



HAL
open science

Exploitation des algorithmes d'apprentissage profond et des séries temporelles d'images satellitaires pour la prédiction de la déforestation

Waytehad Rose Moskolai

► **To cite this version:**

Waytehad Rose Moskolai. Exploitation des algorithmes d'apprentissage profond et des séries temporelles d'images satellitaires pour la prédiction de la déforestation. Traitement du signal et de l'image [eess.SP]. Université Bourgogne Franche-Comté; Université de Maroua, 2022. Français. NNT : 2022UBFCK113 . tel-04121626

HAL Id: tel-04121626

<https://theses.hal.science/tel-04121626v1>

Submitted on 8 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Cotutelle de thèse de Doctorat
UNIVERSITÉ BOURGOGNE FRANCHE-COMTE
UNIVERSITÉ DE BOURGOGNE, France
UNIVERSITE DE MAROUA, Cameroun

Ecole doctorale n°37 - SPIM : Sciences pour l'Ingénieur et Microtechniques
Unité de Formation Doctorale : Sciences de l'Ingénieur

Laboratoires d'accueil :

ImVia (Imagerie et Vision Artificielle), Université de Bourgogne
LaRI (Laboratoire de Recherche en Informatique), Université de Maroua

Doctorat en Informatique

Par

Waytehad Rose MOSKOLAÏ

Exploitation des algorithmes d'apprentissage profond et des séries temporelles d'images satellitaires pour la prédiction de la déforestation

Thèse présentée et soutenue à Dijon, le 16 décembre 2022

Composition du Jury :

Président :	M. Franck MARZANI	Professeur des universités, Université de Bourgogne
Rapporteurs :	Mme Saïda BOUAKAZ	Professeure des universités, Université Claude Bernard Lyon 1
	M. Olivier MONGA	Directeur de recherche, Institut de Recherche pour le Développement, Paris
	M. Clémentin TAYOU DJAMENI	Professeur des universités, Université de Dschang
Examineur :	M. Paul DAYANG	Maître de conférences, Université de Ngaoundéré
Direction de thèse :	M. Albert DIPANDA	Professeur des universités, Université de Bourgogne
	M. KOLYANG	Professeur des universités, Université de Maroua
	M. Wahabou ABDOU	Maître de conférences, Université de Bourgogne

Dédicace

A la mémoire de mon père, parti trop tôt...

Résumé

Ces dernières années, nous assistons à une émergence des méthodes d'apprentissage profond appelées Deep Learning en anglais (DL), qui ont favorisé d'énormes progrès dans divers domaines. Le succès de ces méthodes résulte de la disponibilité sans cesse croissante de grandes quantités d'informations ainsi que des puissances de calcul des ordinateurs. Dans le domaine de la télédétection, on dispose aujourd'hui de volumes considérables d'images satellites grâce au nombre élevé de satellites d'observation de la Terre qui gravitent autour de la planète. Cette disponibilité en images permet de constituer des séries chronologiques de données appelées séries temporelles d'images satellite (STIS). Les STIS peuvent être utilisées pour de multiples applications du monde réel comme la prédiction des changements des classes d'occupation des sols, notamment la déforestation. Dans le cadre de cette thèse, il est question d'exploiter les potentialités des algorithmes d'apprentissage profond et la disponibilité des STIS pour créer des modèles basés sur l'apprentissage profond, capables d'analyser les données historiques d'une zone donnée et de prédire la déforestation dans cette zone. Quatre contributions sont apportées pour ces travaux de thèse. Premièrement, nous définissons un workflow de collecte et de prétraitement par lot des images satellite Sentinel-1 qui nous a permis de mettre à la disposition de la communauté scientifique trois jeux de données d'apprentissage. Deuxièmement, pour déterminer quelle architecture DL est la plus adaptée pour la prédiction des STIS, nous évaluons trois variantes d'architectures LSTM (ConvLSTM, CNN-LSTM, Stack-LSTM) pour la prédiction de la prochaine occurrence dans une séquence d'images. Ensuite, nous montrons par une étude comparative, que les méthodes DL permettent d'obtenir de meilleurs résultats que la méthode classique CA-Markov pour la prédiction des changements des classes d'occupation des sols. Pour finir, nous proposons un modèle basé sur l'architecture hybride CNN-LSTM pour prédire la déforestation autour de la réserve de Biosphère du Dja (Cameroun) en 2025 comme cas d'utilisation. Le modèle entraîné sur des images Sentinel-1A a permis d'estimer une hausse approximative de 6% de déforestation dans cette zone. Pour cette contribution, nous définissons également une stratégie d'augmentation des données permettant d'améliorer les performances du modèle.

Mots clés : Deep learning, Apprentissage Profond, Réseaux de Neurones Artificiels, Analyse prédictive, ConvLSTM, CNN-LSTM, STIS, Déforestation.

Abstract

In recent years, we have witnessed the emergence of deep learning (DL) methods, which have led to enormous progress in various fields. The success of these methods is due to the ever-increasing availability of large amounts of information and the computing power of computers. In the field of remote sensing, we now have considerable volumes of satellite images thanks to the large number of Earth observation satellites orbiting the planet. This availability of images allows to create time series of data called satellite image time series (STIS). STIS can be used for multiple real-world applications such as the prediction of land use and land cover changes, including deforestation. In this thesis, the aim is to exploit the potential of deep learning algorithms and the availability of STIS to create deep learning based models capable of analyzing historical data of a given area and predicting deforestation in that area. Four contributions are made for this thesis. First, we define a workflow for the collection and batch preprocessing of Sentinel-1 satellite images that allowed us to make available to the scientific community three training datasets. Second, to determine which DL architecture is best suited for STIS prediction, we evaluate three variants of LSTM architectures (ConvLSTM, CNN-LSTM, Stack-LSTM) for predicting the next occurrence in an image sequence. Then, we show by a comparative study that DL methods allow to obtain better results than the classical CA-Markov method for land use changes prediction. Finally, we propose a model based on the hybrid CNN-LSTM architecture to predict deforestation around the Dja Biosphere Reserve (Cameroon) in 2025 as a use case. The model trained on Sentinel-1A images allowed us to estimate an approximate 6% increase in deforestation in this area. For this contribution, we also define a data augmentation strategy to improve the model performance.

Keywords : Deep learning, Artificial neural networks, Predictive Analysis, ConvLSTM, CNN-LSTM, STIS, Deforestation.

Table des matières

Dédicace	i
Résumé	iii
Abstract	iv
Table des matières	iv
Liste des algorithmes	ix
Liste des figures	x
Liste des tableaux	xii
Acronymes	xiii
Introduction	1
1 Présentation des concepts et outils	7
1.1 Images satellites	8
1.1.1 Généralités	8
1.1.2 Images optiques / images radar	9
1.1.2.1 Images optiques	9
1.1.2.2 Images radar	10
1.1.3 Notions de résolution	13
1.2 Apprentissage profond	15
1.2.1 Généralités	15
1.2.2 Fonctionnement	18
1.2.3 Principales architectures deep learning	19
1.2.3.1 Réseaux de neurones feedforward	19
1.2.3.2 Réseaux de neurones à convolution	20
1.2.3.3 Réseaux de neurones récurrents	22
1.2.4 Amélioration des modèles deep learning	23
1.2.4.1 Augmentation des données	23
1.2.4.2 Choix des hyperparamètres	24
1.2.4.3 Choix de la fonction d'optimisation	24
1.3 Prédiction des séries temporelles	24
1.3.1 Définitions des concepts	24
1.3.2 Méthodes de prédiction	26

1.3.3	Méthode CA-Markov	27
2	Etat de l'art sur l'utilisation du DL pour la prédiction des STIS	29
2.1	Description des problèmes de prédiction et leurs applications	30
2.1.1	Description des problèmes de prédiction	30
2.1.2	Principales applications	30
2.1.2.1	Prévisions météorologiques	30
2.1.2.2	Prédiction des précipitations	31
2.1.2.3	Analyse spatio-temporelle	31
2.1.2.4	Reconstruction des données manquantes	32
2.2	Méthodes deep learning pour la prédiction des séries temporelles d'images satellites	33
2.2.1	Architectures basées sur les RNN	33
2.2.1.1	Réseaux LSTM ou longue mémoire à court terme	33
2.2.1.2	Réseaux TrajGRU	34
2.2.1.3	Réseaux ConvLSTM	34
2.2.2	Architectures hybrides	35
2.2.2.1	Réseaux LSTM-AdaBoost	35
2.2.2.2	Réseaux GAN-LSTM	36
2.2.2.3	Réseaux ConvLSTM-Auto-Encoder	37
2.2.3	Architectures basées sur les réseaux feedforward	38
2.2.3.1	Réseaux DeepstepFE	38
2.2.3.2	Réseaux CNN	38
2.3	Optimisation et métriques d'évaluation	39
2.3.1	Fonctions d'optimisation	39
2.3.1.1	Algorithmes SGD	40
2.3.1.2	Algorithme RMSProp	40
2.3.1.3	Algorithme Adam	40
2.3.2	Métriques d'évaluation	41
2.3.2.1	Métriques utilisées dans les problèmes de régression	41
2.3.2.2	Métriques utilisées dans les problèmes de vision par ordinateur	43
2.3.2.3	Métriques utilisées dans les problèmes statistiques	43
2.3.2.4	Métriques utilisées pour mesurer la performance	44
2.3.2.5	Métriques utilisées dans les problèmes de classification	44
2.3.2.6	Autres métriques	44
2.4	Limites et propositions de solutions	45
2.4.1	Manque de données d'entraînement	45
2.4.2	Opérations de prétraitement des STIS	45
2.4.3	Conception des architectures DL	47
2.4.4	Réutilisation des modèles	47
3	Constitution d'une base d'images à l'aide d'un workflow	49
3.1	Description des données Sentinel-1	50
3.1.1	Généralités	50
3.1.2	Modes d'acquisition	50
3.1.3	Types de produits	51
3.1.3.1	Produits de Niveau-0	51
3.1.3.2	Produits de Niveau-1	52
3.1.3.3	Produits de Niveau-2	52
3.2	Sites d'études	52
3.2.1	Réserve de Faune de Togodo	52

3.2.2	Parc National de Bouba-Ndjida	54
3.2.3	Réserve de Biosphère du Dja	56
3.3	Collecte et prétraitement	57
3.3.1	Collecte	57
3.3.2	Prétraitement	60
3.3.2.1	Logiciel SNAP	60
3.3.2.2	Workflow de prétraitement	61
3.3.3	Prétraitement automatisé par lot avec GPT	64
3.4	Résultats d'implémentation du workflow	66
3.4.1	Outils utilisés	66
3.4.2	Description des datasets constitués	66
4	Evaluation des architectures LSTM pour la prédiction des STIS	69
4.1	Description des architectures LSTM	70
4.1.1	Réseaux LSTM empilés	71
4.1.2	Réseaux ConvLSTM	72
4.1.3	Réseaux CNN-LSTM	73
4.2	Évaluation des architectures LSTM pour la prédiction des STIS	74
4.2.1	Formulation du problème et méthodologie	74
4.2.2	Structure des modèles et paramètres utilisés	75
4.2.2.1	Configuration des architectures	75
4.2.2.2	Hyperparamètres utilisés	76
4.2.3	Résultats obtenus	76
4.2.3.1	Métriques d'évaluation	77
4.2.3.2	Discussions	77
4.3	Comparaison des méthodes DL et CA-Markov	83
4.3.1	Formulation du problème et description des méthodes	83
4.3.1.1	Méthode CA-Markov	84
4.3.1.2	Méthode DL	85
4.3.2	Structure des modèles et hyperparamètres utilisés	85
4.3.2.1	Modèle CA-Markov	85
4.3.2.2	Modèles DL	86
4.3.3	Résultats obtenus	86
4.3.3.1	Métriques d'évaluation	87
4.3.3.2	Résultats	88
5	Cas d'usage du DL pour la prédiction des changements du couvert forestier	91
5.1	Introduction	92
5.2	Outils et méthodes	94
5.2.1	Données	94
5.2.1.1	Description de la zone d'étude	94
5.2.1.2	Préparation des données	95
5.2.1.3	Augmentation des données	96
5.2.2	Modèle	98
5.2.2.1	Architecture des couches	98
5.2.2.2	Régularisation	99
5.2.2.3	Optimisation des hyperparamètres	100
5.3	Résultats et discussions	101
5.3.1	Outils utilisés	101
5.3.2	Paramètres utilisés	102

5.3.3	Prédiction one-step et validation du modèle	104
5.3.4	Prédiction multi-step pour l'année 2025	108
5.4	Conclusion	109
	Conclusion et perspectives	111
	Liste des publications	115
	Bibliographie	117

Liste des Algorithmes

1	Algorithme général de rétropropagation pour l'optimisation des poids dans un réseau de neurones artificiels.	39
2	Algorithme d'optimisation stochastique Adam.	42
3	Algorithme Hyperband pour l'optimisation des hyperparamètres [151].	101

Liste des figures

1.1	Domaines du spectre électromagnétique [35].	9
1.2	Exemple de polarisations horizontale H (en bleu) et verticale V (en rouge) d'une onde électromagnétique plane.	11
1.3	Rétrodiffusion du signal radar dans un couvert forestier [40].	12
1.4	Comparaison d'images optique et radar d'une zone dans le Parc National de Bouba-Ndjida (Cameroun) : (a) Composition colorée RVB d'une image Sentinel-2B; (b) Composition colorée VV, VH, VH/VV d'une image Sentinel-1A.	13
1.5	Représentation du complexe Wimbledon sur des images avec différentes résolutions spatiales.	14
1.6	Apprentissage d'un système pour la reconnaissance des roses à partir d'exemples de photos (x) et d'étiquette (y).	16
1.7	Relation entre le DL, le ML et l'IA [50].	17
1.8	Représentation simplifiée d'un réseau de neurones artificiels.	18
1.9	Représentation d'un réseau de neurones <i>feedforward</i> à deux couches cachées.	20
1.10	Représentation graphique de la fonction ReLu.	21
1.11	Structure générale d'un réseau de neurones à convolution [53].	22
1.12	Structure générale d'un réseau de neurones récurrents [61].	22
1.13	Principe général d'augmentation de données.	23
1.14	Exemple de prédiction à une étape dans une série temporelle d'images satellites.	25
2.1	Approches de prévision météorologique décrite dans [75].	31
2.2	Image manquante dans une série temporelle d'images satellites.	33
2.3	Architecture de régression par arbre de décision utilisant AdaBoost proposée dans [91].	35
2.4	Configuration LSTM-AdaBoost simplifiée utilisée dans [91].	36
2.5	Architecture du réseau GAN proposé dans [93]	37
2.6	Vue d'ensemble de la structure encodeur-décodeur du réseau suggéré dans [96].	37
2.7	Architecture CNN de la méthode proposée dans [98].	38
2.8	Vue d'ensemble du processus de traitement des données de télédétection.	46
2.9	Aperçu des méthodes de <i>meta-learning</i> [111].	48
3.1	Couverture géographique du satellite Sentinel-1 [112].	50
3.2	Courbe ombrothermique du climat dans la zone de la RFT [115].	53
3.3	Localisation de la réserve de faune de Togodo.	54
3.4	Courbe ombrothermique du climat dans la zone du PNB [115].	55
3.5	Localisation du Parc National de Bouba-Ndjida.	55
3.6	Courbe ombrothermique du climat dans la zone de la RBD [115].	56
3.7	Localisation de la Réserve de Biosphère du Dja.	57

3.8	Convention de nommage des produits Sentinel-1 [112].	59
3.9	Architecture générale du logiciel SNAP.	60
3.10	Graphe de traitement des données à partir du GPF [121].	61
3.11	Workflow des opérations de prétraitement par lot.	62
3.12	Effets d'un filtre appliqué sur une image radar	63
3.13	Illustration des STIS constituées en bande VH pour chacune des trois zones d'étude (PNB, RBD et RFT).	67
4.1	Structure générale d'un simple bloc LSTM.	71
4.2	Architecture d'un réseau Stack-LSTM.	72
4.3	Bloc mémoire ConvLSTM [133].	73
4.4	Structure générale d'un réseau CNN-LSTM.	74
4.5	Évolution de la perte d'apprentissage (MAE) au fil des epochs en fonction du timestep (5 à 10).	78
4.6	Évolutions de la perte de validation (MAE) au fil des époques en fonction du timestep (5 à 10).	79
4.7	Évolution de la fonction MAE au fil des époques en fonction du timestep (5 à 10) pour le modèle ConvLSTM.	80
4.8	Résultats de prédiction sur des images (128 × 128). timestep = 5.	82
4.9	Résultats de prédiction sur des images (128 × 128). timestep = 10	83
4.10	Workflow général de comparaison des méthodes CA-Markov et DL.	84
4.11	Architecture d'un automate cellulaire intégré à la chaîne de Markov.	85
4.12	Résultat de la prédiction avec la méthode CA-Markov.	88
4.13	Images de la différence entre image prédite et image réelle (DI).	90
4.14	Résultat des prédictions à partir des modèles ConvLSTM et CNN-LSTM.	90
5.1	Répartition des plantations d'hévéas près de la RBD [144].	93
5.2	Représentation de la zone d'étude (RBD)	94
5.3	Courbes représentant les variations des valeurs des pixels au cours du temps.	95
5.4	Évolution de la fonction de coût avec le jeu de données d'entraînement et de validation dans un modèle qui fait du surapprentissage.	97
5.5	Augmentation des STIS par fusion d'images.	97
5.6	Architecture hybride CNN-LSTM proposée.	99
5.7	Modèle de réseau de neurones dropout.	100
5.8	Comparaison des valeurs pertes entre pour des combinaisons fonctions de coût + fonction d'optimisation	103
5.9	Évolution de l'erreur pendant l'apprentissage du modèle, avec et sans augmentation de données.	104
5.10	Évolution de la fonction de coût RMSE (apprentissage et validation) pendant l'entraînement du modèle.	105
5.11	Images des classes d'occupation des sols dans la zone d'étude	106
5.12	Séquence d'images utilisée comme entrée pour la prédiction de l'image de 2022.	107
5.13	Résultat de la prédiction du modèle	107
5.14	Principe de la prédiction multi-étapes.	108
5.15	Prédiction multi-step des images de 2023, 2024, 2025.	109

Liste des tableaux

1.1	Plages de longueurs d'ondes utilisées pour l'observation de la Terre	8
1.2	Principales bandes utilisées pour l'observation de la Terre en mode actif.	10
3.1	Paramètres de recherche.	60
3.2	Valeur des paramètres utilisés pour chaque opérateur.	65
3.3	Informations de téléchargement.	66
3.4	Proportions des données manquantes.	67
3.5	Principales caractéristiques des données prétraitées.	67
4.1	Une série temporelle univariée organisée pour un apprentissage supervisé.	75
4.2	Valeurs des métriques d'évaluation pour la zone RFT.	81
4.3	Valeurs des métriques d'évaluation la zone PNB.	81
4.4	Principales classes d'occupation de sol considérées.	86
4.5	Valeurs des hyperparamètres - Méthode DL.	86
4.6	Surfaces des classes d'occupation de sol.	89
4.7	Valeurs des métriques utilisées pour comparer les modèles CA-Markov et DL.	89
5.1	Description des couches et paramètres du réseau CNN-LSTM.	99
5.2	Valeurs des hyperparamètres.	104
5.3	Valeurs des métriques d'évaluation. ACC=Précision.	105
5.4	Surface des classes d'occupation des sols en 2022.	107

Acronymes

AE	Auto-Encoder
ASF	Alaska Satellite Facility
CA	Cellular Automata
CC	coefficient de corrélation
CNN	Convolutional Neural Network
CSI	Critical Success Index
DEM	Digital Elevation Model
DL	Deep Learning
DNN	Deep Neural Network
DSN	Deep Stacking Network
EW	Extra Wide swath
FAR	False Alarm Rate
GAN	Generative Adversarial Network
GPF	Graph Processing Framework
GPU	Graphics Processing Unit
GRD	Ground Range Detected
GRU	Gated recurrent unit
HSS	Heidke Skill Score
IR	Infrarouge
IW	Interferometric Wide swath
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error

ML	Machine Learning
MLP	Multi Layer Perceptron
MNT	Modèle numérique de Terrain
MSE	Mean Squared Error
NDVI	Normalized Difference Vegetation Index
NMBE	Normalized Mean Bias Error
NRMSE	Normalized Root Mean Square Error
PIR	Proche Infrarouge
PNB	Parc National Bouba-Ndjida
PNT	Prévision Numérique du Temps
POD	Probability Of Detection
PSNR	Peak Signal to Noise Ratio
RBD	Réserve de Biosphère du Dja
RELU	Rectified Linear Unit
RFT	Réserve de Faune de Togodo
RMSE	Root Mean Square Error
RMSPROP	Root Mean Square Propagation
RNA	Réseau de Neurones artificiels
RNN	Recurrent Neural Network
RSO	Radar à Synthèse d'Ouverture
RVB	Rouge-vert-Bleu
S1A/S1B	Sentinel-1A/Sentinel-1B
SAM	Spectral Angle Mapper
SAR	Proche Infrarouge
SGD	Stochastic Gradient Descent
SNAP	SentiNel Application Platform
SOT	Satellite d'Observation de la Terre
SSIM	Structural Similarity
STIS	Série Temporelle d'images Satellites
WKT	Well-Known Text

Introduction

Tout en assurant la subsistance de millions de personnes dans le monde, les forêts constituent un rempart vital contre le changement climatique et la perte de biodiversité. Cependant, elles sont menacées. Au cours des 30 dernières années, environ 420 millions d’hectares de forêt ont été perdus dans le monde à cause de la déforestation, phénomène défini comme l’ensemble des pratiques et processus conduisant au recul irréversible des surfaces couvertes de forêts, au profit d’activités non forestières (indépendamment du fait qu’elle soit anthropique ou pas) [1]. L’étude par télédétection de l’évaluation des ressources forestières mondiales montre que la principale cause de la déforestation est l’expansion des terres cultivées, surtout dans les pays les moins développés [2]. Si le rythme de disparition de nos forêts a ralenti de près de 30% entre les années 2000 et 2020, les forêts pluviales tropicales sont toujours très menacées, que ce soit par le pâturage du bétail en Amérique du Sud, ou par l’expansion des terres cultivées (plantations de palmiers à huile par exemple) en Asie. Sur le plan géographique, l’Amérique du Sud et l’Afrique sont les régions qui ont connu les déforestations les plus fortes sur la période 2000-2020 (68 et 49 millions d’hectares respectivement) [3].

