in \mathbb{R}^{N\times F}.$$

Z est la représentation latente et X est la donnée d'entrée de longueur arbitraire N. On peut remarquer que les matrices de poids apprises (W_Q, W_K, W_V) ne dépendent pas de N, la longueur de l'entrée (ni de M, la longueur de la représentation latente). Cela garantit que le mécanisme est agnostique à la taille de l'entrée.

Les scores d'attention sont calculés comme le produit scalaire entre la requête (Q, projection de la représentation latente) et la clé transposée ($K^\top$, projection de la donnée d'entrée), mis à l'échelle par $\sqrt{F}$ pour stabiliser les gradients. Mathématiquement :

$$A = \frac{QK^\top}{\sqrt{F}},\ \text{où } A \in \mathbb{R}^{M\times N}.$$

La matrice de scores d'attention résultante A a une forme de (M, N), indiquant que chacune des M requêtes latentes porte attention à tous les N éléments séquentiels. Puisque A dépend de M (taille de la représentation latente) et de N (longueur de l'entrée), elle s'adapte dynamiquement aux entrées de taille variable.

Les scores d'attention sont normalisés en utilisant la fonction Softmax le long de la dimension de la longueur d'entrée (N). Cette étape garantit que les poids d'attention sur tous les N éléments d'entrée pour chaque requête somment à 1, indépendamment de la taille de l'entrée. Les poids d'attention normalisés sont utilisés pour calculer une somme pondérée de la matrice de valeurs V et sont projetés linéairement sur l'espace latent. Cette étape produit la sortie du module d'attention :

$$O = \text{Softmax}(A)VW_O,\ \text{où } O \in \mathbb{R}^{M\times C} \text{ et } W_O \in \mathbb{R}^{F\times C}.$$

L'idée clé est que la forme de O est la même que celle de la représentation latente donnée. La longueur L de l'entrée affecte le calcul d'attention intermédiaire mais ne se propage pas à la sortie. Cela garantit que la taille de la sortie finale reste cohérente, indépendamment de la longueur de l'entrée.

À aucun moment dans ce pipeline, les matrices de poids apprises (W_Q, W_K, W_V, W_O) ne dépendent de la longueur d'entrée N (ni de la longueur de la représentation latente M). Cette indépendance permet de gérer des entrées de taille arbitraire tout en maintenant une sortie cohérente et évolutive. Cette conception est idéale pour les tâches impliquant des données de longueur variable, comme l'étude de cas présentée dans la Section 4.4.

Dans le décodeur, le rôle des représentations est inversé : les représentations latentes sont utilisées comme clés et valeurs, tandis que l'encodage de la sortie

variable sert de requête. Le mécanisme d'attention du décodeur suit le même principe, produisant une sortie de taille variable, cohérente avec la dimension initiale de l'entrée.

2.5.3 Perceiver et Perceiver IO

Les modèles Perceiver et Perceiver IO, développés par DeepMind et introduits respectivement en 2021, représentent une avancée majeure dans le domaine de l'intelligence artificielle, notamment pour leur capacité à traiter et à intégrer des données multimodales de manière flexible et efficace. Ces architectures innovantes ont été conçues pour surmonter les limitations des modèles traditionnels qui sont souvent spécialisés pour un seul type de données, comme les images, le texte ou l'audio. Leur conception repose sur une approche unifiée permettant de traiter divers types de données, de diverses tailles et formats, en utilisant une architecture cohérente et extensible. Cette capacité à gérer la multimodalité ouvre de nombreuses possibilités dans des domaines aussi variés que la vision par ordinateur, la compréhension du langage naturel, la robotique, la médecine, et bien d'autres encore.

2.5.3.1 Perceiver

Les modèles Perceiver (Jaegle, Gimeno, et al. 2021) se distinguent par leur architecture basée sur un mécanisme d'attention inspiré des Transformers, mais optimisée pour réduire la complexité computationnelle liée au traitement de grandes quantités de données. Contrairement aux Transformers classiques, dont la complexité est quadratique par rapport à la taille de l'entrée, Perceiver utilise une étape d'indexation basée sur l'attention croisée pour transformer les données brutes en une représentation latente de taille fixe souvent de taille inférieure à la donnée d'entrée. Un bloc traditionnel d'auto-attention multi-tête est ensuite chaîné sur la représentation latente. Cette approche permet de capter l'essence de l'information tout en maintenant un coût computationnel raisonnable, même pour des entrées très volumineuses ou de nature très différente.

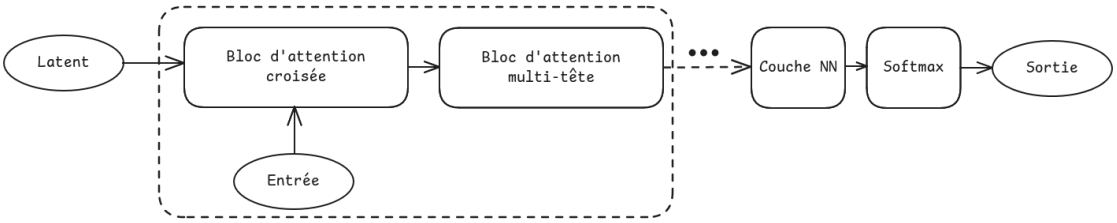


Figure 9 – Architecture du modèle Perceiver

Une caractéristique clé du modèle Perceiver est sa capacité à effectuer une fusion multimodale en traitant plusieurs modalités dans une seule architecture. Par exemple, dans une application de reconnaissance multimodale, il peut analyser une scène visuelle tout en tenant compte du contexte sonore ou textuel associé. Cette capacité à combiner efficacement différentes sources d'information est essentielle pour des tâches complexes telles que la compréhension contextuelle, la recherche d'images par description, ou la traduction multimodale. Par ailleurs, la modularité de l'architecture permet d'ajouter ou de supprimer des modalités sans nécessiter une refonte complète du modèle, ce qui facilite son adaptation à différents cas d'usage. Par exemple, dans le domaine médical, un système basé sur Perceiver pourrait analyser conjointement des images médicales (IRM, radiographies), des dossiers électroniques, et des enregistrements audio de symptômes ou de consultations, pour fournir une synthèse intégrée du diagnostic.

L'application de Perceiver dans la reconnaissance d'images et de textes est particulièrement remarquable. Grâce à sa capacité à associer des descriptions textuelles à des images, il permet de créer des systèmes de recherche d'images par description, où l'utilisateur peut rechercher une image vague ou complexe à partir d'une requête textuelle précise. De même, dans le traitement du langage naturel, il peut comprendre le contexte d'un dialogue en intégrant des données visuelles et auditives pour une meilleure interprétation. La reconnaissance vidéo, par exemple dans la vidéosurveillance ou la surveillance de la sécurité, bénéficie également de cette approche, puisque le modèle peut analyser plusieurs flux simultanément, détecter des comportements suspects ou des événements importants en combinant des données provenant de caméras, de microphones, et de capteurs environnementaux.

Dans le domaine de la robotique, Perceiver offre des possibilités considérables pour la perception environnementale. Un robot équipé d'un tel système peut intégrer des images, des sons, des données tactiles et d'autres capteurs pour former une compréhension globale de son environnement. Cela lui permettrait de naviguer de manière plus sûre et efficace, d'interagir avec des objets ou des personnes, et d'effectuer des tâches complexes dans des environnements dynamiques. La capacité à fusionner plusieurs modalités en temps réel est essentielle pour le développement de robots autonomes capables de percevoir et d'interpréter leur environnement de manière sophistiquée, mimant ainsi certains aspects de la perception humaine.

En médecine, la puissance de Perceiver se traduit par la possibilité d'analyser conjointement une multitude de données médicales. Par exemple, un système pourrait intégrer des images médicales telles que des IRM ou des radiographies, des dossiers de patients contenant des notes écrites, des résultats de tests biologiques, et des enregistrements audio de consultations ou de symptômes rapportés par le

patient. Cette approche intégrée permettrait de fournir un diagnostic plus précis, de détecter des corrélations invisibles pour un examen isolé de chaque donnée, et de proposer des recommandations thérapeutiques adaptées. La capacité à traiter simultanément plusieurs modalités de données constitue une étape clé vers des systèmes de diagnostic assisté par intelligence artificielle plus complets et fiables.

En comparaison avec d'autres architectures, Perceiver se distingue par sa capacité à traiter des entrées volumineuses ou très différentes sans nécessiter de prétraitements ou de découpes spécifiques. Contrairement aux modèles traditionnels qui doivent être adaptés à chaque modalité ou format, Perceiver offre une plateforme unifiée qui peut évoluer avec de nouvelles modalités. Par exemple, alors que les modèles de vision ou de traitement du langage naturel classiques nécessitent souvent des architectures distinctes ou des étapes de fusion complexes, Perceiver intègre toutes ces capacités dans une seule architecture cohérente. Cela facilite la maintenance, la mise à jour, et l'adaptation aux nouveaux défis technologiques ou scientifiques.

Les perspectives futures pour les modèles Perceiver sont vastes. L'un des axes de développement concerne l'amélioration de leur capacité à apprendre de manière plus efficace avec moins de données, en intégrant des techniques d'apprentissage autosupervisé ou semi-supervisé. La montée en puissance des capacités multitâches, où un seul modèle pourrait gérer simultanément plusieurs applications ou types de données, est également envisageable. Par ailleurs, l'intégration de ces modèles dans des systèmes interactifs, autonomes ou en temps réel, tels que des robots ou des assistants intelligents, représente un domaine en pleine expansion. La recherche continue à explorer comment optimiser ces architectures pour qu'elles soient plus rapides, plus économes en ressources, et plus robustes face à la variabilité des données.

2.5.3.2 Perceiver IO

Perceiver IO (Jaegle, Borgeaud, et al. 2021) représente une avancée supplémentaire par rapport au modèle Perceiver en proposant une architecture conçue pour produire des sorties structurées spécifiques à chaque tâche tout en conservant une capacité d'intégration de multiples modalités. Son objectif principal est de percevoir, fusionner et répondre à des contextes variés dans un cadre unifié, en abordant la perception multimodale avec une flexibilité accrue.

L'architecture de Perceiver IO repose sur plusieurs principes fondamentaux. Elle commence par l'encodage des données d'entrée, transformant chaque modalité en représentations numériques via des embeddings ou vecteurs de caractéristiques. Ces

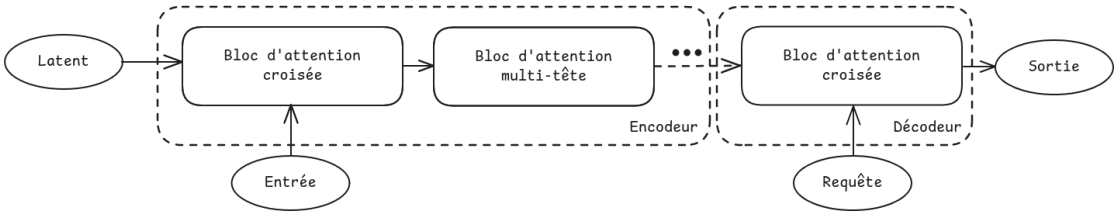


FIGURE 10 – Architecture du modèle Perceiver IO

représentations, issues de sources diverses (images, texte, audio), sont projetées dans un espace latent de taille fixe à l'aide de vecteurs d'indicateurs ("queries"). Cette étape d'indexation réduit la variabilité et la complexité des données tout en conservant leur essence pertinente. Le mécanisme d'attention, avec des couches croisées et d'auto-attention, permet à ces vecteurs latents d'interagir avec les représentations encodées, favorisant une fusion efficace et une synthèse cohérente.

Ce processus aboutit à une représentation interne qui peut être adaptée pour produire une sortie structurée en fonction de la tâche : vecteurs pour la classification, cartes de segmentation ou séquences pour la génération de texte. La conception modulaire des couches finales confère à Perceiver IO une grande flexibilité pour différentes applications, sans nécessiter de modifications majeures dans l'architecture de base.

Contrairement à Perceiver, qui se concentrait principalement sur l'intégration et la compréhension multimodale, Perceiver IO peut générer des sorties variées et structurées en fonction de la tâche, grâce à ses couches finales modulables. L'approche d'encodage et de projection dans un espace latent via des vecteurs d'indicateurs est une innovation qui permet de réduire la variabilité des données tout en conservant leur pertinence. Cette étape d'indexation est une amélioration notable par rapport à la simple fusion de données dans Perceiver.

La capacité de fusion dans Perceiver IO repose sur des couches d'attention croisées entre les vecteurs latents et les représentations encodées, ce qui offre une meilleure efficacité dans l'intégration de données disparates. Sa conception lui permet de traiter efficacement de très grandes quantités de données, même très disparates, tout en produisant des résultats cohérents et structurés. La modularité intégrée simplifie également la conception, offrant une flexibilité accrue pour l'ajout de nouvelles modalités ou tâches avec un effort minimal.

Enfin, cette architecture est conçue pour être scalable et adaptable, ce qui permet de l'étendre à de nouvelles modalités ou applications sans nécessiter de modifications majeures. Les efforts de recherche actuels se concentrent sur l'amélioration de l'efficacité, la réduction des besoins en ressources, et le développement de modèles

multitâches capables de gérer plusieurs applications ou modalités simultanément. La montée en puissance des architectures modulaires et hybrides, combinant Perceiver IO avec d'autres techniques, pourrait conduire à des systèmes encore plus intelligents, autonomes et capables de traiter des situations complexes dans le monde réel.

2.6 Autres avancées et architectures récentes

Au-delà des modèles fondamentaux évoqués précédemment, la recherche en apprentissage profond a permis le développement de plusieurs architectures innovantes et hybrides, qui répondent à des besoins spécifiques ou combinent plusieurs techniques pour améliorer la performance et la flexibilité des modèles.

Parmi ces avancées, les réseaux antagonistes génératifs (GANs) occupent une place centrale. Introduits par (Goodfellow et al. 2020), les GANs se composent de deux réseaux opposés : un générateur qui crée des données synthétiques, et un discriminateur qui distingue les données réelles des données générées. Leur entraînement antagoniste permet de produire des images, vidéos, ou autres types de données synthétiques d'une qualité remarquable, ouvrant la voie à des applications en synthèse d'images, design, augmentation de données, et plus encore.

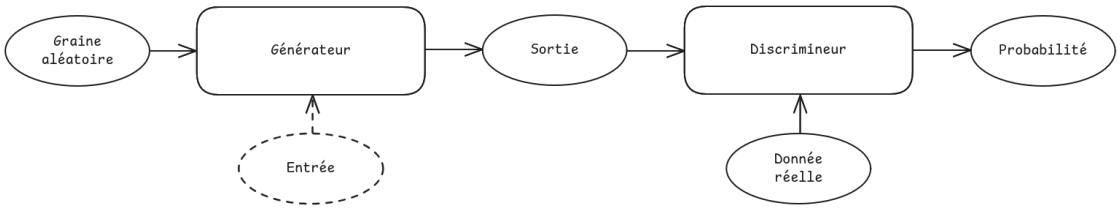


FIGURE 11 – Architecture de l'apprentissage d'un GAN

Les modèles hybrides et multimodaux combinent différentes architectures pour traiter des données complexes ou multi-sources. Par exemple, des architectures intégrant des réseaux convolutifs avec des modules de traitement du langage naturel (transformers) permettent de gérer des données textuelles et visuelles simultanément, favorisant des applications en vision par ordinateur multimédia, systèmes de recommandation, ou assistants intelligents.

Une autre tendance notable concerne les réseaux de neurones profonds à attention, qui ont permis d'améliorer la capacité de modélisation des dépendances à longue portée dans les données séquentielles ou structurées. Ces architectures, notamment les transformers, ont conduit à des modèles de grande envergure capables

d'apprendre des représentations riches et contextualisées, avec des applications en traduction automatique, synthèse vocale, et analyse de texte.

Enfin, les approches auto-supervisées ont émergé comme une réponse à la nécessité d'entraîner des modèles puissants sans dépendre exclusivement de données annotées. Ces méthodes exploitent des tâches prétextes pour apprendre des représentations générales, qui peuvent ensuite être affinées pour des tâches spécifiques avec peu de données annotées, ce qui a considérablement renforcé la capacité d'apprentissage dans divers domaines.

En résumé, ces architectures récentes et hybrides illustrent la dynamique d'innovation constante dans le domaine de l'apprentissage profond, permettant de relever des défis de plus en plus complexes et d'étendre la portée des applications dans des secteurs variés tels que la vision, le traitement du langage, la génération de contenu ou l'intelligence artificielle explicable.

2.7 Conclusion

Cet état de l'art a permis de mettre en lumière la richesse et la diversité des modèles d'apprentissage profond, ainsi que leur évolution rapide depuis leurs origines jusqu'aux architectures modernes et innovantes. Nous avons exploré les principes fondamentaux qui sous-tendent ces techniques, tels que les réseaux de neurones, les auto-encodeurs, et les architectures avancées comme les transformers, tout en illustrant leurs applications concrètes dans des domaines variés tels que la vision, le traitement du langage naturel, la génération de contenu, et bien plus encore.

L'apprentissage profond s'est rapidement imposé comme une composante essentielle de l'intelligence artificielle moderne, grâce à sa capacité à modéliser des relations complexes et à apprendre des représentations hiérarchiques à partir de vastes ensembles de données. Toutefois, cette croissance s'accompagne aussi de défis, notamment en termes de nécessités computationnelles, de transparence, et de biais potentiels.

Les avancées récentes, telles que les modèles génératifs, les architectures hybrides et auto-supervisées, témoignent d'une recherche dynamique orientée vers une meilleure efficacité, une plus grande généralisation, et une adaptation à des tâches de plus en plus complexes. Ces innovations laissent entrevoir des perspectives prometteuses, notamment dans la création de systèmes plus autonomes, explicables, et éthiques.

En somme, l'apprentissage profond continue de transformer le paysage de l'intelligence artificielle, tout en suscitant des enjeux importants pour la recherche, l'industrie, et la société. La compréhension approfondie de ses modèles et de ses

architectures est essentielle pour exploiter pleinement leur potentiel et relever les défis futurs.

Chapitre 3

Première contribution : Compression de signaux temporels pour l'embarqué

L'objectif principal du projet RailMon est d'évaluer l'état des mécanismes d'aiguillages sur les infrastructures ferroviaires en service. Les capteurs installés sur ces derniers enregistrent les réponses vibratoires de ces appareils lors du passage d'un train. La première problématique est de compresser ces signaux temporels sur le dispositif embarqué afin qu'il soit transmis au concentrateur de données avec une consommation d'énergie minimale. Au vu du protocole de communication utilisé, on s'attend à réduire des données vectorielles de 10 000 à 100 000 valeurs sur 256 octets, taille maximale de la charge utile d'un paquet LoRa, soit un rapport de compression entre 150 et 1500. La perte liée à la compression doit permettre la reconstruction de la forme globale du signal.

3.1 Étude exploratoire

3.1.1 Données

Pour tester les modèles présentés précédement, on utilise un jeu de données uniforme de 642 signaux fournis par VOSSLOH. Ces derniers correspondent à des enregistrements vibratoires de trains franchissant un aiguillage ferroviaire sur une fenêtre de septembre 2022 à janvier 2023. L'uniformité du jeu de données vient du fait que ces signaux sont générés par des convois identiques franchissant à une vitesse similaire le même appareil de voie. Les signaux considérés ont une taille variant entre 10 000 et 12 000 échantillons, ce qui nous permet d'évaluer les performances des différents modèles.

L'objectif de cette première étude est d'évaluer les métriques d'apprentissage, les tailles et les qualités de reconstructions afin de comparer les modèles précédement cités.

3.1.2 Transformée de Fourier à Court Terme (STFT)

Plusieurs modèles ont été testés pour reconstruire directement le signal temporel sans pré-traitement. Il s'avère qu'aucun résultat positif n'a été obtenu. L'objectif de la compression n'étant pas de reconstruire parfaitement le signal, on utilise un pré-traitement pour filtrer le bruit.

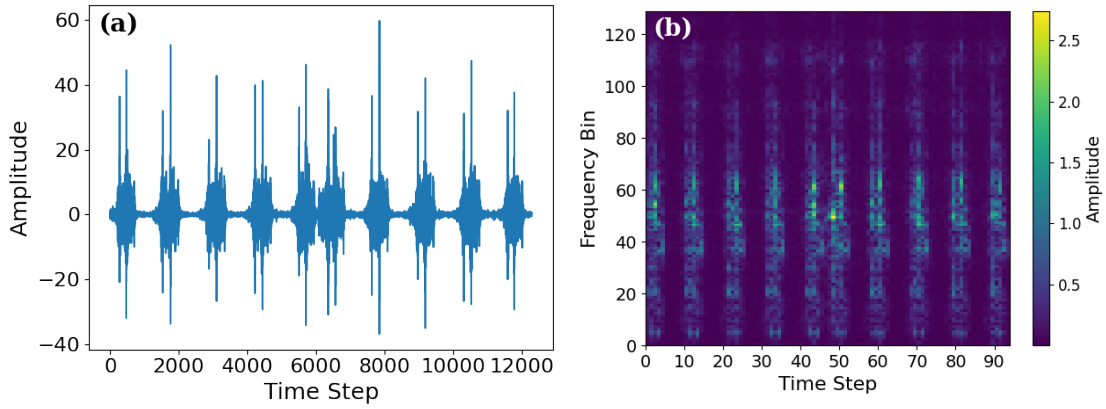
Dans cette étude, nous appliquons la Transformée de Fourier à Court Terme (STFT) pour analyser nos signaux vibratoires. La STFT est avantageuse pour étudier les signaux non stationnaires, car elle fournit une représentation temps-fréquence permettant de capturer événements transitoires et variations de fréquence (Griffin et Lim 1984). Il existe d'autres transformées, dont la transformée en cosinus discrète (DCT) ou la transformée en ondelette. L'avantage de la transformée de Fourier est la corrélation de l'intensité en temps et en fréquence, ainsi que sa simplicité d'implémentation qui la rend très rapide d'exécution, notamment sur certaines cibles embarquées disposant d'un bloc de traitement spécifique.

Le processus consiste à segmenter le signal en fenêtres se chevauchant, puis à appliquer la transformée de Fourier à chaque segment. Cette méthode conserve à la fois des informations temporelles et fréquentielles, permettant d'observer l'évolution des composantes du signal dans le temps. En choisissant judicieusement la taille de la fenêtre, on peut équilibrer résolution temporelle et fréquentielle, faisant de la STFT un outil polyvalent pour analyser des signaux vibratoires complexes. Ainsi, pour un instant temporel t et une pulsation fréquentielle ω, la composante complexe $Z(t, \omega)$ s'écrit :

$$Z(t, \omega) = \sum_{n}^{w[n-t] \neq 0} x[n] w[n-t] e^{-j\omega n}$$

Le module de $Z(t, \omega)$ correspond à l'intensité de la composante fréquentielle de pulsation ω sur la fenêtre centrée en t. C'est cette information, représentée sous la forme d'un spectrogramme, qui constitue les données d'entrée du modèle de compression.

La Figure 12a illustre un exemple de signal dans le domaine temporel, tandis que la Figure 12b présente sa représentation en spectrogramme.



(a) Signal temporel

(b) Spectrogramme

FIGURE 12 – Double représentation des signaux.

3.1.3 Problématique

On cherche à obtenir deux fonctions matricielles f et g représentant respectivement les opérations de compression et de décompression. Soit X la matrice de données à compresser, i.e. le spectrogramme du signal vibratoire, Y sa représentation latente compressée, et X' la décompression de Y. On a :

$$Y = f(X)$$

$$X' = g(Y)$$

$$|X'| = |X|, |Y| \ll |X|, d(X, X') < \epsilon$$

où $|\cdot|$ représente l'opérateur de taille, i.e. le nombre d'élément de la matrice considérée, $d(\cdot, \cdot)$ désigne une fonction de distance entre deux matrices et ϵ la difference entre la donnée décompressée et la donnée originale, i.e. la perte lié à la compression.

Lorsque l'on utilise un modèle autoencodeur, la partie encodage correspond à la fonction de compression f et la partie décodage la fonction de décompression g. L'entraînement de ce type de modèle se réalise donc simplement en minimisant la distance entre la sortie de l'autoencodeur avec son entrée par la fonction de son choix. La validation des performances en reconstruction peut être réalisée de la même façon.

3.1.4 Choix du modèle

Nous présentons ci-dessous un aperçu détaillé des différentes architectures qui ont été testées dans le cadre de cette étude, ainsi que les résultats obtenus pour chacune d'entre elles. Ces architectures ont été sélectionnées en fonction de critères empiriques, en s'appuyant sur des règles de construction éprouvées pour ce type de modèles, afin d'assurer une cohérence et une comparabilité des performances.

Les configurations proposées sont considérées comme étant proches de l'architecture optimale, selon les règles et les principes établis lors de la conception, notamment celles liées à la spécification des modèles et aux meilleures pratiques en matière d'optimisation. Ces règles empiriques ont permis d'orienter la sélection des paramètres, la structure des couches, ainsi que les stratégies d'entraînement, afin d'obtenir des performances maximales tout en maintenant une complexité raisonnable.

Par ailleurs, dans le cadre de cette démarche, d'autres architectures ont été expérimentées mais ne sont pas toutes détaillées dans cette section. Certaines configurations alternatives ont été testées pour explorer différentes approches ou pour valider la robustesse des résultats, mais n'ont pas été retenues comme modèles de référence dans l'analyse finale. Ces expérimentations complémentaires ont permis d'affiner la compréhension des paramètres clés et de confirmer la pertinence des architectures présentées ici.

L'objectif de cette présentation est de fournir une vision claire des choix effectués, des compromis réalisés, et des performances observées, afin de soutenir la démarche d'optimisation et de sélection du modèle le plus adapté à nos besoins spécifiques.