La déforestation a des répercussions majeures sur l’humanité et engendre d’énormes pertes économiques. En effet, la forêt n’est pas que du bois, c’est aussi le deuxième puits de carbone derrière les océans. C’est également un refuge pour la biodiversité (80% de la biodiversité terrestre se trouve en forêt) et un filtre pour l’eau et l’air [2]. Il est donc primordial de préserver les forêts et de lutter contre la déforestation.

Stopper la déforestation d’ici 2030 est en effet l’une des résolutions adoptées lors de la conférence internationale sur le climat COP26. Cette rencontre, qui réunit chaque année les pays signataires de la Convention Cadre des Nations Unies sur le Changement Climatique (CCNUCC), s’est tenue à Glasgow du 1er au 12 novembre 2021. Pour atteindre cet objectif, l’une des solutions à mettre en œuvre est l’imposition des législations sur la protection des zones forestières qui présentent un risque de déforestation élevé. Cela peut se faire à travers des modèles prédictifs capables de prévoir le taux de déforestation dans une zone après un certain temps. Établir des scénarios de déforestation est ainsi utile pour permettre une anticipation de l’avenir et une orientation de prise de décision par les politiques publiques autour des questions de protection et de conservation.

S’il est difficile de prédire l’avenir avec exactitude, il existe cependant des méthodes basées sur des concepts scientifiques, permettant d’estimer les tendances de certains événements qui

n'ont pas encore eu lieu. L'ensemble de ces méthodes porte le nom d'analyse prédictive. Elle fait référence à un domaine scientifique qui consiste à utiliser diverses techniques statistiques et des algorithmes particuliers pour prédire des événements, en analysant les informations passées et actuelles d'un ensemble de données chronologiques [4].

Les données de télédétection (les images satellites), utilisées par des méthodes statistiques et d'apprentissage automatique, ont largement été exploitées pour prédire les phénomènes de déforestation [5, 6, 7, 8, 9, 10]. En particulier, les travaux basés sur les chaînes de Markov pour la prédiction de la déforestation sont très populaires dans la littérature. On note par exemple les travaux publiés par Gosh et al., Tadese et al., Gharaibeh et al., Abijith et al., Liping et al. dans [11, 12, 13, 14, 15]. Cependant, bien que produisant des résultats assez satisfaisants, ces méthodes ont présenté de nombreuses limites qui influent sur la qualité des précisions obtenues. En effet, cette approche requiert des étapes supplémentaires telles que l'identification des données d'une même saison, la production des cartes thématiques par classification d'images, le choix et l'inclusion des données auxiliaires, dont dépendent les performances du modèle [11, 12, 13, 14, 15]. De plus, les méthodes d'apprentissage automatique classiques présentent également des insuffisances dans la résolution des tâches complexes telles que la prédiction de la déforestation. L'intervention d'un expert du domaine des forêts ou de la télédétection est quasi indispensable pour la réalisation des tâches de prédiction, qui peuvent s'avérer très chronophages. Fort de cela, des outils plus efficaces ont donc été introduits depuis quelques années pour obtenir de meilleurs résultats dans le domaine de la télédétection, et ainsi répondre à l'évolution des besoins des utilisateurs : les algorithmes d'apprentissage profond ou algorithmes *Deep Learning* (DL) [16, 17, 18, 19, 20, 21].

Le DL est une branche de l'apprentissage automatique ou *Machine Learning* (ML), qui lui-même est un domaine d'étude de l'intelligence artificielle. Le ML permet à un ordinateur d'apprendre à partir d'exemples et de formuler des prévisions [20, 22]. Les idées à la base du DL ne sont pas nouvelles et datent des années 80, mais pourtant, le DL a commencé sa révolution il y a seulement quelques années suite au célèbre concours de reconnaissance d'images ImageNet, où des algorithmes DL ont eu les meilleurs scores. Depuis lors, l'apprentissage profond rencontre un succès remarquable dans de nombreux domaines tels que la robotique, la vision par ordinateur, le traitement de langage automatique, la médecine [23, 24, 25, 26, 27]. La quintessence de ces algorithmes réside dans leur capacité à détecter automatiquement les caractéristiques importantes à prédire et à traiter des données non structurées comme le son, le texte, les images, etc. Grâce à la disponibilité des données massives (*big data*) et à la disponibilité des puissances de calculs sans cesse croissantes, l'utilisation du DL en analyse prédictive a permis d'aboutir à d'excellentes conclusions dans de nombreux domaines [23, 24]. A partir des séries chronologiques de données, les algorithmes DL apprennent et mémorisent comment les changements se produisent au fil du temps pour ensuite estimer les tendances futures des événements. Une base de données d'entraînement constituée de longues séries permettent en général aux algorithmes de mieux faire l'apprentissage et d'obtenir de meilleures précisions [28].

On note également de nombreux progrès en ce qui concerne le domaine de la télédétection. La communauté dispose aujourd'hui de volumes considérables de données grâce au nombre élevé

de satellites d’observation de la Terre qui gravitent autour de la planète. En effet, selon la base de données satellitaire de *l’Union of Concerned Scientists*, plus de 4 000 satellites étaient opérationnels au 30 avril 2021 (dont environ 24% dédiés à l’observation de la Terre). Avec un temps de revisite des satellites au-dessus d’une zone de plus en plus court, il sera probablement bientôt possible d’obtenir quotidiennement et gratuitement des images satellites, de n’importe quelle zone dans le monde [29]. Par exemple, en 2001, l’Union européenne et l’Agence Spatiale Européenne (ESA) ont lancé un programme afin de doter l’Europe d’une capacité autonome d’observation et de surveillance de la Terre. Depuis 2014, ce programme appelé Copernicus fournit des données à partir d’une constellation de 6 familles de satellites appelée mission « Sentinel » (Sentinel-1 à Sentinel-6). Chacune de ces missions a son propre objectif et ambitionne de fournir des données gratuites accessibles sur Internet. Avec les images acquises, il est possible de constituer des séries chronologiques d’images représentant une même zone donnée. Ces séquences sont appelées Séries Temporelles d’Images Satellite (STIS) et sont utilisées pour de multiples applications du monde réel, telles que la classification, la segmentation, la détection d’anomalies et la prédiction [30, 31]. Compte tenu de cette disponibilité d’images, tant en quantité qu’en qualité, et des capacités des algorithmes DL, on s’attendrait à une utilisation massive des algorithmes DL pour la prédiction des dynamiques de sol tels que la déforestation. Malheureusement malgré les prouesses enregistrées, l’utilisation du DL pour la prédiction des STIS en général, et pour la prédiction de la déforestation en particulier, présente encore quelques limites qui freinent son essor [32]. On peut noter par exemple :

- la complexité des tâches de prétraitement des STIS conduisant au manque de bases de données d’entraînement ;
- la taille et la nature des images satellites ;
- la conception des architectures DL pour la prédiction des séries d’images satellites ;
- la généralisation des modèles de prédiction des STIS ;
- l’augmentation des données dans les STIS ;
- l’application des techniques telles que le *few shot learning*, le *transfer learning* ou le *meta learning*.

Face à ce contexte et aux différentes contraintes associées, l’enjeu ici est de pouvoir exploiter à la fois les potentialités du DL et la disponibilité des STIS pour améliorer la prévention de la déforestation à partir des modèles prédictifs.

L’objectif général de cette thèse est donc d’explorer les différentes pistes méthodologiques et de proposer des outils et méthodes d’apprentissage profond, qui analysent les images satellites à haute résolution, dans le but de prédire l’évolution future du couvert forestier. Détecter la déforestation dans une zone avant même qu’elle n’ait eue lieu permettrait aux décideurs d’anticiper l’application des mesures préventives et de protection des zones susceptibles de présenter un fort taux de déforestation.

Pour atteindre les différents objectifs fixés, plusieurs tâches ont été effectuées au cours de ces travaux. Les principaux résultats obtenus peuvent être répartis en quatre groupes correspondant aux quatre principales contributions de cette thèse :

- création d'un jeu de données constituée de trois STIS Sentinel-1 pour l'entraînement des algorithmes DL ;
- évaluation comparative de trois architectures DL pour la prédiction de la prochaine occurrence dans une séquence temporelle d'images Sentinel-1 ;
- validation des méthodes DL pour la prédiction des changements de classes d'occupation de sol par comparaison à la méthode utilisant les automates cellulaires intégrés aux chaînes de Markov (CA-Markov) ;
- proposition d'un modèle DL pour la prédiction de la déforestation autour de la réserve de Biosphère du Dja (Cameroun).

Le présent manuscrit est organisé autour de cinq chapitres :

Le chapitre 1 présente de façon générale les principaux concepts et outils mentionnés dans le cadre de cette thèse. Nous aborderons premièrement les notions générales sur les images satellites, les différents types d'images (optiques *vs* radar), et les principales caractéristiques de celles-ci. Deuxièmement, les concepts relatifs à l'apprentissage profond, ainsi que les différents types d'architectures DL et leur utilisation sont également évoqués. Enfin, la notion de prédiction des STIS est détaillée et l'une des principales méthodes de prédiction utilisée dans la littérature est présentée (méthode CA-Markov).

Le chapitre 2 est un état de l'art sur l'utilisation du DL pour la prédiction des STIS. Après une description sommaire des problèmes de prédiction en télédétection et des différentes applications décrites dans la littérature, les principales méthodes d'apprentissage, d'optimisation et d'évaluation des modèles DL sont présentés. A la fin de ce chapitre, les principales limites rencontrées dans ce domaine et les pistes de solutions proposées sont expliquées. Un article « État de l'art » a été publié dans une revue internationale pour les travaux menés dans cette partie.

Le chapitre 3 est consacré à la présentation des étapes de collecte et de prétraitement des données. Les images provenant des satellites Sentinel-1 utilisées dans le cadre de cette thèse sont présentées. De même, les zones d'étude ainsi que les méthodes de collecte et de prétraitement automatisées mises en place pour constituer les jeux de données d'entraînement sont décrites. Les caractéristiques des éléments constituant les séries d'images créées et mises à la disposition de la communauté scientifique sont également décrites dans ce chapitre. Ce chapitre correspond à la première contribution répondant au problème du manque de jeux de données d'entraînement et a donné lieu à la publication d'un article dans un journal international avec comité de lecture.

Le chapitre 4 examine des méthodes DL particulières, notamment les architectures *Long Short-Term Memory* (LSTM) pour la création des modèles prédictifs à partir des séries d'images satellites. Plus précisément, les résultats de deux expérimentations ayant permis l'évaluation comparative de ces architectures pour la résolution des tâches de prédiction sont présentés et discutés. La première expérimentation a permis d'évaluer trois types d'algorithmes LSTM et d'orienter le choix vers les architectures CNN-LSTM ou ConvLSTM pour la prédiction des STIS. Par la suite, une deuxième expérimentation a été menée pour comparer les performances

des méthodes DL et celles des méthodes utilisant les chaînes CA-Markov pour la prédiction des changements des classes d'occupation. Les résultats de ces deux expérimentations correspondent à la deuxième et la troisième contribution à ces travaux. Cette phase expérimentale a été valorisée par deux publications dans des conférences internationales.

Le chapitre 5 propose un modèle d'apprentissage profond pour la résolution d'un problème réel, en l'occurrence la prédiction de la déforestation autour de la réserve de biosphère du Dja (Cameroun). Le modèle prédictif proposé est conçu à partir d'une architecture hybride CNN-LSTM et amélioré grâce à une approche d'augmentation de données basée sur la fusion d'image. Après une description de la méthodologie et de la solution proposée, les résultats sont présentés et discutés. Ce chapitre correspond à la dernière contribution de nos travaux de thèse.

Pour finir, une conclusion ressortira les grandes lignes de notre contribution, les limites enregistrées au terme de cette thèse, ainsi que les perspectives envisagées pour les futurs travaux.

Chapitre 1

Présentation des concepts et outils

Objectifs

Pour mieux appréhender des notions importantes abordées dans ce manuscrit et faciliter leur compréhension, les concepts et outils clés utilisés seront présentés dans ce chapitre. Les éléments sur les images satellites, les algorithmes d'apprentissage profond ainsi que les prédictions des séries temporelles d'images satellites seront tour à tour décrits dans les sections qui suivent.

Sommaire

1.1	Images satellites	8
1.1.1	Généralités	8
1.1.2	Images optiques / images radar	9
1.1.3	Notions de résolution	13
1.2	Apprentissage profond	15
1.2.1	Généralités	15
1.2.2	Fonctionnement	18
1.2.3	Principales architectures deep learning	19
1.2.4	Amélioration des modèles deep learning	23
1.3	Prédiction des séries temporelles	24
1.3.1	Définitions des concepts	24
1.3.2	Méthodes de prédiction	26
1.3.3	Méthode CA-Markov	27

1.1 Images satellites

1.1.1 Généralités

La télédétection peut être définie comme l'ensemble des techniques permettant d'observer et d'analyser des objets sans contact matériel avec ceux-ci. Cette observation se fait généralement grâce à des données (images) acquises à partir des capteurs embarqués sur divers types de vecteurs tels que les drones, les avions, les ballons, les bateaux ou les satellites. Lorsque les données sont des images prises par des capteurs installés à bord de satellites, on parle d'imagerie satellitaire ou imagerie spatiale. Les satellites en question font alors partie des satellites appelés Satellites d'Observation de la Terre (SOT) ou satellites de télédétection. Ils permettent de capter des images d'une partie de la Terre et sont utilisés dans plusieurs domaines tels que la météorologie, l'agriculture, la défense, l'utilisation des sols, la foresterie, etc. [33, 34].

Les images satellites et les images classiques sont différentes sur plusieurs aspects. En effet, les images satellites contiennent généralement bien plus d'informations que les photographies classiques. Si l'on observe les deux types d'images, on peut constater que l'image satellite contient généralement beaucoup plus de pixels que l'image numérique classique. Le nombre de canaux disponibles pour chaque type d'images est aussi différent. Une image classique contient trois canaux correspondant respectivement aux couleurs rouge, vert et bleu (RVB). En plus de ces trois canaux, les images satellites peuvent contenir une quinzaine d'autres canaux pour les images multispectrales, voire des centaines pour les images hyperspectrales. De plus, les images satellites sont géo-référencées (à chaque pixel de l'image satellite correspond une coordonnée géographique) et leurs métadonnées contiennent des informations supplémentaires telles que les conditions d'acquisition, l'angle d'incidence, la direction de vol du satellite, etc.

On note aussi une différence concernant la gamme de longueurs d'ondes du spectre électromagnétique (figure 1.1) dans lesquelles les images sont acquises. Tandis que les images classiques relèvent du domaine du visible, les images de télédétection spatiale utilisent principalement trois fenêtres spectrales à savoir le domaine du visible, des infrarouges (IR) notamment le proche, le moyen et le thermique IR, et des hyperfréquences ou micro-ondes. Le tableau 1.1 donne un aperçu des différentes régions spectrales utilisées en observation de la Terre.

TABLEAU 1.1 – Plages de longueurs d'ondes utilisées pour l'observation de la Terre

Domaine spectral	Longueur d'onde	Source	Propriété
Visible	0,4 - 0,7 μm	Solaire	Réflectance
Proche IR	0,7 - 1,1 μm	Solaire	Réflectance
IR Moyen	1,1 - 8 μm	Solaire / Thermique	Réflectance / Température
IR Thermique	8 - 14 μm	Thermique	Température
Micro-ondes	1 mm - 1 m	Thermique / Artificiel	Température / Rugosité

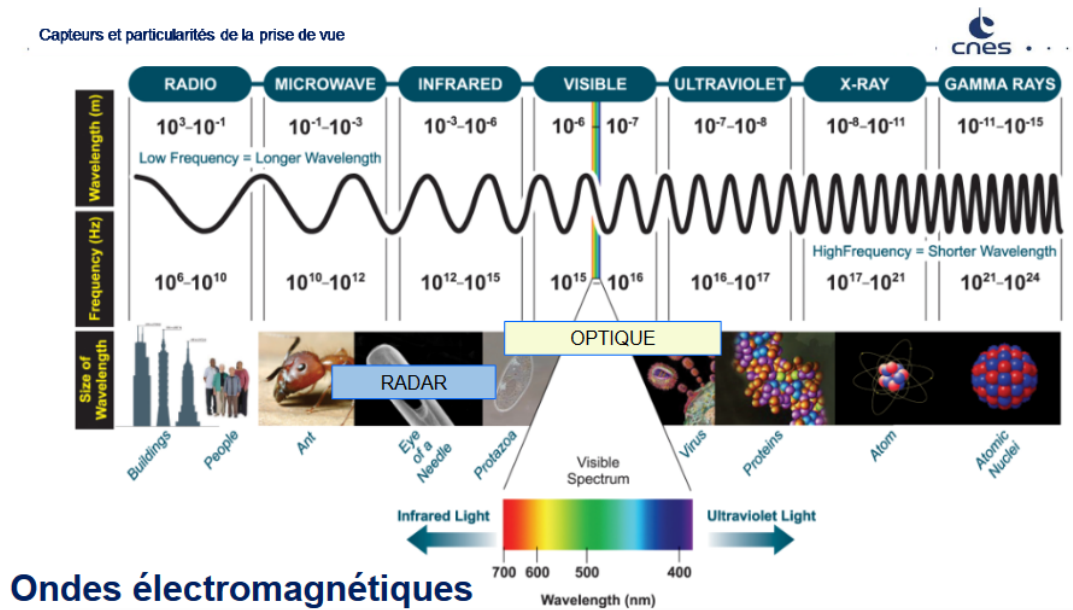


FIGURE 1.1 – Domaines du spectre électromagnétique [35].

La valeur de chaque pixel constituant une image satellite correspond à l'intensité du rayonnement réfléchi par l'objet observé (appelé cible) dans la gamme de longueur d'ondes auxquelles le capteur est sensible. Cette valeur appelée réflectance (ou réflectivité pour les images radar) dépend du type de capteur utilisé et de la nature de la cible (surface lisse, objet rugueux, végétation, eau, etc.). La réflectance est généralement exprimée en pourcentage et est définie comme le rapport entre l'énergie réfléchie et l'énergie incidente.

1.1.2 Images optiques / images radar

En télédétection, on mesure le rayonnement réfléchi par une cible. En fonction du mode d'acquisition des données, on distingue la télédétection passive (dans le domaine du visible et de l'IR) et la télédétection active (dans le domaine des hyperfréquences). Les images issues de la télédétection active sont principalement les images radar et celles de la télédétection passive les images optiques.

1.1.2.1 Images optiques

Dans la télédétection passive, la source d'illumination est le soleil ou l'objet observé en cas de télédétection thermique. Un des inconvénients pour ce mode d'acquisition est qu'il n'est pas possible de prendre des images dans le domaine du visible pendant la nuit. De plus, un autre inconvénient en télédétection passive est le fait que la qualité des images prises par les capteurs optiques dépend fortement des conditions météorologiques (présence de nuage, de brouillard, de pluie, etc.). Si le satellite se trouve par exemple dans une zone fortement nuageuse à l'instar des zones tropicales, les images seront difficilement exploitables à cause de la présence fréquente et dense des nuages qui vont cacher la cible. Comme exemples de satellites embarquant des capteurs optiques on peut citer SPOT, Landsat, Sentinel-2, Ikonos, Quickbird, etc. Les images

optiques n'ayant pas été utilisées dans le cadre de nos travaux, des détails supplémentaires sur ce type d'images ne seront pas données ici. Plus de détails sur les images optiques sont donnés dans [34, 35, 36].

1.1.2.2 Images radar

Pour pallier les insuffisances des capteurs passifs, un autre type de capteurs capables de prendre des images de jour comme de nuit, indépendamment des conditions météorologiques peut être utilisé. Il s'agit des capteurs dits « actifs », tels que les radars, qui émettent des rayonnements sur les objets et mesurent l'énergie réfléctée. Ils sont utilisés en télédétection active et ont l'avantage d'être peu perturbés par les conditions atmosphériques [37].