3.1.4.1 Réseau de neurones conventionnel

La construction des architectures de réseaux de neurones repose sur un ensemble de règles empiriques, validées par l'expérience et les bonnes pratiques du domaine. Ainsi, le rapport entre le nombre de neurones de deux couches successives est généralement maintenu proche de 4. Cette proportion permet de favoriser une réduction progressive de la dimensionalité des données, tout en évitant une perte brutale d'information entre les couches.

Par ailleurs, le nombre de neurones dans les couches cachées est souvent choisi comme une puissance de 2. Cette convention facilite non seulement l'optimisation des calculs sur les architectures matérielles modernes, mais contribue aussi à une meilleure organisation et une plus grande efficacité de l'apprentissage.

Une autre observation clé concerne l'importance relative des couches au sein du réseau : les couches les plus profondes ont tendance à jouer un rôle moins critique

que les premières couches dans l'extraction des caractéristiques. Pour cette raison, on autorise un rapport plus élevé entre le nombre de neurones des couches profondes et celles qui les précèdent, ce qui permet de réduire la complexité du modèle sans sacrifier significativement ses performances.

Enfin, le choix des fonctions d'activation est également guidé par des principes empiriques. Pour l'encodeur, on privilégie la fonction ReLU (Rectified Linear Unit), dont la propriété de parcimonie permet de sélectionner et de conserver les informations les plus pertinentes. À l'inverse, le décodeur utilise généralement une fonction sigmoïde, mieux adaptée à la reconstruction des données, car elle permet de produire des valeurs comprises dans un intervalle borné, souvent nécessaire pour restituer fidèlement les entrées originales.

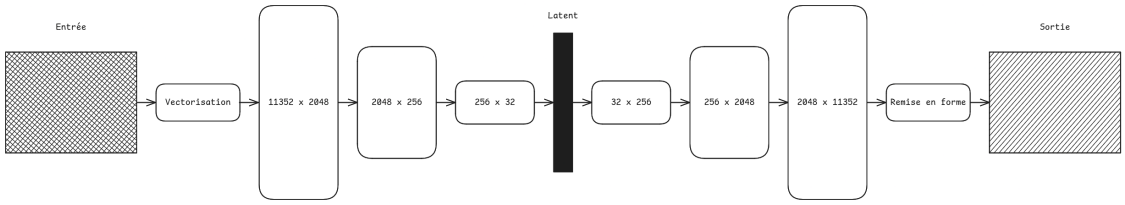


FIGURE 13 – Architecture du modèle de réseaux de neurones conventionnels évalué

Ainsi, on aboutit à l'architecture suivante :

- Encodeur :
  - Couche d'entrée : 11352 neurones = 88 fenêtres de 129 composantes fréquentielles
  - Première couche cachée : 2048 neurones = 11352 / 4 arrondi à la puissance de 2 la plus proche
  - Première couche d'activation : ReLU
  - Deuxième couche cachée : 256 neurones = 2048 / 8
  - Deuxième couche d'activation : ReLU
  - Couche de sortie : 32 neurones = 256 / 8
  - Couche d'activation en sortie : ReLU
- Décodeur :
  - Couche d'entrée : 32 neurones
  - Première couche cachée : 256 neurones
  - Première couche d'activation : Sigmoïde
  - Deuxième couche cachée : 2048 neurones
  - Deuxième couche d'activation : Sigmoïde
  - Couche de sortie : 11352 neurones
  - Couche d'activation en sortie : Sigmoïde

L'auto-encodeur utilisé, présenté en Figure 13, est symétrique à l'exception des biais. L'encodeur utilise 23 783 712 paramètres, le décodeur 23 795 032 paramètres, pour un total de 47 578 744 paramètres. L'apprentissage est réalisé en utilisant 32 signaux par lot, i.e. les gradients de 32 signaux en entrée servent pour une rétro-propagation et un pas d'apprentissage de l'optimiseur. La Figure 14 présente l'évolution des métriques d'entraînement.

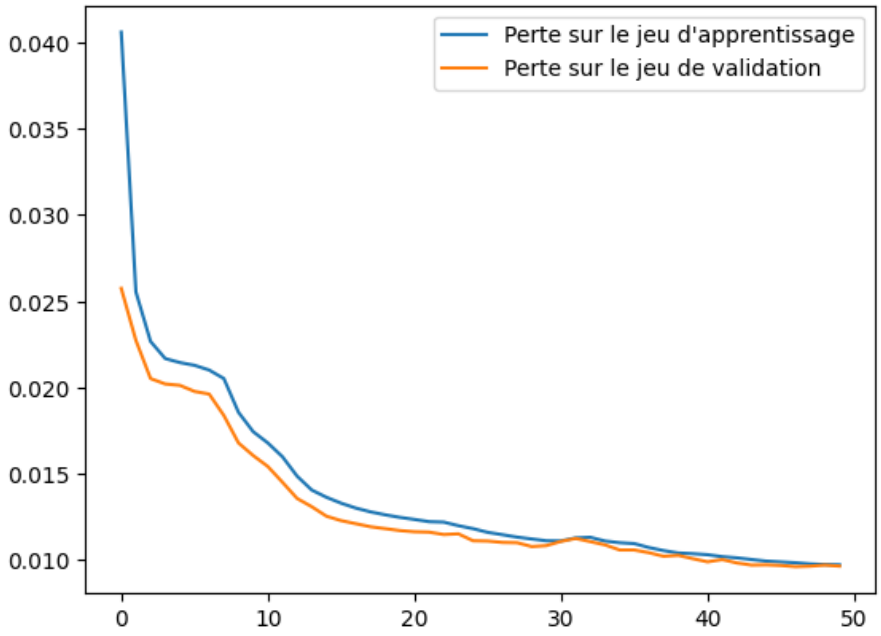


FIGURE 14 – Courbe d'apprentissage du modèle de réseaux de neurones conventionnels

Les métriques obtenues sur le spectrogramme reconstruit sont :
— erreur quadratique : 0,009636
— similarité en cosinus : 91,04 %
— PSNR : 32,56
— SSIM : 0,8841
On présente sur la Figure 15 des exemples de reconstructions en spectrogramme.

3.1.4.2 Réseau de neurones convolutif

Là encore, nous appliquons plusieurs règles empiriques pour guider la conception de nos architectures d'autoencodeurs convolutionnels. La première consiste à respecter tant que possible la symétrie du modèle : on préfèrera donc effectuer les convolutions sur des matrices de tailles paires. Cela permet d'assurer que la phase de déconvolution, ou transposition de convolution, puisse reconstruire l'ensemble

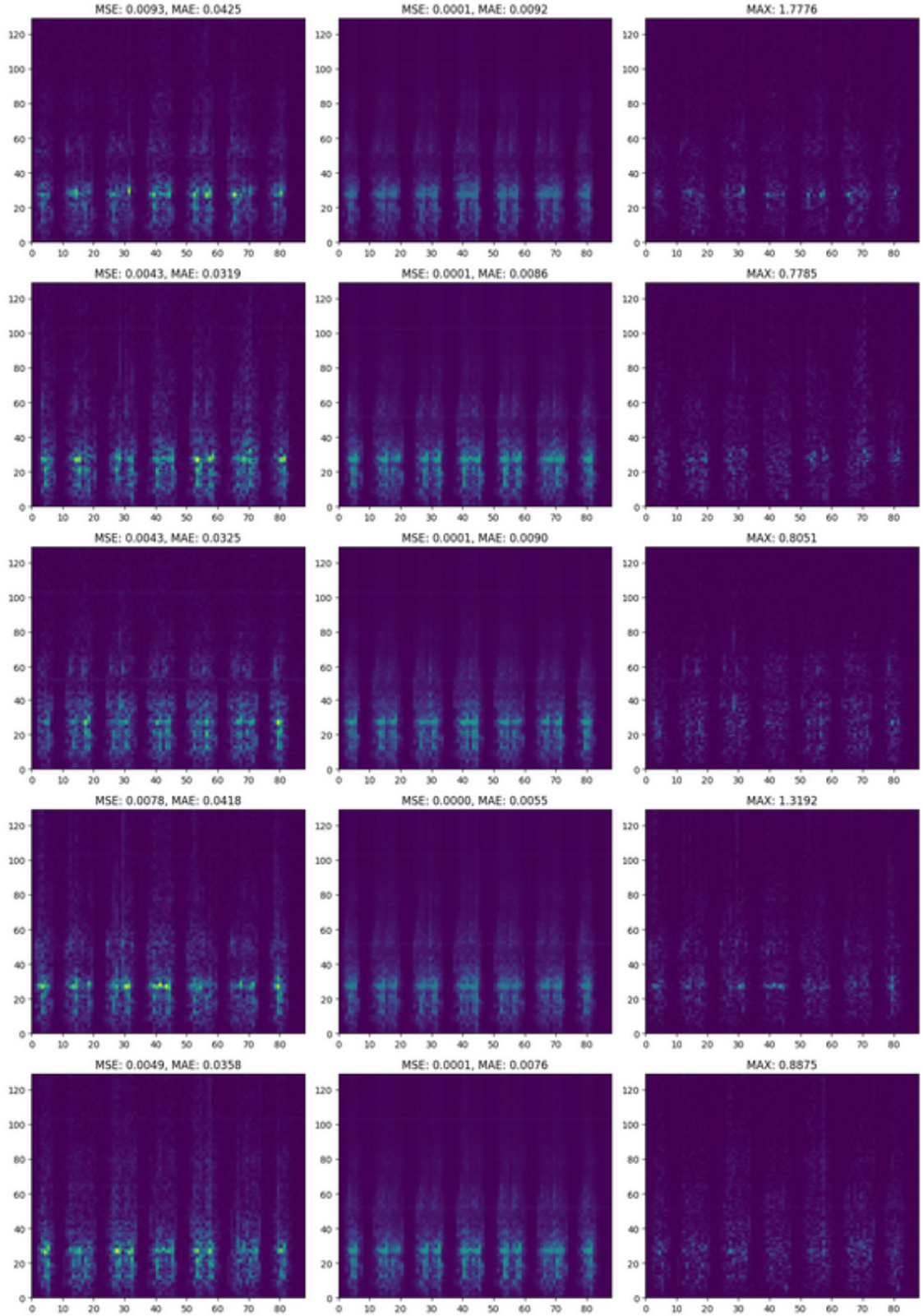


FIGURE 15 – Exemple de reconstructions en spectrogramme par le réseau de neurones conventionnel (gauche : original, centre : reconstruit, droite : erreur)

des données d'origine sans perte d'information. En conséquence, la taille des noyaux de convolution est choisie de manière à respecter cette contrainte, facilitant ainsi un processus de reconstruction précis.

De plus, il est observé que plus une couche est profonde dans le réseau, plus le nombre de canaux doit être élevé. Cette règle vise à augmenter la capacité d'expression du modèle à mesure que l'information progresse vers les couches de traitement supérieur, permettant une extraction de caractéristiques plus riches et plus abstraites.

Enfin, on respectera les règles utilisées dans la section précédente pour la partie comportant les couches de neurones conventionnels.

Dans ce modèle, l'application d'une couche de convolution de noyau W avec un biais b sur une matrice X est défini par la formule suivante :

$$Y = b + \sum_c W_c \star X_c$$

où c représente le canal considéré en entrée et le produit $\star$ correspond au produit de corrélation croisé, i.e. le glissement du noyau sur les dimensions de la matrice en entrée.

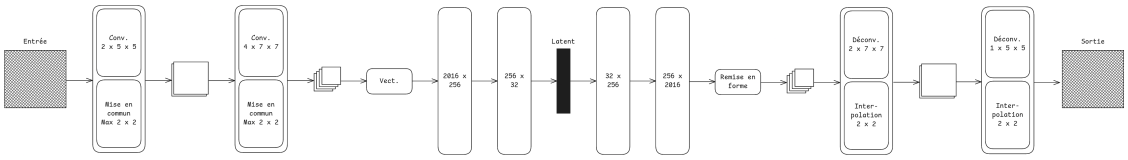


FIGURE 16 – Architecture du modèle de réseaux de neurones convolutifs évalué

L'architecture présentée sur la Figure 16 est la suivante :

- Encodeur :
  - Couche d'entrée : 88 × 129 sur 1 canal
  - Première couche de convolution : 2 noyaux 5 × 5
  - Première couche d'activation : ReLU
  - Première couche de mise en commun : maximum sur 2 × 2
  - Deuxième couche de convolution : 4 noyaux 7 × 7
  - Deuxième couche d'activation : ReLU
  - Deuxième couche de mise en commun : maximum sur 2 × 2
  - Couche de vectorisation
  - Couche cachée : 256 neurones
  - Couche de sortie : 32 neurones
- Décodeur :
  - Couche d'entrée : 32 neurones

— Première couche cachée : 256 neurones
— Deuxième couche cachée : 2016 neurones
— Couche de remise en forme : 18 × 28 sur 4 canaux
— Première couche d'interpolation : 2 × 2
— Première couche de déconvolution : 2 noyaux 7 × 7
— Première couche d'activation : Sigmoïde
— Deuxième couche d'interpolation : 2 × 2
— Deuxième couche de déconvolution : 1 noyau 5 × 5
— Deuxième couche d'activation : Sigmoïde
— Couche de remplissage : 1 colonne de zéros à droite

L'auto-encodeur utilisé est symétrique à l'exception des biais et de la couche de remplissage pour rajouter la dernière composante fréquentielle perdue lors de la première convolution. L'encodeur utilise 525 024 paramètres, le décodeur 527 005 paramètres, pour un total de 1 052 029 paramètres. L'apprentissage est réalisé en utilisant 32 signaux par lot. La Figure 17 présente l'évolution des métriques d'entraînement.

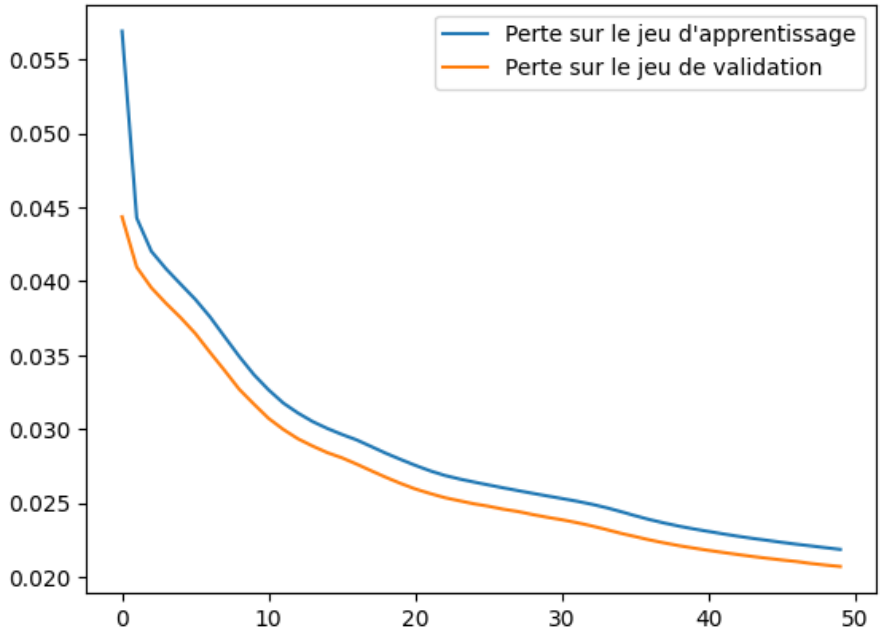


FIGURE 17 – Courbe d'apprentissage du modèle de réseaux de neurones convolutif

Les métriques obtenues sur le spectrogramme reconstruit sont :
— erreur quadratique : 0,02158
— similarité en cosinus : 81,42 %
— PSNR : 29,06

— SSIM : 0,7572

On présente sur la Figure 18 des exemples de reconstructions en spectrogramme.

3.1.4.3 RNN et LSTM

Le modèle RNN n'a pas donné de résultats satisfaisants, les signaux étant de trop grandes tailles pour être reconstruits correctement. On présente donc ici les résultats du modèle LSTM.

La cellule LSTM utilisée est cohérente avec l'implémentation originale (Hochreiter et Schmidhuber 1997). L'encodeur utilise 132 096 paramètres et le décodeur 133 128 paramètres pour un total de 265 224 paramètres. La principale différence à noter par rapport aux modèles précédents est le long temps de convergence lors de l'apprentissage et la présence d'instabilités au cours de l'optimisation des paramètres (Figure 19).

Les métriques obtenues sur le spectrogramme reconstruit sont :

— erreur quadratique : 0,01737

— similarité en cosinus : 59,70 %

— PSNR : 30,00

— SSIM : 0,6420

On présente sur la Figure 20 des exemples de reconstructions en spectrogramme.

3.1.4.4 Perceiver IO

Au sein du bloc de traitement, il est nécessaire de fixer deux paramètres : la dimension des caractéristiques internes (F) et la dimension de la couche cachée du MLP (H). La dimension F est choisie pour se situer entre les dimensions des deux entrées considérées. Cette approche permet de préserver une représentation suffisamment riche pour capturer les informations essentielles, tout en évitant une redondance ou une perte excessive de détails. Par ailleurs, la dimension H est quand à elle fixée à quatre fois la dimension de la requête.

Le modèle testé (Figure 21) présente des blocs d'attention mono-tête, contrairement à l'implémentation conventionnelle. L'encodeur ne présentant pas de blocs de traitement supplémentaires, il est nettement plus léger que le décodeur : 46 004 paramètres pour l'encodeur et 156 299 paramètres pour le décodeur, pour un total de 202 303 paramètres. La donnée compressée en sortie de l'encodeur a pour dimensions 4 par 8 et l'apprentissage est réalisé en utilisant 32 signaux par lot. L'encodage positionnel utilisé, combinaison de sinus et de cosinus, est de dimension 32. Il est utilisé à la fois en concaténation de l'entrée pour l'encodeur et comme requête de sortie pour le décodeur. La dimension de sortie étant identique à celle

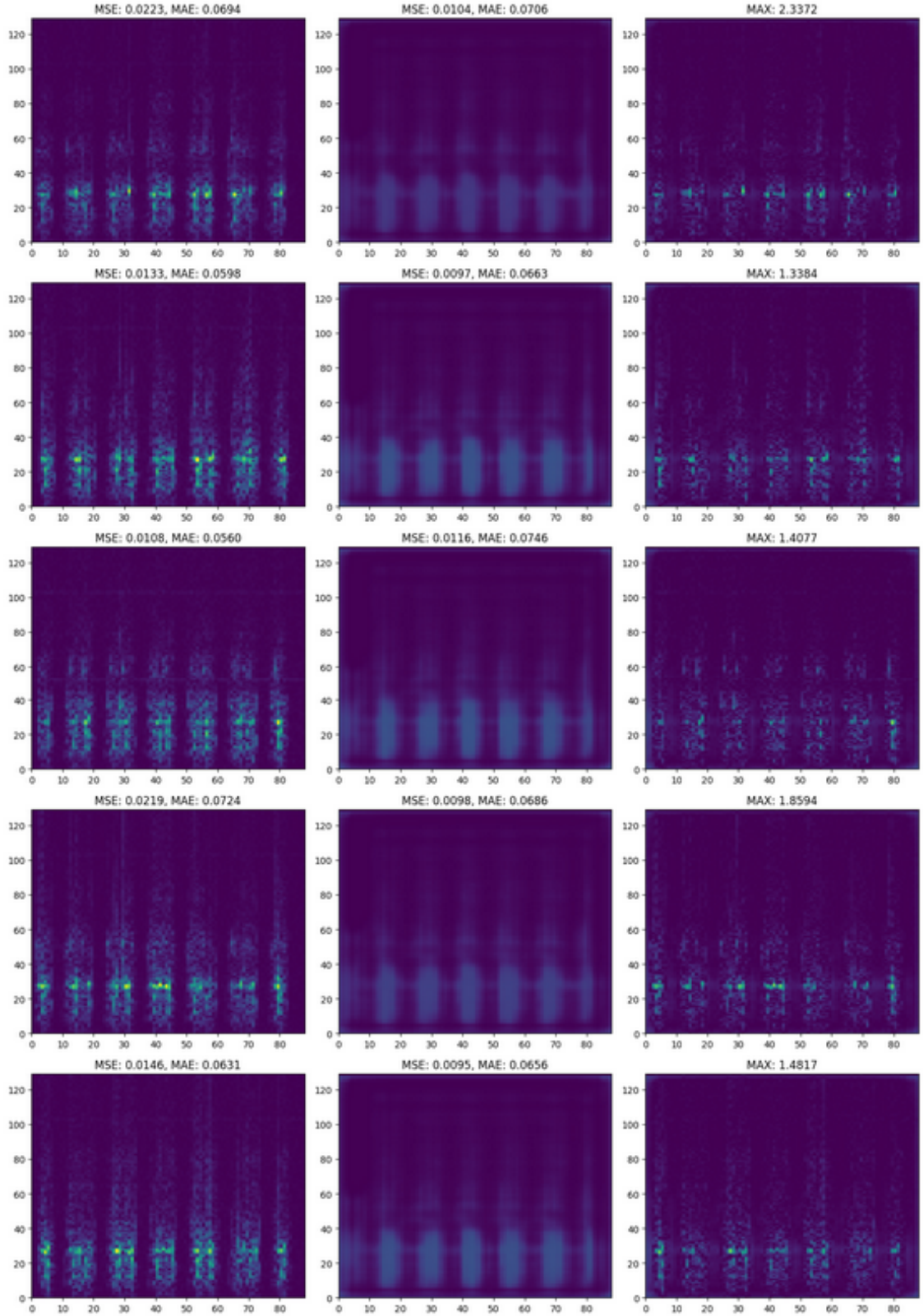


FIGURE 18 – Exemple de reconstructions en spectrogramme par le réseau de neurones convolutif (gauche : original, centre : reconstruit, droite : erreur)

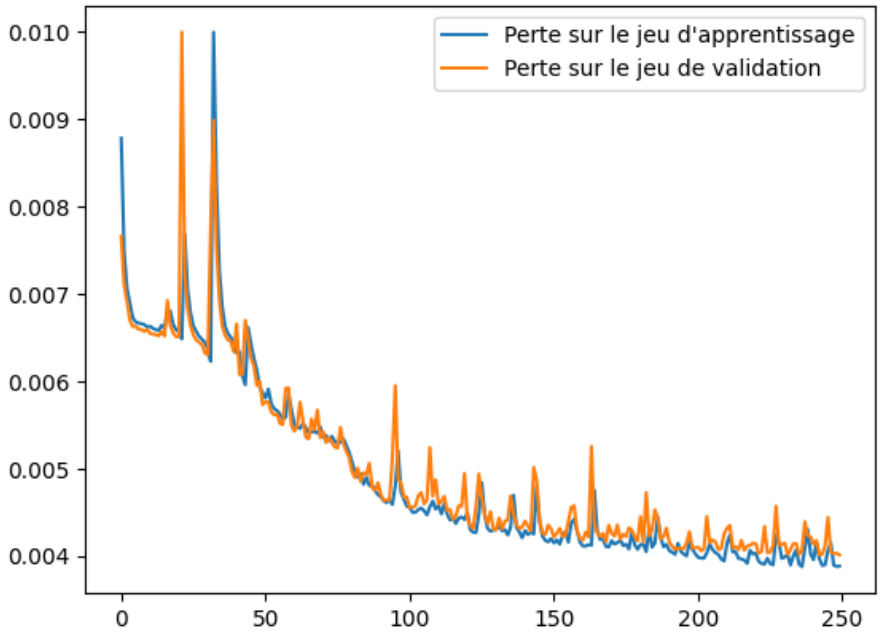


FIGURE 19 – Courbe d'apprentissage du modèle LSTM

en entrée et donc différente de la dimension de l'encodage positionnel, la somme de la sortie du bloc d'attention avec la requête n'est pas effectuée au sein du décodeur. La Figure 22 présente l'évolution des métriques d'entraînement.

Comme montré sur les courbes d'apprentissage, le modèle Perceiver IO converge plus rapidement que les autres modèles. La fonction de perte utilisée diffère également par rapport aux précédents entraînements et se présente comme suit :

$$\text{Loss}(x, y) = MSE(x, y) + (1 - \text{Cosine Similarity}(x, y))$$

Les métriques obtenues sur le spectrogramme reconstruit sont :

— erreur quadratique : 0,01211
— similarité en cosinus : 88,96 %
— PSNR : 31,57
— SSIM : 0,7082

On présente sur la Figure 23 des exemples de reconstructions en spectrogramme.