Le domaine des hyperfréquences utilisé par un capteur radar est nettement plus large que celui du visible et de l'infrarouge. Les bandes de longueurs d'ondes variant de 0,75 *cm* à 1,3 *m* environ, y sont identifiées par des lettres (Ka, K, Ku, X, C, S, L, P) et interagissent différemment selon la nature de la cible. Le tableau 1.2 présente les différentes bandes avec les plages de longueurs d'ondes associées en radar.

TABLEAU 1.2 – Principales bandes utilisées pour l'observation de la Terre en mode actif.

Bandes	Longueur d'onde en <i>cm</i>	Fréquence en <i>GHz</i>
P	30 - 130	0,3 - 1
L	15 - 30	1 - 2
S	7,5 - 15	2 - 4
C	3,75 - 7,5	4 - 8
X	2,4 - 3,75	8 - 12,5
Ku	1,67 - 2,4	12,5 - 18
K	1,11 - 1,67	18 - 26,5
Ka	0,75 - 1,11	26,5 - 40

Un autre élément important à considérer avec les capteurs actifs est l'orientation du champ électromagnétique pendant le processus de prise de vue (polarisation). Pour la majorité des capteurs radars, la transmission et la réception peuvent se faire de façon verticale (V) ou horizontale (H). La figure 1.2 présente un exemple de polarisations H et V d'une onde électromagnétique. Selon l'orientation de transmission et de réception des ondes, on peut distinguer les polarisations :

- VV - polarisations V pour la transmission et V pour la réception ;
- VH - polarisations V pour la transmission et H pour la réception ;
- HV - polarisations H pour la transmission et V pour la réception ;
- HH - polarisations H pour la transmission et H pour la réception.

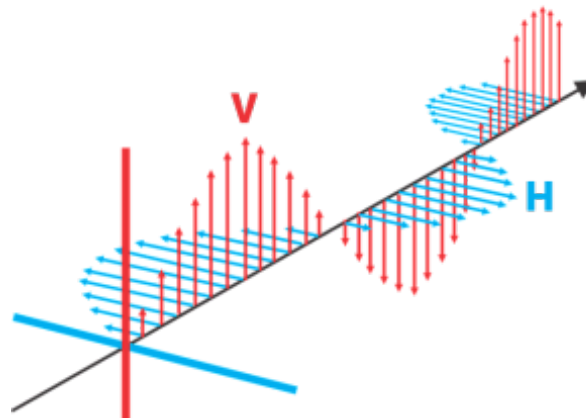
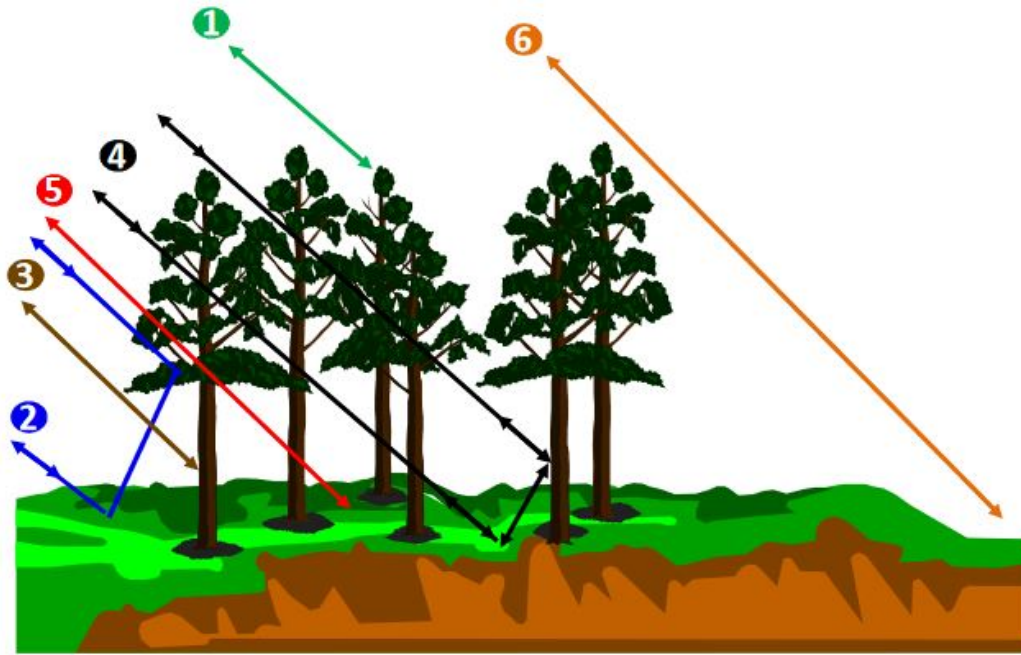


FIGURE 1.2 – Exemple de polarisations horizontale H (en bleu) et verticale V (en rouge) d'une onde électromagnétique plane.

Le capteur radar est un capteur qui fonctionne en mode actif, c'est-à-dire que le capteur illumine une portion du globe terrestre avec une onde électromagnétique et enregistre en retour les échos réfléchis par les objets qui s'y trouvent. Le mot radar signifie en effet *Radio Detection and Ranging* ou en français détection par onde radio et mesure de la distance. Le principe de fonctionnement des capteurs radar peut se résumer de la façon suivante : l'antenne radar émet de très brèves impulsions électromagnétiques latéralement au cours de son déplacement. Ces impulsions sont lancées à intervalles réguliers et fixes en direction des cibles au sol. Par la suite, les cibles réfléchissent les ondes reçues et l'émetteur enregistre les échos avec un certain décalage [38].

En fonction de la longueur d'onde utilisée par le capteur, le signal sera plus ou moins réfléchi selon la nature de l'objet visé. Ainsi les bandes P et L, ayant de grandes longueurs d'ondes peuvent facilement pénétrer la végétation, plus spécifiquement le couvert forestier ; contrairement aux bandes plus courtes (bande X) qui sont réfléchies par le sommet des arbres en raison de leur forte sensibilité à la rugosité de la surface du sol [39]. De façon générale, la longueur d'onde correspond à la capacité de pénétration, et lorsqu'en particulier l'imagerie radar est utilisée pour l'étude du couvert forestier, on distingue globalement trois types de diffusion comme présenté sur la figure 1.3 :

- diffusion de volume par le couvert forestier ;
- diffusion de surface ou de sol atténuée par la végétation ;
- interaction sol et végétation.



- | | |
|-------------------------------------|-----------------------------------|
| 1) Diffusion directe de la couronne | 4) Interaction tronc-sol multiple |
| 2) Interaction directe tronc-sol | 5) Diffusion du sol atténuée |
| 3) Diffusion du tronc | 6) Diffusion directe du sol |

FIGURE 1.3 – Rétrodiffusion du signal radar dans un couvert forestier [40].

Les systèmes fonctionnant sous ce principe complexe d'acquisition de données visent à illuminer plusieurs fois un objet par le capteur en mouvement et sont généralement qualifiés de radar à synthèse d'ouverture (RSO) ou *Synthetic Aperture Radar* (SAR) en anglais. Il s'agit des principaux types de capteurs en télédétection radar. Chaque image acquise selon ce principe est constituée de pixels dont les valeurs sont des nombres complexes. La partie réelle de ce nombre (amplitude) correspond à l'intensité réfléchiée selon la cible tandis que la partie imaginaire (phase) est liée au temps de propagation des ondes (distance) entre le capteur et l'objet visé [41].

Au cours du processus d'acquisition, des phénomènes d'interférence aléatoires surviennent et donnent à l'image formée un effet poivre-sel appelé chatoiement ou *speckle* [42]. Lorsque les interférences s'additionnent, on parle d'interférences constructives, par opposition aux interférences destructives qui ont lieu quand elles se soustraient.

Il existe de nombreuses méthodes permettant de réduire le *speckle* par des opérations de filtrage [43]. On note par exemple le filtre Lee, le filtre Sigma de Lee, le filtre de Lee amélioré, le filtre de Frost, le filtre médian, le filtre de Novak, etc. [44, 45, 46]. Le principe de base du filtrage du chatoiement est de sélectionner des pixels voisins ayant des caractéristiques de diffusion similaires à celles du pixel central, et d'appliquer un filtre.

Dans le cadre de cette thèse, le filtre amélioré de Lee [47] a été utilisé car il est simple et efficace en termes de calculs et permet de réduire efficacement le *speckle* sur les images radar, tout en conservant la netteté et les détails de celles-ci. Ce filtre défini par l'équation 1.1, est une version modifiée de celui proposé par Lee et al. dans [46].

$$z(k, l) = x(k, l)v(k, l) \quad (1.1)$$

où $z(k, l)$ représente la valeur en intensité ou en amplitude du pixel d'une image radar en position (k, l) , $x(k, l)$ est la réflectance (sans bruit), et $v(k, l)$ représente le bruit de speckle [47].

De façon visuelle, les images radar sont généralement très éloignées d'une vision naturelle et sont difficiles à interpréter, car ce qui y est présenté ne correspond pas forcément à ce qui peut être vu à l'œil nu, contrairement aux images optiques. La figure 1.4 représente par exemple l'aspect d'une même zone prise par des capteurs optiques et radars. Les surfaces rugueuses sur les images radar sont généralement représentées par des pixels de forte intensité, contrairement aux surfaces lisses qui ont tendance à être sombres sur les images (routes, eaux calmes). En ce qui concerne la végétation, elle est généralement grise sur les images en bande VV ou VH, car rugueuse à l'échelle des ondes centimétriques [39].

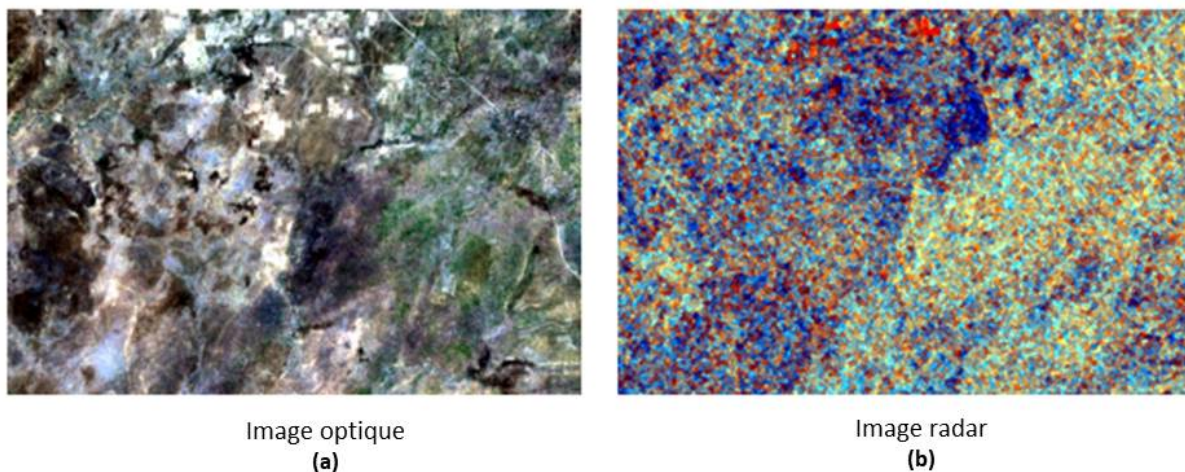


FIGURE 1.4 – Comparaison d'images optique et radar d'une zone dans le Parc National de Bouba-Ndjida (Cameroun) : (a) Composition colorée RVB d'une image Sentinel-2B ; (b) Composition colorée VV, VH, VH/VV d'une image Sentinel-1A.

On compte aujourd'hui plusieurs satellites qui embarquent des capteurs radar opérationnels. ALOS, ASAR, Sentinel-1, ENVISAT sont quelques exemples de satellites fonctionnant en mode actif.

1.1.3 Notions de résolution

Lorsque l'on veut choisir une image satellite pour traiter un problème, il est important de vérifier quelques caractéristiques fondamentales liées au capteur utilisé, notamment les informations sur les différents types de résolution (spatiale, radiométrique, spectrale, temporelle).

Toutefois, il est difficile de trouver un capteur unique possédant de très bonnes caractéristiques sur l'ensemble des types de résolution. C'est pour cela que le type de capteur sera choisi en fonction de l'usage que l'on veut faire des images.

- La résolution spatiale : l'un des premiers éléments à connaître concernant une image satellite est la résolution spatiale du capteur utilisé. Elle détermine le détail qu'il est possible de discerner sur une l'image. Cette résolution dépend des caractéristiques du capteur et de l'altitude de la plate-forme d'acquisition. Une image de résolution $10\text{ m} \times 10\text{ m}$ signifie qu'un pixel sur l'image représente $10\text{ m} \times 10\text{ m}$ au sol. Pour désigner la résolution d'un capteur ou d'une image satellite, on utilise généralement la taille du pixel. Les résolutions des images satellites varient de l'ordre des kilomètres pour les images basses résolutions, aux centimètres pour les images très hautes résolution spatiales qui sont comparables à des images aériennes. La figure 1.5 présente par exemple comment apparaît le complexe de tennis Wimbledon à Londres (Royaume-Uni) sur des images selon différentes résolutions spatiales (250 m à 0,3 m). Ces résolutions correspondent à celles des capteurs Aqua-MODIS (250 m), Landsat-8 (30 m), Sentinel-2 (10 m), PlanetScope (3 m), Pleiades (0,5 m) et Worldview-4 (0,3 m).

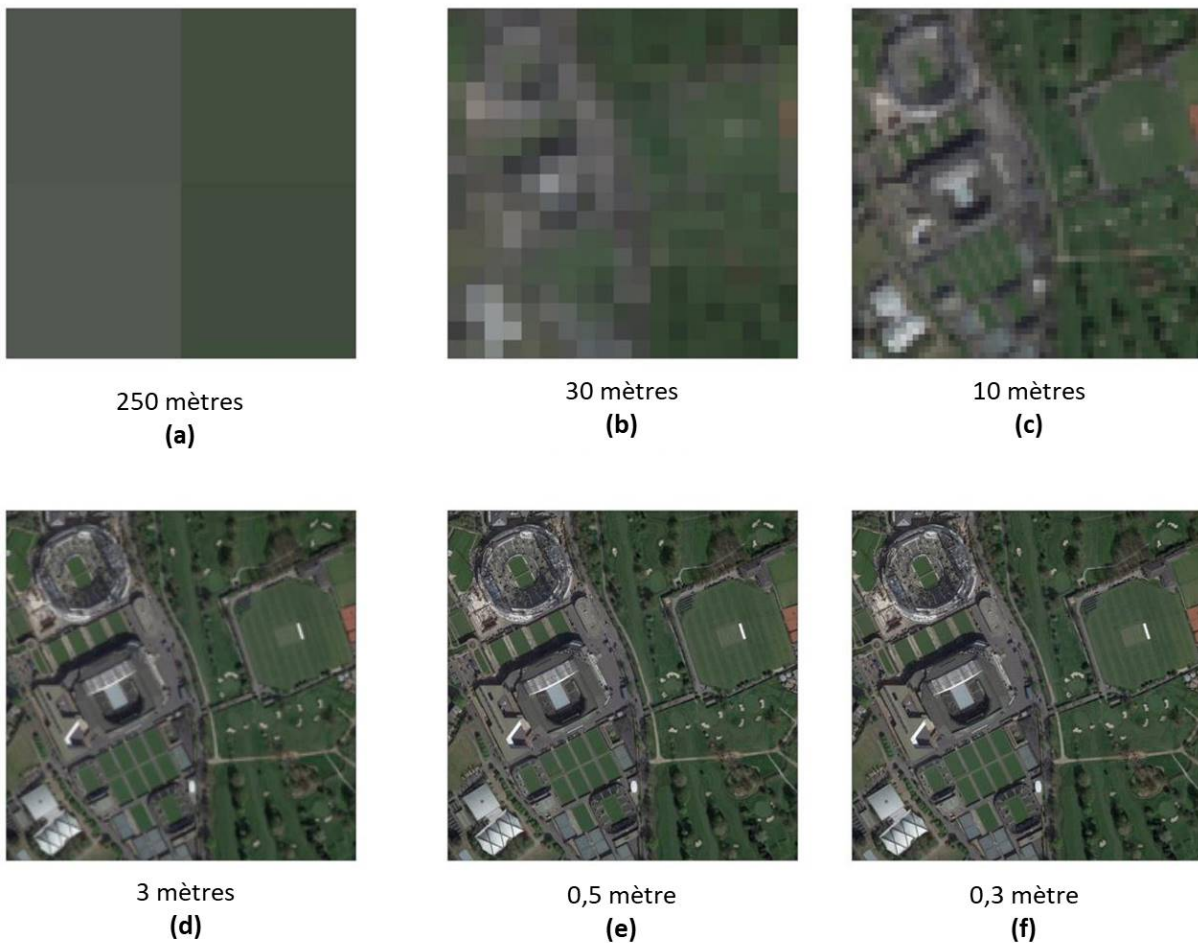


FIGURE 1.5 – Représentation du complexe Wimbledon sur des images avec différentes résolutions spatiales : (a) 205 m ; (b) 30 m ; (c) 10 m ; (d) 3 m ; (e) 0,5 m ; (f) 0,3 m [36].

- La résolution spectrale : elle correspond aux plages de longueurs d’onde auxquelles le capteur est sensible. Plus la résolution spectrale est fine, plus les fenêtres des différents canaux du capteur sont étroites, et plus la détection et la différenciation des objets sont précises. Les capteurs hyperspectraux sont des capteurs multispectraux plus développés, capables de détecter un grand nombre de bandes spectrales très fines (des centaines). Les capteurs PRISMA et HISUI sont des exemples d’instruments hyperspectraux (avec respectivement 250 et 185 bandes), tandis que Sentinel-2 et Landsat 8 sont des exemples de capteurs multispectraux (avec respectivement 13 et 11 bandes).
- La résolution radiométrique : il s’agit de la capacité du capteur à distinguer entre deux niveaux d’énergie. Le signal diffusé par les cibles et enregistré par le capteur est codé sous forme de bits, produisant une image en niveaux de gris. Le nombre de bits utilisés pour enregistrer l’image déterminera le nombre maximum de niveaux d’intensité disponibles. Par exemple, les capteurs Landsat 4 et SPOT-HRV sont codés sur 8 bits (256 niveaux de gris), tandis que les capteurs IKONOS et QuickBird sont codés sur 11 bits.
- La résolution temporelle : cette résolution correspond au temps de revisite du capteur au-dessus d’une zone. En d’autres termes, il s’agit de la durée qui s’écoule entre deux acquisitions de la même scène. La résolution temporelle des images Sentinel-1(A et B) est de 6 jours et 5 jours pour les images Sentinel-2(A et B).

Pour le cas particulier des systèmes RSO, la résolution du capteur se définit selon deux dimensions qui sont la résolution en portée et la résolution en azimut [48]. Contrairement au cas des systèmes optiques, la résolution spatiale d’un radar varie en fonction des propriétés spécifiques du rayonnement des hyperfréquences et des effets géométriques.

1.2 Apprentissage profond

1.2.1 Généralités

L’apprentissage profond, généralement connu sous le nom de Deep Learning (DL), est une partie de l’Intelligence Artificielle (IA) et de l’apprentissage automatique ou Machine Learning (ML), qui permet à un système d’apprendre à résoudre un problème par lui-même à partir d’exemples, contrairement à la programmation classique où des instructions prédéterminées sont exécutées. Le paradigme d’apprentissage le plus populaire en DL est l’apprentissage supervisé où la machine apprend à réaliser une tâche à partir d’exemples qui lui sont fournis. Dans la suite de ce document, il sera question de l’apprentissage supervisé. Par exemple, un modèle peut être entraîné pour reconnaître des roses après qu’on lui ait montré un grand nombre de photos de fleurs. De façon générale, la machine peut apprendre une relation $f : x \rightarrow y$ qui relie x à y après avoir analysé des millions d’exemples d’associations $x \rightarrow y$. La figure 1.6 illustre l’exemple de reconnaissance des roses. Après avoir fourni au système le *dataset*¹ constitué des images étiquetées de différentes fleurs, celui-ci est capable de reconnaître une rose sur une nouvelle photo inconnue au système.

1. Un *dataset* représente l’ensemble des données exemples de ce que le modèle doit apprendre.

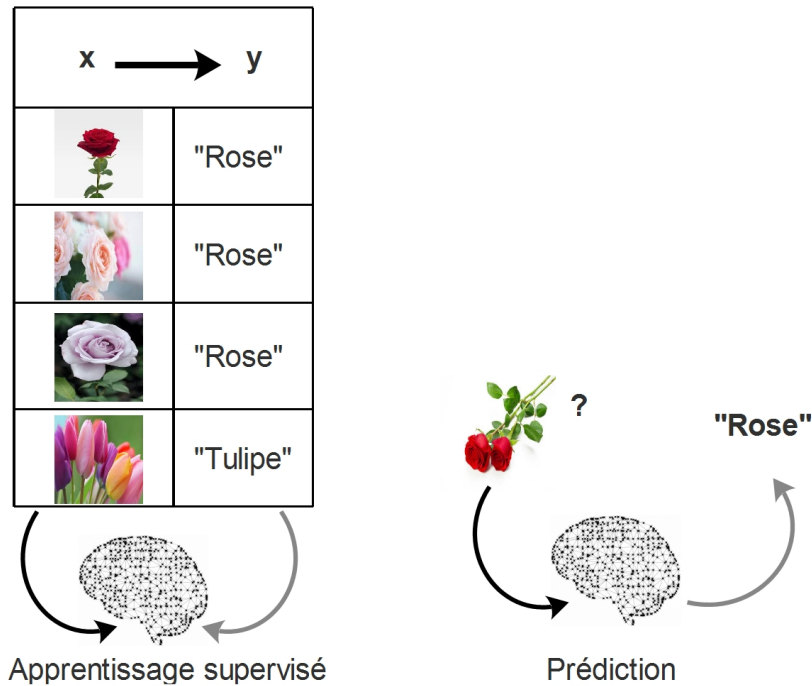


FIGURE 1.6 – Apprentissage d’un système pour la reconnaissance des roses à partir d’exemples de photos (x) et d’étiquette (y).