3.1.5 Synthèse

La Table 3.1 présente la synthèse des performances des quatre modèles testés. En termes de qualité de reconstruction, le modèle NN affiche les meilleures performances sur toutes les métriques : une erreur quadratique moyenne (MSE) de

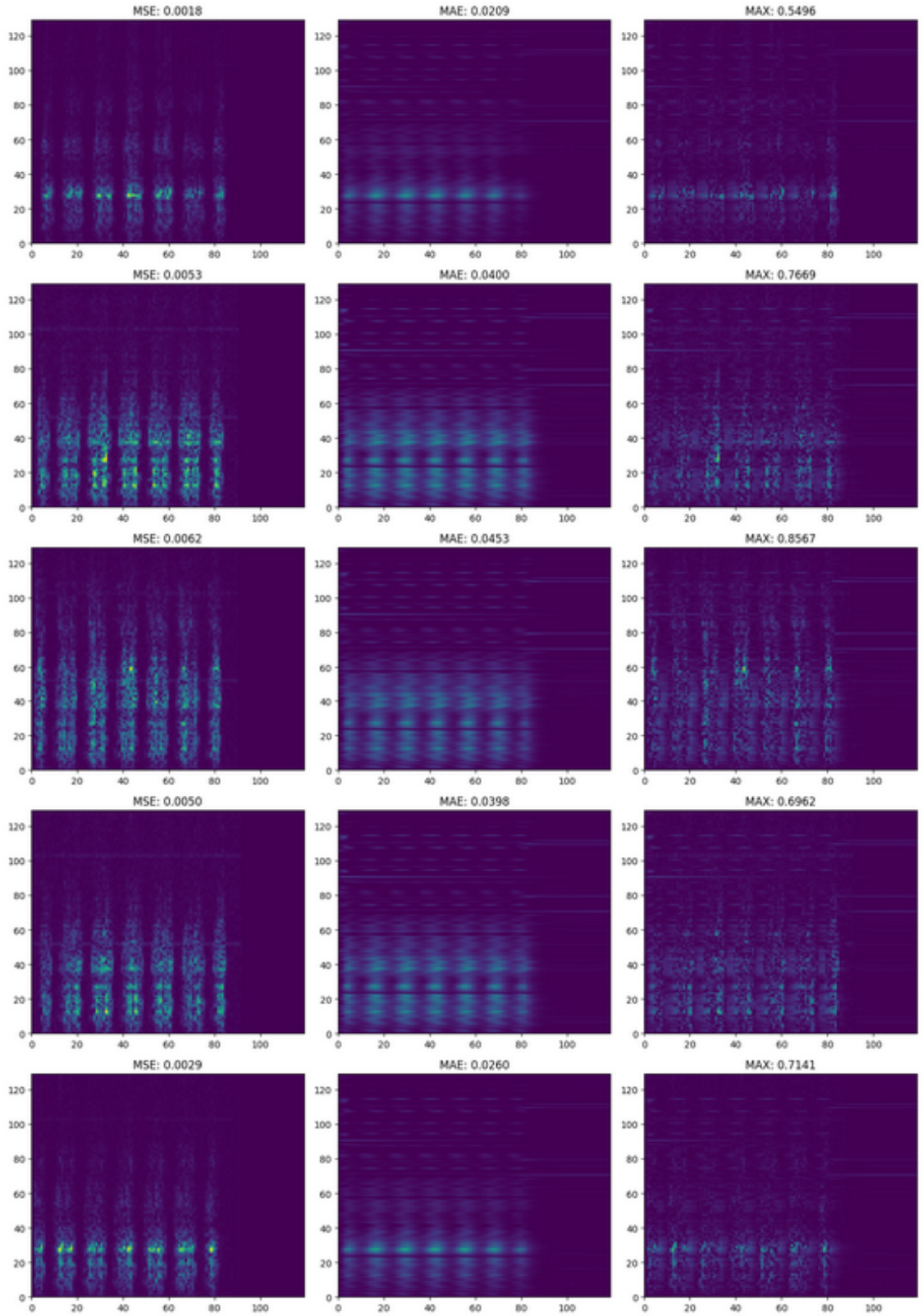


FIGURE 20 – Exemple de reconstructions en spectrogramme par le modèle LSTM (gauche : original, centre : reconstruit, droite : erreur)

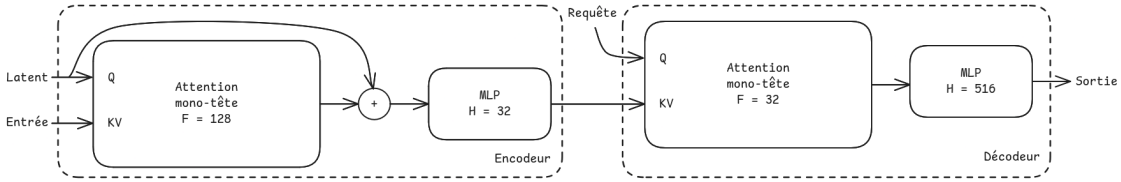


FIGURE 21 – Architecture du modèle Perceiver IO étudié

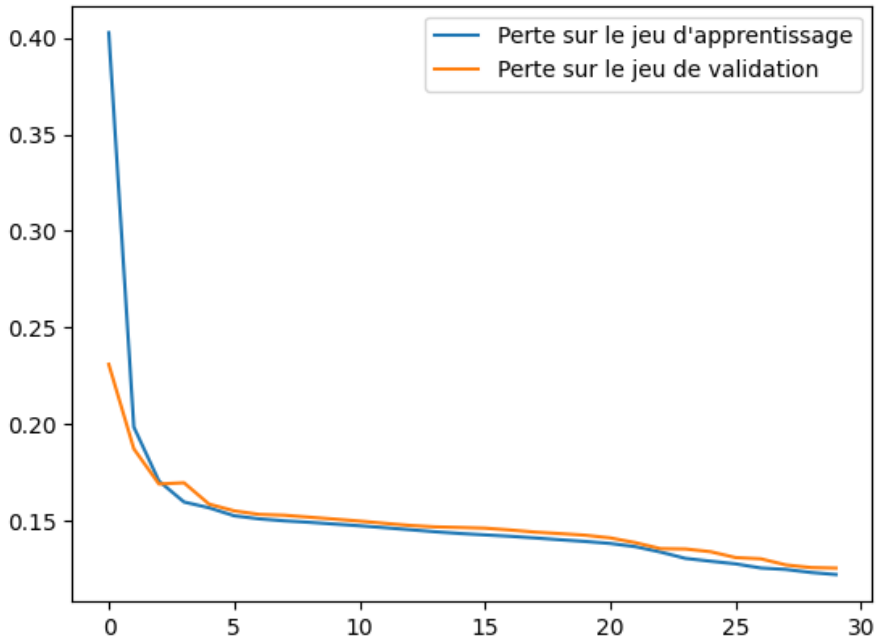


FIGURE 22 – Courbe d'apprentissage du modèle Perceiver IO

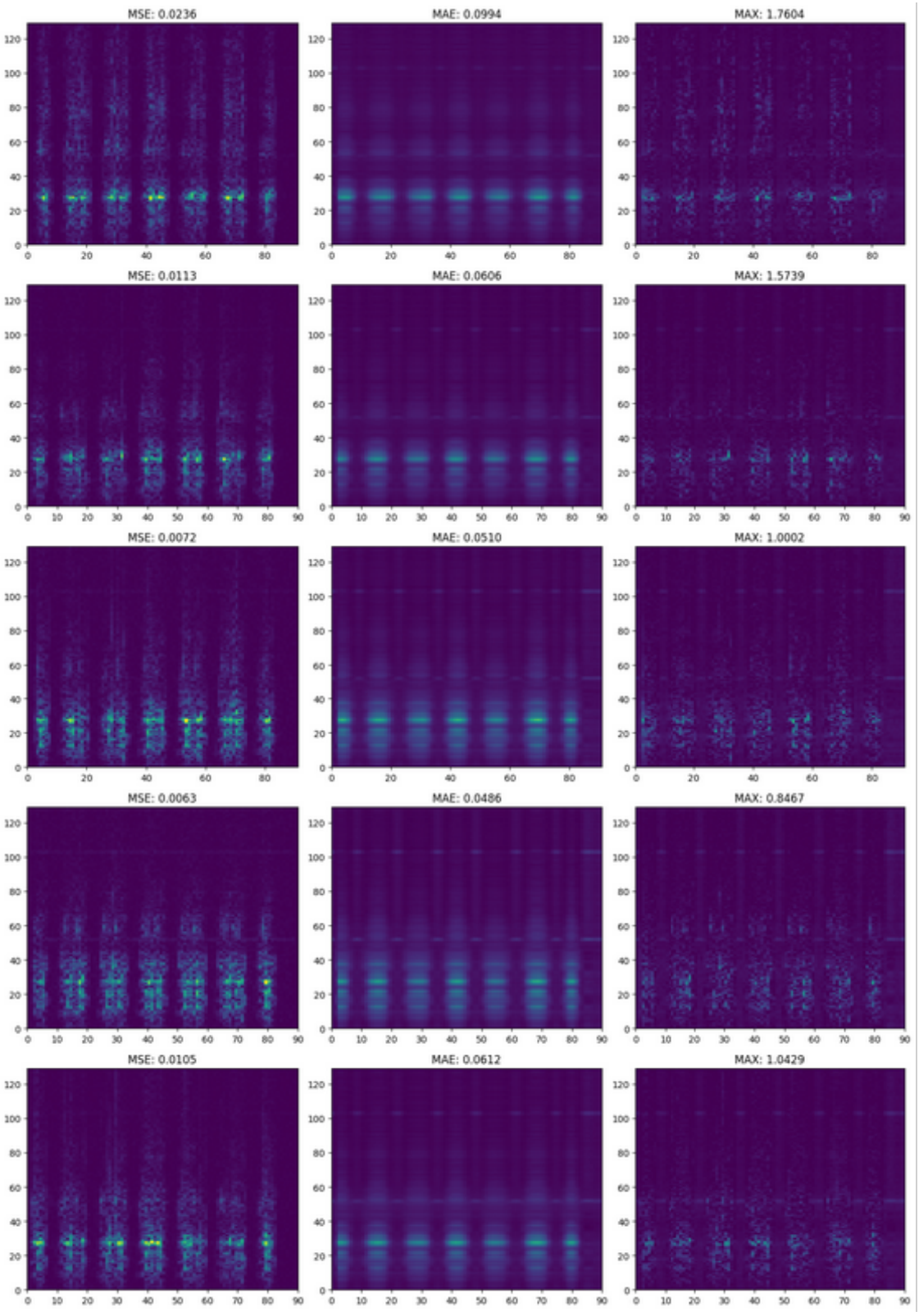


FIGURE 23 – Exemple de reconstructions en spectrogramme par le modèle Perceiver IO (gauche : original, centre : reconstruit, droite : erreur)

0,009636, une similarité en cosinus de 91,04 %, un rapport signal sur bruit de crête (PSNR) de 32,56 et un indice de similarité structurelle (SSIM) de 0,8841. Le modèle Perceiver IO arrive en deuxième position avec des résultats très proches : un MSE de 0,01211, une similarité en cosinus de 88,96 %, un PSNR de 31,57 et un SSIM de 0,7082. Les modèles CNN et LSTM obtiennent des performances inférieures, notamment le LSTM qui présente une similarité en cosinus particulièrement basse (59,70 %).

TABLE 3.1 – Synthèse des performances des différents modèles testés

Modèle	MSE	Similarité en cosinus	PSNR	SSIM
NN	**0,009636**	**91,04 %**	**32,56**	**0,8841**
CNN	0,02158	81,42 %	29,06	0,7572
LSTM	0,01737	59,70 %	30,00	0,6420
Perceiver IO	0,01211	88,96 %	31,57	0,7082

La Table 3.2 présente les tailles des différents modèles testés. Côté complexité, le modèle NN se distingue par son nombre très élevé de paramètres, totalisant près de 48 millions, ce qui le rend lourd et peu adapté aux environnements contraints. À l'inverse, Perceiver IO se révèle extrêmement léger avec seulement 202 303 paramètres, soit environ 200 fois moins que le NN. Le LSTM et le CNN occupent une position intermédiaire, avec respectivement 265 224 et 1 052 029 paramètres.

TABLE 3.2 – Synthèse des tailles des différents modèles testés en nombre de paramètres

Modèle	Encodeur	Décodeur	Total
NN	23 783 712	23 795 032	47 578 744
CNN	525 024	527 005	1 052 029
LSTM	132 096	**133 128**	265 224
Perceiver IO	**46 004**	156 299	**202 303**

Le choix de Perceiver IO s'explique par son excellent compromis entre qualité de reconstruction et efficacité computationnelle. Il offre des performances proches du NN et du CNN, tout en étant bien plus léger et donc plus adapté aux applications où les ressources sont limitées. De plus, son architecture basée sur les mécanismes d'attention, lui permet de traiter des données de taille variable, ce qui en fait un modèle particulièrement adaptée à notre problématique.

3.2 Optimisation de la taille du modèle

Pour réaliser l'embarquement du modèle Perceiver IO, on présente ci-dessous les différentes optimisations appliquées à ce dernier afin de réduire la taille nécessaire ainsi que les opérations à effectuer. Ces modifications ont pour but de ne pas modifier le résultat obtenu par l'inférence.

On présente sur la Figure 24 l'architecture originel du bloc de traitement utilisé dans Perceiver IO.

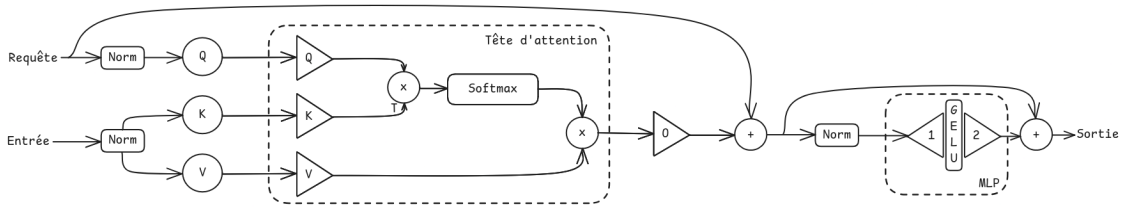


FIGURE 24 – Architecture du bloc de traitement du modèle Perceiver IO

3.2.1 Tête d'attention

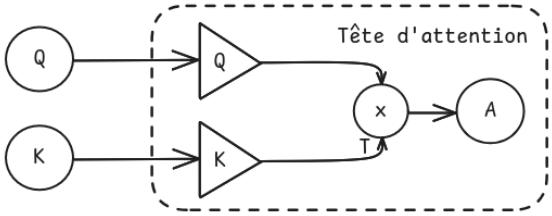


FIGURE 25 – Architecture du calcul des scores d'attention

On s'intéresse au calcul des scores d'attention au sein de la tête d'attention tel que présenté dans la Figure 25. Avec les notation de la Section 2.5.2, on peut écrire matriciellement ce calcul de la façon suivante :

$$A = \frac{QK^T}{\sqrt{F}}$$

Si l'on développe les matrices Q et V par rapport aux matrices de poids respectives W_Q et W_K et aux matrices d'entrées normalisées X_Q et X_K, on obtient :

$$\begin{aligned} A &= \frac{X_Q W_Q (X_K W_K)^T}{\sqrt{F}} \\ &= \frac{X_Q W_Q W_K^T X_K^T}{\sqrt{F}} \end{aligned}$$

Le produit $W_Q W_K^T$ peut être pré-calculé, ce qui donne :

$$A = \frac{X_Q W_{QK} X_K^T}{\sqrt{F}},\ W_{QK} = W_Q W_K^T$$

que l'on peut également écrire comme :

$$A = \frac{\left(X_K W_{QK}^T X_Q^T\right)^T}{\sqrt{F}}$$

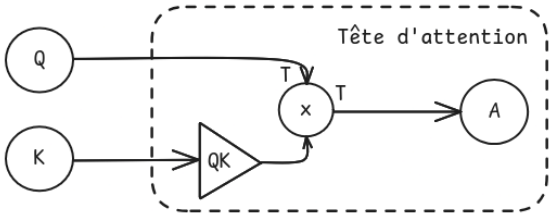


FIGURE 26 – Architecture optimisée du calcul des scores d'attention

L'architecture optimisée est présentée sur la Figure 26. Les matrices de poids W_Q et W_K sont fusionnées sur la dimension des caractéristiques internes F. La réduction de taille, i.e. le rapport de la taille optimisée par rapport à la taille originale, peut s'exprimer ainsi :

$$\begin{aligned} r &= \frac{D.C}{D.F + C.F} \\ &= \frac{1}{F}\frac{D.C}{D + C} \end{aligned}$$

$$X_Q \in \mathbb{R}^{N \times D},\ X_K \in \mathbb{R}^{M \times C},\ C \leq F \leq D$$

3.2.2 Optimisations mono-tête

Le bloc de traitement utilisé pour le modèle de compression ne comporte qu'une unique tête d'attention. Ce choix permet l'application de plusieurs optimisations supplémentaires permettant de réduire encore la taille du modèle final.

3.2.2.1 Normalisation

Avant l'entrée dans la tête d'attention, on dispose de couches de normalisation qui réalise la fonction vectorielle suivante :

$$\begin{aligned}\vec{y} &= \frac{\vec{x} - E(\vec{x})}{\sqrt{V(\vec{x}) + \epsilon}} \odot \vec{\gamma} + \vec{\beta} \\ &= \underline{\vec{x}} \odot \vec{\gamma} + \vec{\beta}, \underline{\vec{x}} = \frac{\vec{x} - E(\vec{x})}{\sqrt{V(\vec{x}) + \epsilon}}\end{aligned}$$

où $E(\cdot)$ représente l'espérance vectorielle (i.e. la moyenne des éléments du vecteur), $V(\cdot)$ la variance et ϵ un biais évitant une potentielle division par zéro. $\vec{\gamma}$ et $\vec{\beta}$ sont des vecteurs de poids de même longueur que $\vec{x}$ représentant une couche proche d'un linéaire avec biais, à la différence que le produit $\odot$ est une multiplication terme à terme. On notera $\underline{\vec{\cdot}}$ un vecteur centré-réduit.

Soit $\overrightarrow{x_q^i}$ la ligne i de la matrice de requête en entrée X_q et $\overrightarrow{x_Q^i}$ la ligne i de sa normalisation X_Q. Si l'on exprime $\overrightarrow{Q^i}$ la ligne i de la matrice Q par rapport à $\overrightarrow{x_q^i}$, on obtient :

$$\begin{aligned}\overrightarrow{Q^i} &= \overrightarrow{x_Q^i} W_Q \\ &= (\overrightarrow{\underline{x}_q^i} \odot \vec{\gamma} + \vec{\beta}) W_Q \\ &= (\overrightarrow{\underline{x}_q^i} \odot \vec{\gamma}) W_Q + \vec{\beta} W_Q\end{aligned}$$

Le terme $(\overrightarrow{\underline{x}_q^i} \odot \vec{\gamma}) W_Q$ peut s'exprimer ainsi :

$$\begin{aligned}(\overrightarrow{\underline{x}_q^i} \odot \vec{\gamma}) W_Q &= \left(\begin{pmatrix}\underline{x}_q^{i,1} & \underline{x}_q^{i,2} & \dots & \underline{x}_q^{i,m}\end{pmatrix} \odot \begin{pmatrix}\gamma^1 & \gamma^2 & \dots & \gamma^m\end{pmatrix}\right) W_Q \\ &= \begin{pmatrix}\underline{x}_q^{i,1}\gamma^1 & \underline{x}_q^{i,2}\gamma^2 & \dots & \underline{x}_q^{i,m}\gamma^m\end{pmatrix} \begin{pmatrix} W_Q^{1,1} & W_Q^{1,2} & \dots & W_Q^{1,m} \\ W_Q^{2,1} & W_Q^{2,2} & \dots & W_Q^{2,m} \\ \dots & \dots & \dots & \dots \\ W_Q^{n,1} & W_Q^{n,2} & \dots & W_Q^{n,m} \end{pmatrix} \\ &= \begin{pmatrix}\underline{x}_q^{i,1}\gamma^1 W_Q^{1,1} + \underline{x}_q^{i,2}\gamma^2 W_Q^{2,1} + \dots + \underline{x}_q^{i,m}\gamma^m W_Q^{m,1} & \dots\end{pmatrix} \\ &= \begin{pmatrix}\underline{x}_q^{i,1} & \underline{x}_q^{i,2} & \dots & \underline{x}_q^{i,m}\end{pmatrix} \begin{pmatrix} \gamma^1 W_Q^{1,1} & \gamma^1 W_Q^{1,2} & \dots & \gamma^1 W_Q^{1,m} \\ \gamma^2 W_Q^{2,1} & \gamma^2 W_Q^{2,2} & \dots & \gamma^2 W_Q^{2,m} \\ \dots & \dots & \dots & \dots \\ \gamma^m W_Q^{n,1} & \gamma^m W_Q^{n,2} & \dots & \gamma^m W_Q^{n,m} \end{pmatrix} \\ &= \overrightarrow{\underline{x}_q^i} W_{\gamma Q}, W_{\gamma Q} = \begin{pmatrix} \gamma^1 W_Q^{1,1} & \gamma^1 W_Q^{1,2} & \dots & \gamma^1 W_Q^{1,m} \\ \gamma^2 W_Q^{2,1} & \gamma^2 W_Q^{2,2} & \dots & \gamma^2 W_Q^{2,m} \\ \dots & \dots & \dots & \dots \\ \gamma^m W_Q^{n,1} & \gamma^m W_Q^{n,2} & \dots & \gamma^m W_Q^{n,m} \end{pmatrix}\end{aligned}$$

On a donc :

$$Q^i = \overrightarrow{x_q^i} W_{\gamma Q} + \overrightarrow{w_{\beta Q}},\ \overrightarrow{w_{\beta Q}} = \vec{\beta} W_Q$$

La matrice $W_{\gamma Q}$ est de même forme que W_Q et $\overrightarrow{w_{\beta Q}}$ est de longueur F, la dimension des caractéristiques internes. Cette optimisation a pour intérêt la suppression du vecteur $\vec{\gamma}$ et de la multiplication terme à terme, ce qui compense la potentielle plus grande taille de $\overrightarrow{w_{\beta Q}}$ par rapport à $\vec{\beta}$. Architecturalement, la couche de normalisation est remplacée par une couche de centrage-réduction et le linéaire qui suit est équipé d'un biais. Ce même procédé peut également s'appliquer aux entrées K et V.

3.2.2.2 Requête fixe

On reprend le calcul des scores d'attention tel que présenté sur la Figure **??**. Au sein de l'encodeur, la matrice de requête en entrée du modèle est une matrice latente vierge d'information ne contenant qu'un encodage positionnel. Cette matrice étant fixe, elle peut être fusionnée avec sa matrice de poids associée. On obtient alors :

$$\begin{aligned} A &= \frac{(X_Q W_{QK}) X_K^T}{\sqrt{F}} \\ &= \frac{(X_K (X_Q W_{QK})^T)^T}{\sqrt{F}} \\ &= \frac{(X_K W_{XQK})^T}{\sqrt{F}},\ W_{XQK} = (X_Q W_{QK})^T \end{aligned}$$

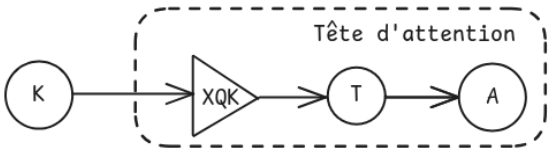


FIGURE 27 – Architecture optimisée du calcul des scores d'attention avec l'entrée du bloc de traitement

Le produit d'attention se présente alors comme une simple couche de réseaux de neurones linéaire, comme montré sur la Figure 27. La réduction de taille ainsi obtenue s'exprime alors :

$$\begin{aligned} r &= \frac{D.M}{M.C + C.D} \\ &= \frac{1}{C} \frac{D.M}{D + M} \end{aligned}$$

3.2.2.3 Matrice de sortie

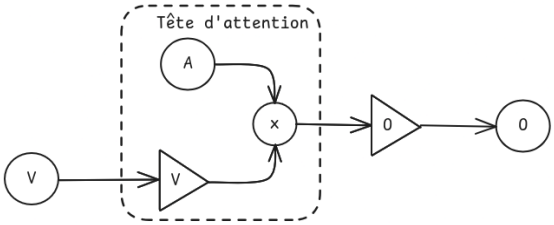


FIGURE 28 – Architecture de la sortie de la tête d'attention

La Figure 28 représente la sortie de la tête d'attention avec les scores d'attentions A tel que défini dans la section précédente. Du fait de l'unique tête d'attention, la dimension en entrée du linéaire de sortie de poids W_O est identique à celle en sortie de la tête, la concaténation habituellement présente à cet endroit étant caduque. On obtient alors :

$$\begin{aligned} O &= \text{softmax}(A)X_V W_V W_O \\ &= \text{softmax}(A)X_V W_{VO},\ W_{VO} = W_V W_O \end{aligned}$$

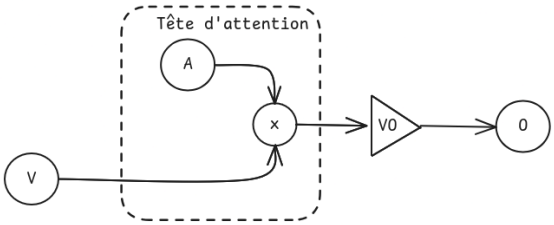


FIGURE 29 – Architecture optimisée de la sortie de la tête d'attention

Comme au sein de la tête d'attention, le produit de W_V avec W_O peut être pré-calculé (Figure 29), ce qui donne la réduction de taille suivante :

$$\begin{aligned} r &= \frac{D.C}{D.F + F.C} \\ &= \frac{1}{F}\frac{D.C}{D + C} \end{aligned}$$

3.2.3 Conclusion

Les optimisations présentées ci-dessus permettent de réduire le nombre de couches linéaires au sein du bloc d'attention. L'encodeur perd également un produit matriciel. L'architecture optimisée est présentée sur la Figure 30.

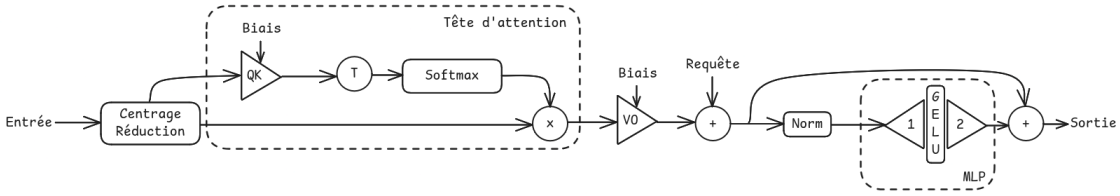


FIGURE 30 – Architecture optimisée du bloc de traitement Perceiver IO

3.3 Embarquement du modèle

3.3.1 Présentation de la cible embarquée

Le modèle doit être embarqué sur une cible ARM dont le microcontrôleur est fabriquée par STMicroelectronics : le STM32L451CE. Appartenant à la famille L4 (*Low Power*), c'est un microcontrôleur adapté pour les applications aux ressources énergétiques limitées, fournissant un cœur ARM Cortex-M4 à 80 MHz avec une unité de calcul flottant (FPU). Le dispositif embarqué est équipé de 512 Ko de Flash et de 160 Ko de mémoire vive (RAM).