Pour mettre en œuvre un modèle d’apprentissage automatique, quatre étapes fondamentales sont nécessaires [49] :

- créer le *dataset* constitué des exemples que la machine doit apprendre. Ce *dataset* doit inclure la ou les variables caractéristiques x (*features* en anglais) et la variable cible y (ou *target*).
- concevoir un modèle avec des paramètres aléatoires permettant de produire des prédictions sur de nouvelles valeurs.
- développer ou utiliser une fonction de coût qui permettra de mesurer la différence (perte) entre la valeur prédite par le modèle et la valeur réelle du *dataset*. Cette différence est également utilisée pour mesurer la performance du modèle.
- créer ou utiliser un algorithme d’apprentissage (fonction d’optimisation) pour obtenir les paramètres du modèle qui minimise la perte.

Les architectures DL sont souvent appelées réseaux de neurones profonds car ils sont basés sur le concept de réseaux de neurones artificiels (RNA). Les RNA désignent des structures composées de nœuds (ou neurones), conçus pour reproduire certaines caractéristiques des neurones biologiques à partir des ordinateurs. Dans sa forme classique, un RNA consiste généralement en une interconnexion de neurones divisée en trois couches : une couche pour l’entrée des données, une couche pour la sortie et une couche cachée [25, 26]. Dans les architectures de réseaux de neurones profonds, il y a plusieurs couches cachées, et les connexions entre les neurones et les couches sont également plus complexes que dans les RNA classiques [32]. Ces interconnexions

permettent au modèle d'effectuer un apprentissage à partir des informations reçues et de résoudre des problèmes qui sont similaires de la meilleure façon possible.

Parmi les différentes techniques du ML, les algorithmes DL se sont avérés extrêmement puissants et sont largement utilisés dans plusieurs domaines de la vie courante. On les utilise notamment dans le domaine de la conduite automatisée, des traducteurs automatiques, de la détection des fraudes, des systèmes de reconnaissance faciale, de la recherche médicale, de la télédétection, et bien d'autres encore [20, 22]. La plupart des grandes firmes technologiques telles que IBM, Amazon, Adobe, Microsoft, Google ou Baidu s'y mettent aujourd'hui.

Le ML et le DL sont deux concepts qui se ressemblent à bien des égards et qui sont souvent confondus. Cependant, leur mode de fonctionnement diffère en certains points qu'il est important de relever. Par exemple, contrairement à l'apprentissage classique qui nécessite une étape supervisée d'extraction des caractéristiques représentatives (*feature extraction*), les informations importantes sont déterminées automatiquement dans les systèmes DL grâce à la profondeur du réseau de neurones [20, 22]. Cette capacité à s'adapter à des fonctions complexes et la capacité d'abstraction permettant d'extraire automatiquement les caractéristiques représentatives des données sans étape de *feature engineering* constitue l'un des éléments caractérisant les algorithmes DL. Une autre différence entre le ML et le DL se situe au niveau du type de données utilisées. En effet, les données non-structurées, telles que les données de télédétection, ne peuvent pas être manipulées directement par les algorithmes ML sans prétraitement préalable. La figure 1.7 permet de donner un aperçu global de la relation entre l'IA, le ML et le DL.

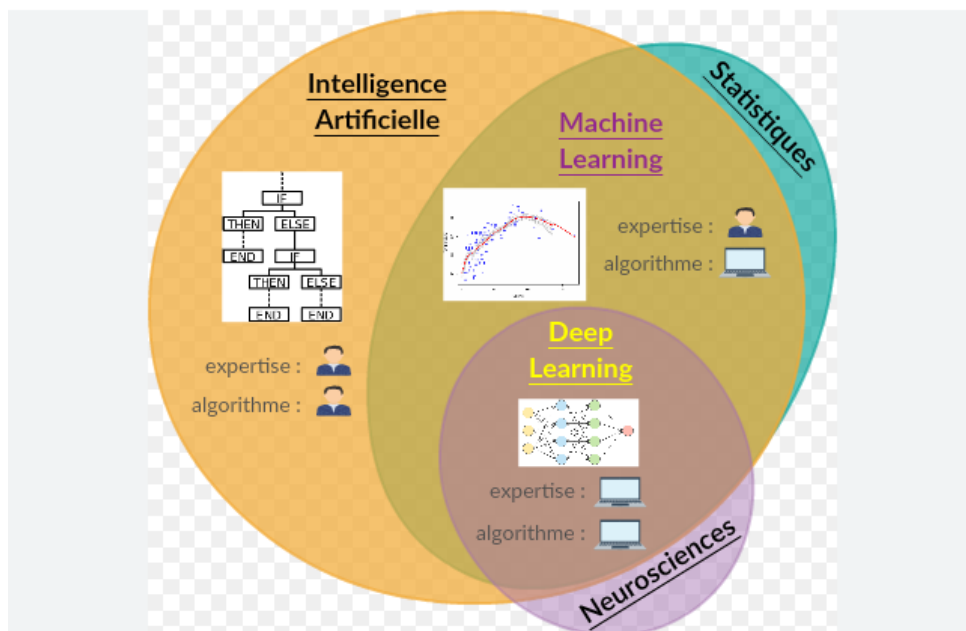


FIGURE 1.7 – Relation entre le DL, le ML et l'IA [50].

1.2.2 Fonctionnement

La figure 1.8 donne la représentation simplifiée d'un RNA. Ce réseau comprend :

Une couche d'entrée qui reçoit les variables d'entrée x_1, x_2, \dots, x_n . Chaque valeur correspond en principe à un neurone dans le réseau.

Des connexions entre les neurones de la couche d'entrée et ceux de la couche cachée, à l'exemple des synapses d'un réseau de neurones humain. Chacune de ces connexions porte un poids w_1, w_2, \dots, w_n qui est généralement initialisé avec des valeurs faibles mais non nulles. Ce sont les seules valeurs qui peuvent être changées dans le modèle pendant l'apprentissage et sont considérées comme les paramètres du modèle.

Une couche cachée où le neurone a la valeur de la somme pondérée des valeurs d'entrées aux différents poids, à laquelle on applique une fonction d'activation ϕ . Il existe plusieurs fonctions d'activation parmi lesquelles la fonction unité linéaire rectifiée ou *Rectified Linear Unit* (ReLU), la fonction sigmoïde, la fonction exponentielle normalisée ou *softmax*, la fonction tangente hyperbolique (\tanh) [25], etc.

Une couche de sortie avec un neurone qui contient la valeur en sortie (prédiction y'). Cette valeur sera comparée à la valeur réelle attendue y et permettra de calculer l'erreur C à partir de la fonction de coût f . La valeur en sortie peut être binaire, continue ou catégorique.

La fonction de coût f , encore appelée fonction de perte, définit la stratégie de mesure de la différence entre les valeurs réelles et les valeurs prédites par le modèle. Les fonctions de coût les plus utilisées sont le *Mean Square Error* (MSE), le *Mean Absolute Error* (MAE), le *Root Mean Squared error* (RMSE), etc. [27]. Le calcul de ces fonctions sera décrit à la section 2.3.2.1.

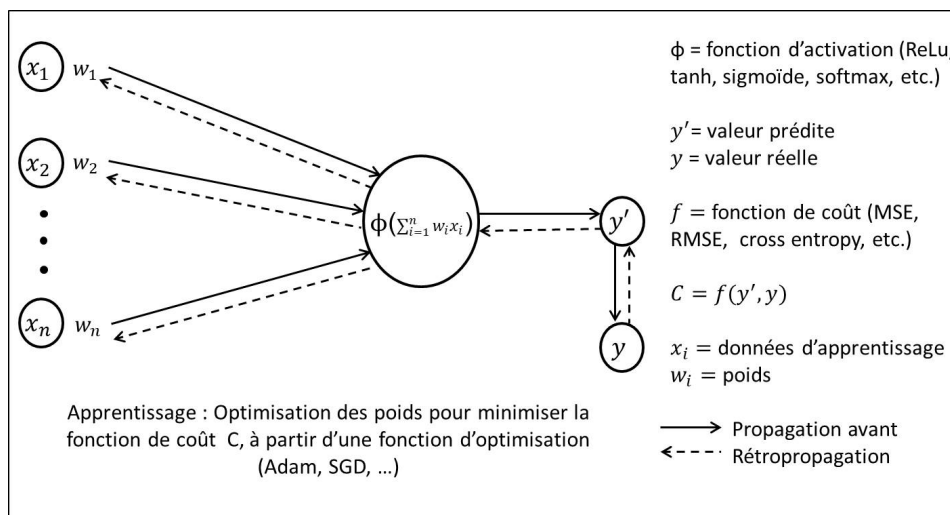


FIGURE 1.8 – Représentation simplifiée d'un réseau de neurones artificiels.

L'apprentissage dans un RNA se fait généralement en 7 étapes et peut se résumer par « l'ajustement des paramètres (poids) à partir des fonctions d'optimisation permettant de minimiser

l'erreur obtenue par la fonction de coût ».

- Étape 1 : Les poids sont initialisés avec des valeurs proches de 0, mais non nulles.
- Étape 2 : La première observation (les données) est envoyée à travers la couche d'entrée, avec une variable par neurone. Dans la pratique, toutes les variables sont mises sur la même échelle (standardisation ou normalisation) avant de les utiliser.
- Étape 3 : Propagation vers l'avant à travers les différentes couches jusqu'à la prédiction d'une valeur : les neurones sont activés en fonction des poids initiaux (application de la fonction d'activation).
- Étape 4 : Comparaison entre la valeur prédite et la valeur réelle et mesure de l'erreur avec la fonction de coût.
- Étape 5 : Rétropropagation de l'erreur à travers le réseau et mise à jour (ajustement) des poids de façon à obtenir une erreur minimale, grâce à une fonction d'optimisation.
- Étape 6 : Répétition des étapes 1 à 5 après chaque observation (apprentissage renforcé) ou après un certain nombre d'observations (apprentissage par lots).
- Étape 7 : Lorsque toutes les données sont passées dans le réseau, il s'agit d'une époque. Plusieurs époques sont exécutées autant que nécessaire, jusqu'à l'obtention d'une erreur acceptable.

1.2.3 Principales architectures deep learning

Les modèles DL sont basés sur les RNA. Il existe plusieurs types d'architectures DL et chacune d'entre elles est plus ou moins adaptée à un type de tâche particulière. Pour des problèmes plus complexes, plusieurs architectures sont souvent combinées pour obtenir de bons résultats. De plus, certains auteurs ont développé des méthodes dites hybrides qui consistent à combiner des techniques différentes. Dans cette section, les principales typologies de réseaux utilisés en apprentissage profond sont présentées. Un accent sera mis sur les architectures clés utilisées dans le cadre de cette thèse, notamment les réseaux de neurones à convolution et les réseaux de neurones récurrents.

1.2.3.1 Réseaux de neurones feedforward

Un réseau de neurones multicouches encore appelé *Multi-Layer Perceptron* (MLP) ou *feed-forward network*, est considéré comme la première et la plus simple des architectures DL [27]. Il se compose d'une couche d'entrée qui reçoit le signal, d'une couche de sortie pour la prédiction, et de plusieurs couches cachées qui sont le véritable moteur de calcul du réseau. Le nombre de neurones dans chaque couche est variable et les neurones sont tous entièrement connectés. Cela signifie que chaque neurone d'une couche est connecté à tous les neurones de la couche suivante comme le montre la figure 1.9. Ces réseaux sont généralement utilisés pour les problèmes de régression et de classification, impliquant une très grande quantité de données d'entraînement [51]. Cependant, ils ne sont pas assez efficaces pour les tâches avancées de vision par ordinateur. En effet, les images ou vidéos ne peuvent pas être utilisées directement comme entrée dans un réseau

MLP sans être prétraitées au préalable. De plus, comme chaque perceptron est connecté à tous les autres perceptrons du réseau, le nombre total de paramètres peut devenir très élevé. Les MLP appartiennent à la classe des algorithmes d'apprentissage supervisé et utilisent la rétro-propagation pour résoudre les problèmes d'apprentissage non linéaires. Par rapport à d'autres modèles, le temps de traitement avec les MLP est généralement inférieur au temps d'exécution pris par d'autres modèles DL [32].

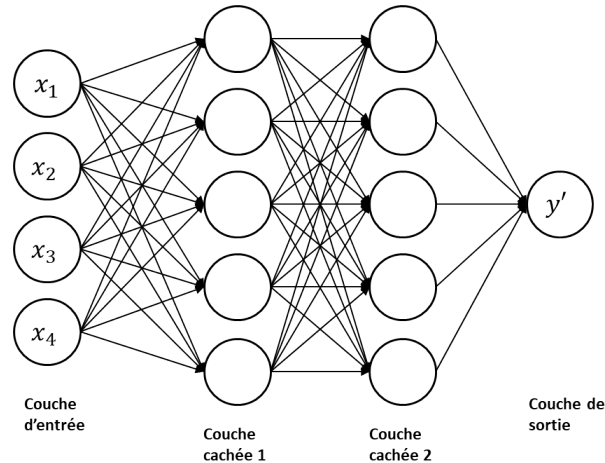


FIGURE 1.9 – Représentation d'un réseau de neurones *feedforward* à deux couches cachées.

1.2.3.2 Réseaux de neurones à convolution

Pour pallier les insuffisances des réseaux MLP dans la résolution de certaines tâches complexes, les réseaux de neurones convolutifs, encore appelés réseaux de neurones à convolution ou en anglais *Convolutional Neural Networks* (CNN), ont été conçus [52]. Il s'agit d'un type particulier de réseau utilisé notamment pour les données structurées en deux dimensions (2D) ou plus, comme les images, les vidéos ou les séries de données [52, 25]. Cette architecture est généralement utilisée dans le domaine de la vision par ordinateur et est appliquée dans plusieurs domaines tels que la détection d'objets, la reconnaissance de visages sur les médias sociaux, la détection d'anomalies, etc.

Le principe général de fonctionnement du CNN repose sur le fait d'appliquer des opérations successives (filtres) aux données d'entrée afin d'en extraire des caractéristiques importantes ; ce qui permet de réduire la taille des données. Ensuite, le résultat final est mis dans un seul vecteur qui est soumis à un MLP. Contrairement aux architectures MLP, les couches ne sont pas entièrement connectées dans un CNN. Un réseau CNN utilisé pour des données 2D, se compose généralement de quatre couches de base : couche de convolution, couche de *Pooling*, couche de *Flattening* et une couche entièrement connectée [52].

Couche de convolution : les étapes de convolution permettent d'extraire les caractéristiques représentatives de l'image par des opérations mathématiques de convolution ou filtrage. Les images obtenues après les opérations de convolutions sont appelées *features map*. Durant cette phase, on applique généralement une fonction d'activation telle que la fonction ReLu, pour créer une discontinuité (interrompt la linéarité) sur l'image filtrée. La fonction ReLu représentée

par la figure 1.10 est très souvent utilisée dans les réseaux CNN car son calcul est rapide. Elle renvoie x si x est supérieur à 0, 0 sinon.

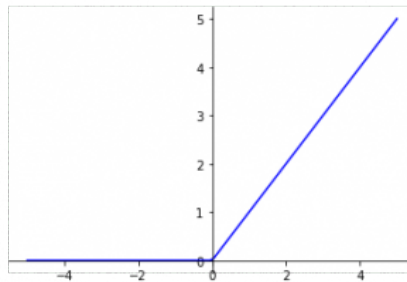


FIGURE 1.10 – Représentation graphique de la fonction ReLu.

Couche de Pooling (mise en commun) : parfois appelées *subsampling* ou *downsampling*, les étapes de *pooling* réduisent la taille de chaque donnée mais conservent les informations importantes. Elles confèrent au réseau la propriété d'invariance spatiale, c'est-à-dire une souplesse qui lui permet de reconnaître par exemple un objet sur une image malgré les déformations (rotation, luminosité, etc.). Généralement, une couche de *pooling* est insérée après chaque couche de convolution et permet dans la pratique de réduire le surapprentissage. La fonction *Max-Pooling* est la fonction la plus couramment utilisée pour l'étape de *pooling*, bien qu'il en existe d'autres comme la fonction *Min-Pooling* et la fonction *Average-Pooling*.

Couche de Flattening (aplatissement) : au cours de cette étape, les valeurs des matrices sont disposées sous forme de colonne dans un vecteur. Cette disposition des valeurs est nécessaire pour faire passer chacune d'elles dans des neurones d'un RNA.

Couche entièrement connectée : pour cette étape, il est question de rajouter un tout nouveau réseau de neurones qui va être complètement connecté aux données issues de la couche précédente. Une autre fonction est généralement utilisée tout à la fin du réseau, notamment la fonction *softmax*. Elle permet par exemple d'avoir une somme de probabilités à 1 pour un problème de classification.

Un réseau CNN prend en entrée des images de taille $W \times H \times D$, où W est sa largeur en pixels, H sa hauteur en pixels et D le nombre de canaux de l'image ($D=1$ pour une image en noir et blanc et $D = 3$ pour une image en couleurs).

La figure 1.11 présente un exemple de réseau CNN de reconnaissance des caractères écrits à la main. Ce réseau prend en entrée des images de taille $28 \times 28 \times 1$ et comprend deux couches de convolution avec des filtres de taille 5×5 . Chaque couche de convolution est suivie d'une couche *Max-Pooling* de taille 2×2 . Cette structure du réseau permet d'obtenir n^2 images de taille 4×4 qui sont concaténées dans un vecteur par la couche *Flattening*. Enfin, le vecteur obtenu après les opérations de convolution est connecté à un MLP qui va classifier l'image fournie en entrée au réseau.

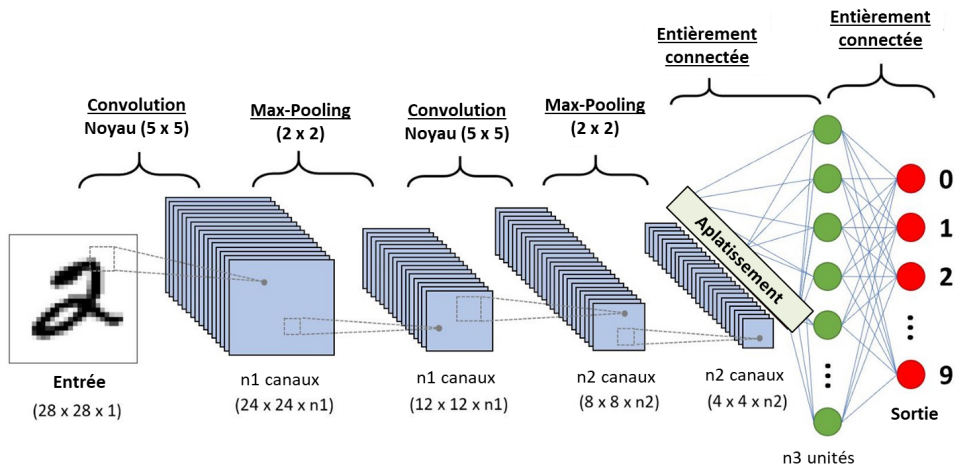


FIGURE 1.11 – Structure générale d’un réseau de neurones à convolution [53].

1.2.3.3 Réseaux de neurones récurrents

Les réseaux de neurones récurrents correspondent aux *Recurrent Neural Network* (RNN) en anglais. Ils sont généralement utilisés lorsque les données d’entrée sont séquentielles, comme dans les séries chronologiques, le son, ou le traitement du langage naturel. Ils ont des capacités d’apprentissage exceptionnelles et sont adaptés à la conception de modèles prédictifs [54, 55, 56, 57]. En effet, comme présenté sur la figure 1.12, ce réseau comprend dans sa structure une boucle de rétroaction permettant de prendre en compte les informations des couches précédentes, et de faire les prédictions. Cependant, deux problèmes principaux peuvent survenir lors de l’utilisation des architectures RNN : le problème des gradients évanescents (*vanishing gradients*) ou des gradients explosifs (*exploding gradients*) [25]. Pour surmonter ces limitations, deux variantes des RNN ont été développées : le *Long Short-Term Memory* (LSTM) [58, 59] et, plus récemment, le *Gated Recurrent Unit* (GRU) [60]. Le LSTM permet de résoudre les problèmes des RNN par l’introduction de cellules mémoire à travers un système de portes (*gates*) et d’états (*states*). Quant au GRU, il s’agit en quelque sorte d’une version simplifiée du LSTM, avec moins de paramètres. Au cours de cette thèse, seules les architectures LSTM ont été utilisées et seront décrites plus en détails dans le chapitre 4 (section 4.1).