Pour réaliser les tests d'embarquement, nous disposons d'une carte d'évaluation du même constructeur, mais possédant des périphériques plus performants (Figure 31). Le microcontrôleur utilisé est le STM32H755ZI-Q, de la famille H7 (*High Performance*), fournissant un cœur ARM Cortex-M7 à 480 MHz et un cœur ARM Cortex-M4 à 240 MHz avec 2 Mo de Flash et 1 Mo de RAM.

3.3.2 Implémentation du modèle en embarqué

L'implémentation originale du modèle prévoit l'intégration d'un algorithme d'apprentissage en ligne au sein de la cible embarquée afin d'adapter le modèle en temps réel aux trains utilisant le système d'aiguillage. Il est donc primordial de disposer d'une librairie permettant la réalisation de la rétropropagation au sein du microcontrôleur. Plusieurs alternatives ont été étudiées, dont notamment TFLite et PyTorch Mobile. Cependant, la grande majorité d'entre elles n'autorise que l'inférence. Il a donc été décidé d'utiliser une librairie ciblant le langage Rust : Burn.

Rust est un langage de programmation moderne axé sur la sécurité et la performance. Concurrent des langages comme C ou C++, Rust cherche à éliminer les erreurs de gestion mémoire grâce à son système de propriété et d'emprunt, qui assure une gestion automatique et sûre de la mémoire sans nécessiter de ramasse-miettes. Rust offre une exécution plus rapide et un contrôle précis des ressources tout en disposant d'une syntaxe moderne et expressive, ainsi qu'une grande partie des fonctionnalités de langages de plus haut niveau. Sa cible d'usage est centrée sur

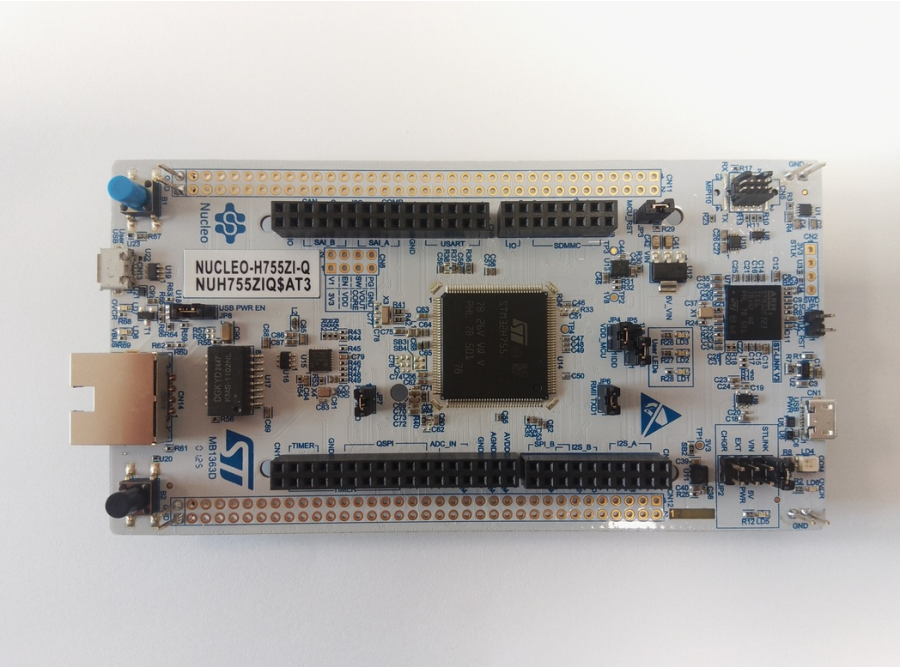


FIGURE 31 – STM32H755ZI-Q : carte d'évaluation utilisée pour les tests d'embarquement

les applications système, dont les logiciels embarqués. Le compilateur Rust supporte les cibles ARM via l'infrastructure LLVM (*Low Level Virtual Machine*).

Burn est une librairie d'apprentissage profond écrite nativement en Rust et supportant un grand nombre de platformes. Burn peut réaliser des entraînements et des inférences sur différentes cibles GPU et CPU via différentes librairies. On peut notamment citer l'interfaçage sur navigateur web via WebGL pour utiliser des modèles d'IA sans installation logicielle préalable. Cependant, Burn peut aussi être utilisé sur des cibles embarquées via la librairie NdArray, équivalent de NumPy sous Python.

Nous estimons que l'inférence du modèle nécessite environ 2 millions d'opérations en virgule flottante pour une entrée de taille (100, 129), correspondant à un signal temporel d'environ 12 000 échantillons. Cela suggère que la plupart des microcontrôleurs équipés d'une unité de calcul en virgule flottante (FPU) comme le STM32L451CE (fonctionnant à 80 MHz), pourraient effectuer l'inférence en moins d'une seconde. Cette estimation est basée sur le nombre de toutes les opérations mathématiques nécessaires pour les différentes couches de notre modèle. Cependant, la charge computationnelle réelle peut être plus élevée en fonction de la cible matérielle et de la mise en œuvre des fonctions mathématiques nécessaires, en particulier les fonctions exponentielles et tangentes hyperboliques. Pour la même taille

d'entrée, l'empreinte mémoire de l'inférence, qui est la somme des tailles de toutes les matrices intermédiaires, est inférieure à 16 Ko.

3.3.3 Résultats

Le modèle embarqué ne comporte que la partie encodeur réalisant la compression du signal temporel. Ce dernier comporte 47 746 paramètres pour une taille de 186,12 Ko (en *float32*). Avec les optimisations présentées dans la Section 3.2, ce modèle n'utilise plus que 5284 paramètres, soit 20,66 Ko. L'embarquement a été réalisé sur l'équipement présenté précédemment en appliquant les contraintes mémoires propres au dispositif final. Le logiciel embarqué comportant le code d'inférence ainsi que les poids du modèle utilise près de 90 % de la Flash. À l'inférence, la mémoire vive est utilisé pour le stockage du signal temporel et de son spectrogramme. Il se trouve également que le modèle est porté en RAM : en effet, Burn est conçu pour permettre l'entraînement des modèles sur l'ensemble des cibles supportées, même lorsque cela n'est pas demandé. Ainsi, les tenseurs contenant les poids du modèle ne peuvent exister qu'au sein de la mémoire vive. Cela n'est pas un problème dans le cas d'un apprentissage en ligne, mais peut augmenter la consommation mémoire si seule l'inférence est nécessaire. Une autre alternative qui a été étudiée dans le cadre où aucun apprentissage n'est réalisé sur le capteur se base sur la librairie standard CMSIS-NN. CMSIS décrit une interface standard d'accès aux périphériques pour les microcontrôleurs ARM. Ce standard permet un portage plus aisé des logiciels embarqués entre différentes platformes ARM et offre aussi un accès à des implémentations optimisées d'opérations usuelles. La librairie CMSIS-NN fournit les fonctions de base nécessaires à la définition de tous modèles d'apprentissage profond. Ces dernières supportent l'utilisation du jeu d'instruction DSP (*Digital Signal Processing*) qui offre des opérations SIMD (*Single Instruction Multiple Data*) permettant de paralléliser les traitements. Une implémentation en C utilisant ces instructions optimisées donnera de meilleures performances et une consommation mémoire plus faible dans le cas de l'inférence et pourrait être plus adaptée au dispositif final.

3.4 Conclusion

Les évaluations des modèles de reconstruction ont montré que le modèle utilisant les réseaux de neurones conventionnels offre les meilleures performances sur toutes les métriques, mais sa complexité le rend peu adapté aux environnements contraints. Le Perceiver IO, bien que légèrement moins performant, se distingue

par sa légèreté et sa capacité à traiter des données de taille variable grâce à son architecture attentionnelle et représente un excellent compromis entre qualité de reconstruction et efficacité computationnelle.

Dans le cadre de l'embarquement, la partie encodeur du modèle de compression a été optimisé pour réduire sa taille à près de 20 Ko, respectant les contraintes mémoires du dispositif. Les tests ont été réalisés sur une carte d'évaluation et ont prouvé la portabilité du modèle choisi. Toutefois, si seule l'inférence est nécessaire, d'autres alternatives plus efficaces existent, améliorant les performances et réduisant la consommation mémoire.

Chapitre 4

Seconde contribution : Compression et estimation de la dégradation

Ce chapitre expose la seconde contribution, qui consiste en deux approches complémentaires l'une de l'autre visant à l'estimation de l'état de santé d'un aiguillage. La première intègre une partie détection d'anomalie dans le modèle de compression présenté au chapitre précédent. La seconde approche vise à estimer numériquement la dégradation via un modèle dédié prenant en entrée les signaux préalablement compressés.

4.1 Introduction

L'analyse est un domaine crucial en ingénierie, largement utilisé pour surveiller la santé des systèmes mécaniques et détecter les anomalies qui pourraient indiquer des défaillances potentielles. En identifiant les premiers signes de défauts, des stratégies de maintenance prédictive peuvent être mises en œuvre pour améliorer la fiabilité, minimiser les temps d'arrêt et prévenir les réparations coûteuses. L'efficacité de la surveillance des motifs de vibration pour la maintenance prédictive a été bien documentée (Brandt 2023). La nature non intrusive de l'analyse des vibrations en fait une solution attrayante dans divers secteurs, y compris le génie maritime (Li et al. 2024), le pétrole (Chu et al. 2024), les chemins de fer (Z. Wang et al. 2020) et l'aérospatiale (Peixin et al. 2021).

Avec la complexité croissante des machines et l'augmentation des données générées par les capteurs, le traitement efficace et la représentation des signaux vibratoires sont devenus des défis clés. Les méthodes traditionnelles d'analyse des vibrations peinent à gérer les données bruyantes et de haute dimension, nécessitant des modèles capables de capturer efficacement les caractéristiques essentielles tout en réduisant la redondance (Jang et al. 2023). Les techniques de compression non seulement atténuent les contraintes de stockage et de transmission, mais améliorent

également les tâches en aval telles que la classification des défauts et la détection des anomalies (Choi et al. 2024). Ces avancées sont particulièrement précieuses pour les applications industrielles en temps réel, où l'efficacité des données est cruciale.

Pour relever ces défis, les techniques avancées d'apprentissage automatique sont devenues des outils puissants pour l'analyse des vibrations, extrayant des motifs significatifs à partir de grandes quantités de données de capteurs pour une compression efficace et une détection des défauts. Une représentation compacte mais informative des signaux réduit les goulots d'étranglement des données tout en préservant les informations diagnostiques. De plus, l'apprentissage adaptatif aide à distinguer les variations normales des véritables anomalies, améliorant la détection des défauts dans des environnements industriels complexes. Les techniques d'apprentissage automatique et d'apprentissage profond offrent plusieurs avantages dans l'analyse des vibrations : la capacité à traiter de grands volumes de données efficacement, à identifier des motifs complexes et à s'adapter à des conditions évolutives (AlShorman et al. 2020 ; Liu et al. 2018). Au fil des ans, de nombreux modèles ont été proposés pour la détection et le diagnostic des défauts (Zhu et al. 2023 ; C. Wang et al. 2024). Des approches telles que le clustering K-means (Yiakopoulos et al. 2011), les réseaux de neurones convolutifs (CNN) (W. Zhang et al. 2017) et les réseaux Long Short-Term Memory (B. Zhang et al. 2019) ont démontré une grande précision dans la détection des défauts. Plus récemment, les auto-encodeurs profonds combinés à l'apprentissage d'ensemble (Y. Zhang et al. 2020) ont montré une robustesse dans des conditions bruyantes.

À notre connaissance, la littérature existante traite la compression et la détection d'anomalies comme des tâches distinctes, ce qui peut entraîner des calculs redondants et une augmentation de la consommation de ressources, particulièrement problématique dans les applications industrielles en temps réel. De plus, les contraintes des environnements embarqués, y compris la mémoire et la puissance limitées, rendent difficile le déploiement efficace de modèles séparés. Pour relever ces défis, nous proposons une architecture unifiée basée sur Perceiver IO qui intègre la compression des signaux de vibration et la détection d'anomalies dans un seul cadre. Nos contributions clés sont triples :

1. nous développons un plongement dans l'espace latent qui extrait des représentations compactes mais informatives des données de vibration, atteignant une haute efficacité de compression sans compromettre les caractéristiques diagnostiques critiques ;
2. nous concevons un mécanisme de détection d'anomalies adaptatif qui se concentre sélectivement sur les motifs pertinents, filtrant le bruit et

améliorant la fiabilité du diagnostic de défaut dans les environnements industriels ;

3. nous proposons un cadre unifié et léger qui combine efficacement la compression et la détection d'anomalies, le rendant idéal pour les systèmes embarqués à ressources limitées. Cette intégration garantit que notre approche est bien adaptée aux applications industrielles à grande échelle où le traitement en temps réel et la précision sont primordiaux.

Ce chapitre est organisé comme suit : la Section 4.1.1 passe en revue les travaux connexes sur l'analyse des vibrations, la détection des défauts et la compression des signaux ; la Section 4.2 présente l'architecture du modèle Perceiver IO, les étapes de prétraitement des données, les mécanismes d'attention pour la compression et la détection d'anomalies, et la mise en œuvre de l'apprentissage contrastif pour améliorer la séparabilité des caractéristiques ; la Section 4.3 décrit le contexte de la recherche, le jeu de données et la formation spécifique de notre modèle ; les résultats sont discutés dans la Section 4.4 ; et le chapitre conclut avec la Section 4.6, qui couvre les conclusions et les perspectives d'avenir.

4.1.1 Travaux associés

Les techniques d'apprentissage profond ont démontré un potentiel considérable dans l'analyse des signaux de vibration, en particulier pour la détection de défauts dans les systèmes mécaniques. Ces dernières années, plusieurs études ont exploré l'utilisation de modèles d'apprentissage profond pour prédire l'état de santé des systèmes à partir des signaux de vibration. Par exemple, (Lin et al. 2019) a évalué divers réseaux de neurones d'apprentissage profond pour la prédiction de la rugosité de surface, en utilisant l'analyse des signaux de vibration. Leurs résultats ont montré que les modèles d'apprentissage profond, en particulier les réseaux de neurones convolutifs (CNN), peuvent extraire avec succès des caractéristiques des signaux de vibration et fournir des prédictions précises. De même, (Chen et Lee 2021) ont étudié comment les CNN peuvent améliorer l'analyse des signaux de vibration pour des applications telles que le diagnostic de défauts et l'estimation de la rugosité de surface. Ils utilisent à la fois des modèles CNN 1D et 2D pour traiter les données brutes de vibration et les représentations temps-fréquence, améliorant ainsi la précision prédictive. Leur approche intègre des techniques d'optimisation pour affiner les performances des CNN, démontrant son efficacité dans la détection des défauts et l'évaluation des conditions d'usinage.

Pour le diagnostic de défauts de roulements, (X. Wang et al. 2021) ont fusionné des données vibro-acoustiques avec un réseau 1D-CNN, montrant des améliorations

significatives de la précision diagnostique par rapport aux méthodes traditionnelles. Une autre étude (Dolenc et al. 2016) a proposé une approche distribuée pour le diagnostic de défauts de roulements basée sur les signaux de vibration. Cette méthode a utilisé plusieurs capteurs pour collecter des données de vibration, améliorant le processus de diagnostic de défauts en réduisant l'impact du placement des capteurs et du bruit environnemental.

Plus récemment, l'utilisation de l'intelligence artificielle explicable (XAI) a gagné en traction dans l'analyse des signaux de vibration pour le diagnostic de défauts. (Chen et Lee 2020) a exploré l'utilisation de la XAI pour le diagnostic de défauts de roulements, où ils ont employé des modèles d'apprentissage automatique avec des caractéristiques d'interprétabilité. Cette approche permet une compréhension plus approfondie du processus de prise de décision, ce qui est particulièrement bénéfique dans des applications critiques comme la maintenance industrielle.

La revue des méthodes existantes démontre que, bien que des avancées significatives aient été réalisées dans la compression des signaux de vibration et le diagnostic de défauts, de nombreux défis subsistent. Les approches traditionnelles traitent souvent la compression et la détection d'anomalies séparément, ce qui conduit à des inefficacités et à une augmentation des demandes computationnelles, en particulier dans les applications industrielles en temps réel. De plus, de nombreuses techniques existantes peinent à équilibrer l'efficacité de la compression avec la préservation des caractéristiques critiques de détection de défauts, surtout lorsqu'elles traitent des données bruyantes ou de haute dimension. Ces défis sont encore exacerbés dans les environnements embarqués, où des contraintes strictes de mémoire et d'énergie limitent la faisabilité des modèles complexes. Pour répondre à ces limitations, ce travail propose un modèle unifié basé sur Perceiver IO qui intègre à la fois une compression efficace et une détection d'anomalies tout en restant léger et computationnellement efficace pour les dispositifs embarqués à ressources limitées.

4.1.2 Techniques Conventionnelles de Détection d'Anomalie

La détection d'anomalies représente un enjeu crucial dans de nombreux secteurs industriels, de la cybersécurité à la maintenance prédictive, en passant par la surveillance de systèmes complexes. Identifier rapidement et avec précision des comportements ou des événements inhabituels permet non seulement d'éviter des défaillances coûteuses, mais aussi d'assurer la sécurité et la fiabilité des opérations. Historiquement, les techniques conventionnelles de détection d'anomalie ont constitué la pierre angulaire de ces processus, en s'appuyant sur des méthodes statistiques, des modèles mathématiques ou des approches basées sur l'analyse de données. Leur

simplicité, leur robustesse, et leur interprétabilité en font encore aujourd'hui des outils indispensables, notamment dans le contexte de la surveillance des infrastructures ferroviaires, où la détection précoce de défauts peut éviter des accidents majeurs.

Les méthodes statistiques constituent la première catégorie de techniques de détection d'anomalie. Elles reposent sur l'hypothèse que les données normales suivent une certaine distribution probabiliste, souvent gaussienne, ce qui permet d'établir des seuils ou des tests pour identifier les valeurs atypiques. Le test de Grubbs, par exemple, est un test statistique classique permettant d'identifier des valeurs extrêmes dans un échantillon en supposant une distribution normale. Il est particulièrement adapté pour détecter des valeurs aberrantes dans des séries temporelles issues de capteurs, comme ceux déployés dans la surveillance ferroviaire. De même, le Z-Score mesure la déviation d'une donnée par rapport à la moyenne, en unités d'écart-type. Lorsqu'une valeur présente un Z-Score élevé, cela indique qu'elle s'écarte fortement de la norme attendue, constituant potentiellement une anomalie. Ces techniques sont rapides, simples à mettre en œuvre, et très efficaces pour des distributions de données bien modélisées, mais leur performance peut diminuer face à des données non stationnaires ou présentant des distributions complexes.

Les méthodes basées sur des modèles ont permis d'étendre la capacité de détection d'anomalies en utilisant des représentations plus sophistiquées. La régression linéaire, par exemple, modélise la relation entre plusieurs variables d'entrée et une variable cible, permettant de repérer des écarts importants entre la valeur observée et la valeur prédite comme des indicateurs d'anomalies. Dans le contexte ferroviaire, cela peut correspondre à la prédiction de vibrations ou de températures à partir de paramètres opératoires, avec des écarts importants signalant un problème potentiel. La technique ARIMA (AutoRegressive Integrated Moving Average) constitue une autre approche, particulièrement adaptée aux séries temporelles. En modélisant la dynamique temporelle des signaux vibratoires ou des paramètres de fonctionnement, ARIMA permet de prévoir leur évolution et d'identifier les écarts par rapport aux prévisions comme des anomalies. Ces méthodes sont efficaces pour suivre des processus stochastiques et détecter des déviations dans des données structurées dans le temps.

L'émergence de l'apprentissage automatique a permis d'introduire des techniques plus avancées, capables de capturer des motifs complexes et non linéaires dans les données. Parmi celles-ci, l'Isolation Forest se distingue par sa simplicité et sa performance dans la détection d'anomalies dans de grandes bases de données. Son principe repose sur l'isolation aléatoire des observations : les anomalies, étant rares ou différentes, sont généralement plus faciles à isoler en peu de partitions. Utilisée dans

la détection d'anomalies à grande échelle, elle est particulièrement adaptée pour la surveillance continue de multiples capteurs ou infrastructures. Les autoencoders, quant à eux, sont des réseaux de neurones conçus pour apprendre une représentation compacte des données en tentant de les reconstruire. Lorsqu'un échantillon est anormal, sa reconstruction sera généralement moins fidèle, ce qui permet de le repérer en mesurant l'erreur de reconstruction. Ces modèles sont utilisés avec succès pour la détection d'anomalies dans des images, des séries temporelles ou des signaux vibratoires, par exemple dans la maintenance prédictive des équipements ferroviaires.

L'application concrète de ces techniques dans la surveillance des infrastructures ferroviaires illustre leur importance stratégique. La surveillance vibratoire constitue une approche privilégiée pour diagnostiquer l'état des composants mécaniques tels que rails, roues, roulements et moteurs. En mesurant en continu les vibrations à l'aide de capteurs installés le long des voies ou sur des composants critiques, il devient possible d'identifier en temps réel des anomalies, comme des fissures, des désalignements ou des usures prématurées. La transformée de Fourier (FFT) permet, par exemple, de décomposer les signaux vibratoires en composantes fréquentielles, facilitant la détection de fréquences anormales associées à des défauts spécifiques. L'analyse des enveloppes, une autre technique avancée, extrait les composantes haute fréquence du signal vibratoire, révélant des défauts de contact ou des fissures incipientes dans les roulements ou les rails. Par ailleurs, l'intégration de modèles prédictifs, notamment ceux issus de l'apprentissage automatique comme les réseaux de neurones récurrents (RNN), permet de modéliser le comportement normal des systèmes et de détecter précocement des écarts indicatifs de défaillances potentielles.

L'utilisation de ces techniques dans des applications concrètes a permis d'améliorer la sécurité et la fiabilité des réseaux ferroviaires. La surveillance en temps réel des rails, par exemple, permet de détecter des fissures ou des affaissements avant qu'ils ne deviennent critiques, évitant ainsi des incidents majeurs. La maintenance prédictive des roulements, en détectant précocement des signes d'usure ou de dommage, réduit les coûts de réparation et limite les interruptions de service. De même, le contrôle des moteurs électriques via l'analyse vibratoire contribue à prévenir les pannes coûteuses en alertant sur des déséquilibres ou des désalignements. Ces techniques, souvent combinées avec des méthodes d'analyse avancée, constituent une véritable révolution dans la gestion proactive des infrastructures ferroviaires, permettant une intervention ciblée et planifiée plutôt qu'une maintenance réactive.

En conclusion, les techniques conventionnelles de détection d'anomalie, qu'elles soient statistiques, modélisées ou basées sur l'apprentissage automatique, jouent un

rôle essentiel dans la surveillance et la maintenance des systèmes complexes. Leur simplicité, leur efficacité et leur capacité à fournir des diagnostics précoces en font des outils incontournables pour assurer la sécurité, la performance et la pérennité des infrastructures. Dans un contexte où les enjeux de sécurité et de disponibilité sont cruciaux, l'intégration de ces méthodes dans des systèmes de surveillance intelligents constitue une étape incontournable vers des infrastructures plus résilientes. À l'avenir, le développement de techniques hybrides combinant méthodes classiques et apprentissage automatique, ainsi que l'intégration de capteurs plus précis et plus nombreux, promettent d'améliorer encore la détection d'anomalies, ouvrant la voie à une gestion proactive et intelligente des infrastructures critiques.

4.1.3 Détection d'Anomalie par Auto-Encodage

La détection d'anomalies constitue une étape essentielle dans la surveillance et la maintenance de systèmes variés, allant des infrastructures industrielles à la cybersécurité. Parmi les méthodes modernes, les auto-encodeurs (autoencoders) se sont révélés particulièrement efficaces pour identifier des comportements inhabituels en exploitant leur capacité à apprendre une représentation compacte des données normales. Leur principe repose sur l'apprentissage de la reconstruction des données d'entrée : en mesurant l'écart entre l'entrée originale et sa reconstruction, il devient possible de détecter des anomalies comme des écarts significatifs. Cette approche est particulièrement avantageuse car elle ne nécessite pas forcément d'étiquettes pour les anomalies, ce qui la rend adaptée à des contextes où celles-ci sont rares ou difficiles à définir.

Les auto-encodeurs classiques sont la première étape de cette famille. Leur architecture se compose d'un encodeur, qui réduit la dimensionnalité des données en produisant une représentation latente, et d'un décodeur, qui tente de reconstruire les données d'origine à partir de cette représentation. L'entraînement consiste à minimiser une fonction de perte, généralement l'erreur quadratique moyenne (MSE), afin que le modèle apprenne à représenter efficacement les données normales. Lors de l'utilisation, si l'auto-encodeur rencontre un échantillon présentant une erreur de reconstruction élevée, cela indique généralement qu'il s'agit d'une anomalie, car le modèle n'a pas appris à représenter ces comportements inhabituels. Cette technique est largement employée pour la détection de fraude, la surveillance de qualité ou la détection d'intrusions réseau.

Les Variational Autoencoders (VAE) représentent une extension probabiliste des auto-encodeurs classiques, introduisant une contrainte sur la distribution des représentations latentes. Plutôt que d'apprendre une simple représentation, le VAE

impose que cette dernière suive une distribution gaussienne standard, en utilisant une divergence de Kullback-Leibler (KL). Lors de l'inférence, un échantillon avec une faible probabilité sous le modèle indique une anomalie potentielle. Les VAE sont particulièrement adaptés pour la génération d'images, la détection d'anomalies dans le domaine médical ou l'analyse de séries temporelles, où la modélisation probabiliste apporte une meilleure capacité à distinguer les comportements inhabituels.