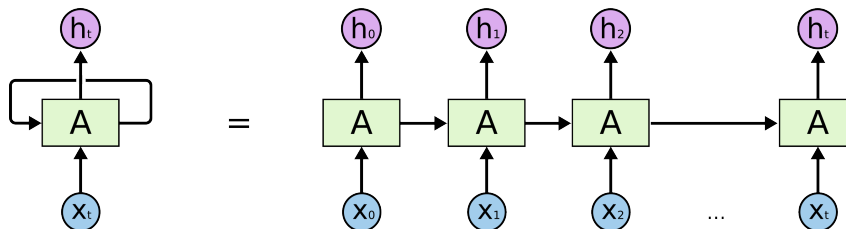


FIGURE 1.12 – Structure générale d’un réseau de neurones récurrents [61].

1.2.4 Amélioration des modèles deep learning

Les capacités de prédiction et les performances des modèles DL dépendent de nombreux facteurs tels que la qualité ou la quantité des données, la structuration des architectures DL, les algorithmes DL choisis, les fonctions d'activation et d'optimisation utilisées, les valeurs des hyperparamètres, etc. [62]. Parmi les techniques d'amélioration les plus utilisées dans la littérature, on note l'augmentation des données, le choix des hyperparamètres et des fonctions d'optimisation.

1.2.4.1 Augmentation des données

Tout comme en ML, les modèles DL ont besoin de données pour la phase d'apprentissage et constituent un rouage essentiel dans tout projet IA. Dans la plupart des cas, un modèle formé avec plus de données est généralement plus performant qu'un modèle ayant eu moins de données pour l'apprentissage. En effet, avec un jeu de données de grande taille, le modèle est capable de mieux identifier les relations existantes entre les données. Cependant, il faut noter que la quantité des données d'entraînement à elle seule ne revêt pas un caractère crucial pour les modèles. La qualité des données existantes est également une condition nécessaire pour escompter de bonnes performances du modèle.

Pour améliorer les modèles grâce aux données, la technique la plus utilisée est l'augmentation des *datasets*. Elle consiste à utiliser des stratégies pour augmenter artificiellement le nombre des données d'entraînement à partir des données existantes [62]. Comme illustré sur la figure 1.13, l'objectif est d'appliquer des fonctions de transformation sur les images initiales afin d'obtenir une base de données avec un volume plus important que celui de départ.

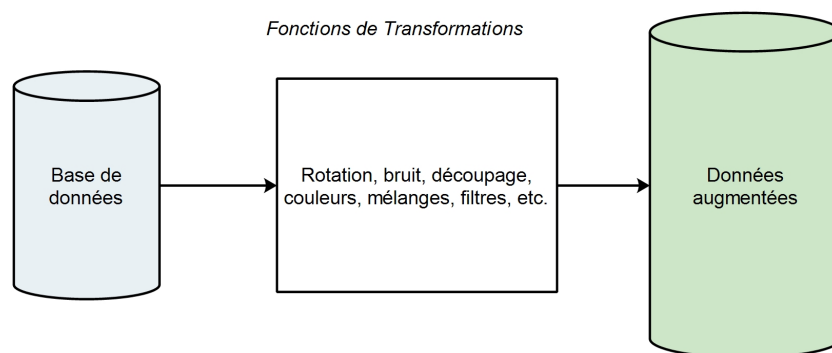


FIGURE 1.13 – Principe général d'augmentation de données.

Les techniques d'augmentation d'images reposent soit sur les manipulations basiques faites sur les images telles que les transformations géométriques, les augmentations de l'espace couleur, les filtres à noyau, le mélange d'images, l'effacement aléatoire, soit sur la création de nouvelles images réalistes à partir des algorithmes DL tels que les réseaux antagonistes génératifs ou l'apprentissage par transfert [62].

1.2.4.2 Choix des hyperparamètres

Les hyperparamètres sont des valeurs définies par l'utilisateur avant le début du processus d'apprentissage. Ils sont des éléments importants qui influent sur les performances, la vitesse et la qualité des modèles DL. On distingue trois grands types d'hyperparamètres pour les modèles DL. Les hyperparamètres de couches (taille et nombre des noyaux, nombre de couches, méthodes d'activation des couches, etc.), les hyperparamètres de compilation du modèle (fonction d'optimisation, fonction de coût, taux d'apprentissage, etc.) et les hyperparamètres d'exécution du modèle (taille du lot, nombre d'époques (epochs), nombre d'étapes par epoch, etc.).

Choisir les meilleurs hyperparamètres d'un modèle de façon manuelle en effectuant plusieurs tests peut constituer une tâche fastidieuse. Pour remédier à cette difficulté, des méthodes d'optimisation automatiques ont été proposées. Parmi les techniques de base on peut citer la technique du *Grid Search* (la recherche en grille), la technique du *Random Search* (la recherche aléatoire) ou des techniques plus évoluées telles que l'optimisation bayésienne ou hyperbande [63].

1.2.4.3 Choix de la fonction d'optimisation

Le processus d'ajustement des poids pour produire des prédictions de plus en plus précises sur les données est connu sous le nom d'optimisation des paramètres. Ce processus peut être considéré comme une méthode scientifique dans laquelle une hypothèse est formulée et testée par rapport à la réalité. Puis cette hypothèse est affinée ou remplacée (éventuellement plusieurs fois) pour mieux décrire les événements dans le monde.

Le problème du choix des poids appropriés pour le modèle est une tâche ardue, car un modèle d'apprentissage profond se compose généralement de millions de paramètres. Il est donc nécessaire de choisir un algorithme d'optimisation adapté au type d'application traitée. Parmi les fonctions d'optimisation les plus utilisées dans la littérature, les fonctions RMSProp, Adam, ou Momentum sont mentionnés [64].

1.3 Prédiction des séries temporelles

1.3.1 Définitions des concepts

Les séries temporelles, encore appelées séries chronologiques, sont des suites de données acquises à des intervalles de temps réguliers et successifs (minutes, jours, années, etc.). Dans une série x_t à n éléments, chaque observation x est associée à un indice temporel t . En analyse de séries temporelles, la prédiction des événements futures (ou prévisions) constitue l'un des problèmes traités. En effet, contrairement aux modèles de prédiction classiques (classification ou régression) où les estimations sont faites sur la base des valeurs d'une nouvelle observation (prédiction du prix de vente d'une maison par exemple), la prédiction des séries temporelles utilise l'historique des données passées pour prédire les valeurs futures (prédiction du cours des actions de la bourse par exemple ou prédiction des conditions météorologiques) [65]. Cette dépendance temporelle entre les observations ajoute une certaine complexité dans la résolution

des problèmes de prédictions de séries temporelles. La figure 1.14 illustre un exemple simple de prédiction dans une série temporelle d'images de télédétection.

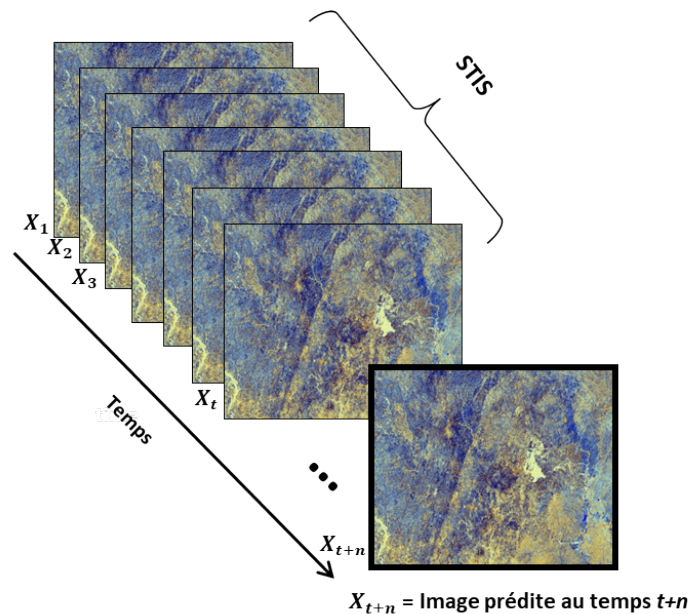


FIGURE 1.14 – Exemple de prédiction à une étape dans une série temporelle d'images satellites.

Pour mieux appréhender certaines notions du champ lexical de la prédiction des séries temporelles, qui sont par ailleurs utilisées dans la suite de ce document, certains éléments nécessitent d'être précisés :

Entrées et sorties : lorsqu'on parle d'entrée dans ce contexte, il s'agit de la configuration des données utilisées par le modèle pour faire une prédiction unique. Par exemple, considérant une série d'observations, le modèle peut être entraîné pour recevoir une séquence d'éléments de taille n en entrée et produire en sortie du modèle une prédiction [19]. L'entrée d'un modèle ne correspond donc pas à l'ensemble des données utilisées pour l'entraînement, mais plutôt des données utilisées pour réaliser une prédiction.

Variables endogènes ou variables exogènes : une variable d'entrée est endogène si elle est affectée par d'autres variables du système et que la variable de sortie en dépend. Par contre, une variable est dite exogène si elle est indépendante des autres variables du système et que la variable de sortie en dépend.

Série univariée ou série multivariée : une série univariée désigne une série avec une variable unique mesurée dans le temps, contrairement à une série temporelle multivariée qui possède plusieurs variables.

Prédiction one-step ou prédiction multi-step : lorsqu'un problème consiste à prédire le prochain élément au temps $t+1$ dans une série temporelle, on parle de modèle de prédiction *one-step*. Au contraire, dans un modèle prédictif *multi-step*, la prévision peut se faire sur plusieurs pas de temps $t+2, t+3, \dots$. Plus le nombre de pas de temps à projeter dans le futur est important, plus le problème est difficile à résoudre, puisque à chaque pas, la prédiction se fait sur la base de valeurs prédites (comportant des erreurs).

Modèle dynamique ou modèle statique : il est possible de développer un modèle une seule fois et de l'utiliser à plusieurs reprises pour faire des prévisions : on parle de modèle statique. Inversement, dans un modèle dynamique, il est possible de mettre à jour un modèle existant avant chaque prédiction. En général, il est préférable d'utiliser des modèles dynamiques. Cependant, en fonction du domaine ou des limitations de l'algorithme choisi, il n'est pas souvent possible de mettre en œuvre de tels modèles.

Série contiguë ou non-contiguë : une série temporelle où les acquisitions des observations sont uniformes dans le temps peut être décrite comme contiguë. Dans le cas contraire, la série est non-contiguë et un formatage spécifique des données peut être nécessaire pour rendre les observations uniformes dans le temps. Le manque d'uniformité des observations peut être causé par des valeurs manquantes ou corrompues. Il peut également s'agir d'une série où les observations ne sont disponibles que sporadiquement ou à des intervalles de temps de plus en plus ou de moins en moins espacés.

De façon générale, tout problème de prédiction peut être catégorisé dans l'un des trois grands groupes [66] :

- *one-to-sequence* : à partir d'une entrée à un élément, le modèle génère une sortie sous forme de séquence (plusieurs éléments). Par exemple, dans un système permettant de générer des phrases pendant l'écriture d'un texte, un mot reçu en entrée permet de générer une succession de mots (phrase) ;
- *sequence-to-one* : ici, les entrées sont des séquences et une seule occurrence est produite en sortie. C'est par exemple le cas d'un problème de prédiction du prochain élément d'une séquence donnée en entrée ;
- *sequence-to-sequence* : dans ce cas, les entrées et les sorties sont toutes deux des séquences, comme dans les tâches de traduction de texte par exemple.

1.3.2 Méthodes de prédiction

L'estimation des futures valeurs dans une série chronologique se fait toujours à partir de l'observation des valeurs précédentes. Dans certains cas, des variables exogènes peuvent être également utilisées pour améliorer les précisions des prédictions. Réaliser des prédictions très précises constitue un réel défi. Cependant, plusieurs méthodes de prédiction permettant d'obtenir des résultats très satisfaisants existent. Pour la modélisation des séries chronologiques linéaires, les principales méthodes sont les modèles de régression, les lissages exponentiels et d'autres modèles linéaires qui sont très populaires depuis une trentaine d'années. En effet, la prévision des séries temporelles a longtemps été dominée par des méthodes linéaires univariées, basées sur le principe d'auto-régression comme ARIMA (*Autoregressive integrated moving average*), car elles sont efficaces pour de nombreux problèmes [67]. Cependant ces méthodes classiques présentent quelques limites [19]. En effet, elles se focalisent essentiellement sur :

- les données complètes : les données manquantes ou corrompues ne sont généralement pas prises en charge ;

- les relations linéaires : supposer une relation linéaire exclut les distributions conjointes plus complexes ;
- la dépendance temporelle fixe : la relation entre les observations à différents moments, ainsi que le nombre d’observations décalées fournies en entrée, doit être spécifié ;
- les données univariées : de nombreux problèmes du monde réel comportent des variables d’entrée multiples.
- les prédictions *one-step* : certains problèmes peuvent nécessiter des prévisions avec un horizon temporel long.

Par la suite, des modèles de prédiction multivariés ont été développés. A titre d’exemple, on note la version étendue du modèle auto-régressif (AR) ou le modèle vectoriel auto-régressif (*Vector Auto-Regressive*, VAR). Ces modèles sont encore populaires de nos jours et sont utilisés soit indépendamment, soit en combinaison avec d’autres techniques telles que les RNA [68]. L’objectif des modèles multivariés est d’améliorer la qualité des prédictions en utilisant des données supplémentaires provenant d’autres sources d’information, mais en gardant la même logique que les modèles univariés.

Pour des problèmes de prédiction assez complexes, les capacités des modèles DL semblent bien adaptées. Si l’on dispose des ressources suffisantes, les réseaux de neurones devraient être en mesure de surpasser les capacités des méthodes de prévision linéaires classiques, étant donné leur capacité à apprendre des relations complexes arbitraires des entrées aux sorties. Cependant, en ce qui concerne les problèmes nécessitant l’utilisation des données spatio-temporelles, le modèle CA-Markov est largement utilisé dans la littérature. Il s’agit d’un modèle hybride composé des chaînes de Markov et des automates cellulaires (*Cellular Automata*) [69]. La section suivante présente plus en détail cette approche qui sera considérée comme référence dans nos travaux et dont les performances seront comparées à celles des modèles DL proposés.

1.3.3 Méthode CA-Markov

Les automates cellulaires (*Cellular Automata*, CA) et les chaînes de Markov sont deux concepts utilisés dans de nombreux travaux scientifiques. Lorsqu’ils sont fusionnés pour former le modèle hybride communément appelé CA-Markov, ils constituent un puissant outil de prédiction pour de nombreux problèmes pratiques, notamment dans le domaine spatio-temporel [11]. Ces outils sont utilisés par exemple pour caractériser les changements de l’utilisation et de la couverture des sols, prédire la déforestation et la dynamique de l’urbanisation, étudier l’évolution des zones humides et arides, estimer les récoltes, etc. [32].

Dans un modèle CA-Markov, la partie chaîne de Markov s’occupe des changements temporels entre les classes d’occupation de sol, tandis que la partie automates cellulaires, contrôlée par des règles locales déterminées, s’occupe de la dynamique spatiale [70]. En effet, un modèle de chaîne de Markov est un processus stochastique qui analyse la probabilité de passage d’un état à un autre, c’est-à-dire que l’état d’un système au temps t_n est prédit à partir de l’état dans lequel se trouve le système au temps t_{n-1} [71]. L’état à un instant particulier t dépend donc exclusivement de l’état au temps $t - 1$ et non des états précédents c’est-à-dire aux instants $t - 2$,

$t - 3$ et ainsi de suite [11].

D'après le principe de fonctionnement des chaînes de Markov, les états des cellules adjacentes ne dépendent pas de l'état de la cellule observée. Ainsi, l'utilisation de ce modèle de Markov seul, ne tenant pas compte de la distribution spatiale de chaque catégorie, s'est avérée insuffisante pour traiter les problèmes spatio-temporels. Par conséquent, le modèle CA a été mis en œuvre pour prendre en compte la nature spatiale des données [72, 11]. Ainsi, dans un automate cellulaire, les changements d'état d'une cellule dépendent non seulement de son état précédent mais aussi de l'état des cellules qui lui sont immédiatement adjacentes.

Dans la pratique, la prédiction à l'aide du modèle CA-Markov se fait généralement en analysant deux images prises à deux différentes dates (différentes années généralement). Une image de référence est également utilisée pour valider la méthode. Les cartes thématiques sont d'abord produites à partir d'images pour identifier les différentes classes d'occupation de sol de la zone étudiée (forêt, agriculture, urbaine, sol nu, etc.). Cependant, il est important d'avoir des images classifiées avec une bonne précision pour permettre au modèle prédictif d'être plus efficace [11]. Des données exogènes telles que le modèle numérique de terrain ou la carte de réseau routier, sont éventuellement aussi utilisées, en fonction de leur implication dans le phénomène étudié. Au cours du processus, certains éléments qui impactent la sortie du modèle (comme la matrice de probabilité de transition) sont produits. La validation du modèle se fait par comparaison entre l'image prédite et l'image réelle correspondante (nombre de pixels appartenant à chaque classe d'occupation de sol). Lorsque le pourcentage global de prédiction est acceptable, alors une nouvelle prédiction pour une date plus éloignée peut être faite à partir du modèle créé [11].

Les principaux concepts et outils liés aux images satellites et à l'apprentissage profond ont été abordés dans ce chapitre. Dans la première partie, les principales caractéristiques des images satellites et particulièrement des images radar utilisées dans le cadre de ces travaux ont été présentées. Le mode d'acquisition des images radar permettant d'obtenir des images sans nuages, ainsi que les notions de polarisation et de résolution ont été décrits. Dans la deuxième partie consacrée à l'apprentissage profond, les principales architectures ainsi que les méthodes d'amélioration des modèles DL ont été présentées. Les architectures LSTM qui sont généralement utilisées lorsque les données d'entrée sont séquentielles ont été retenues pour la suite des travaux. De plus, les éléments liés à la prédiction des STIS ont été définis et les méthodes de prédiction décrites. Il ressort de cette partie que la méthode CA-Markov est largement utilisée dans la littérature et produit des résultats assez satisfaisants. Dans la suite des travaux, les méthodes DL et la méthode CA-Markov seront comparées. Dans le chapitre 2, les principaux travaux de la littérature qui traitent de la thématique des prédictions des STIS à partir des algorithmes DL seront étudiés et présentés.

Chapitre 2

Etat de l'art sur l'utilisation du DL pour la prédiction des séries temporelles d'images satellites

Objectifs

L'utilisation du deep learning sur les images satellites présente des spécificités qu'il est important de noter. L'objectif de ce chapitre « Etat de l'art » est d'étudier et présenter les récentes publications scientifiques relatives à l'application des techniques de l'apprentissage profond pour la prédiction des séries temporelles d'images satellites. Plus précisément, les problèmes de prédiction et les principales applications rencontrées dans la littérature, ainsi que les méthodes d'apprentissage, d'optimisation et d'évaluation des modèles DL sont présentés. A la fin de ce chapitre, les limites majeures rencontrées dans ce domaine et les pistes de solutions proposées sont données.

Sommaire

2.1	Description des problèmes de prédiction et leurs applications	30
2.1.1	Description des problèmes de prédiction	30
2.1.2	Principales applications	30
2.2	Méthodes deep learning pour la prédiction des séries temporelles d'images satellites	33
2.2.1	Architectures basées sur les RNN	33
2.2.2	Architectures hybrides	35
2.2.3	Architectures basées sur les réseaux feedforward	38
2.3	Optimisation et métriques d'évaluation	39
2.3.1	Fonctions d'optimisation	39
2.3.2	Métriques d'évaluation	41
2.4	Limites et propositions de solutions	45
2.4.1	Manque de données d'entraînement	45
2.4.2	Opérations de prétraitement des STIS	45
2.4.3	Conception des architectures DL	47
2.4.4	Réutilisation des modèles	47

2.1 Description des problèmes de prédiction et leurs applications

2.1.1 Description des problèmes de prédiction

Soit X_t une image satellite, représentant une zone donnée, acquise à l'instant t . Une Série Temporelle d'Images Satellites (STIS) ou *Satellite Image Time Series* (SITS) en anglais, est une séquence d'images satellites acquises à de différents temps consécutifs [73]. Les images peuvent provenir de sources différentes mais représentent toujours la même scène. Ainsi, une STIS peut être définie selon l'équation 2.1 :

$$SITS = \langle X_1, X_2, \dots, X_{t-1}, X_t, X_{t+1}, \dots, X_{t+n} \rangle \quad (2.1)$$

où X_t représente l'image X au temps t .

Lorsqu'on parle de STIS représentant une partie de la Terre, l'expression « cube de données d'observation de la Terre » est généralement utilisée [74]. Dans littérature, la prédiction dans les STIS ne consiste pas toujours à prévoir un événement. Dans certains cas, elle est liée à la fusion de données ou à la reconstruction de données manquantes dans les séquences. Lorsque les données prédites concernent des événements futurs, le mot « prévision » (*forecasting*) est également utilisé, et le terme « *nowcasting* » est souvent utilisé pour désigner des prévisions à très court terme [32].

2.1.2 Principales applications

Les techniques DL sont utilisées pour résoudre plusieurs types de problèmes notamment les prévisions météorologiques, la prédiction des précipitations, l'analyse spatio-temporelle, ou la reconstruction des données manquantes.

2.1.2.1 Prévisions météorologiques

Les tâches de prévision météorologique consistent à prédire des conditions futures de l'atmosphère (la température, l'humidité et le vent) pour un lieu et à un moment donné. Ces tâches sont d'une très grande importance pour la gestion des catastrophes. Avec le développement de la technologie des satellites météorologiques, les images satellites permettent de caractériser la distribution des nuages et sont utilisées pour suivre l'évolution des systèmes météorologiques à grande échelle [75]. Outre l'approche classique de la prévision numérique du temps (PNT), les techniques DL sont utilisées de nos jours par les *data scientists* pour la prévision météorologique et permettent d'obtenir des résultats plus précis [75].

La figure 2.1 illustre les deux concepts de prévision météorologique : l'approche fondée sur la théorie et l'approche fondée sur les données. L'approche fondée sur la théorie est axée sur la compréhension des mécanismes physiques et utilise des modèles basés sur la PNT pour les prédictions, tandis que l'approche fondée sur les données utilise les relations spatiales et temporelles entre les données météorologiques et les techniques DL. On peut observer sur la figure 2.1

que la méthode théorique comprend trois étapes (l'assimilation des données, l'exécution du modèle PNT et la visualisation), tandis que l'approche axée sur les données utilise directement les données et les algorithmes DL pour faire des prédictions [75].

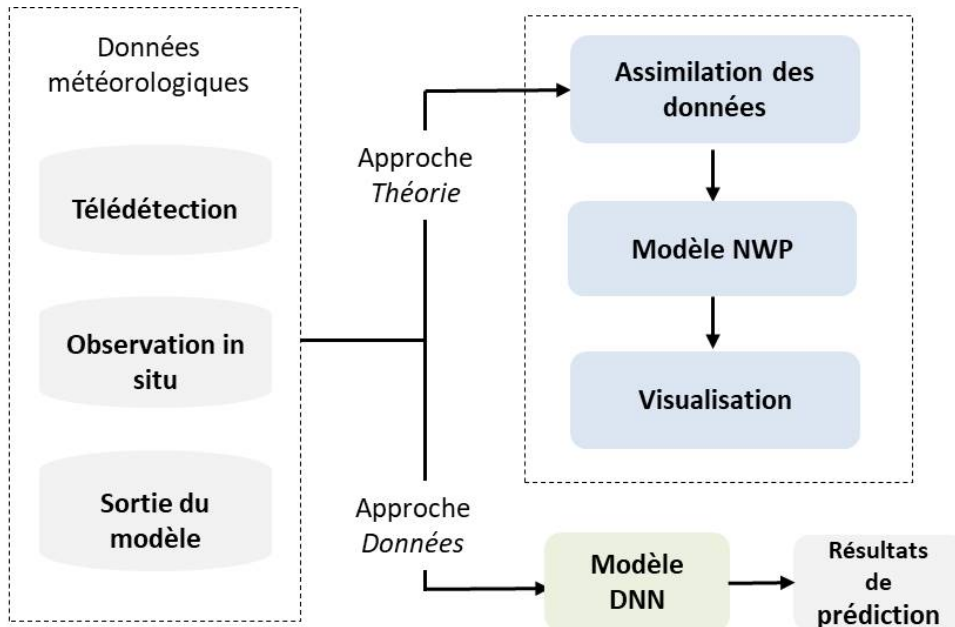


FIGURE 2.1 – Approches de prévision météorologique décrite dans [75].

2.1.2.2 Prédiction des précipitations

La prédiction immédiate des précipitations est une partie importante et essentielle en météorologie. L'objectif est de prédire l'intensité de la pluie ou de la neige dans une région donnée sur une période de temps relativement courte (par exemple, 0 à 6 heures) [76]. Pour atteindre cet objectif, des séries de cartes observées appelées « echo radar », sont utilisées pour prédire en sortie une séquence de longueur fixe d'images, représentant les états futurs de la zone étudiée. Les cartes « echo radar » sont des images qui peuvent être converties en cartes d'intensité de pluie à l'aide de la relation Marshall-Palmer ou de la relation Z-R [77]. Du point de vue de l'apprentissage automatique, ce problème est considéré comme un problème de prédiction de séquences spatio-temporelles. Récemment, des études ont montré que les techniques DL surpassaient les méthodes classiques existantes dans ce domaine, à savoir les méthodes basées sur la PNT et les méthodes basées sur l'extrapolation des « echo radar » [78, 79]. En effet, la disponibilité d'une grande quantité de données et le développement des modèles appropriés pour l'apprentissage de bout en bout justifient les bonnes performances atteintes.

2.1.2.3 Analyse spatio-temporelle

L'analyse spatio-temporelle fait référence à l'étude de données collectées à la fois dans l'espace et dans le temps. Elle décrit un événement qui s'est produit à un endroit spécifique au fil du temps. Avec le développement des processeurs de plus en plus performants, les techniques DL sont

aujourd'hui largement utilisées pour l'analyse des données spatio-temporelles qui sont de nature très lourdes. Les applications spécifiques comprennent la prédiction de la couverture neigeuse, de l'indice foliaire [80], de la végétation [81], de la température de surface de la mer [82], des récoltes, etc. En général, les prédictions sont faites à partir d'indices qui sont calculés à l'aide des valeurs des pixels des images brutes. Les indices cartographiques sont des images obtenues à partir de certaines bandes d'images multispectrales. Ils mettent en évidence un phénomène particulier à étudier, par des calculs spécifiques à chaque type d'indices (indices de végétation, indices des sols, indices d'humidité, etc.).

Par exemple, l'indice de végétation par différence normalisée ou *Normalized Difference Vegetation Index* (NDVI) calculé à partir de la réflectance des canaux rouge (R) et proche infrarouge (PIR), permet d'étudier la végétation sur une image (quantité, qualité). Le NDVI est défini par l'équation (2.2) et ses valeurs sont comprises entre -1 et 1. Les valeurs négatives représentent généralement des surfaces sans végétation comme la neige, les nuages ou les surfaces d'eau, tandis que les valeurs positives représentent les formations végétales. Plus la végétation est verte et dense, plus les valeurs des pixels du NDVI sont importantes. Pour les sols nus, le NDVI a des valeurs proches de zéro.

$$NDVI = \frac{PIR - R}{PIR + R} \quad (2.2)$$

2.1.2.4 Reconstruction des données manquantes

L'observation de la Terre par télédétection est l'un des moyens les plus importants pour obtenir des informations sur des objets ou surfaces du globe. Parfois, en raison de certains problèmes tels que le dysfonctionnement interne des capteurs du satellite, la faible fréquence temporelle, les mauvaises conditions atmosphériques ou d'autres problèmes spécifiques au capteur, les séries d'images satellites acquises souffrent d'informations manquantes. Par exemple, les données peuvent être confrontées au problème des pixels « morts » sur les images, d'une couverture neigeuse trop importante [83], ou d'images manquantes à certaines dates comme présenté sur la figure 2.2. On peut en effet observer dans cette illustration que, les informations de l'image X_{t+3} , qui auraient dû être enregistrées au temps $(t + 3)$, sont manquantes. Ce qui pourrait entraîner une réduction de la qualité de l'exploitation des données. Des modèles pour reconstituer ces données manquantes sont alors proposés afin d'avoir une séquence complète [83].

A ce jour, plusieurs techniques de reconstruction d'informations manquantes ont été proposées. Ces approches sont catégorisées en quatre grands groupes, à savoir les méthodes spatiales, les méthodes spectrales, les méthodes temporelles et les méthodes spatio-temporelles spectrales [83].

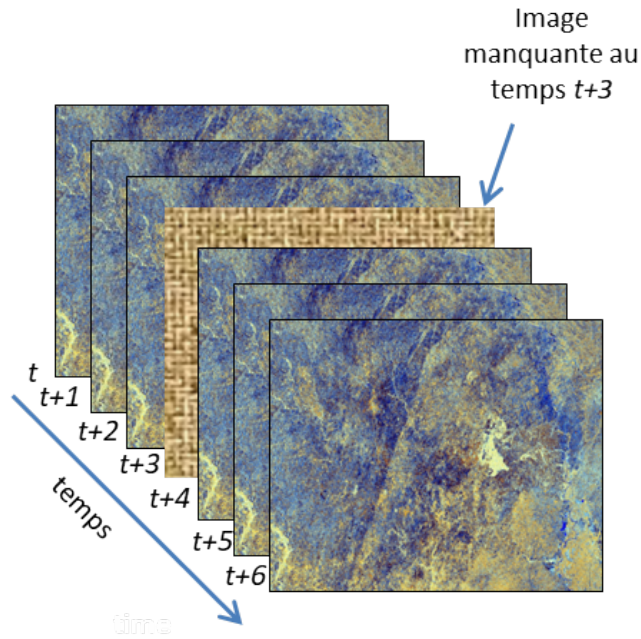


FIGURE 2.2 – Image manquante dans une série temporelle d'images satellites.

2.2 Méthodes deep learning pour la prédiction des séries temporelles d'images satellites

Dans cette section, les principales approches DL utilisées dans la littérature pour la prédiction des STIS sont présentées, ainsi que quelques exemples de leur utilisation dans des problèmes du monde réel. Les modèles basés sur les RNN, les réseaux *feedforward* et sur des architectures hybrides sont présentés.

2.2.1 Architectures basées sur les RNN

2.2.1.1 Réseaux LSTM ou longue mémoire à court terme

Le réseau LSTM a été conçu pour résoudre le problème de la disparition du gradient dans les RNN. Contrairement aux réseaux récurrents standards qui ne comportent qu'une seule couche d'activation, les architectures LSTM comportent quatre couches qui interagissent de manière particulière : la porte d'information, la porte d'oubli, la porte d'entrée et la porte de sortie (toutefois dans les configurations réelles, il peut y avoir plus de portes).

Bien que les architectures LSTM aient été initialement conçues et adaptées pour les données 1D, elles ont également été utilisées pour la prédiction des STIS. Par exemple dans [84], les auteurs ont utilisé un modèle LSTM simple pour la reconstruction des pixels manquants dans des images optiques couvertes par des nuages. Dans un tel modèle, les prédictions sont effectuées pixel par pixel. Les résultats obtenus étant tout à fait satisfaisants, il a été conclu que les réseaux LSTM sont également utilisables pour ce type de problème.

De manière similaire, des réseaux LSTM multicouches ont été utilisés dans [85] pour la prédiction *sequence-to-sequence* de données météorologiques. Avant l'étape d'apprentissage, certaines tâches ont été effectuées pour avoir des séries de données complètes. Par exemple, les valeurs manquantes dans la séquence étaient remplacées par la valeur précédente.

Plus récemment, un nouveau modèle basé sur l'architecture LSTM, pour la prédiction du rendement de la culture de soja à partir d'images satellites et de données météorologiques, a été présenté dans [86]. Les auteurs ont comparé les performances de leur modèle à celles obtenues par les algorithmes multivariés des moindres carrés ordinaires et de la forêt aléatoire (*Random Forest*). Après l'analyse des résultats obtenus, il a été conclu que les STIS combinées aux données météorologiques pouvaient améliorer considérablement la précision des modèles de prédiction des productions du soja.

2.2.1.2 Réseaux TrajGRU

Les architectures ConvLSTM et ConvGRU sont des réseaux LSTM et GRU respectivement, dans lesquelles des opérations de convolutions ont été introduites au niveau des cellules mémoires au lieu des simples opérations matricielles qu'on retrouve dans les cellules LSTM classiques. L'utilisation des réseaux ConvLSTM et ConvGRU pour la prédiction des précipitations ont montré certaines limites. Pour y remédier, les réseaux TrajGRU (*Trajectory-Gated Recurrent Unit*), inspirés des ConvGRU ont été conçus [77]. La particularité du modèle mis en place est sa capacité à apprendre activement la position des connexions récurrentes qui varie. Contrairement au RNN convolutif classique, les connexions récurrentes sont déterminées dynamiquement au fil du temps dans le modèle TrajGRU.

Les résultats des travaux ont montré que ce nouveau modèle était plus efficace que le ConvGRU classique pour la prédiction immédiate des précipitations. Cependant, pour améliorer la précision de la prédiction dans un contexte de données à grande échelle, un nouveau modèle de prédiction de type *sequence-to-sequence*, basé sur le TrajGRU, le ConvGRU et le ConvLSTM, a été proposé dans [87].

2.2.1.3 Réseaux ConvLSTM

Cette architecture a été introduite en 2015 par Xingjian shi et al. dans [76]. Les auteurs ont proposé un modèle pour résoudre un problème de prédiction météorologique, à savoir la prédiction des intensités de précipitations futures. Aux mêmes fins, les auteurs dans [88] ont conçu un nouveau modèle appelé DeepRain basé sur l'architecture ConvLSTM. Les réseaux ConvLSTM ont également été utilisés pour d'autres applications telles que décrites dans [82, 66, 89] (prédiction spatio-temporelle et prédiction des températures de surface de la mer à court et moyen terme). De façon générale, les résultats de ces expérimentations ont montré que l'utilisation des réseaux ConvLSTM pour des tâches de prédiction des STIS est préférable à celle des techniques traditionnelles telles que le *Fully connected LSTM*, la méthode *ROVER*, la régression linéaire, ou les réseaux MLP.

2.2.2 Architectures hybrides

Pour des problèmes de prédiction plus complexes, au lieu d'utiliser une seule architecture, plusieurs peuvent être combinées pour améliorer la qualité des résultats. Parfois, des architectures DL sont alors utilisées en association avec des algorithmes de ML classiques. De telles architectures sont appelées « architectures hybrides ». Elles font référence à une catégorie de méthodes qui intègrent les avantages de différents modèles utilisés individuellement [90].

2.2.2.1 Réseaux LSTM-AdaBoost

L'architecture LSTM-AdaBoost a été conçue pour la prédiction des températures de surface de la mer dans [91]. Cette architecture consiste en la combinaison de deux types de structures différentes, à savoir le réseau LSTM et les algorithmes de *boosting adaptatif* (Adaboost). AdaBoost est une méthode d'apprentissage automatique utilisée pour faire des classifications. Elle est généralement utilisée en combinaison avec de nombreux autres types d'algorithmes de ML pour améliorer les performances des modèles prédictifs. Le terme « boosting adaptatif » est utilisé car les poids sont réaffectés à chaque instance. L'algorithme de Adaboost alloue plus de poids aux instances difficiles à classer et moins à celles qui sont déjà bien traitées. Cet algorithme peut être utilisé à la fois pour les problèmes de régression et de classification. La fonction de régression dans le modèle LSTM-AdaBoost est l'arbre de décision, dont la puissance a été démontrée pour les prédictions [91]. La figure 2.3 donne une représentation simple de la partie AdaBoost de l'architecture proposée dans [91].

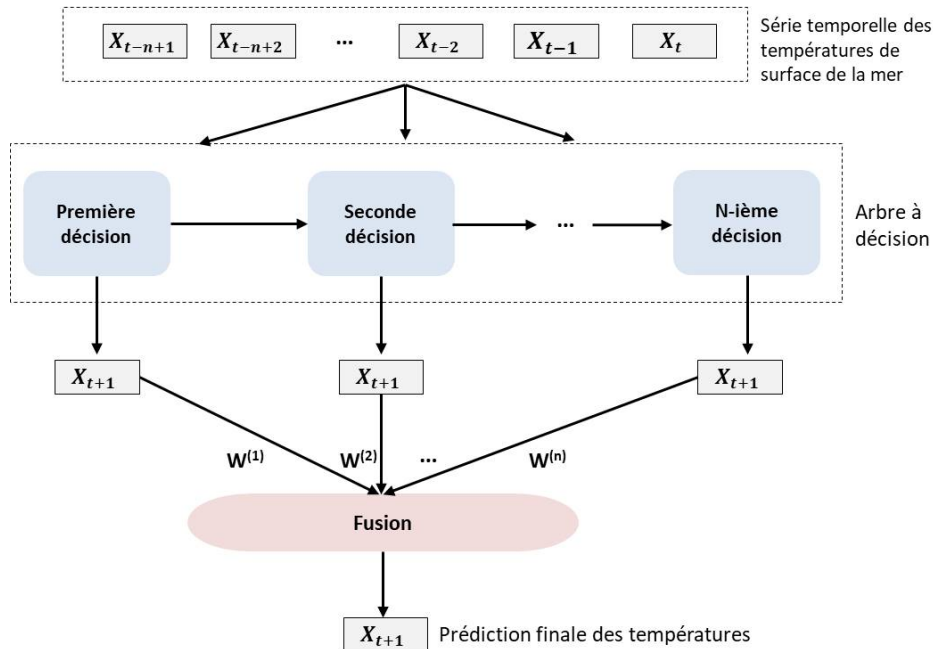


FIGURE 2.3 – Architecture de régression par arbre de décision utilisant AdaBoost proposée dans [91].

Avant l'utilisation des données par ce réseau, les séquences de données ont été désaisonnalisées et normalisées. L'expérimentation pour la prédiction des températures à court et moyen

terme a montré que les résultats obtenus avec l'architecture hybride ont surpassé ceux obtenus avec les algorithmes LSTM, Adaboost, ou SVR (*Support Vector Regression*) séparément. Un aperçu de l'ensemble du modèle LSTM-AdaBoost est illustré à la figure 2.4. La séquence de données en entrée est d'abord fournie aux deux réseaux LSTM et AdaBoost pour des prédictions indépendantes. Ensuite, leurs prédictions respectives sont combinées pour calculer la moyenne qui est considérée comme la prédiction finale [91].

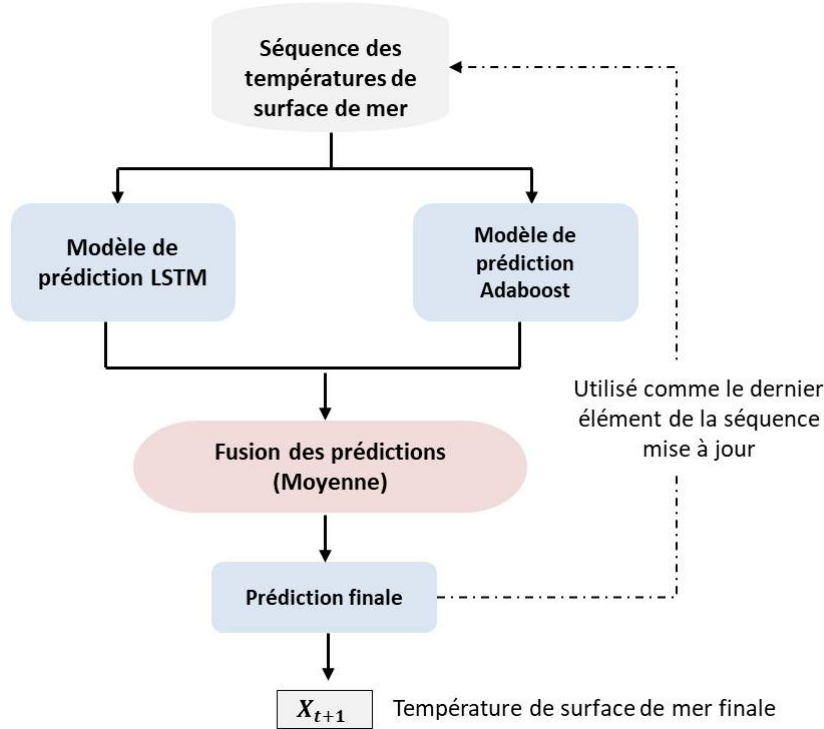


FIGURE 2.4 – Configuration LSTM-AdaBoost simplifiée utilisée dans [91].

2.2.2.2 Réseaux GAN-LSTM

Les réseaux adverses génératifs sont un type d'architecture DL très célèbre qui a connu un succès impressionnant depuis leur création en 2014 [92]. Ces réseaux appelés *Generative Adversarial Network* (GAN) en anglais, ont pour objectif de générer du contenu proche de la réalité (des images par exemple). Le modèle GAN est constitué de deux réseaux (générateur et discriminateur) qui s'opposent l'un à l'autre afin de générer une instance artificielle de données ressemblant fortement à une instance réelle. Le générateur crée de nouvelles données envoyées au discriminateur qui détermine si elles sont vraies ou générées, grâce à une base de données constituée de données réelles. Ainsi, le premier réseau apprend à créer des éléments de plus en plus réalistes tandis que le second apprend à reconnaître au mieux les données réelles [92]. On utilise généralement ces architectures dans plusieurs applications telles que la génération d'objets 3D, les systèmes de traduction images-textes, les sous-titrages, etc.

L'architecture GAN-LSTM consiste en la combinaison d'une architecture d'apprentissage supervisée avec une architecture non supervisée, à savoir les réseaux LSTM et les réseaux (GAN) respectivement. Cette méthode hybride, telle que proposée dans [93], combine les compétences du

générateur fournies par le GAN et la capacité de prévision du réseau LSTM pour la prédiction. La partie GAN du modèle, représentée par la figure 2.5, génère des images à partir d'un ensemble de données aléatoires, tandis que le discriminateur porte le bon jugement sur la nature des données (générées ou réelles)[93].