Les auto-encodeurs débruiteurs (Denoising Autoencoders) ont été conçus pour améliorer la robustesse face aux données bruitées. Pendant l'entraînement, un bruit aléatoire est ajouté aux données d'entrée, et le modèle apprend à reconstruire les versions propres de ces données. Lorsqu'un échantillon présente un bruit ou une déviation importante, l'auto-encodeur aura du mal à produire une reconstruction fidèle, ce qui permet d'identifier ces anomalies. Cette technique est très utilisée dans la détection d'anomalies dans les images, les signaux audio ou encore les données issues de capteurs, où le bruit peut masquer ou simuler des défauts réels.

Les auto-encodeurs robustes intègrent des mécanismes pour limiter leur sensibilité aux anomalies présentes dans les données d'entraînement. En utilisant des fonctions de perte robustes ou en pondérant différemment les échantillons selon leur erreur, ces modèles réduisent l'impact des anomalies lors de l'apprentissage, évitant ainsi un apprentissage biaisé. Ils sont particulièrement utiles dans les domaines financiers, les réseaux sociaux ou la surveillance, où les données peuvent contenir des valeurs extrêmes ou bruitées.

Une autre utilisation innovante consiste à combiner les auto-encodeurs avec des techniques de clustering. En apprenant une représentation latente des données, ils facilitent la segmentation en groupes ou clusters via des algorithmes comme K-means. Les échantillons qui ne s'intègrent pas bien dans ces clusters, c'est-à-dire qui sont éloignés des centres, peuvent être considérés comme des anomalies. Cette approche est utile pour la segmentation de clients, la détection de comportements suspects ou la détection de fraudes.

Dans le contexte de la détection d'anomalies, ces méthodes trouvent de nombreuses applications concrètes. En finance, les auto-encodeurs détectent des transactions frauduleuses en identifiant celles qui s'écartent de la norme. Dans la surveillance d'infrastructures, comme les réseaux électriques ou les systèmes industriels, ils permettent de repérer précocement des défaillances via l'analyse de signaux vibratoires ou électriques. Les auto-encodeurs débruiteurs, en particulier, sont efficaces pour analyser des données de capteurs bruitées, tandis que les modèles robustes améliorent la fiabilité de la détection en présence de données bruitées ou contaminées. Enfin, l'analyse de séries temporelles et la détection d'intrusions réseau

bénéficient aussi largement de ces techniques, en permettant une identification précise des comportements anormaux ou suspectés.

En résumé, les auto-encodeurs offrent une solution flexible, efficace et adaptable pour la détection d'anomalies dans divers domaines. Leur capacité à apprendre des représentations compactes et à différencier facilement les comportements normaux des anomalies en fait des outils de choix pour la surveillance proactive, la maintenance prédictive, ou encore la sécurité informatique. À mesure que la recherche progresse, de nouvelles variantes et applications émergeront probablement, renforçant encore leur rôle dans la détection automatique d'événements inhabituels.

4.1.4 Compression et détection d'anomalie

Dans un monde en constante évolution où la collecte et l'analyse de données jouent un rôle central dans de nombreux domaines, la gestion efficace des flux d'informations est devenue un enjeu crucial, notamment dans le contexte des systèmes embarqués. Ces dispositifs, souvent intégrés à des objets connectés, des capteurs industriels ou des appareils portables, doivent fonctionner sous des contraintes strictes en termes de puissance de calcul, de mémoire, de consommation énergétique et de bande passante. Ces limites rendent difficile le traitement direct de volumes importants de données, ce qui impose le développement d'outils capables de réduire la volumétrie tout en conservant l'information pertinente, tout en permettant une détection fiable d'éventuelles anomalies.

Une approche particulièrement innovante consiste à utiliser des outils capables de réaliser simultanément deux fonctions essentielles : la compression des données pour en diminuer le volume, et la détection en temps réel de comportements déviants ou inhabituels pouvant indiquer un dysfonctionnement, une intrusion ou un évènement critique. Parmi ces outils, les autoencodeurs, une classe de réseaux de neurones non supervisés, ont émergé comme une solution particulièrement adaptée. Leur capacité à apprendre une représentation compacte et efficiente des données, tout en étant capables d'identifier rapidement toute divergence par rapport à cette représentation, en fait des candidats idéaux pour répondre aux exigences des environnements embarqués où la simplicité, la rapidité et la faible consommation sont des impératifs.

Les autoencodeurs s'appuient sur une architecture composée principalement d'un encodeur, d'un goulot d'étranglement et d'un décodeur. L'encodeur extrait une représentation de dimension réduite, ou vecteur latent, à partir de l'entrée initiale. Ce vecteur, ou code, doit contenir l'essentiel de l'information pertinente pour la reconstruction. Le décodeur, quant à lui, tente de reconstruire le signal d'origine à partir

de cette représentation comprimée. L'apprentissage du réseau consiste à minimiser la différence entre l'entrée et sa reconstruction, généralement via une fonction de perte comme l'erreur quadratique moyenne. Une fois cette étape terminée, l'autoencodeur peut être utilisé dans deux modes : pour compresser efficacement les données en passant par la vecteur latent, ou pour détecter des anomalies en évaluant la qualité de la reconstruction d'une nouvelle donnée. Si l'erreur de reconstruction dépasse un seuil prédéfini, cela indique généralement que la donnée est inhabituelle ou déviante par rapport à ce que le réseau a appris comme étant « normal ».

La capacité des autoencodeurs à réduire la dimensionnalité de données complexes, tout en conservant leur structure essentielle, leur confère une aptitude naturelle à la compression. La représentation latente, en capturant l'information pertinente, permet de diminuer considérablement la taille des données à stocker ou à transmettre, tout en évitant la perte d'informations cruciales. Cela se traduit concrètement par une utilisation plus efficiente de la bande passante, une réduction de la consommation d'énergie lors de la transmission ou du traitement, et un gain en rapidité pour les opérations d'analyse. La reconstruction étant précise pour les données normales, mais médiocre pour celles qui sont déviantes, le mécanisme de détection d'anomalies repose sur la mesure de cette erreur. En effet, si une nouvelle donnée, non vue lors de l'entraînement, présente une erreur de reconstruction élevée, cela indique qu'elle s'écarte du profil appris comme étant « normal ». Ce processus, non supervisé, ne requiert pas d'étiquetage préalable, ce qui constitue un avantage considérable dans de nombreux scénarios où les anomalies sont rares ou difficiles à définir à l'avance.

L'intérêt majeur des autoencodeurs dans les environnements embarqués réside dans leur capacité à répondre simultanément à deux besoins critiques : la compression des données pour réduire la charge sur le réseau ou le stockage, et la détection en temps réel de comportements anormaux pour assurer la sécurité, la maintenance prédictive ou la qualité du fonctionnement. Leur déploiement sur des dispositifs à ressources limitées est rendu possible par la conception de modèles légers, peu profonds, ou par l'utilisation de techniques de compression de modèles. La compatibilité avec des frameworks d'optimisation pour microcontrôleurs ou FPGA permet d'intégrer ces réseaux directement dans des systèmes embarqués, évitant ainsi la nécessité d'infrastructures coûteuses ou volumineuses.

Cette capacité d'adaptation locale permet également de mettre à jour en continu les modèles, en intégrant de nouvelles données provenant de l'environnement spécifique du système. Cela favorise une meilleure précision et une fiabilité accrue, même dans des contextes où les conditions opérationnelles évoluent rapidement. Des exemples concrets illustrent cette efficacité : la surveillance de réseaux de capteurs environnementaux pour détecter des anomalies dans la qualité de l'air ou de

l'eau, le suivi de la santé de machines industrielles pour anticiper des défaillances, le contrôle qualité dans la fabrication où la détection de pièces défectueuses doit être rapide et précise, ou encore la surveillance des batteries dans des véhicules électriques ou des bâtiments intelligents.

Cependant, malgré leurs nombreux avantages, les autoencodeurs présentent aussi des limites. Leur performance dépend fortement de l'adéquation du corpus d'entraînement, qui doit représenter fidèlement l'état « normal » de l'environnement. Dans un contexte où cet environnement évolue ou où la collecte de données représentatives est difficile, la fiabilité peut diminuer. La conception de modèles légers, tout en conservant la capacité de détection, constitue un défi technique. La détermination d'un seuil précis pour la détection d'anomalies, adaptée aux environnements bruités ou très variables, n'est pas triviale et nécessite souvent une calibration fine. En outre, le déploiement en temps réel implique une optimisation du code, une gestion efficace de la consommation d'énergie, et parfois un compromis entre performance et ressources disponibles.

Les progrès récents en deep learning, notamment avec l'utilisation de réseaux autoencoder convolutionnels ou récurrents, ont permis d'améliorer la robustesse et la capacité d'adaptation de ces outils. La mise en œuvre de techniques d'apprentissage en ligne, de compression de modèles ou encore d'optimisation pour microcontrôleurs ouvre la voie à une intégration plus large dans des environnements ultra-contraints. Par ailleurs, des approches hybrides combinant autoencodeurs avec d'autres méthodes telles que l'apprentissage semi-supervisé ou l'intégration de règles métier sont en cours de développement pour renforcer la fiabilité et la précision des détections tout en maintenant une compression efficace.

En somme, ces outils combinant la capacité à réduire la volumétrie des données et à détecter en temps réel des comportements atypiques offrent une solution innovante pour la gestion intelligente des flux d'informations dans les systèmes embarqués. Leur adoption croissante, soutenue par les avancées technologiques en deep learning et en architecture matérielle, laisse entrevoir un avenir où ces réseaux seront intégrés de manière ubiquitaire dans une multitude d'applications industrielles, médicales, environnementales ou de domotique. La poursuite de la recherche doit viser à optimiser ces outils pour qu'ils répondent toujours mieux aux exigences croissantes en termes de performance, de fiabilité, de faible consommation d'énergie et de simplicité d'intégration, contribuant ainsi à la concrétisation d'objets et de systèmes plus intelligents, autonomes et économes en ressources.

4.2 Méthodologie proposée

La surveillance des signaux de vibration est essentielle pour identifier les défauts dans les systèmes mécaniques, mais la grande dimensionnalité et le bruit inhérent aux données brutes présentent souvent des défis importants. Les modèles traditionnels ont souvent du mal à traiter efficacement de tels signaux sans perdre d'informations critiques liées aux défauts. Pour pallier ces limitations, nous proposons un modèle basé sur Perceiver IO (Jaegle, Borgeaud, et al. 2021), reconnu pour sa capacité à gérer des données variées et de haute dimension.

En exploitant les forces de Perceiver IO, notre modèle utilise une représentation dans l'espace latent afin de réduire considérablement les exigences computationnelles et l'empreinte mémoire, tout en conservant les structures essentielles du signal. Son mécanisme d'attention, agnostique au domaine, permet une gestion flexible de formats d'entrée et de sortie arbitraires, assurant une adaptabilité à divers scénarios de détection de défauts par vibration. Cette conception est particulièrement efficace pour isoler les caractéristiques importantes du bruit, améliorant ainsi la précision et la fiabilité de la détection.

Une amélioration clé de ce modèle consiste à intégrer, dans un cadre unifié, la compression et la détection d'anomalies. En utilisant des mécanismes basés sur l'attention, le modèle identifie dynamiquement les motifs indicatifs de défauts et élimine les interférences de fond. De plus, nous incorporons l'apprentissage contrastif pour affiner la détection d'anomalies, supprimant ainsi la nécessité d'un modèle séparé de prise de décision. Ces innovations simplifient l'architecture, tout en augmentant l'efficacité et la sensibilité aux défauts.

Les sections suivantes détaillent l'application de ce modèle amélioré à la compression des signaux vibratoires et à la détection d'anomalies basée sur l'apprentissage contrastif, après un prétraitement initial des signaux dans le domaine temporel. Cette approche unifiée atteint des performances satisfaisantes en termes d'efficacité computationnelle et de précision dans la détection de défauts, se positionnant comme une solution avancée pour la surveillance de l'état industriel.

4.2.1 Architecture du Perceiver IO proposé

Comme illustré dans la Figure 32, l'architecture globale comporte trois composants principaux : la compression des signaux vibratoires, la détection d'anomalies via l'attention, et la reconstruction des signaux. Initialement, les signaux de vibration subissent un prétraitement par la STFT, comme indiqué précédemment. Ensuite, ces représentations en spectrogramme sont projetées dans un espace latent via deux étapes clés : (1) un module d'attention croisée qui échange entre les

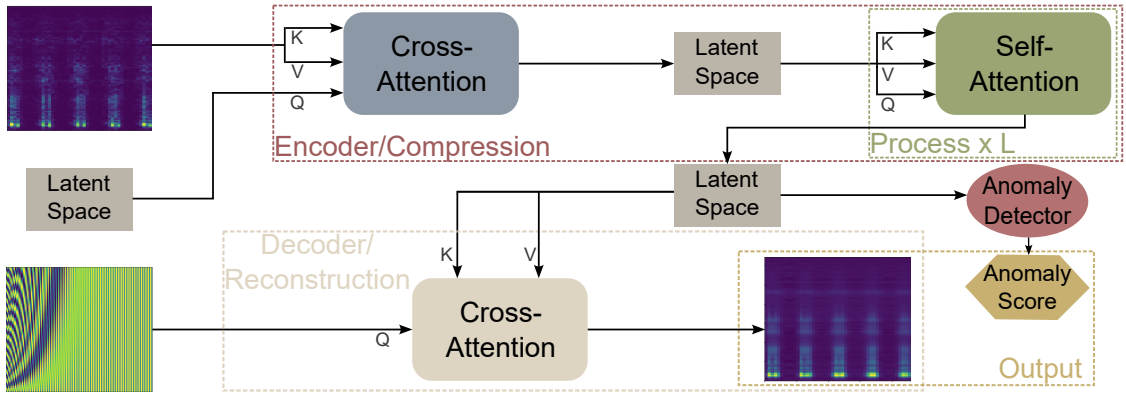


FIGURE 32 – Architecture Perceiver IO pour la compression et la détection d'anomalies.

signaux d'entrée et l'espace latent, et (2) une série d'auto-attentions au sein de cet espace. Cette étape capture des relations complexes tout en réduisant la dimensionnalité, tout en conservant les caractéristiques critiques. Ce module joue le rôle d'encodeur dans le modèle Perceiver IO.

Une fois la représentation compressée, elle est dirigée vers deux modules distincts. Le premier, dédié à la détection d'anomalies, utilise une simple fonction linéaire basée sur l'Analyse en Composantes Principales (PCA). Il identifie les écarts par rapport aux motifs normaux en analysant la représentation latente. Le second module, le décodeur, reconstruit la représentation en spectrogramme du signal à partir de la représentation compressée, en utilisant un mécanisme d'attention croisée avec un encodage positionnel.

Pour chaque module d'attention, un Perceptron Multicouche (MLP) et des couches de normalisation sont intégrés. Le MLP comprend deux couches entièrement connectées avec une activation GELU, permettant d'étendre la dimension à H, puis de la ramener à la dimension initiale. Les couches de normalisation centrent et réduisent les données via des moyennes et écarts-types appris, stabilisant l'entraînement. La sortie finale fournit à la fois le signal reconstruit et un score d'anomalie, facilitant la détection en temps réel. Cette architecture équilibre efficacement compression et détection précise.

La structure modulaire — encodeur, décodeur et détecteur d'anomalies — est adaptée aux environnements embarqués. Seuls l'encodeur et le détecteur sont déployés sur le dispositif, permettant un traitement local à faible charge, tandis que le décodeur, plus coûteux, peut fonctionner en aval pour la reconstruction si nécessaire. Cela assure une détection à faible latence tout en maintenant la capacité de vérification ou d'analyse ultérieure.

4.2.2 Apprentissage contrastif

L'apprentissage contrastif est une technique d'apprentissage automatique axée sur l'apprentissage de représentations en comparant des échantillons de données, afin de différencier des paires similaires et dissemblables. Le principe consiste à rapprocher les représentations de paires positives (échantillons similaires) et à éloigner celles de paires négatives (échantillons dissemblables). La perte par triplet, ou Large Margin Nearest Neighbor (LMNN) (Weinberger et Saul 2009), est une fonction de perte essentielle dans ce contexte. Elle vise à organiser les points dans un espace d'intégration en assurant que les points similaires soient proches, tandis que les dissemblables restent séparés par une marge définie.

Dans notre approche, la perte contrastive évalue la similarité entre deux matrices latentes, en utilisant uniquement une partie sélectionnée (certaines colonnes) de chaque matrice pour le calcul. Cela repose sur l'observation que les caractéristiques clés pour la détection d'anomalies diffèrent souvent de celles nécessaires à la reconstruction du spectrogramme. En séparant ces caractéristiques, le modèle minimise l'impact négatif de la perte contrastive sur le processus de reconstruction, assurant que celles essentielles à la reconstruction restent inchangées. Ainsi, la performance globale et la précision du modèle sont améliorées.

Lorsque deux vecteurs proviennent du même cluster, la perte cherche à minimiser leur distance, favorisant la similarité interne. Inversement, pour des vecteurs issus de clusters différents, la perte impose une contrainte de marge, garantissant une séparation suffisante dans l'espace latent. Cette séparation est essentielle pour une détection précise des anomalies sans recourir à un classificateur dédié. Notre implémentation modifie le cadre traditionnel LMNN ou perte par triplet (Weinberger et Saul 2009) : la marge, généralement calculée dynamiquement dans la formulation originale, est fixée à une valeur constante. Cela évite que la perte ne devienne trop influente, permettant au modèle d'apprendre des distinctions significatives entre clusters tout en maintenant une capacité de reconstruction précise. La description détaillée de cette approche est fournie dans la Section 4.3.3. Cette méthode sur mesure améliore la robustesse et l'efficacité globale dans la détection d'anomalies.

4.3 Configuration Expérimentale

4.3.1 Étude de Cas sur la Surveillance Ferroviaire

Cette recherche a été développée dans le contexte de la surveillance des infrastructures ferroviaires, en abordant spécifiquement les défis de la surveillance des

systèmes d'aiguillages et de croisements (Kassa et al. 2021 ; Hamadache et al. 2019). Dans notre contexte industriel, le système de surveillance ferroviaire met en œuvre une architecture hiérarchique de flux de données conçue pour optimiser la surveillance des infrastructures ferroviaires tout en répondant aux contraintes de déploiement à distance et de prise de décision en temps réel. Les signaux de vibration sont d'abord capturés par des capteurs IoT autonomes (situés au niveau des systèmes d'aiguillages et de croisements) équipés de capacités de récupération d'énergie et d'algorithmes de traitement embarqués. Plutôt que de transmettre des données brutes, ces capteurs effectuent un traitement local pour extraire des métadonnées et détecter des anomalies à la source. Cette stratégie réduit considérablement l'utilisation de la bande passante et permet des réponses immédiates aux problèmes potentiels.

Les informations traitées sont ensuite transmises à des unités modulaires de passerelles d'observation virtuelles, qui agissent comme des concentrateurs de données. Ces unités agrègent les entrées de plusieurs capteurs et effectuent une analyse intermédiaire avant de transmettre les informations critiques à des serveurs centralisés. Au plus haut niveau de l'architecture, des algorithmes d'IA avancés effectuent une analyse globale sur le réseau pour prédire les besoins de maintenance et coordonner les interventions proactives.

Cette approche de traitement distribué est essentielle, car elle réduit la surcharge de communication dans les environnements ferroviaires éloignés, soutient la prise de décision locale en temps réel pour les événements critiques de sécurité, minimise les exigences de stockage des données et améliore la résilience du système en distribuant l'intelligence à travers le réseau.

Alors que le système de surveillance ferroviaire plus large incorpore un flux de données hiérarchique complet, notre travail se concentre spécifiquement sur le développement de modèles d'IA légers pour la compression des signaux et la détection d'anomalies, capables de fonctionner directement sur des nœuds de capteurs embarqués.

4.3.2 Description du Jeu de Données

Pour développer et valider le modèle, un jeu de données de signaux de vibration a été collecté dans diverses conditions de fonctionnement du réseau ferroviaire. En raison des politiques de protection des données, certains détails du processus de collecte de données sont omis. Les signaux ont été acquis à l'aide d'accéléromètres montés sur plusieurs systèmes d'aiguillages et de croisements ferroviaires. Le jeu de données se compose de 13 907 enregistrements de séries temporelles, avec des

fréquences d'échantillonnage allant de 3 100 à 3 350 Hz. Parmi ceux-ci, 5 000 sont anormaux, et chaque enregistrement dure entre 3 et 26 secondes.

Les signaux ont été enregistrés dans différentes conditions, incluant des variations de types de trains, du nombre de wagons, des vitesses et de la température ambiante. Pour prétraiter les données, nous avons appliqué une procédure simple de rognage pour éliminer le bruit du début et de la fin de chaque signal. De plus, la STFT, comme détaillé dans la Section 3.1.2, a été utilisée pour une extraction supplémentaire de caractéristiques.

4.3.3 Entraînement du Modèle

Pour assurer la séparabilité des anomalies par rapport aux données normales, un apprentissage par contraste supervisé est introduit en parallèle de l'entraînement à la reconstruction. Le but de la partie contrastive est d'élargir la marge entre les deux clusters pour permettre une discrimination rapide des échantillons sans nécessiter un modèle de décision complet. L'Algorithme 1 montre la boucle d'entraînement au sein de chaque époque.

Algorithm 1 Itération d'apprentissage

for each (x_1, x_2) **do**
  $z_1 \leftarrow f(x_1)$
  $z_2 \leftarrow f(x_2)$
  $\hat{x_1} \leftarrow f^{-1}(z_1)$
  $L_R \leftarrow 1 - \frac{1}{N}\frac{x_1 \hat{x_1}^\top}{||x_1||.||\hat{x_1}||} + \frac{1}{ND}\sum (x_1 - \hat{x_1})^2$
  $L_C \leftarrow \begin{cases} \frac{1}{ND}\sum (\bar{z_1} - \bar{z_2})^2 \text{ s'ils sont de la même classe} \\ \textbf{min}(\text{MARGIN} - \frac{1}{ND}\sum (\bar{z_1} - \bar{z_2})^2, 0) \text{ sinon} \end{cases}$
  if fin du batch **or** plus de données **then**
    $L_G \leftarrow \lambda_1 L_R + \lambda_2 L_C$
    Rétropropager L_G et mettre à jour les poids
  end if
end for

Chaque échantillon d'entraînement est associé à un autre échantillon pour former des paires pour le calcul de la perte contrastive. Notre implémentation utilise une liste d'indices mélangés pour assurer des appariements aléatoires et diversifiés. Les paires de signaux restent fixes pendant tout le processus d'entraînement.

Le code itère sur ces indices mélangés, sélectionnant l'échantillon original et son homologue apparié dans le jeu de données. Pour chaque paire d'entrées (x_1, x_2),

l'encodeur f mappe ces entrées à leurs représentations latentes z_1 et z_2. Le décodeur f^{-1} reconstruit ensuite x_1 en $\hat{x_1}$. La perte de reconstruction L_R mesure la qualité de la reconstruction en combinant un terme de similarité cosinus et un terme d'erreur quadratique moyenne (MSE). La similarité cosinus évalue l'alignement dans l'espace des caractéristiques, tandis que le MSE assure la précision pixel par pixel.

En parallèle, la perte contrastive évalue la similarité entre des parties sélectionnées des deux matrices latentes. Ces sous-matrices, appelées $\bar{z_1}$ et $\bar{z_2}$, sont comparées en fonction de leur clustering. Pour les paires du même cluster, L_C minimise la distance entre les vecteurs latents. Pour les paires de clusters différents, elle impose la marge fixe donnée pour maintenir la séparation dans l'espace latent.

À la fin de chaque lot, ou lorsqu'il n'y a plus de données, la perte globale L_G — une combinaison linéaire de L_R et L_C — est rétropropagée pour mettre à jour les poids du modèle. Les coefficients λ dans cette combinaison contrôlent l'influence relative de chaque terme de perte, permettant un réglage fin de leur impact sur l'entraînement.

Après avoir complété la boucle d'entraînement pour une époque, le modèle est évalué sur les données de validation pour calculer les pertes de validation. Les données de validation sont traitées de manière similaire aux données d'entraînement, mais sans mise à jour des paramètres du modèle. Les pertes de validation sont moyennées et utilisées pour calculer la perte de validation globale.

4.4 Résultats et Discussion

Notre objectif final est de parvenir à l'implémentation du modèle dans une cible embarquée avec des capacités de calcul et une capacité mémoire limitées. Par conséquent, nous avons réduit l'architecture pour accommoder notre cas d'utilisation sans perdre trop de performance sur les métriques. La matrice latente a une forme de 8 lignes (M) et 32 colonnes (C). Le module d'attention croisée de l'encodeur a une dimension de caractéristique (F) de 128 canaux et son MLP a une couche cachée (H) de 256 neurones. Nous avons choisi d'inclure un bloc de processus avec une taille de caractéristique de 64 canaux et une taille cachée de 64 neurones. Le décodeur a une dimension de caractéristique de 128 canaux et une couche cachée de 512 neurones. Tous les modules d'attention sont à tête unique pour réduire l'empreinte mémoire du modèle. Dans les expériences suivantes, les modèles ont été entraînés en utilisant l'optimiseur Adam avec un taux d'apprentissage de 0,001 sur 50 époques et une taille de lot de 256 signaux.

Pour garantir une évaluation robuste, nous avons employé une stratégie de validation croisée en 3 plis pour entraîner tous les modèles. Dans chaque pli, 9 271

signaux ont été utilisés pour l'entraînement et 4 636 pour les tests. Les ensembles d'entraînement et de test étaient équilibrés pour contenir un nombre égal de signaux normaux et anormaux.