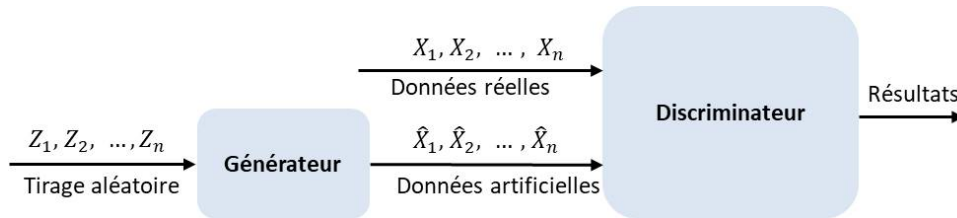


FIGURE 2.5 – Architecture du réseau GAN proposé dans [93]

Dans la même idée, une architecture hybride similaire au GAN-LSTM a été conçue dans [94]. Il s'agit du *Generative Adversarial-ConvGRU* (GA-ConvGRU) [94]. L'architecture proposée possède deux systèmes d'apprentissage adversaires : un générateur basé sur le ConvGRU et un discriminateur basé sur un réseau de neurones à convolution. Les deux systèmes du modèle ont permis une extrapolation plus réaliste et plus précise que le GAN-LSTM [93].

2.2.2.3 Réseaux ConvLSTM-Auto-Encoder

Cette architecture est constituée de deux réseaux fonctionnant en mode encodeur-décodeur. Tout d'abord, la partie encodeur du modèle reçoit et code la série temporelle d'images. Ensuite, les données codées et les états du réseau sont utilisés par le décodeur pour prédire les valeurs une par une. Cette approche a été proposée, par exemple pour prédire la prochaine séquence d'images d'une STIS dans [95]. Pour obtenir de meilleurs résultats, des cellules ConvLSTM ont été introduites à la fois dans l'encodeur et le décodeur du réseau. La qualité des images prédites en termes de résolution n'étant pas très bonne, les auteurs ont suggéré en perspective l'introduction d'un GAN dans le modèle pour remédier à cette insuffisance.

De même, un modèle DL composé d'un réseau encodeur-décodeur avec des unités LSTM convolutives a également été proposé dans [96]. Comme le montre la figure 2.6, les derniers états et la cellule de sortie de l'encodeur sont copiés dans le réseau du décodeur. L'encodeur et le décodeur de cette architecture peuvent être constitués de plusieurs couches LSTM empilées, la sortie d'une couche étant l'entrée de la couche suivante [96].

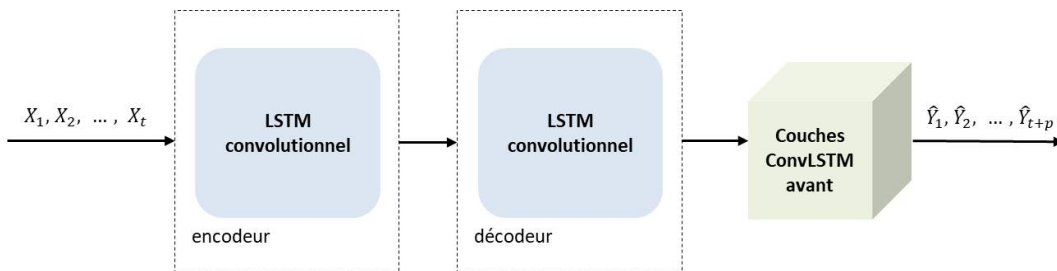


FIGURE 2.6 – Vue d'ensemble de la structure encodeur-décodeur du réseau suggéré dans [96].

2.2.3 Architectures basées sur les réseaux feedforward

2.2.3.1 Réseaux DeepstepFE

Pour reconstituer des données manquantes dans une séquence d'images, Monidipa Das et Soumaya Ghoh ont proposé un modèle DL basé sur une architecture MLP appelé DeepstepFE [97]. DeepstepFE est inspiré du modèle deep-step proposé dans [81] pour la reconstruction des images NDVI manquantes dans une séquence. En effet, les deux modèles, dérivés du modèle DSN (*Deep Stacking Network*), ont été conçus par les mêmes auteurs, l'un étant une version améliorée de l'autre. Selon les résultats obtenus, le modèle DeepstepFE a montré de meilleures performances et avait un temps d'exécution à la fois plus compétitif et raisonnable, comparativement aux autres approches DL classiques pour la prédiction des données manquantes.

2.2.3.2 Réseaux CNN

Dans les problèmes de prédiction des STIS, on peut remarquer que le CNN est généralement utilisé dans le cadre de la fusion de données ou la reconstruction de données manquantes. Par exemple, dans [98], les auteurs ont exploré la fusion d'images optiques et radar (Sentinel-2 et Sentinel-1) pour la reconstruction de NDVI manquant. Pour y parvenir, un modèle basé sur les réseaux CNN a été développé. Les résultats de cette étude ont montré qu'il existe une forte relation entre les données radar et le NDVI, qui peut être capturée par un modèle CNN. Selon la configuration de cette architecture représentée par la figure 2.7, une série d'images Sentinel-1, passe par le réseau CNN et produit en sortie l'image NDVI du capteur Sentinel-2 correspondante. Le réseau proposé est constitué de trois couches convolutives ($L = 3$) et le pas temporel (Δ) entre les images en entrée est de 12 jours.

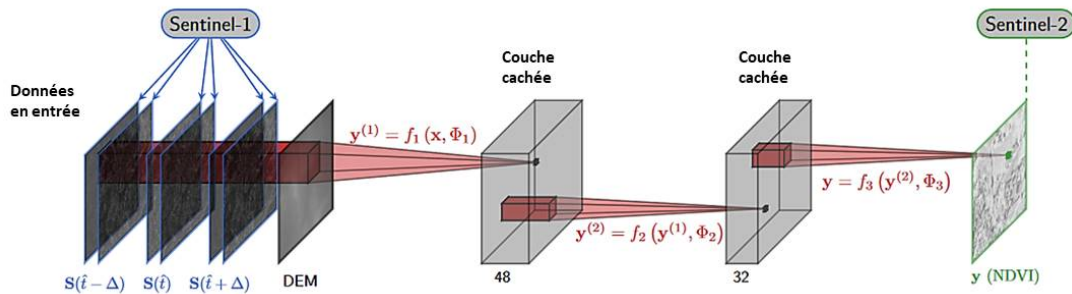


FIGURE 2.7 – Architecture CNN de la méthode proposée dans [98].

Le modèle CNN proposé dans [98] est défini selon la fonction décrite par l'équation 2.3.

$$y = f(x, \Phi) = f_L(f_{L-1}(\dots f_1(x, \Phi_1), \dots, \Phi_{L-1}), \Phi_L) \quad (2.3)$$

où x représente l'ensemble de données en entrée, y est le NDVI estimé, $L = \{1, 2, 3\}$ est le numéro de la couche, et $\Phi \triangleq (\Phi_1, \dots, \Phi_L)$ désigne l'ensemble des paramètres à apprendre [98].

Un autre cas d'utilisation du CNN est présenté dans [83] où un système de prédiction spatial-temporel-spectral basé sur l'architecture CNN est proposé pour la reconstruction des données manquantes rencontrées dans les séries d'images satellitaires. Trois tâches principales de reconstruction de données manquantes (causées notamment par le dysfonctionnement des capteurs MODIS et Landsat ETM ou par un fort taux de couverture nuageuse sur les images) ont été considérées par ces travaux. Les résultats obtenus ont montré que le modèle CNN proposé est apte à donner de bons résultats pour la reconstruction des lignes mortes dans la bande-6 du capteur Aqua-MODIS.

2.3 Optimisation et métriques d'évaluation

Cette section présente les algorithmes d'optimisation et les métriques de performance utilisés dans la littérature pour la prédiction des STIS.

2.3.1 Fonctions d'optimisation

Au cours de l'étape d'apprentissage des réseaux de neurones, des algorithmes d'optimisation sont utilisés pour minimiser l'erreur (différence entre la sortie prédite et la sortie réelle). Cette optimisation se fait par le réajustement ou la mise à jour des poids afin d'obtenir des prédictions plus précises. L'algorithme de rétropropagation pour la mise à jour des poids est décrit par l'algorithme 1 [27]. Dans la littérature, plusieurs stratégies d'optimisation existent et le choix de l'algorithme le plus approprié pour un modèle DL est crucial [99, 64].

Algorithme 1 : Algorithme général de rétropropagation pour l'optimisation des poids dans un réseau de neurones. *network* représente le vecteur initial des paramètres.

Entrées : *network*, training-records, learning-rate

Sorties : *network*

```

1 network ← initialiser aléatoirement les poids;
2 tant que network ne converge pas faire
3   pour chaque exemple de données d'entraînement faire
4     network-output ← neural-network-output (network, exemple) ; // Calculer la
       sortie pour l'élément donné en entrée
5     exemple-err ← target-output – network-output ; // Calculer l'erreur et la
       différence Delta pour les neurones dans la couche de sortie
6      $W_{j,i} \leftarrow W_{j,i} + \alpha \times \alpha_j \times Err_i \times g'(input_{sum_i})$  ; // Mettre à jour les poids menant
       à la couche de sortie
7     pour chaque couche suivante du réseau faire
8        $\Delta_j \leftarrow g'(input_{sum_j}) \sum_i W_{j,i} \Delta_i$  ; // calculer l'erreur à chaque nœud
9        $W_{k,j} \leftarrow W_{k,j} + \alpha \times \alpha_k \times \Delta_j$  ; // mettre à jour les poids menant à la couche
10 retourner network ; // Paramètres résultants
```

La descente de gradient est l'un des algorithmes d'optimisation les plus populaires et constitue la manière la plus courante d'optimiser les réseaux de neurones. Il en existe trois variantes de base, qui dépendent de la quantité de données utilisées pour la descente de gradient : la des-

cente de gradient par lots, la descente de gradient stochastique ou *Stochastic Gradient Descent* (SGD), et la descente de gradient par mini-lots. Bien que la fonction SGD se soit avérée être une méthode d'optimisation efficace et efficiente pour les algorithmes DL, certaines variantes ont été introduites, notamment le moment adaptatif (Adam) et le *Root Mean Square Propagation* (RMSProp) [100, 101]. Dans la littérature, les algorithmes SGD, Adam et RMSProp susmentionnés ont été largement utilisés pour l'entraînement des modèles de prédiction dans les STIS. Ils seront présentés dans les paragraphes qui suivent.

2.3.1.1 Algorithmes SGD

Dans les algorithmes SGD, tous les paramètres sont mis à jour pour chaque itération, au lieu de calculer le gradient de la fonction de coût pour l'ensemble des données d'apprentissage. Le cœur de cette approche consiste à minimiser une fonction qui peut être écrite comme la somme de fonctions différentiables. Elle convient aux problèmes d'optimisation sans contrainte et s'est avérée très efficace dans une variété de réseaux de neurones profonds [81, 83, 86]. L'équation (2.4) décrit la règle de mise à jour du SGD.

$$W_{k+1} = W_k - \eta \Delta J(W_k) \quad (2.4)$$

où W_{k+1} est la valeur mise à jour après la k -ième itération, W_k est la valeur initiale à la k -ième itération, η représente le pas, et ΔJ est le gradient de la fonction J .

2.3.1.2 Algorithme RMSProp

La fonction d'optimisation *Root Mean Square Propagation* (RMSProp) proposée par Geoffrey Hinton est généralement un choix approprié pour les RNN [96, 94]. Dans l'algorithme RMSProp, le taux d'apprentissage de chaque paramètre est automatiquement ajusté par une moyenne des gradients carrés à décroissance exponentielle. L'auteur du RMSProp suggère de fixer la valeur par défaut du taux d'apprentissage à $\eta = 0.001$.

L'équation (2.5) définit l'optimiseur RMSProp.

$$\begin{aligned} E[g^2](t) &= \beta E[g^2](t-1) + (1-\beta) \left(\frac{\delta c}{\delta w}\right)^2 \\ w_{ij}(t) &= w_{ij}(t-1) - \frac{\eta}{\sqrt{E[g^2]}} \frac{\delta c}{\delta w_{ij}} \end{aligned} \quad (2.5)$$

où $E[g]$ est la moyenne mobile au carré des gradients, $\frac{\delta c}{\delta w}$ est le gradient de la fonction de coût par rapport aux poids, w_{ij} est le poids du neurone j vers le neurone i , η est le taux d'apprentissage, et β est le paramètre de moyenne mobile.

2.3.1.3 Algorithme Adam

La fonction *Adaptive Moment Estimation Adam* est de loin l'une des fonctions d'optimisation les plus appréciées pour les algorithmes DL. Cet algorithme combine les capacités du RMSProp

et de la fonction *Momentum* décrite dans [64]. Dans la littérature, la plupart des études utilisent la fonction *Adam* pour la prédiction des STIS. En effet, cet algorithme est simple à mettre en œuvre, efficace sur le plan informatique, nécessite peu de mémoire et de réglage, et est approprié pour les problèmes avec des gradients très bruités ou épars. La fonction *Adam* est définie par l'équation (2.6) et l'algorithme 2 [100].

$$\begin{aligned}
 v_t &= \beta_1 * v_{t-1} - (1 - \beta_1) * g_t \\
 s_t &= \beta_2 * s_{t-1} - (1 - \beta_2) * g_t^2 \\
 \Delta w_t &= -\eta \frac{v_t}{\sqrt{s_t + \epsilon}} * g_t \\
 w_{t+1} &= w_t + \Delta w_t
 \end{aligned} \tag{2.6}$$

où :

η désigne le taux d'apprentissage initial ;

g_t représente le gradient au temps t ;

v_t représente la moyenne exponentielle du gradient ;

s_t désigne la moyenne exponentielle du carré du gradient ;

β_1 et β_2 sont des hyperparamètres ;

et chaque paramètre w^j est remplacé par w pour plus de clarté.

2.3.2 Métriques d'évaluation

Le choix des métriques d'évaluation appropriées est une étape cruciale pour tout projet en apprentissage automatique. En effet, ces mesures permettent d'évaluer la performance du modèle et de déterminer s'il peut produire de bonnes prédictions sur de nouvelles données (capacité de généralisation). Dans la littérature, différentes métriques sont testées pour évaluer les modèles prédictifs des STIS. On classera ces mesures selon six groupes d'utilisation.

2.3.2.1 Métriques utilisées dans les problèmes de régression

Dans la plupart des travaux analysés, on peut observer qu'au moins une des mesures de régression suivantes est utilisée pour l'évaluation des modèles proposés : l'erreur quadratique moyenne ou *Mean Squared Error* (MSE), l'erreur absolue moyenne ou *Mean Absolute Error* (MAE), et le coefficient de détermination noté R^2 (équivalent au carré du coefficient de corrélation de Pearson r). Ces métriques définies respectivement par les équations (2.7), (2.8) et (2.9), sont considérées en effet comme les meilleures pour l'évaluation des problèmes de régression. En outre, certaines versions dérivées, telles que la racine de l'erreur quadratique moyenne ou *Root Mean Square Error* (RMSE), l'erreur moyenne absolue en pourcentage ou *Mean Absolute Percentage Error* (MAPE), l'erreur quadratique moyenne normalisée ou *Normalized Root Mean Square Error* (NRMSE) et le biais moyen normalisé ou *Normalized Mean Bias Error* (nMBE),

Algorithme 2 : Algorithme d'optimisation stochastique Adam. g_t représente le produit par élément $g_t \odot g_t$. Les meilleurs paramètres par défaut pour les problèmes d'apprentissage automatique testés sont $\alpha = 0,001$, $\beta_1 = 0,9$, $\beta_2 = 0,999$ et $\epsilon = 10^{-8}$. Toutes les opérations sur les vecteurs sont par éléments. $\beta_1 t$ et g_t désignent β_1 et β_2 à la puissance t .

Données : α : Taille du pas

Données : $\beta_1, \beta_2 \in [0, 1]$: Taux de décroissance exponentielle pour les estimations du moment

Données : $f(\theta)$: Fonction objectif stochastique avec paramètres

Données : θ_0 : Vecteur initial des paramètres

```

1  $m_0 \leftarrow 0$  : Initialiser le vecteur du 1er moment;
2  $v_0 \leftarrow 0$  : Initialiser le vecteur du 2ème moment;
3  $t \leftarrow 0$  : Initialiser le pas de temps;
4 tant que  $\theta_t$  ne converge pas faire
5    $t \leftarrow t + 1$ ;
6    $g_t \leftarrow \Delta_{\theta} f(\theta_{t-1})$ ; // Obtenir les gradients par rapport à l'objectif stochastique au
   pas de temps  $t$ 
7    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ ; // Mise à jour de l'estimation du premier moment biaisé
8    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ ; // Mise à jour de l'estimation biaisée du second moment
   brut
9    $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ ; // Calculer la première estimation du moment corrigée du biais
10   $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ ; // Calcul de la deuxième estimation brute du moment corrigée du
   biais
11   $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$ ; // Mise à jour des paramètres
12 retourner  $\theta_t$ ; // Paramètres résultants

```

sont également utilisées pour évaluer les performances des modèles. Plus de détails sur les métriques de régression sont disponibles dans [102].

Pour deux images x et y de taille $r \times c$, la MSE est définie de la façon suivante :

$$MSE = \frac{1}{r \times c} \sum_{i=0}^{r-1} \sum_{j=0}^{c-1} (x(i, j) - y(i, j))^2 \quad (2.7)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |(p_i - o_i)| \quad (2.8)$$

où n représente la taille de l'échantillon, p_i représente les valeurs prédites et o_i les valeurs observées.

$$R^2 = r^2 = \left(\frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \right)^2 \quad (2.9)$$

où x_i et y_i représentent l'intensité du i^{ieme} pixel pour la première et la seconde image respectivement. \bar{x} et \bar{y} sont l'intensité moyenne de la première et de la deuxième image respectivement, et n la taille de l'échantillon.

2.3.2.2 Métriques utilisées dans les problèmes de vision par ordinateur

Les métriques suivantes, généralement utilisées dans les problèmes de vision par ordinateur, sont également utilisées par certains auteurs : le *Peak Signal to Noise Ratio* (PSNR), l'indice de similarité structurelle ou *Structural Similarity* (SSIM), le *Spectral Angle Mapper* (SAM) et la différence entre les images ou *Difference Image* (DI) [83, 84, 97, 98].

Le PSNR calcule le rapport signal/bruit maximal en décibels entre deux images. Son but est d'évaluer la qualité entre l'image originale et l'image compressée par la valeur du rapport. Plus le PSNR est élevé, plus la qualité de l'image reconstruite est bonne [103]. Elle est exprimée par l'équation (2.10).

$$PSNR = 10 \times \log_{10}\left(\frac{d^2}{MSE}\right) \quad (2.10)$$

où d est la gamme dynamique du signal et MSE est l'erreur quadratique moyenne définie par l'équation (4.6). Dans le cas standard d'une image codée sur 8 bits, $d = 255$.

Quant à l'indice SSIM, il permet de mesurer la similarité structurelle entre les images prédites et les images originales. Les valeurs proches de 1 indiquent de bonnes similarités [104] et celle proches de 0 une mauvaise similarité. L'équation (2.11) donne la définition mathématique du SSIM.

$$SSIM(x, y) = \frac{(2\bar{x}\bar{y} + c_1)(2\sigma_{xy} + c_2)}{(\bar{x}^2 + \bar{y}^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (2.11)$$

où x, y sont les images comparées, \bar{x} \bar{y} représentent les moyennes de x et y , σ_{xy} est la covariance croisée des images, σ_x^2 et σ_y^2 représentent les variances, et c_1, c_2 sont des constantes utilisées pour stabiliser la division lorsque le dénominateur est très faible [103].

En ce qui concerne la métrique SAM, elle est basée sur un concept physique qui mesure la similarité angulaire entre le spectre de chaque pixel de l'image et les spectres de référence [105].

Enfin, le DI résulte simplement des opérations de soustraction entre les valeurs de chaque pixel de l'image d'entrée et les valeurs des pixels correspondant à l'image prédite. Un DI sombre avec des valeurs proches de 0 indique que la prédiction est bonne.

2.3.2.3 Métriques utilisées dans les problèmes statistiques

Les métriques présentées dans cette section sont généralement utilisées dans les problèmes rencontrés en statistique ou en probabilité. Il s'agit du taux de fausses alarmes *False Alarm Rate* (FAR), de la probabilité de détection *Probability Of Detection* (POD), de l'indice de réussite critique *Critical Success Index* (CSI), du score de compétence *Heidke Skill Score* (HSS), du coefficient de corrélation (CC) et de la corrélation de Pearson. Les métriques POD, FAR et CSI, qui sont les plus utilisées dans la littérature, sont définies par les équations suivantes :

$$POD = \frac{TP}{TP + FP} \quad (2.12)$$

$$FAR = \frac{FN}{TP + FN} \quad (2.13)$$

$$CSI = \frac{TP}{TP + FP + FN} \quad (2.14)$$

où TP , FP et FN représentent respectivement le nombre de vrais positifs, le nombre de faux positifs et le nombre de faux négatifs respectivement dans les résultats des prédictions.

2.3.2.4 Métriques utilisées pour mesurer la performance

En plus des mesures classiques utilisées pour les problèmes en apprentissage profond, certains auteurs ont relevé d'autres valeurs telles que le temps d'apprentissage du modèle, le temps d'exécution ou le taux d'utilisation de la mémoire de l'ordinateur pendant l'étape d'apprentissage. Dans le cadre de notre étude, ces valeurs sont appelées métriques de performance. Elles ont été utilisées dans le cadre des travaux présentés dans [96, 81, 83].