4.4.1 Performance des Modèles de Compression de Données et d'Apprentissage Profond

L'échantillonnage compressé (CS) est un paradigme d'acquisition de signaux qui permet la reconstruction de signaux clairsemés ou compressibles à partir de mesures significativement moins nombreuses que celles requises par le théorème de Nyquist (Candès et Wakin 2008). Dans le traitement des signaux de vibration, le CS a été utilisé pour réduire les demandes de transmission et de stockage de données dans les réseaux de capteurs sans fil en exploitant la parcimonie inhérente des signaux de vibration mécanique dans des domaines de transformation appropriés (Yin et al. 2024), (Huang et al. 2024).

Pour établir une référence de comparaison avec notre modèle Perceiver IO proposé, nous avons implémenté la méthode d'échantillonnage compressé de (Huang et al. 2024), qui combine un algorithme amélioré d'apprentissage de dictionnaire K-SVD avec une poursuite adaptative parcimonieuse (SAMP) pour l'estimation automatique de la parcimonie. Nous avons sélectionné cette méthode car elle est spécifiquement conçue pour la compression de signaux dans les réseaux de capteurs sans fil, ce qui correspond étroitement à notre cas d'utilisation. Notre implémentation a utilisé 128 atomes de dictionnaire, entraînés sur 50 itérations. Le SAMP a été configuré pour fonctionner sur 800 itérations de reconstruction avec une tolérance de $1e^{-4}$.

Nous avons sélectionné 727 signaux de notre jeu de données correspondant à un seul type de train fonctionnant à une vitesse fixe, afin de simuler un jeu de données presque homogène où les signaux sont très similaires les uns aux autres. Chaque signal a été rogné pour contenir exactement 8000 échantillons. La méthode a été évaluée avec un taux de compression de 0,5, ce qui signifie que 4000 mesures ont été acquises à partir de signaux de 8000 échantillons. Cette configuration nous a permis de comparer ses performances avec notre approche de compression basée sur l'apprentissage profond proposée.

Pour garantir que les deux méthodes fonctionnent avec le même taux de compression, nous avons augmenté la dimension de l'espace latent du modèle Perceiver IO de 8 lignes et 32 colonnes à 32 lignes et 125 colonnes.

Cette comparaison se concentre sur la compression des signaux. Dans le cas du Perceiver IO, l'algorithme calcule uniquement la perte de reconstruction L_R (et

$\lambda_1 = 1$), tandis que la perte contrastive L_C, qui fait partie de l'objectif d'entraînement complet, est omise dans ce contexte.

TABLE 4.1 – Métriques d'entraînement pour K-SVD et Perceiver IO

	Temporel		Fréquentiel	
	K-SVD	Perceiver IO	K-SVD	Perceiver IO
Erreur quadratique	4.960±0.113	7.984±0.183	0.019±0.001	0.011±0.000
Similarité	0.186±0.000	0.000±0.003	0.585±0.026	0.803±0.003

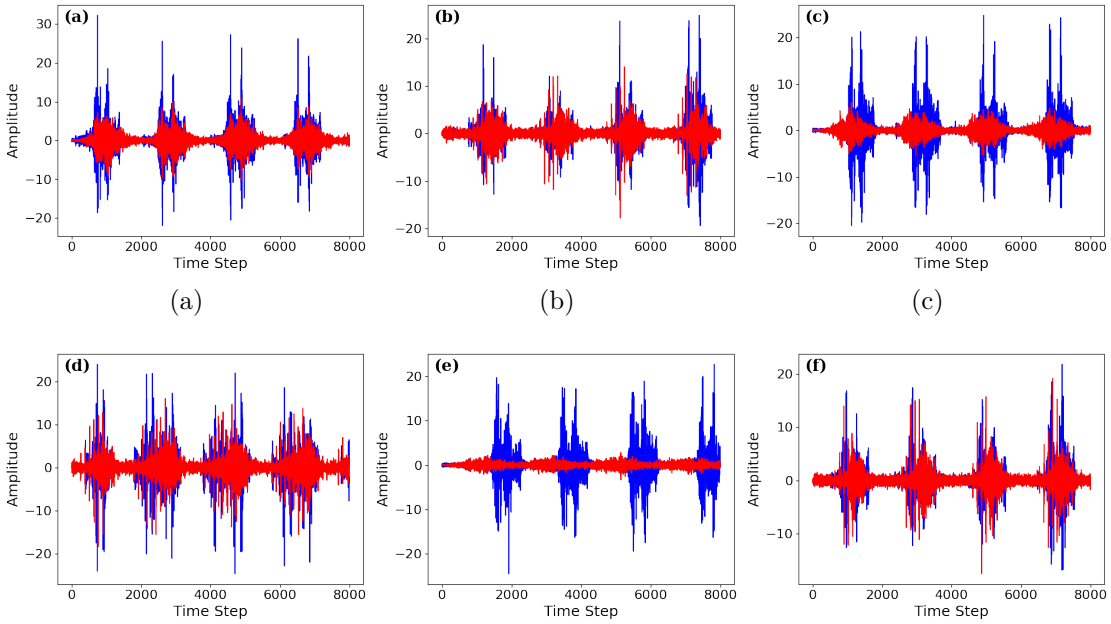


FIGURE 33 – Reconstructions de signaux d'exemple. (a), (c) et (e) : en utilisant l'algorithme K-SVD. (b), (d) et (f) : en utilisant l'algorithme Perceiver IO.

Dans le cas de l'algorithme K-SVD, la sortie est un signal reconstruit dans le domaine temporel. En revanche, notre modèle produit la représentation en spectrogramme du signal. Pour permettre une comparaison entre les deux approches, nous avons appliqué l'algorithme de reconstruction de phase RTPGHI (Real-Time Phase Gradient Heap Integration) (Pruša et Søndergaard 2016) et un algorithme classique d'inversion de STFT pour convertir la sortie de notre modèle en domaine temporel. La Table 4.1 rapporte l'erreur quadratique moyenne (MSE) et la similarité cosinus entre les signaux originaux et reconstruits dans les domaines temporel

et temps-fréquence pour l'algorithme K-SVD et notre approche proposée. La Figure 33 montre des exemples de signaux temporels reconstruits. Les graphiques de gauche correspondent aux reconstructions K-SVD, tandis que ceux de droite montrent les sorties du modèle Perceiver IO. Pour chaque exemple, les signaux originaux et reconstruits sont tracés ensemble pour évaluer visuellement la qualité de la reconstruction.

Les métriques du domaine temporel présentées dans le tableau suggèrent que les reconstructions K-SVD sont significativement meilleures que celles produites par notre modèle. Notamment, la similarité cosinus est proche de zéro. Cependant, le problème réside dans le fait que ces métriques ne représentent pas avec précision la similarité entre la reconstruction et le signal original. L'utilisation de l'algorithme RTPGHI pour la reconstruction temporelle introduit des décalages de phase qui dégradent les deux métriques. Par exemple, un décalage de phase de 90 degrés réduit la similarité cosinus à zéro, tandis qu'un décalage de phase de 180 degrés maximise l'erreur quadratique moyenne. Ces distorsions sapent la validité des comparaisons dans le domaine temporel.

Pour fournir une évaluation plus précise de la qualité de reconstruction des deux modèles, des métriques temps-fréquence sont également calculées. Dans ce contexte, l'erreur quadratique moyenne représente la reconstruction précise de l'intensité de chaque composante fréquentielle à un moment donné, se rapportant ainsi à l'amplitude du signal temporel et à la fidélité du contenu spectral. Inversement, la similarité cosinus évalue la préservation de la forme globale du signal temporel. Les deux métriques indiquent de meilleures performances pour notre modèle, ce qui correspond à la comparaison visuelle des échantillons de reconstruction montrés dans la Figure 33. Le K-SVD peut reconstruire avec un bon niveau de fidélité (Figure 33a), mais il a clairement du mal à reconstruire les signaux impliquant des décalages temporels (Figures 33c, 33e). Cette limitation est courante parmi les algorithmes d'échantillonnage compressé, qui reposent généralement sur des dictionnaires fixes et supposent la parcimonie dans une certaine base. De telles hypothèses rendent difficile pour ces méthodes de gérer des jeux de données hétérogènes avec diverses variations de signaux, y compris des décalages temporels. De plus, à mesure que la longueur du signal augmente, la complexité de la reconstruction précise croît, conduisant souvent à une performance dégradée.

En raison de ces défis, nous avons restreint notre jeu de données à un ensemble pseudo-homogène de signaux pour garantir des comparaisons équitables et significatives entre les modèles. Pour les entraînements suivants, nous utilisons l'ensemble complet de données disponibles, qui inclut des signaux de sept types de trains différents se déplaçant à des vitesses comprises entre 90 et 110 km/h. Nous avons

observé que, bien que la méthode fonctionne bien, la gestion d'une diversité significativement plus grande de motifs de signaux peut nécessiter d'augmenter la taille de l'espace latent et, par conséquent, le modèle global. Cette limitation reflète un compromis courant dans les modèles d'IA entre complexité et généralisation.

Dans les deux modèles, les pics correspondant aux passages d'essieux ne sont pas bien reconstruits. Cependant, notre modèle présente des artefacts induits par l'algorithme de reconstruction de phase (Figures 33b, 33d, 33f). Ces artefacts pourraient potentiellement être atténués en ajustant finement les paramètres du spectrogramme, notamment en réduisant la longueur du pas. Cela peut conduire à une meilleure fidélité de reconstruction et à des distorsions liées à la phase réduites, mais augmentera l'empreinte mémoire.

4.4.2 Résultats sur l'Apprentissage par Contraste

La fonction de perte globale, qui est une combinaison linéaire de métriques normalisées, a été utilisée pour contrôler l'influence de chaque métrique lors de la rétropropagation. Les valeurs de λ_1 et λ_2 sont initialement choisies arbitrairement. Ces hyperparamètres contrôlent le compromis entre l'optimisation de deux objectifs concurrents : la reconstruction du signal et l'apprentissage par contraste. Augmenter l'un par rapport à l'autre donne la priorité à l'objectif correspondant pendant l'entraînement. Dans nos expériences, nous fixons $\lambda_1 = 1$, c'est-à-dire une influence de 100 % à la fois sur les pertes MSE et de similarité cosinus, et nous varions λ_2 pour identifier une valeur qui sépare efficacement les classes (normal versus anomalie) tout en préservant la qualité de la reconstruction en spectrogramme. Pour simplifier, nous faisons référence à λ_2 comme poids contrastif.

Table 4.2 – Résumé des pertes et scores F1 suivant le poids alloué à l'entraînement contrastif

Poids	Erreur quadratique	Similarité	Perte en contrastif	Précision	Score F1 (Normal)	Score F1 (Anomalie)
0.00	**0.0178**	**0.8140**	0.9792	0.4926	0.5262	0.5144
	±0.0006	**±0.0042**	±0.1755	±0.2710	±0.2680	±0.2492
0.05	0.0210	0.7905	0.0373	0.9981	0.9984	0.9953
	±0.0016	±0.0068	±0.0141	±0.0008	±0.0004	±0.0029
0.10	0.0385	0.6267	0.0075	0.9992	0.9994	0.9986
	±0.0027	±0.0123	±0.0046	±0.0006	±0.0005	±0.0016

Poids	Erreur quadratique	Similarité	Perte en contrastif	Précision	Score F1 (Normal)	Score F1 (Anomalie)
0.15	0.0400 ±0.0013	0.6215 ±0.0073	0.0072 ±0.0033	0.9994 ±0.0005	0.9991 ±0.0007	0.9995 ±0.0003
0.20	0.0402 ±0.0007	0.6192 ±0.0106	0.0040 ±0.0030	**0.9999 ±0.0001**	**0.9998 ±0.0001**	**0.9998 ±0.0001**
0.25	0.0413 ±0.0010	0.6028 ±0.0063	0.0096 ±0.0078	0.9990 ±0.0010	0.9987 ±0.0014	0.9991 ±0.0008
0.30	0.0386 ±0.0006	0.6261 ±0.0018	**0.0015 ±0.0008**	0.9997 ±0.0001	0.9997 ±0.0000	0.9997 ±0.0002
0.35	0.0395 ±0.0017	0.6107 ±0.0025	0.0021 ±0.0006	0.9996 ±0.0002	0.9995 ±0.0003	0.9996 ±0.0002
0.40	0.0409 ±0.0007	0.6129 ±0.0047	0.0057 ±0.0035	0.9986 ±0.0009	0.9984 ±0.0014	0.9985 ±0.0006
0.45	0.0367 ±0.0024	0.6238 ±0.0290	0.1502 ±0.2089	0.9822 ±0.0240	0.9854 ±0.0191	0.9766 ±0.0320

La Table 4.2 présente un résumé des métriques d'évaluation calculées pour une série de notre modèle proposé entraîné avec différents poids contrastifs. Chaque ligne correspond à une configuration distincte du modèle, où le poids contrastif est incrémenté de 0,00 à 0,45. Les colonnes rapportent la moyenne et l'écart type des métriques de performance à travers les plis, y compris la perte MSE, la perte de similarité cosinus et la perte contrastive. De plus, la performance de classification est représentée par la précision et le score F1 pour les classes normale et anomalie.

Les métriques présentées dans ce tableau indiquent que toutes les configurations, à l'exception des deux premières et de la dernière, donnent des résultats similaires. La première configuration sans apprentissage contrastif nous donne les métriques de reconstruction de base (0,0178 pour MSE et 81,40 % pour la similarité cosinus). La précision de classification et les scores F1 pour la détection normale et d'anomalie s'améliorent significativement avec le poids contrastif, atteignant des valeurs presque parfaites autour des poids de 0,20 et plus. Notamment, à un poids de 0,20, le modèle atteint une précision de 99,99 %, avec des scores F1 dépassant 0,999 pour les deux classes, tandis que les métriques de reconstruction (MSE et similarité cosinus) se stabilisent à environ 0,04 pour MSE et 61 % pour la similarité cosinus, indiquant un compromis minimal en qualité de reconstruction. Au-delà du poids de 0,20, bien que certaines configurations comme 0,30 et 0,35 montrent également d'excellentes performances, elles ne surpassent pas la configuration de 0,20 en termes

d'équilibre global et de cohérence à travers toutes les métriques. Par conséquent, le poids contrastif de 0,20 est choisi pour son équilibre optimal entre la qualité de reconstruction et la précision de classification, démontrant la performance la plus robuste et fiable.

Pour garantir un modèle global léger et efficace, nous évitons d'incorporer des modèles complexes supplémentaires pour la détection d'anomalies et utilisons plutôt un classificateur linéaire simple. Chaque signal est classé en fonction de la première composante PCA dérivée de l'espace latent comme montré dans la Figure 32. Cette composante est comparée à un seuil de décision pour déterminer la classification. Ce seuil est calculé en premier lieu en calculant la moyenne de la première composante PCA pour les signaux normaux et la moyenne de la première composante PCA pour les signaux anormaux, puis en prenant la moyenne de ces deux moyennes. Comparé à des classificateurs plus lourds tels que les Machines à Vecteurs de Support (SVM) et les Forêts Aléatoires, cette approche offre des avantages computationnels significatifs. Les SVM nécessitent souvent des fonctions de noyau complexes et peuvent devenir intensifs en ressources avec des espaces latents de haute dimension. Les Forêts Aléatoires, bien que puissantes, impliquent plusieurs arbres de décision, ce qui augmente à la fois l'utilisation de la mémoire et le temps d'inférence. En revanche, notre méthode réduit la surcharge en se concentrant sur une seule frontière de décision linéaire, la rendant très efficace pour notre cas d'utilisation en temps réel et contraint en ressources.

Table 4.3 – Métriques de classification et d'entraînement

		Sans Contrastif		Avec Contrastive		Contrastif partiel	
		Normal	Anormal	Normal	Anormal	Normal	Anormal
Classification	Précision	0.43 ±0.24	0.24 ±0.17	0.998 ±0.001	0.999 ±0.000	0.992 ±0.002	0.997 ±0.000
	Rappel	0.29 ±0.19	0.38 ±0.23	0.999 ±0.000	0.999 ±0.000	0.995 ±0.001	0.995 ±0.001
	Score F1	0.35 ±0.22	0.30 ±0.19	0.998 ±0.001	0.999 ±0.000	0.994 ±0.001	0.996 ±0.000
Entraînement	Erreur quadratique	0.017±0.000		0.040±0.003		0.036±0.002	
	Similarité	0.813±0.012		0.621±0.014		0.647±0.025	

Perte en contrastif	0.840±0.049	0.058±0.076	0.025±0.019

Pour isoler les contributions de l'apprentissage contrastif et de l'architecture à double module des autres variations architecturales, nous comparons notre méthode proposée à un modèle Perceiver IO de base qui partage la même structure sous-jacente. La Table 4.3 présente les métriques de classification et de reconstruction pour le même modèle entraîné avec trois fonctions de perte différentes. La première perte n'incorpore pas l'apprentissage contrastif ($\lambda_2 = 0$). Les deuxième et troisième modèles incluent l'apprentissage contrastif avec $\lambda_2 = 0,2$, mais diffèrent dans l'entrée utilisée pour calculer la perte contrastive L_C. Le deuxième modèle applique l'apprentissage contrastif sur toute la matrice latente ; c'est-à-dire que la perte contrastive L_C est calculée en utilisant toutes les valeurs 8×32 de la matrice latente. Le troisième, appelé apprentissage contrastif partiel, applique l'apprentissage contrastif uniquement sur les 8 premières colonnes de la matrice latente — spécifiquement, une sous-matrice de taille 8×8 de la matrice latente complète 8×32 est utilisée pour calculer L_C.

La Figure 34 illustre la distribution des échantillons de notre jeu de données projetés sur les deux premières composantes principales (ligne supérieure), ainsi que les histogrammes des projections sur la première composante principale (ligne inférieure). Chaque visualisation correspond au modèle obtenu à partir de la moyenne des plis de la configuration respective.

Le modèle sans perte contrastive démontre de faibles performances de classification. La précision moyenne, le rappel et le score F1 pour la classe anormale sont particulièrement bas (0,24, 0,38 et 0,30), indiquant que de nombreuses anomalies sont mal classées. Ces résultats suggèrent que les caractéristiques extraites de la première composante PCA ne sont pas bien séparées, rendant la détection d'anomalies difficile pour la fonction de détection simple. Comme le montre la Figure 34a, le nuage de points PCA ne révèle aucun regroupement distinct entre les signaux normaux et anormaux. De manière cohérente, l'histogramme de la Figure 34b démontre que le seuil (ligne verte) ne sépare pas efficacement ces deux classes lorsqu'il est utilisé comme fonction linéaire de détection d'anomalies.

En revanche, l'application de l'apprentissage contrastif améliore considérablement les performances de classification, atteignant des valeurs de précision et de rappel presque parfaites de 0,999 pour les signaux anormaux. Cette amélioration est corroborée visuellement sur les Figures 34c, 34d, où les signaux normaux et anormaux sont clairement séparés le long de l'axe de la composante principale.

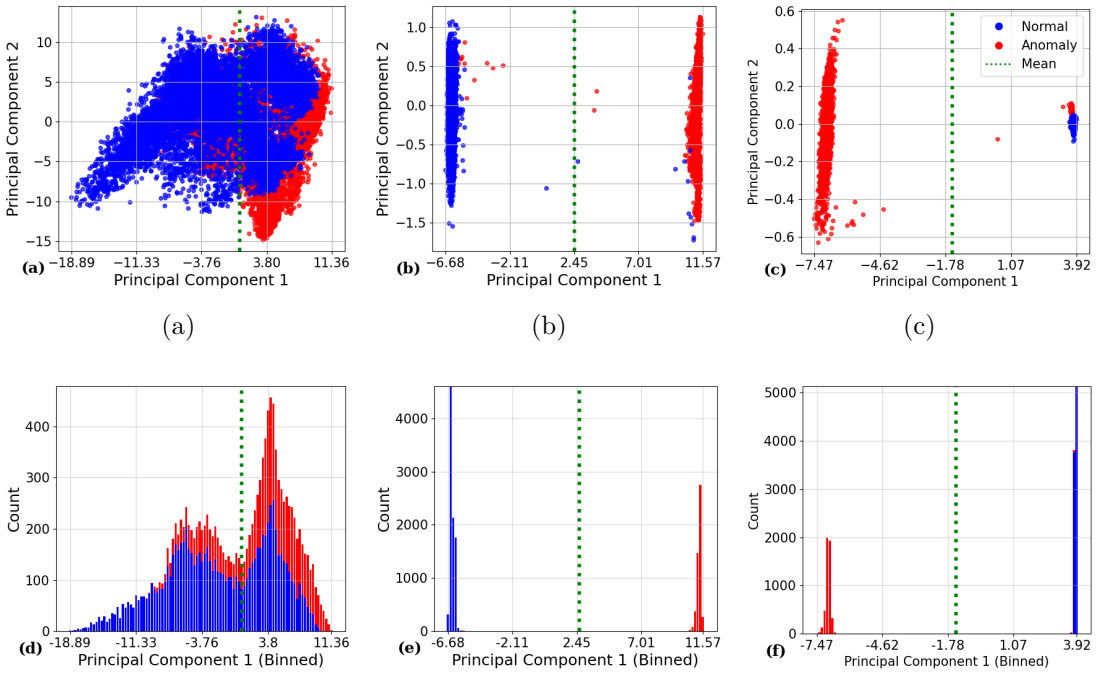


FIGURE 34 – Projection des deux premières composantes PCA : (a) sans apprentissage contrastif, (b) avec apprentissage contrastif et (c) avec apprentissage contrastif partiel. Histogramme de la première composante PCA : (d) sans apprentissage contrastif, (e) avec apprentissage contrastif et (f) avec apprentissage contrastif partiel.

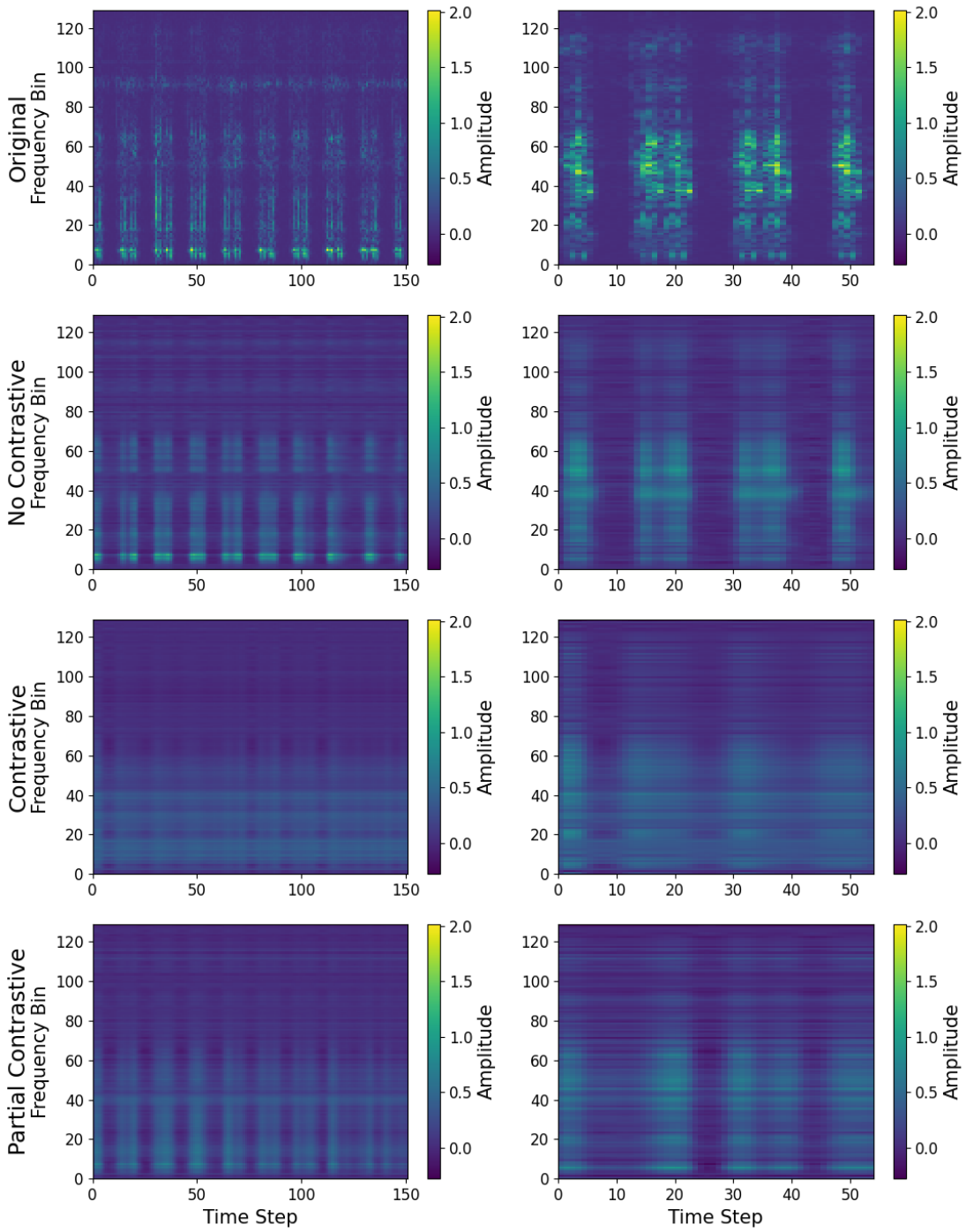


FIGURE 35 – Reconstructions en spectrogramme des trois modèles présentés pour deux exemples de signaux.

L'apprentissage contrastif partiel, représenté sur les figures, donne des métriques de classification légèrement inférieures par rapport à l'apprentissage contrastif complet. Néanmoins, il apporte des améliorations notables en termes de qualité de reconstruction, avec une augmentation d'environ 2,5 % de la similarité, comme nous pouvons le voir sur la Table 4.3. La Figure 35 illustre clairement que les reconstructions produites par le modèle d'apprentissage contrastif partiel (dernière ligne) présentent une fidélité supérieure par rapport à celles générées par le modèle d'apprentissage contrastif complet (troisième ligne). De plus, il est évident que l'introduction de l'apprentissage contrastif affecte négativement la qualité de reconstruction par rapport aux modèles entraînés sans celui-ci (deuxième ligne). Étant donné que les performances de classification restent largement comparables entre les modèles, avec les deux modèles d'apprentissage contrastif mal classant en moyenne 50 échantillons, ce compromis est jugé bénéfique pour l'application prévue.

Les performances de compression ont été évaluées en utilisant des métriques clés, y compris le taux de compression, qui reflète la taille des données compressées par rapport à l'original. De plus, nous avons considéré le rapport signal sur bruit de crête (PSNR), une métrique importante dans la compression d'images, d'audio et de vidéo. Notre modèle a atteint un PSNR de 20,46 avec un taux de compression de 88. En comparaison, le modèle construit dans (Jaegle, Borgeaud, et al. 2021), avec 20 millions de paramètres, a atteint un PSNR de 26,97. Malgré le fait que notre modèle n'ait pas atteint la même qualité de reconstruction, il est 80 fois plus petit, démontrant l'efficacité de l'architecture Perceiver IO même avec des modèles plus petits et des signaux flexibles de taille non fixe.

L'utilisation de l'apprentissage contrastif s'avère cruciale pour améliorer la séparabilité des caractéristiques, ce qui permet au modèle de regrouper les signaux de manière efficace, facilitant ainsi un processus de détection d'anomalies simple mais très précis, même en présence de classes déséquilibrées avec une réduction contrôlée de la qualité de compression. Les solides capacités de généralisation du modèle démontrent son efficacité non seulement dans l'étude de cas de l'industrie ferroviaire, mais aussi pour des applications potentielles dans d'autres domaines.

4.4.3 Environnements embarqués

TABLE 4.4 – Comparaison des tailles de modèles suivant ses paramètres

Modèle	Base	A	B	C	D	E	F	G	H	I	J	K
Latent (M)	8	4	16									
Latent (C)	32	16	64									
Encodeur (F)	128			64	256							
Encodeur (H)	256			128	512							
Decodeur (F)	128					64	256					
Decodeur (H)	512					256	1024					
Têtes	1							3				
Traitements	1								0	3		
Traitement (F)	64										32	128
Traitement (H)	64										32	128
Encodeur (KB)	314	240	463	183	578	314	314	775	264	414	290	364
Decodeur (KB)	645	628	678	645	645	323	1288	875	645	645	645	645
Total (KB)	960	869	1141	828	1223	638	1603	1651	910	1059	935	1009

La Table 4.4 présente plusieurs variantes du modèle avec différentes configurations comparées à la référence. Bien que certains modèles (A, C, E, H, et J) réduisent la taille du modèle et que d'autres l'augmentent, trouver un équilibre entre ces extrêmes est crucial pour les environnements embarqués. Dans les modèles A et B, la taille de la matrice latente détermine le taux de compression, avec des formes plus petites donnant des taux plus élevés. Les modèles C, D, et H à K ajustent la taille de l'encodeur, tandis que E et F modifient le décodeur. Dans notre cas d'utilisation, seule une partie du modèle s'exécute sur une cible embarquée. Étant donné l'architecture asymétrique de Perceiver IO, nous pouvons équilibrer stratégiquement

le calcul et la taille entre l'encodeur et le décodeur pour minimiser le composant embarqué. Le modèle G démontre notre choix d'attention à tête unique, qui réduit significativement la taille du modèle. Puisque nous privilégions un encodeur compact par rapport au décodeur, l'utilisation d'une seule tête d'attention conduit à un modèle plus petit, car les MLP dans l'encodeur sont plus petits que ceux dans le décodeur.

Pour montrer les capacités d'intégration de notre modèle, nous avons entraîné une variante rationalisée qui incorpore des paramètres des modèles A, C, E et H. Le nouveau modèle a une architecture compacte, avec l'encodeur mesurant 106,01 Ko et le décodeur à 307,63 Ko, résultant en une taille totale de modèle de 413,64 Ko. Ce modèle a atteint un MSE de 0,026 et une similarité cosinus de 72,7 %. De plus, il a produit un PSNR de 21,87. Pour référence, bien que non rapporté précédemment, le modèle plus grand sans apprentissage contrastif a atteint un PSNR plus élevé de 23,72, indiquant une amélioration de la qualité de reconstruction due à une plus grande capacité du modèle. Ce modèle optimisé est conçu pour s'intégrer dans une cible embarquée tout en maintenant des niveaux de performance comparables à des alternatives plus grandes et moins compactes.

4.5 Estimation de la dégradation

La détection d'anomalie présentée dans la section précédente ne permet pas d'estimer quantitativement l'état de santé du système d'aiguillage. On propose dans cette section d'implémenter un indicateur de dégradation à partir des représentations latentes fournies par le modèle de compression de la première contribution. La dégradation est une donnée présente en faible importance au sein des représentations latentes par rapport aux autres variables nécessaires à la reconstruction (vitesse et type de train notamment). Le calcul de la dégradation à partir de ces données consiste donc en l'extraction de cette information résiduelle des représentations latentes.

La dégradation n'évoluant pas significativement entre plusieurs signaux consécutifs, on propose d'établir l'estimation de cet indicateur sur un ensemble de signaux, i.e. une fenêtre temporelle glissante sur les échantillons, pour éliminer les dépendances aux variables de reconstruction de plus grande importance. En utilisant des contraintes dérivées de l'apprentissage par contraste présenté précédement, on entraîne un modèle à l'architecture similaire, en intégrant ce principe de fenêtre glissante.

4.5.1 Mesure de la Dégradation dans le Contexte Ferroviaire

La surveillance de la dégradation des composants ferroviaires joue un rôle clé dans la maintenance préventive et la gestion des actifs, permettant d'éviter des pannes coûteuses et d'assurer la sécurité du réseau. Plusieurs techniques conventionnelles sont couramment employées pour évaluer l'état des infrastructures et des équipements, chacune apportant des avantages spécifiques en fonction du type de défaut ou de dégradation recherché.

L'inspection visuelle demeure l'une des méthodes les plus simples et les plus répandues. Elle consiste en une observation directe par des experts pour repérer des signes visibles de dégradation tels que fissures, corrosion, déformations ou usures. Bien que facile à mettre en œuvre, cette méthode présente des limites, notamment la dépendance à l'œil humain et l'incapacité à détecter des défauts internes ou précoces.

L'analyse vibratoire est une technique avancée qui permet de détecter la dégradation mécanique en surveillant les vibrations générées par les composants en fonctionnement. Des capteurs placés sur les roulements, moteurs ou rails enregistrent les signaux vibratoires, lesquels sont décomposés via la transformée de Fourier (FFT) ou analysés par l'analyse des enveloppes pour identifier des fréquences anormales ou des défauts comme fissures ou désalignements. Cette méthode est particulièrement adaptée pour la surveillance des roulements, moteurs et rails, car elle peut révéler des défaillances incipientes avant qu'elles ne deviennent critiques.

Les ultrasons sont utilisés pour détecter des défauts internes ou mesurer l'épaisseur des matériaux. En émettant des ondes ultrasonores dans les composants, on analyse les échos réfléchis pour repérer fissures, inclusions ou variations d'épaisseur. Cette technique permet d'inspecter efficacement les rails, soudures et structures métalliques, notamment pour déceler des défauts non visibles à l'œil nu.

Les courants de Foucault offrent une méthode sans contact pour la détection des défauts de surface ou sous-surface dans les matériaux conducteurs. En induisant un champ magnétique variable dans le matériau, ils permettent d'identifier des fissures ou corrosions sans endommager la pièce, ce qui est très utile pour l'inspection des rails, roues ou composants métalliques.

La thermographie infrarouge complète ce panel en mesurant la distribution de température à la surface des composants. En capturant les émissions de chaleur en fonctionnement, cette technique permet de révéler des anomalies thermiques comme des zones de surchauffe ou des frottements excessifs, indiquant des dégradations ou des défauts mécaniques, notamment dans les moteurs, roulements ou systèmes de freinage.

Dans le contexte ferroviaire, ces techniques sont appliquées à divers composants. Pour les rails, l'inspection visuelle, l'ultrason et les courants de Foucault sont utilisés pour repérer fissures, corrosion ou défauts internes. La surveillance des roulements combine l'analyse vibratoire pour détecter usure ou dommages et la thermographie pour identifier des surchauffes. Les moteurs électriques sont surveillés par vibration et thermographie afin de détecter déséquilibres, surchauffe ou défaillances mécaniques. Enfin, les structures de support telles que ponts et viaducs sont inspectées visuellement pour détecter fissures ou déformations, et par ultrasons pour mesurer l'épaisseur des matériaux et repérer des défauts internes.

En conclusion, la mesure de la dégradation dans le secteur ferroviaire repose sur une combinaison de techniques complémentaires permettant une évaluation précise de l'état des infrastructures et équipements. L'intégration de ces méthodes favorise la détection précoce des défauts, la planification proactive des interventions de maintenance, et la prolongation de la durée de vie des actifs ferroviaires. Avec l'évolution des technologies, notamment l'analyse avancée de données et l'intelligence artificielle, ces approches continueront à s'améliorer, renforçant la sécurité, la fiabilité et la durabilité du réseau ferroviaire.

4.5.2 Compression et estimation d'indicateur de dégradation

Face à l'essor massif des objets connectés, des capteurs et des dispositifs intelligents, la gestion efficace des flux de données devient une nécessité stratégique. Ces systèmes, souvent déployés dans des environnements où les ressources matérielles et énergétiques sont limitées, doivent non seulement traiter, stocker et transmettre des volumes croissants d'informations, mais aussi fournir des indications précises sur l'état de leurs composants ou leur environnement. C'est dans ce contexte qu'interviennent des outils innovants capables de réaliser simultanément deux fonctions cruciales : la compression efficace des données et l'estimation d'indicateurs clés, tels que la dégradation d'un composant, la détection d'anomalies ou l'état de santé global d'un système.

Les enjeux sont doubles. La compression permet de réduire la volumétrie des données pour faciliter leur stockage ou leur transmission sur des réseaux à bande passante limitée, tout en économisant l'énergie consommée lors des opérations de traitement. Parallèlement, l'estimation d'indicateurs offre une lecture synthétique et fiable de l'état d'un système ou d'un dispositif, permettant des interventions rapides et ciblées. Leur combinaison constitue une véritable solution intégrée, adaptée aux contraintes spécifiques des environnements embarqués.

Les systèmes embarqués, qu'il s'agisse de capteurs industriels, de dispositifs IoT ou d'équipements médicaux, doivent fonctionner en temps réel, souvent dans des conditions difficiles et avec des ressources très restreintes. La capacité à traiter de grandes quantités de données en local, pour ensuite transmettre uniquement l'information essentielle, est un enjeu majeur pour la fiabilité et la performance des applications. Par ailleurs, la détection précoce de dégradations ou de déviations par rapport à un fonctionnement normal peut prévenir des défaillances coûteuses, voire critiques.

Ainsi, la solution repose sur des outils capables d'apprendre à représenter efficacement la structure des données normales, tout en étant sensibles aux écarts signifiants. En réduisant la dimension des données via compression, ils permettent une gestion plus efficiente des ressources. En même temps, en évaluant l'erreur de reconstruction ou d'estimation associée à chaque nouvelle donnée, ils fournissent un indicateur fiable pour la détection d'anomalies ou de dégradations.

Un avantage majeur des autoencodeurs dans cette tâche réside dans leur nature non supervisée. Ils n'ont pas besoin d'un grand nombre d'échantillons étiquetés pour apprendre à distinguer le "normal" de l'"anormal". La seule condition est d'avoir un corpus représentatif des données "normales" pour entraîner le modèle. Lorsqu'un nouveau signal est analysé, l'erreur de reconstruction sert de critère de détection. Si elle dépasse un seuil prédéfini, cela indique une probable anomalie ou dégradation.

Ce mécanisme est particulièrement précieux dans les environnements où les anomalies sont rares ou difficiles à anticiper, et où le coût d'une défaillance est élevé. Il permet une surveillance continue, en temps réel, et une réponse immédiate à toute déviation suspecte. La simplicité de cette approche, combinée à sa robustesse, en fait une solution privilégiée pour les applications embarquées.

Dans le secteur industriel, la surveillance de machines ou d'équipements via des capteurs de vibration, de température ou de pression illustre parfaitement cette approche. Par exemple, un autoencodeur entraîné sur des données normales d'un moteur peut réduire la taille des flux de données tout en détectant rapidement toute déviation indicative d'un problème mécanique. Si un capteur signale une vibration anormale, la reconstruction échoue ou est moins précise, déclenchant une alerte pour intervention préventive.

Dans le domaine médical, la surveillance continue de paramètres physiologiques tels que le rythme cardiaque ou la saturation en oxygène, combinée à une compression efficace des flux, permet de réduire la consommation d'énergie et la bande passante tout en assurant une détection rapide de tout changement significatif. Cela facilite la prise en charge à distance, notamment dans des contextes de télésurveillance ou de dispositifs portables.

L'un des atouts majeurs des autoencodeurs est leur capacité à être déployés sur des microcontrôleurs ou des FPGA, grâce à des techniques d'optimisation et de compression de modèles. La possibilité d'apprentissage en ligne, où le modèle s'adapte en continu à l'environnement, offre une flexibilité supplémentaire face à l'évolution des conditions opérationnelles.

Cependant, cette approche n'est pas sans défis. La performance dépend fortement de la qualité et de la représentativité des données d'entraînement. Une évolution rapide du contexte ou la présence de données non représentatives peut réduire la fiabilité du modèle. La calibration précise des seuils de détection, la gestion du bruit et la conception de modèles légers mais efficaces constituent autant de défis techniques. La recherche avance toutefois rapidement, notamment avec le développement de réseaux autoencodeurs convolutionnels ou récurrents, optimisés pour la faible consommation et la faible puissance de calcul.

Les avancées en deep learning ouvrent des perspectives prometteuses. La compression de modèles, l'apprentissage en ligne, et l'intégration avec d'autres techniques, comme l'apprentissage semi-supervisé ou l'intégration de règles métier, permettent d'améliorer la fiabilité, la précision et la robustesse des systèmes. La mise en œuvre d'approches hybrides, combinant autoencodeurs avec des techniques de détection supervisée ou semi-supervisée, offre aussi de nouvelles voies pour renforcer la détection d'anomalies.

En somme, les outils capables de conjuguer compression et estimation d'indicateurs sont devenus des composants clés pour la gestion intelligente et autonome des systèmes embarqués. Leur développement continu, soutenu par les progrès technologiques, contribuera à rendre ces dispositifs plus efficaces, fiables et économes en ressources dans une multitude d'applications industrielles, médicales ou environnementales. La recherche vise désormais à rendre ces outils encore plus adaptatifs, légers et précis, afin d'assurer une intégration fluide dans des objets connectés toujours plus intelligents et autonomes.

4.5.3 Méthodologie

Le modèle proposé est d'architecture identique au modèle de compression. Cependant, il est conçu pour prendre en entrée et reconstruire une séquence de 16 latents consécutifs dans le temps issus de ce même modèle de compression. Cette approche permet au système d'analyser et de modéliser efficacement la dynamique temporelle des données, en éliminant les facteurs trop variables d'un latent à l'autre et en mettant en valeur les informations présentes dans la majorité des entrées. Comme pour le modèle de détection d'anomalie, on sépare le latent en deux portions dis-

tinctes : une partie porte l'information commune sur les latents proposés et l'autre les variations entre latents.

L'information à extraire des latents correspond à l'usure moyenne provoquée par le passage des trains pendant la fenêtre temporelle donnée, i.e. la différence entre le niveau de dégradation avant et après la fenêtre temporelle contenant les latents choisis. Pour réaliser l'apprentissage du modèle, on utilise un jeu de données provenant d'un système sain, remplacé peu de temps avant le premier échantillon. Cet aiguillage considéré comme neuf ne se dégrade que très peu sur le début de sa plage de vie. On applique donc au cours de l'entraînement une perte proche de l'apprentissage contrastif qui consiste à minimiser les deltas de dégradation donnés par le modèle. En se concentrant sur ces données de référence, le modèle apprend à représenter et à reproduire des comportements normaux, ce qui facilite la détection de toute déviation ultérieure.

4.5.3.1 Facteurs d'équilibrage des pertes

Les fonctions de pertes utilisées au cours de l'entraînement sont toutes deux des erreurs quadratiques. Contrairement à l'apprentissage contrastif présenté dans le chapitre précédent, les fonctions de pertes de reconstruction et contrastive sont identiques. Cependant, elles ne s'appliquent pas sur la même donnée : le latent pour la partie contrastive et la reconstruction pour la perte globale. Comme les deux cibles ne sont pas de même taille, les deux pertes ne sont pas normalisées. On introduit pour cela des facteurs d'équilibrage adaptatifs permettant à l'optimiseur de ne pas priviligier l'une des deux pertes par rapport à l'autre. On a :

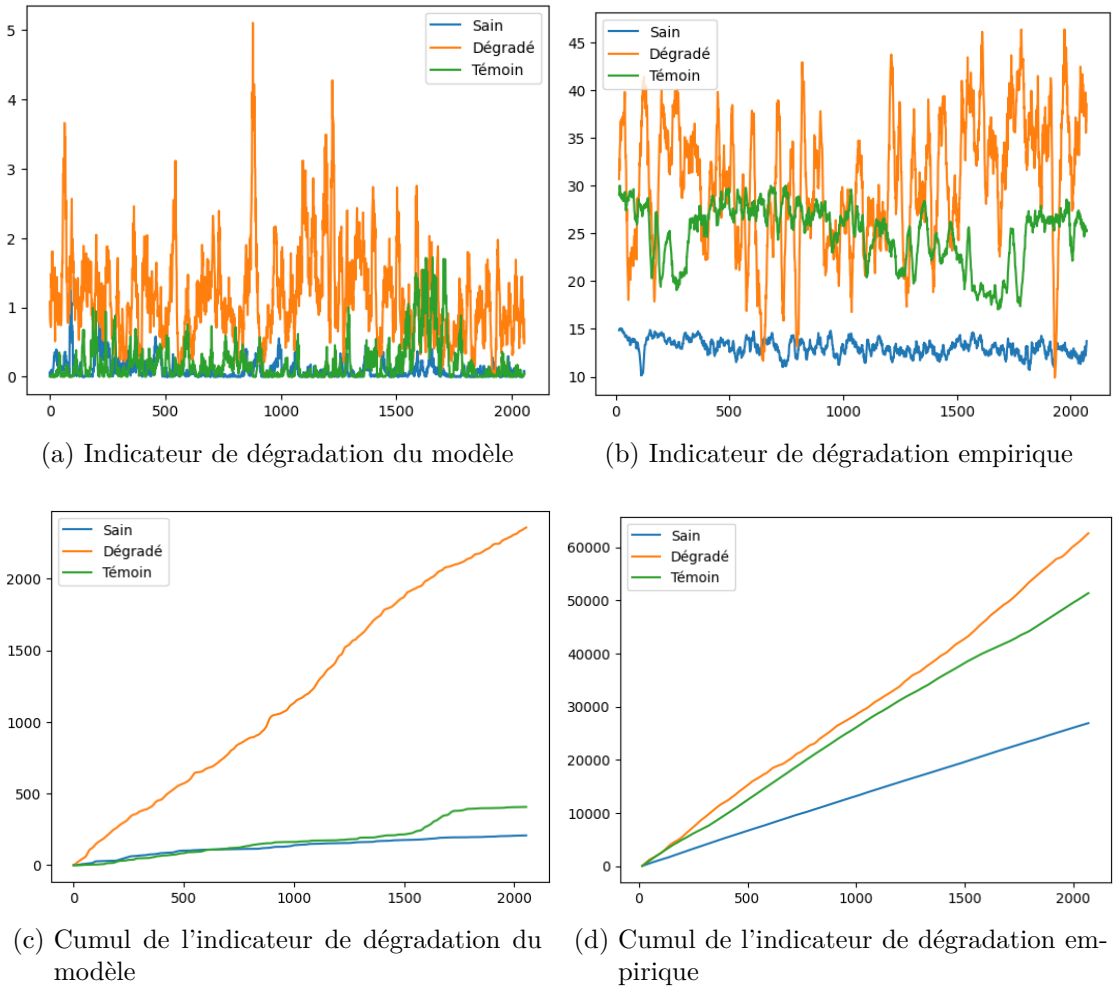
$$\begin{aligned} w_R &= L_R(o) \\ w_C &= L_C(l) \\ L_G(o,l) &= w_C L_R(o) + w_R L_C(l) \end{aligned}$$

Les deux termes de la somme formant la fonction de perte globale L_G sont égaux par valeur : l'optimiseur équilibrera l'apprentissage entre les deux fonctions de pertes sommées. Sans ces facteurs d'équilibrage, la perte contrastive est nettement inférieure à la perte en reconstruction ce qui amène le modèle à ne pas généraliser suffisamment l'extraction de l'information de dégradation.

4.5.4 Résultats

Après entraînement, on évalue le modèle sur trois jeux de données différents :

1. Un jeu de données provenant d'un équipement sain : c'est celui utilisé pour l'apprentissage ;
2. Un jeu de données provenant d'un équipement dégradé : ce dernier est un aiguillage usé qui a été remplacé peu de temps après la plage de mesure ;
3. Un jeu de données provenant d'un équipement en bonne santé : ce jeu de données sera appelé jeu témoin.



(a) Indicateur de dégradation du modèle

(b) Indicateur de dégradation empirique

(c) Cumul de l'indicateur de dégradation du modèle

(d) Cumul de l'indicateur de dégradation empirique

FIGURE 36 – Comparaison des indicateurs de dégradation

On propose sur la Figure 36a les sorties du modèle de dégradation pour les trois jeux de données. En comparaison, on propose sur la Figure 36b un indicateur statistique empirique utilisé par VOSSLOH pour évaluer expérimentalement le niveau de santé d'un aiguillage. Si les jeux de données dits sain et dégradé semblent effectivement exhiber des indicateurs respectivement faibles et élevés, le jeu dit témoin

semble plus proche du comportement dégradé selon l'indicateur empirique, alors que notre modèle dit l'inverse.

Les indicateurs considérés étant considérés comme des deltas de dégradations, les Figures 36c, 36d montrent la somme cumulée des valeurs au cours du temps. Le comportement du jeu témoin apparaît très clairement comme similaire au jeu sain selon notre modèle. À contrario, l'indicateur empirique montre encore un comportement du jeu témoin proche du jeu dégradé.

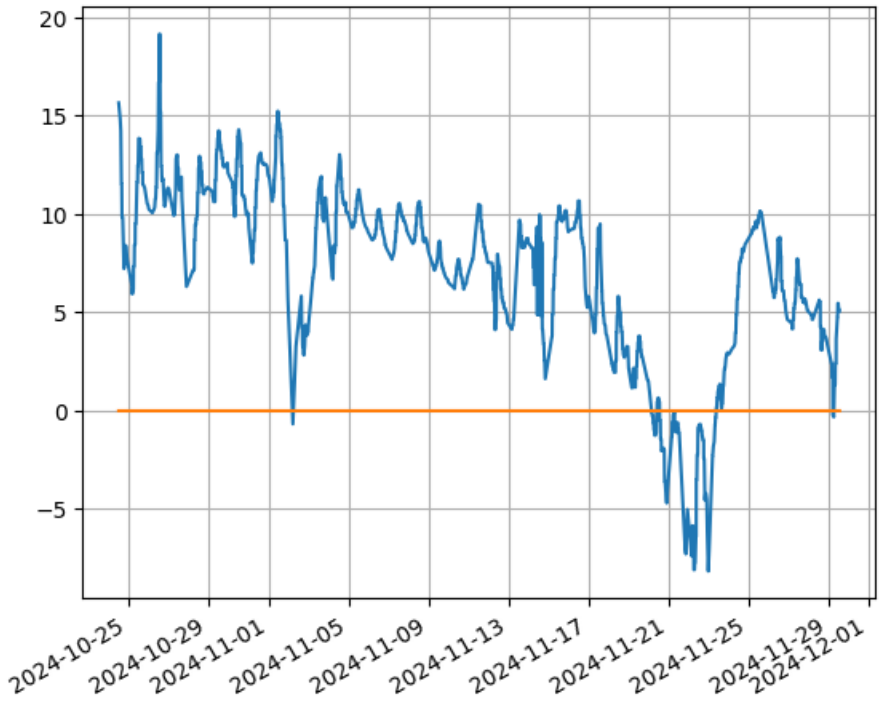


FIGURE 37 – Évolution de la température sur la plage de temps correspondant au jeu témoin

Il est également important de noter que la température exerce un impact significatif sur les deux indicateurs, empirique et modélisé. Si l'on observe les courbes entre les abscisses 1600 et 1700, on remarque un changement de comportement très visible sur notre modèle, et moins sur l'indicateur empirique. Il se trouve que sur cette plage de temps, la température est tombé en dessous de 0°C pendant une longue durée (Figure 37). Ce passage de la température dans le négatif affecte le comportement du système d'aiguillage et/ou de ses fondations, ce qui modifie les réponses vibratoires obtenues.

Cet effet thermique complique toute tentative de correction ou d'ajustement du modèle, car sans disposer de davantage de données, il est impossible d'isoler ou de compenser efficacement l'influence de la température. En conséquence, toute

amélioration ou adaptation du modèle nécessiterait la collecte de données supplémentaires afin de mieux comprendre cette dépendance et d'intégrer ces variations dans le processus d'apprentissage ou de calibration.

4.6 Conclusion

Notre premier modèle intégrant la détection d'anomalie dans le modèle de compression montre de bonnes performances dans l'analyse des vibrations et la détection de défauts grâce au mécanisme d'attention, qui isole les caractéristiques critiques et filtre le bruit. Le mécanisme de sortie basé sur les requêtes améliore la précision dans l'identification des anomalies, telles que l'usure ou le désalignement, dans des ensembles de données étendus et bruyants. La scalabilité du modèle et ses capacités contrastives améliorent la précision de la détection de défauts et l'adaptabilité à travers divers domaines industriels. De plus, l'intégration de la compression et de la détection d'anomalies dans un cadre unifié réduit la dimensionnalité des données, accélère les calculs et diminue les exigences de mémoire, ce qui le rend idéal pour les applications de surveillance en temps réel et de maintenance prédictive. Cette combinaison de caractéristiques établit notre modèle comme un outil robuste et efficace pour l'analyse des vibrations et le diagnostic de défauts dans le monde réel.

Le second modèle développé démontre une capacité à estimer la dégradation moyenne d'un aiguillage à partir des latents du modèle de compression. Grâce à la stratégie d'apprentissage contrastif équilibrée et à l'utilisation de données issues d'un système sain, le modèle parvient à distinguer efficacement les comportements normaux des déviations significatives.

Cependant, les résultats mettent en lumière une sensibilité notable aux variations de température, qui influencent fortement les résultats de notre indicateur de dégradation. Il est nécessaire de disposer de données supplémentaires, par exemple sur une année entière afin d'enrichir le jeu d'apprentissage pour intégrer ces effets environnementaux. Une telle démarche permettrait d'améliorer la robustesse des résultats obtenus.

Chapitre 5

Conclusion générale

Ce chapitre présente les solutions apportés au modèle de compression afin de propager l'information de détection d'anomalie au sein d'un réseau de capteurs.

5.1 Synthèse

Les travaux menés dans le cadre de cette thèse ont permis de développer une approche globale et innovante pour répondre aux enjeux de la surveillance intelligente des infrastructures ferroviaires, en s'appuyant sur des algorithmes d'intelligence artificielle embarqués. La problématique centrale visait à concevoir des solutions capables de compresser, analyser et interpréter les signaux vibratoires issus de capteurs autonomes, tout en respectant des contraintes strictes de consommation énergétique et de ressources computationnelles. L'objectif était double : réduire la transmission de données brutes en faveur d'un traitement local efficace, et estimer la dégradation des systèmes à partir des informations extraites.

Les résultats obtenus démontrent tout d'abord la faisabilité et l'efficacité d'architectures neuronales allégées, adaptées aux environnements embarqués. Le modèle Perceiver IO s'est imposé comme un compromis optimal entre performance et légèreté, offrant des capacités de reconstruction et de détection d'anomalies proches des modèles plus lourds, tout en restant compatible avec les limitations des microcontrôleurs cibles. Son architecture attentionnelle, couplée à des mécanismes de compression et de détection unifiés, a permis de réduire significativement la dimensionnalité des données et les besoins en mémoire, ouvrant la voie à des applications de maintenance prédictive en temps réel.

L'intégration de la détection d'anomalies dans le modèle de compression permet une décision au plus tôt dans la chaîne de traitement des signaux vibratoires tout en évitant l'utilisation d'un modèle de classification annexe, ce qui accélère les calculs, diminue les exigences de mémoire et réduit la consommation énergétique du dispositif embarqué.

Enfin, l'intégration d'un modèle d'estimation de la dégradation, basé sur un apprentissage contrastif équilibré, a confirmé la capacité à distinguer les comportements normaux des déviations significatives. Cependant, ces travaux ont également révélé des défis persistants, notamment la sensibilité des indicateurs aux variations environnementales, comme la température. Ces observations soulignent l'importance de poursuivre la collecte de données sur des périodes plus longues et dans des conditions variées, afin d'affiner la robustesse des modèles face aux perturbations externes.

5.2 Perspectives

Pour conclure cette thèse, on propose ici différentes perspectives de poursuite des recherches menées sur les modèles présentés dans les sections précédentes.

5.2.1 Quantification

Les optimisations présentées dans la première contribution (Section 3) permettent une réduction drastique de la taille du modèle embarqué. Néanmoins, dans le cas où ces optimisations ne suffisent pas, il est possible de quantifier les poids du modèle afin de réduire davantage l'espace mémoire consommé.

La quantification consiste à réduire la précision des représentations numériques des poids d'un modèle d'apprentissage profond, généralement en passant de formats à virgule flottante (*float32* par exemple) à des formats à moindre précision (*int8* ou autre). Cette technique vise à diminuer la taille du modèle, à accélérer les calculs et à réduire la consommation énergétique, tout en maintenant une performance acceptable. La quantification est particulièrement utile pour déployer des modèles sur des dispositifs à ressources limités tels que les capteurs embarqués ou les systèmes IoT, où l'efficacité computationnelle et la consommation d'énergie sont cruciales.

La quantification peut aussi être appliquée aux données en entrée et/ou en sortie du modèle. L'utilité dans notre application est de réduire la taille en mémoire des spectrogrammes afin de disposer de plus de mémoire vive pour le reste du logiciel, d'accélérer les calculs, mais également d'améliorer les performances de reconstruction de notre modèle.

5.2.2 Apprentissage en ligne

Comme présenté dans la première contribution (Section 3), l'implémentation d'un algorithme d'apprentissage en ligne au sein du capteur permettrait d'améliorer les

performances de reconstruction, notamment sur des trains aux motifs rares ou uniques comme les trains de marchandises.

L'apprentissage en ligne (*online learning*) désigne une méthode dans laquelle le modèle est continuellement mis à jour à mesure que de nouvelles données arrivent, plutôt que d'être entraîné en une seule fois sur un jeu de données fixe. Cette approche permet au modèle de s'adapter en temps réel aux évolutions du système qu'il surveille, en intégrant progressivement les nouvelles observations dans son processus d'apprentissage. Elle est particulièrement utile dans les contextes où les données sont dynamiques, non stationnaires ou difficiles à collecter en masse à l'avance.

L'utilisation d'un tel algorithme n'a pas été possible en raison des contraintes matérielles de l'équipement embarqué, mais il est potentiellement possible d'adapter son utilisation pour l'appliquer du côté décodage de notre auto-encodeur. En effectuant la rétropropagation sur un modèle inversé où le décodeur est chaîné à un encodeur fixe identique à celui porté sur les capteurs, il pourrait être possible d'améliorer les reconstructions en comparant les latents. Si cela s'avérait impossible, cette comparaison permettrait au moins d'évaluer approximativement la perte subite par la donnée vibratoire à cause du modèle de compression.

5.2.3 Réseaux de capteurs

La détection d'anomalie présentée dans la seconde contribution (Section 4) peut être parasitée par des défauts liés au train exciteur. Ces défauts uniques à chaque train peuvent être identifiés par l'introduction d'un réseau de capteurs propageant les caractéristiques des signaux vibratoires mesurés. La présence d'une anomalie sur un ensemble de signaux provenant du même convoi conforte l'idée selon laquelle la cause de la détection n'est pas lié aux systèmes d'aiguillage ou aux capteurs eux-mêmes, mais bien à l'unique train exciteur.

Les réseaux de capteurs, également connus sous le nom de réseaux de capteurs sans fil, sont des systèmes composés de nombreux dispositifs de capteurs autonomes qui collaborent pour surveiller et interagir avec leur environnement. Ces réseaux sont devenus un élément clé dans de nombreuses applications, allant de la surveillance environnementale à la domotique, en passant par la santé et l'industrie. Leur capacité à collecter, traiter et transmettre des données en temps réel les rend indispensables dans le monde moderne.

Un réseau de capteurs est généralement composé de plusieurs nœuds capteurs, d'un ou plusieurs nœuds puits et d'un système de gestion central. Chaque nœud capteur est équipé de divers capteurs pour mesurer des paramètres physiques tels que la température, l'humidité, la pression, le mouvement, etc. Ces nœuds sont éga-

lement équipés d'un microcontrôleur pour traiter les données, d'une mémoire pour stocker les informations et d'un module de communication sans fil pour transmettre les données. Les capteurs collectent des données de leur environnement immédiat. Ces données peuvent être de nature diverse, allant de mesures environnementales à des informations sur l'état de machines industrielles. Les données collectées sont traitées localement par le microcontrôleur du nœud capteur. Ce traitement peut inclure des opérations de filtrage, d'agrégation et de compression pour réduire la quantité de données à transmettre. Les données traitées sont ensuite transmises sans fil à un nœud puits. Les nœuds capteurs utilisent des protocoles de communication sans fil tels que Zigbee, Bluetooth Low Energy (BLE), ou LoRa pour transmettre les données. Ces protocoles sont conçus pour être économe en énergie, ce qui est crucial pour les nœuds capteurs qui fonctionnent souvent sur batterie. Le nœud puits collecte les données de tous les nœuds capteurs et les transmet à un système de gestion central. Ce système peut être un serveur local ou un service cloud. Les données sont ensuite analysées pour en extraire des informations utiles. Des techniques d'analyse de données et d'intelligence artificielle peuvent être utilisées pour détecter des motifs, des anomalies ou des tendances. En fonction des résultats de l'analyse, des actions peuvent être déclenchées. Par exemple, dans un système de surveillance environnementale, une alerte peut être envoyée si des niveaux de pollution dangereux sont détectés. Dans un système de domotique, les lumières peuvent être allumées ou éteintes en fonction de la présence de personnes dans une pièce.

Les réseaux de capteurs ont une large gamme d'applications dans divers domaines. Ils sont largement utilisés pour surveiller l'environnement. Ils peuvent être déployés pour surveiller la qualité de l'air, la qualité de l'eau, les niveaux de bruit, et d'autres paramètres environnementaux. Par exemple, dans les forêts, les réseaux de capteurs peuvent être utilisés pour détecter les incendies de forêt à un stade précoce, permettant une intervention rapide. Dans l'agriculture, les réseaux de capteurs peuvent être utilisés pour surveiller les conditions du sol, l'humidité, la température et d'autres paramètres qui affectent la croissance des cultures. Ces informations peuvent être utilisées pour optimiser l'irrigation, la fertilisation et d'autres pratiques agricoles, améliorant ainsi la productivité et réduisant les coûts. Dans le domaine de la santé, les réseaux de capteurs sont de plus en plus utilisés. Par exemple, des capteurs portables peuvent être utilisés pour surveiller en continu les signes vitaux des patients, tels que la fréquence cardiaque, la pression artérielle et les niveaux de glucose dans le sang. Ces données peuvent être transmises à des professionnels de la santé pour une surveillance à distance et une intervention rapide en cas de besoin. Dans l'industrie, les réseaux de capteurs sont utilisés pour la surveillance des machines et des processus de fabrication. Ils peuvent détecter des anomalies

dans le fonctionnement des machines, prévoir les pannes et optimiser les processus de production. Cela permet de réduire les temps d'arrêt, d'améliorer la qualité des produits et d'augmenter l'efficacité globale. Les réseaux de capteurs sont au cœur des systèmes de domotique et des bâtiments intelligents. Ils permettent de surveiller et de contrôler divers aspects d'un bâtiment, tels que l'éclairage, le chauffage, la ventilation et la climatisation (CVC), la sécurité et la consommation d'énergie. Par exemple, des capteurs de mouvement peuvent être utilisés pour allumer ou éteindre les lumières en fonction de la présence de personnes dans une pièce, tandis que des capteurs de température peuvent être utilisés pour réguler le chauffage et la climatisation. Dans le domaine du transport et de la logistique, les réseaux de capteurs sont utilisés pour surveiller l'état des véhicules, des marchandises et des infrastructures. Par exemple, des capteurs peuvent être utilisés pour surveiller la température et l'humidité dans les conteneurs de transport de marchandises périssables, assurant ainsi leur qualité et leur sécurité. Dans les véhicules, des capteurs peuvent être utilisés pour surveiller l'état des moteurs, des freins et d'autres composants critiques.

Malgré leurs nombreux avantages, les réseaux de capteurs font face à plusieurs défis et limites. Les nœuds capteurs fonctionnent souvent sur batterie et doivent être économe en énergie pour prolonger leur durée de vie. La consommation d'énergie est un défi majeur, en particulier pour les applications où les nœuds sont déployés dans des environnements difficiles à atteindre pour le remplacement des batteries. La communication sans fil dans les réseaux de capteurs peut être affectée par divers facteurs, tels que les interférences, les obstacles physiques et les conditions environnementales. Cela peut entraîner des pertes de données et des retards dans la transmission des informations. Les réseaux de capteurs sont souvent déployés dans des environnements sensibles et collectent des données potentiellement confidentielles. La sécurité et la vie privée sont donc des préoccupations majeures. Les données doivent être protégées contre les accès non autorisés et les attaques malveillantes. Les réseaux de capteurs doivent être capables de s'adapter à des environnements dynamiques et de gérer un grand nombre de nœuds. La scalabilité est un défi majeur, en particulier pour les applications où le nombre de nœuds peut varier considérablement au fil du temps. Les réseaux de capteurs doivent souvent être intégrés avec d'autres systèmes et technologies. L'interopérabilité est un défi majeur, en particulier pour les applications où les réseaux de capteurs doivent interagir avec des systèmes hérités ou des technologies propriétaires.

Les réseaux de capteurs sont des systèmes puissants et polyvalents qui jouent un rôle crucial dans de nombreuses applications modernes. Leur capacité à collecter, traiter et transmettre des données en temps réel les rend indispensables dans des

domaines tels que la surveillance environnementale, l'agriculture de précision, la santé, l'industrie, la domotique et le transport. Cependant, ils font face à plusieurs défis, notamment en matière de consommation d'énergie, de communication sans fil, de sécurité, de scalabilité et d'interopérabilité. Malgré ces défis, les réseaux de capteurs continuent d'évoluer et de s'améliorer, ouvrant la voie à de nouvelles applications et innovations.

Une possibilité d'implémentation d'un réseau de capteurs pour cette problématique utiliserai un réseau de neurones à graphes (GNN : *Graph Neural Network*) dont les nœuds représentent les capteurs. Au sein du graphe, chaque capteur est connecté à l'ensemble des capteurs voisins, i.e. il existe une voie reliant directement les deux systèmes d'aiguillages concernés sans appareil de voie intermédiaire équipé d'un capteur. Lorsqu'un capteur détecte et mesure le signal vibratoire d'un train, les caractéristiques (la signature) de ce dernier est transmis aux capteurs adjacents. Si l'un de ces capteurs détecte et mesure un signal vibratoire, ce dernier peut comparer la signature de ce dernier avec la ou les signatures reçus de ses voisins, potentiellement identifier un train parmi ces dernières et adapter sa détection d'anomalie (ou tout autre fonction implémentée au sein du capteur) en conséquence.

Bibliographie

AlShorman, Omar, Muhammad Irfan, Nordin Saad, et al. 2020. « A review of artificial intelligence methods for condition monitoring and fault diagnosis of rolling element bearings for induction motor ». *Shock and vibration* 2020 (1) : 8843759.

Brandt, Anders. 2023. *Noise and vibration analysis : signal analysis and experimental procedures*. John Wiley & Sons.

Candès, Emmanuel J., et Michael B. Wakin. 2008. « An introduction to compressive sampling ». *IEEE signal processing magazine* 25 (2) : 21-30.

Chen, Han-Yun, et Ching-Hung Lee. 2020. « Vibration signals analysis by explainable artificial intelligence (XAI) approach : application on bearing faults diagnosis ». *IEEE Access* 8 : 134246-56.

———. 2021. « Deep learning approach for vibration signals applications ». *Sensors* 21 (11) : 3929.

Choi, Jong Hyun, Sung Kyu Jang, Woon Hyung Cho, Seokbae Moon, et Hyeongkeun Kim. 2024. « Motor PHM on Edge Computing with Anomaly Detection and Fault Severity Estimation through Compressed Data Using PCA and Autoencoder ». *Machine Learning and Knowledge Extraction* 6 (3) : 1466-83.

Chu, Thuy, Tan Nguyen, Hyunsang Yoo, et Jihoon Wang. 2024. « A review of vibration analysis and its applications ». *Heliyon*.

Dolenc, Boštjan, Pavle Boškoski, et Dani Juričić. 2016. « Distributed bearing fault diagnosis based on vibration analysis ». *Mechanical Systems and Signal Processing* 66 : 521-32.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, et al. 2020. « Generative adversarial networks ». *Commun. ACM* (New York, NY, USA) 63 (11) : 139-44. https://doi.org/10.1145/3422622.

Griffin, Daniel, et Jae Lim. 1984. « Signal estimation from modified short-time Fourier transform ». *IEEE Transactions on acoustics, speech, and signal processing* 32 (2) : 236-43.

Hamadache, Moussa, Saikat Dutta, Osama Olaby, Ramakrishnan Ambur, Edward Stewart, et Roger Dixon. 2019. « On the fault detection and diagnosis of railway switch and crossing systems : An overview ». *Applied Sciences* 9 (23) : 5129.

Hochreiter, Sepp, et Jürgen Schmidhuber. 1997. « Long short-term memory ». *Neural computation* 9 (8) : 1735-80.

Huang, Qingqing, Zonghua Li, Yan Han, et al. 2024. « Compressed sensing based on an improved K-SVD for vibration signal compression reconstruction in wireless sensor networks ». *IEEE Transactions on Instrumentation and Measurement*.

Hubel, D. H., et T. N. Wiesel. 1959. « Receptive fields of single neurones in the cat's striate cortex ». *The Journal of Physiology* 148 (3) : 574-91. https ://doi.org/ht tps://doi.org/10.1113/jphysiol.1959.sp006308.

Jaegle, Andrew, Sebastian Borgeaud, Jean-Baptiste Alayrac, et al. 2021. « Perceiver io : a general architecture for structured inputs & outputs ». *arXiv preprint arXiv :2107.14795*.

Jaegle, Andrew, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, et Joao Carreira. 2021. « Perceiver : General perception with iterative attention ». *International conference on machine learning*, 4651-64.

Jang, Jun-Gyo, Chun-Myoung Noh, Sung-Soo Kim, Sung-Chul Shin, Soon-Sup Lee, et Jae-Chul Lee. 2023. « Vibration data feature extraction and deep learning-based preprocessing method for highly accurate motor fault diagnosis ». *Journal of Computational Design and Engineering* 10 (1) : 204-20.

Kassa, Elias, Jan Saramota, et Amund Skavhaug. 2021. « Monitoring of switches and crossings/tracks using smart sensors ». *Intelligent Quality Assessment of Railway Switches and Crossings*, 149-66.

Le Cun, Y., B. Boser, J. S. Denker, et al. 1989. « Handwritten digit recognition with a back-propagation network ». *Proceedings of the 3rd International Conference on Neural Information Processing Systems* (Cambridge, MA, USA), NIPS'89, 396-404.

LeCun, Yann, Patrick Haffner, Léon Bottou, et Yoshua Bengio. 1999. « Object recognition with gradient-based learning ». In *Shape, contour and grouping in computer vision*, 319-45. Springer.

Lecun, Yann, Larry Jackel, Corinna Cortes, et al. 2000. « Learning Algorithms For Classification : A Comparison On Handwritten Digit Recognition ». *The Statistical Mechanics Perspective*, juillet.

Li, Xinbin, Yajun Xu, Jing Liu, et al. 2024. « Vibration analysis of the propulsion shaft system considering dynamic misalignment in the outer ring ». *Journal of Sound and Vibration* 589 : 118612.

Lin, Wan-Ju, Shih-Hsuan Lo, Hong-Tsu Young, et Che-Lun Hung. 2019. « Evaluation of deep learning neural networks for surface roughness prediction using vibration signal analysis ». *Applied Sciences* 9 (7) : 1462.

Liu, Ruonan, Boyuan Yang, Enrico Zio, et Xuefeng Chen. 2018. « Artificial intelligence for fault diagnosis of rotating machinery : a review ». *Mechanical Systems and Signal Processing* 108 : 33-47.

McCulloch, Warren S., et Walter Pitts. 1943. « A logical calculus of the ideas immanent in nervous activity ». *The Bulletin of Mathematical Biophysics* 5 (4) : 115-33. https://doi.org/10.1007/BF02478259.

Peixin, GAO, YU Tao, Yuanlin Zhang, WANG Jiao, et ZHAI Jingyu. 2021. « Vibration analysis and control technologies of hydraulic pipeline system in aircraft : a review ». *Chinese Journal of Aeronautics* 34 (4) : 83-114.

Pruša, Zdenek, et Peter L. Søndergaard. 2016. « Real-time spectrogram inversion using phase gradient heap integration ». *Proc. Int. Conf. Digital Audio Effects (DAFx-16)*, 17-21.

Rosenblatt, Frank. 1958. « The perceptron : a probabilistic model for information storage and organization in the brain. » *Psychological review* 65 (6) : 386.

Rumelhart, David E., Geoffrey E. Hinton, et Ronald J. Williams. 1986. « Learning representations by back-propagating errors ». *nature* 323 (6088) : 533-36.

Wang, Chuanhao, Yongjian Sun, et Xiaohong Wang. 2024. « Image deep learning in fault diagnosis of mechanical equipment ». *Journal of Intelligent Manufacturing* 35 (6) : 2475-515.

Wang, Xin, Dongxing Mao, et Xiaodong Li. 2021. « Bearing fault diagnosis based on vibro-acoustic data fusion and 1D-CNN network ». *Measurement* 173 : 108518.

Wang, Zhiwei, Yao Cheng, Guiming Mei, Weihua Zhang, Guanhua Huang, et Zhonghui Yin. 2020. « Torsional vibration analysis of the gear transmission system of high-speed trains with wheel defects ». *Proceedings of the Institution of Mechanical Engineers, Part F : Journal of Rail and Rapid Transit* 234 (2) : 123-33.

Weinberger, Kilian Q., et Lawrence K. Saul. 2009. « Distance metric learning for large margin nearest neighbor classification ». *Journal of Machine Learning Research* 10 (9) : 207-44. http://jmlr.org/papers/v10/weinberger09a.html.

Yiakopoulos, Christos T., Konstantinos C. Gryllias, et Ioannis A. Antoniadis. 2011. « Rolling element bearing fault detection in industrial environments based on a K-means clustering approach ». *Expert Systems with Applications* 38 (3) : 2888-911.

Yin, Yuhua, Zhiliang Liu, Qiang Zhang, Yong Qin, et Mingjian Zuo. 2024. « A Data Compression Method With an Encryption Feature for Safe and Lightweight Vibration Condition Monitoring ». *IEEE Internet of Things Journal*.

Zhang, Bin, Shaohui Zhang, et Weihua Li. 2019. « Bearing performance degradation assessment using long short-term memory recurrent network ». *Computers in Industry* 106 : 14-29.

Zhang, Wei, Gaoliang Peng, Chuanhao Li, Yuanhang Chen, et Zhujun Zhang. 2017. « A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals ». *Sensors* 17 (2) : 425.

Zhang, Yuyan, Xinyu Li, Liang Gao, Wen Chen, et Peigen Li. 2020. « Ensemble deep contractive auto-encoders for intelligent fault diagnosis of machines under noisy environment ». *Knowledge-Based Systems* 196 : 105764.

Zhu, Zhiqin, Yangbo Lei, Guanqiu Qi, et al. 2023. « A review of the application of deep learning in intelligent fault diagnosis of rotating machinery ». *Measurement* 206 : 112346.

Guillaume LEBONVALLET

Doctorat : Optimisation et Sûreté des Systèmes

Année 2025

Intelligence artificielle embarquée pour la surveillance des infrastructures ferroviaires

Cette thèse s'inscrit dans le cadre du projet RailMon, porté par l'entreprise Vossloh et financé par Bpifrance dans le cadre du Plan d'Investissement d'Avenir (PIA4). L'objectif du projet est de développer un système de télésurveillance intelligente permettant le suivi des infrastructures ferroviaires, afin d'anticiper les besoins en maintenance et d'optimiser les opérations de maintenance. Pour ce faire, des accéléromètres connectés et autonomes en énergie ont été déployés sur les aiguillages ferroviaires, éléments particulièrement sensibles du réseau afin d'enregistrer les signaux vibratoires générés lors du passage des trains. Nous proposons un modèle léger et efficace basé sur Perceiver IO, un autoencodeur modulaire exploitant les mécanismes d'attention, afin de compresser les signaux sur ces capteurs aux ressources limitées, puis de les reconstruire après transmission. Nous intègrerons également en son sein une capacité de détection précoce des signes de dégradation liés à l'usure des infrastructures, tout en préservant la légèreté du traitement embarqué. Enfin, nous introduisons un indicateur de vitesse de dégradation des aiguillages, fondé sur un apprentissage non supervisé sous contraintes, permettant d'évaluer l'évolution de leur état à partir des signaux mesurés.

Mots clés : apprentissage non supervisé (intelligence artificielle) – réseaux neuronaux (informatique) – données, compression (télécommunications) – maintenance conditionnelle – systèmes embarqués (informatique) – trains, vibrations.

Embedded Artificial Intelligence for Railways Infrastructures Monitoring

This thesis is part of the RailMon project, led by the company Vossloh and funded by Bpifrance as part of the Future Investment Plan (PIA4). The goal of the project is to develop an intelligent remote monitoring system for railway infrastructure, aimed at anticipating maintenance needs and optimizing maintenance operations. To achieve this, energy-autonomous connected accelerometers have been deployed on railway switches, which are particularly sensitive elements of the network, to record the vibratory signals generated during train passages. We propose a lightweight and efficient model based on Perceiver IO, a modular autoencoder leveraging attention mechanisms, to compress signals from these resource-constrained sensors and then reconstruct them after transmission. The system will also include an early detection capability for signs of degradation associated with infrastructure wear, while maintaining lightweight on-board processing. Finally, we introduce a deterioration speed indicator for railway switches, based on constrained unsupervised learning, which allows for the assessment of their condition evolution from the measured signals.

Keywords: unsupervised learning (machine learning) – neural networks (computer science) – data compression (telecommunication) – conditional maintenance – embedded computer systems – railroads, trains, vibration.