2.3.2.5 Métriques utilisées dans les problèmes de classification

Pour évaluer la performance du modèle de fusion de données, les auteurs dans [106] ont utilisé les mesures communément employées pour les problèmes de classification (précision, Kappa, et F-score).

La précision est une mesure permettant de calculer la proportion de réelles identifications positives. Elle est exprimée par l'équation (2.15), où TP représente le nombre de vrais positifs et FP le nombre de faux positifs.

$$Précision = \frac{TP}{TP + FP} \quad (2.15)$$

La métrique Kappa Cohen (K) sert généralement à mesurer la performance du classifieur en la comparant à celle d'un autre. Elle est définie par l'équation (2.16), où P_o est la proportion de concordance observée et P_e celle de concordance aléatoire.

$$k = \frac{P_o - P_e}{1 - P_e} \quad (2.16)$$

La métrique F -*measure* ou F -*score* (score F1) est la moyenne harmonique de la précision et du rappel. Elle est utilisée en général pour des problèmes à deux classes au moins, et est définie mathématiquement par l'équation (2.17) où TP est le nombre de vrais positifs, FN le nombre de faux négatifs et FP le nombre de faux positifs.

$$F - score = \frac{TP}{TP + \frac{1}{2}(FN + FP)} \quad (2.17)$$

2.3.2.6 Autres métriques

Dans [85, 89], les auteurs ont plutôt proposé de nouvelles mesures d'évaluation, et ont obtenu de meilleures performances pour leur prédiction par rapport à celles obtenues avec les mesures

classiques. Cette démarche a servi à démontrer l'influence des métriques d'évaluation dans la performance des modèles.

2.4 Limites et propositions de solutions

Les techniques d'apprentissage profond sont aujourd'hui utilisées dans la plupart des applications de la vie réelle et le domaine de la télédétection ne fait pas exception. Cependant, les études concernant l'utilisation d'algorithmes DL pour la prédiction des STIS sont encore récentes et présentent certains verrous qu'il est nécessaire de prendre en compte. Cette section résume les principaux facteurs qui limitent l'utilisation du DL pour la prédiction du STIS.

2.4.1 Manque de données d'entraînement

Pour obtenir des résultats satisfaisants avec les algorithmes DL, il est nécessaire d'utiliser un grand ensemble de données d'entraînement. Cependant, contrairement aux modèles qui utilisent des images classiques, il y a un manque important de données de télédétection étiquetées en général. En effet, il est très facile par exemple d'obtenir des bases de données comportant de milliers d'images naturelles sur internet pour la plupart des catégories (fleurs, animaux, fruits, etc.). Pour des problèmes de classification en télédétection, des efforts sont quand même faits et on retrouve actuellement quelques jeux de données disponibles, même si le problème de quantité d'éléments dans ces bases de données persiste. Mais lorsqu'il s'agit des problèmes de prédiction sur des STIS d'observation de la terre, la communauté fait face à un réel manque de données prétraitées pour l'entraînement des modèles DL. Pourtant, la disponibilité des jeux de données d'entraînement permettrait d'encourager la recherche dans le domaine et de valider les modèles proposés par les scientifiques.

Il est vrai qu'il existe plusieurs missions satellites qui fournissent une grande quantité de données. Cependant, la collecte et le prétraitement de ces images afin de constituer de longues séries temporelles pour la prédiction des STIS n'est pas une tâche simple [74]. Les images d'observation de la Terre sont complexes et variées (notamment en raison des métadonnées géographiques, de leur taille, dimension, ou de leur caractère multispectral), contrairement aux ensembles de données d'imagerie traditionnelles qui sont constitués d'éléments relativement simples (taille, dimension, etc.)[107]. De plus, l'utilisation du DL pour prédiction des STIS (spécialement pour la prédiction des changements des classes d'utilisation de sol) est encore à cette date un pan de recherche assez récent, ce qui expliquerait également ce manque de données. Pour faire face à cette difficulté, il est donc nécessaire de constituer et rendre disponible publiquement des jeux de données « prêtes à l'emploi ».

2.4.2 Opérations de prétraitement des STIS

Grâce aux satellites qui gravitent actuellement autour de la Terre, il est possible d'obtenir des images avec des temps de revisite très courts. Dans leur état initial, ces images contiennent une grande quantité d'informations et ne sont pas directement utilisables dans la pratique sans subir certains traitements.

Pour constituer une STIS, chaque image de la série doit être téléchargée, prétraitée et traitée pour en extraire les informations utiles avec lesquelles l'utilisateur final veut travailler [108]. Par exemple, des compositions colorées ou des indices de végétation peuvent être produits avant que les séries ne soient utilisées comme jeux de données d'entraînement. La figure 2.8 présente le processus de traitement des données de télédétection [108]. Tout d'abord, les images sont acquises à partir des plateformes des fournisseurs d'images. Ensuite, les données sont traitées par trois opérations principales, notamment l'étape de prétraitement (corrections radiométriques, atmosphériques ou géométriques, etc.), l'étape de traitement à valeur ajoutée (fusion, mosaïque, filtrage, etc.) et l'étape d'abstraction de l'information (classification, segmentation, composition colorée, etc.). Enfin, les données traitées sont utilisées pour de grandes applications thématiques de télédétection telles que la prédiction de changement, la détection d'évènements, la classification de l'utilisation des terres, la surveillance de la sécheresse, etc. [32]

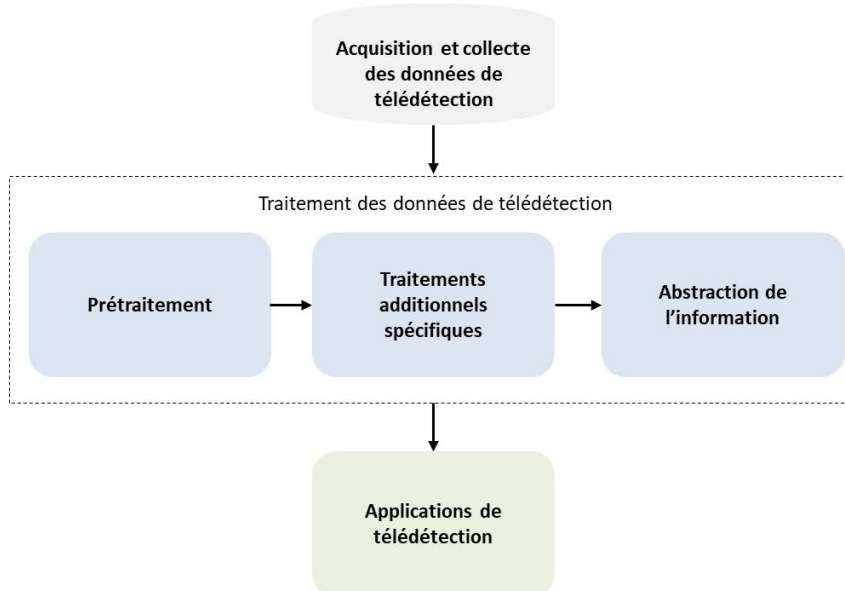


FIGURE 2.8 – Vue d'ensemble du processus de traitement des données de télédétection.

Les images satellites étant complexes et variées, leur traitement nécessite un temps important et généralement l'intervention d'un expert dans le domaine de la télédétection. En effet, l'expert saurait par exemple quelle information extraire en fonction du type de problème à résoudre. De plus, lorsque des données optiques sont utilisées pour construire les séries temporelles, la présence de nuages sur les images est un problème important. Bien qu'il existe certaines techniques pour supprimer les nuages des images, la qualité des données résultantes est souvent altérée, ce qui entraîne également une diminution de la précision des prédictions. Par conséquent, une alternative aux images nuageuses est l'utilisation d'images radar. Cependant, cette solution représente également un autre défi car la manipulation des images radar est plus complexe que celle des images optiques [109].

Comme évoqué dans la section 2.4, la disponibilité des jeux de données d'entraînement permettrait d'encourager la recherche dans le domaine et de valider les modèles proposés. Mais lorsque des prédictions doivent être faites des STIS, il n'est en général pas possible d'utiliser des

données représentant une autre localité pour l'apprentissage du modèle. Face à cette difficulté, il est recommandé de proposer et de rendre publics des outils (workflow, scripts) permettant de faciliter le téléchargement et le prétraitement automatique des images, sans l'intervention d'un expert en télédétection. Cela permettrait un gain de temps considérable et de mettre facilement à la disposition du public des *datasets* pour l'entraînement des modèles DL.

2.4.3 Conception des architectures DL

Dans certains cas, des architectures plus profondes (plus complexes), avec un très grand nombre de données d'apprentissage, permettent aux modèles d'obtenir des résultats satisfaisants et de bien généraliser sur des problèmes complexes. Cependant, les STIS sont constitués d'images extrêmement grandes en termes de dimensions et certaines architectures sont difficilement utilisables avec ce type de données à grande échelle. Par exemple, il a été montré dans [66] que l'utilisation de l'architecture ConvLSTM n'est pas recommandée lorsque la taille des images et la longueur des séquences augmentent. Un redimensionnement important des images permet d'utiliser moins de ressources mais diminue également la qualité des données prédites. Or, la qualité des images obtenue en sortie en termes de résolution est importante, surtout pour des cas particuliers des problèmes de prédictions de changement de l'utilisation des sols par exemple. En effet, pour une meilleure compréhension du phénomène étudié, les images prédites doivent conserver le maximum d'informations et avoir la meilleure résolution possible.

Pour ce faire, il est donc nécessaire de concevoir des architectures DL adéquates ou de trouver des méthodes spéciales pour pouvoir réaliser de bonnes prédictions des STIS, sans perdre les informations essentielles et tout en assurant une consommation de ressources informatiques raisonnable. Une façon de traiter les images de grande taille par exemple est souvent de les diviser en zones plus petites qui sont utilisées comme jeux de données d'entrée pour l'apprentissage du modèle. Des stratégies de prétraitement et de post-traitement pourraient également être mises en place pour restaurer la qualité des images prédites. Pour des utilisateurs non aguerris, il serait également intéressant d'intégrer les algorithmes DL dans des bibliothèques ou logiciels de traitement d'images, sous forme de boîte noire, pour faciliter les prédictions des STIS.

2.4.4 Réutilisation des modèles

L'apprentissage par transfert (*transfer learning*) est une méthode en apprentissage automatique qui permet de résoudre un problème en utilisant des connaissances acquises lors de la résolution d'un autre problème assez similaire. Il s'agit en quelque sorte d'un transfert de connaissances d'un modèle à un autre. Par exemple, pour résoudre un problème de reconnaissance de camions, un modèle entraîné pour la reconnaissance de voitures peut être utilisé comme point de départ d'apprentissage. Ce type d'apprentissage est surtout retrouvé dans les domaines du traitement de langage naturel et de la vision par ordinateur. En effet, ces domaines nécessitent généralement des temps de calcul élevés ainsi que des ressources importantes [32].

Dans le cadre de l'utilisation des images naturelles, les modèles DL sont généralement réutilisables sans difficultés majeure quel que soit le type d'image. En revanche, ce n'est pas le cas

pour les images satellites. En effet, un modèle entraîné par exemple sur des images optiques de 30 mètres de résolution ne produira pas les mêmes résultats avec des images radar ayant les mêmes résolutions ou de résolutions différentes. De plus, des modèles entraînés sur le même type de données mais correspondant à des zones géographiques différentes ne produiront pas les mêmes résultats. L'application de l'apprentissage par transfert pour la prédiction des STIS n'est donc pas évidente. Comme piste de solution, il serait intéressant d'explorer les méthodes de *meta-learning*, qui permettent aux modèles d'apprendre à apprendre [110, 111]. Ce qui permettrait par exemple de considérer différentes zones géographiques comme des tâches différentes et de concevoir des modèles pouvant être réutilisés pour de nouvelles zones. La figure 2.9 proposée par Hospedales et al. dans [111], donne un aperçu des principales méthodes de *meta-learning*.

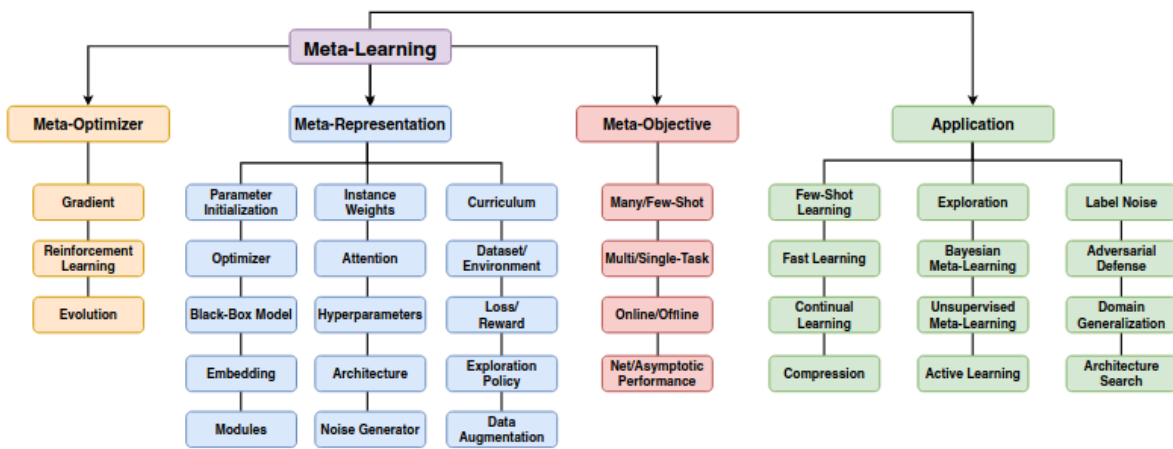


FIGURE 2.9 – Aperçu des méthodes de *meta-learning* [111].

Dans ce chapitre, les travaux qui portent sur l'utilisation DL pour la prédiction des STIS ont été étudiés. Les principaux domaines d'application et les éléments tels que les fonctions d'optimisation et les métriques d'évaluation utilisées par les auteurs ont été présentés. Le manque de jeux de données publics constitués d'images d'observation de la Terre et la difficulté de prétraitement des images satellites ont été mentionnés comme des limites à cet axe de recherche. Dans le chapitre 3, un workflow de collecte et de prétraitement par lot d'images Sentinel-1 est proposé comme solution à ce verrou. Ce workflow a permis de constituer trois jeux de données d'entraînement mis à la disposition de la communauté scientifique.

Chapitre 3

Constitution d'une base d'images à l'aide d'un workflow de collecte et de prétraitement automatisé par lot de données

Objectifs

L'acquisition et le prétraitement des données constituent une phase très importante dans tout processus d'apprentissage. Les images optiques étant très souvent couvertes de nuages surtout dans les zones tropicales (forestières), des images radar ont été utilisées dans le cadre de cette thèse. L'objectif de ce chapitre est de présenter d'abord les caractéristiques des données utilisées dans nos travaux, notamment les images issues des satellites Sentinel-1. Ensuite, les zones d'études choisies sont présentées. Enfin, les outils automatisés mis en place pour faciliter la collecte des images et les différents types de traitements appliqués à ces images pour constituer les jeux de données d'entraînement sont décrits.

Sommaire

3.1	Description des données Sentinel-1	50
3.1.1	Généralités	50
3.1.2	Modes d'acquisition	50
3.1.3	Types de produits	51
3.2	Sites d'études	52
3.2.1	Réserve de Faune de Togodo	52
3.2.2	Parc National de Bouba-Ndjida	54
3.2.3	Réserve de Biosphère du Dja	56
3.3	Collecte et prétraitement	57
3.3.1	Collecte	57
3.3.2	Prétraitement	60
3.3.3	Prétraitement automatisé par lot avec GPT	64
3.4	Résultats d'implémentation du workflow	66
3.4.1	Outils utilisés	66
3.4.2	Description des datasets constitués	66

3.1 Description des données Sentinel-1

3.1.1 Généralités

Les satellites Sentinel-1 sont les premiers de la série Sentinel comprenant 6 missions (Sentinel-1 à Sentinel-6) d'observation de la Terre. Il s'agit d'une constellation de deux satellites, Sentinel-1A (S1A) et Sentinel-1B (S1B), qui partagent le même plan orbital avec un déphasage de 180° . Les satellites S1A et S1B ont été lancés respectivement le 3 avril 2014 et le 25 avril 2016. Chaque satellite survole une même zone tous les 12 jours, ce qui permet aux deux satellites de fournir des images d'une zone tous les 6 jours, de jour comme de nuit, quelles que soient les conditions météorologiques. Comme présenté sur la figure 3.1, la totalité du globe est couverte par les satellites Sentinel-1.

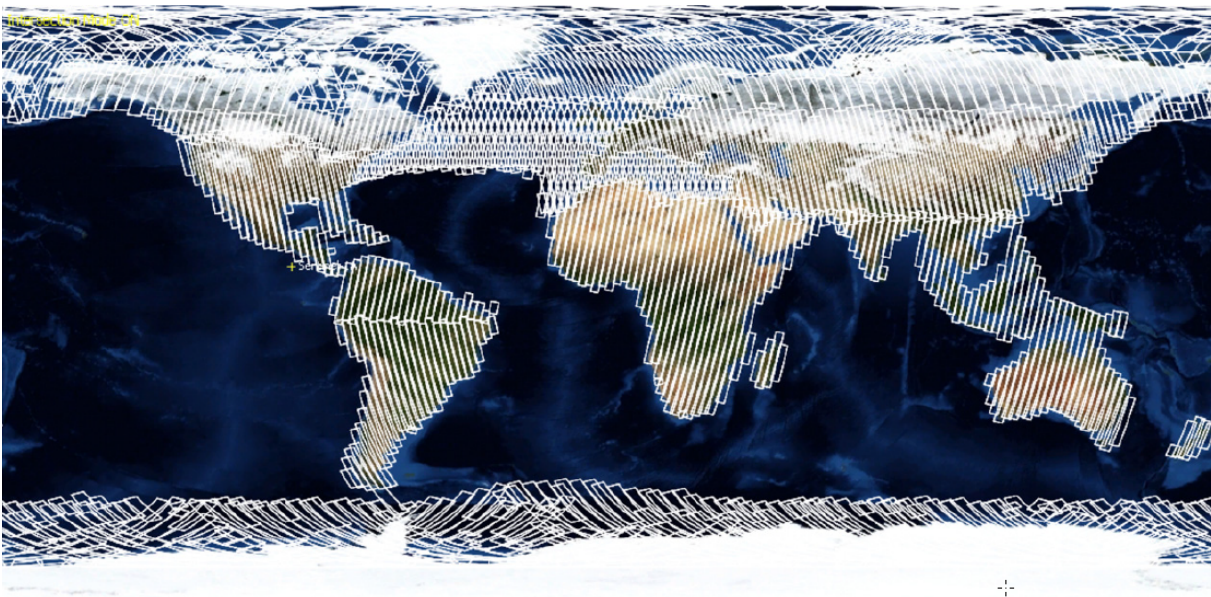


FIGURE 3.1 – Couverture géographique du satellite Sentinel-1 [112].

Les images radar S1 sont utilisées pour de nombreuses applications, notamment pour la cartographie des terres émergées : forêts, eau et sols, agriculture, etc. Pour l'étude des ressources forestières abordées dans le cadre de cette thèse, les images S1 peuvent jouer un rôle important grâce à la détection des coupes claires et partielles, à la classification des types de forêts, à l'estimation de la biomasse et à la détection des feux de forêt, etc. Les séries d'images peuvent également permettre la détection et la prédiction des phénomènes de déforestation dans le monde, et ainsi soutenir les actions de gestion durable des forêts.

3.1.2 Modes d'acquisition

Les satellites Sentinel-1 sont dotés d'instruments de type radar à synthèse d'ouverture (RSO) qui fonctionnent en bande C. Ces capteurs appelés C-SAR, permettent d'acquérir les données couvrant de larges zones (jusqu'à $400\text{ km} \times 400\text{ km}$) et pouvant atteindre $5\text{ m} \times 5\text{ m}$ de résolution selon 4 modes exclusifs :

- mode *Stripmap* (SM) : il s’agit d’un mode d’imagerie RSO standard avec une couverture de $80\text{ km} \times 80\text{ km}$ et une résolution de $5\text{ m} \times 5\text{ m}$. Le mode SM est uniquement utilisé pour les petites îles et sur demande pour des événements extraordinaires tels que la gestion des urgences.
- mode *Interferometric Wide swath* (IW) : dans ce mode considéré comme le mode par défaut sur les terres, la superficie de zone couverte (fauchée) est de $250\text{ km} \times 250\text{ km}$ et la résolution est de $5\text{ m} \times 20\text{ m}$. Les principaux domaines d’application des données issues de ce mode comprennent la foresterie, l’agriculture, la cartographie des changements urbains, etc. Il s’agit du mode d’acquisition des données utilisées pour nos expérimentations.
- mode *Extra Wide swath* (EW) : le mode EW fournit une couverture très large de la fauchée au détriment de la résolution spatiale. Ici les prises de vue se font sur une largeur de 400 km avec une résolution de $20\text{ m} \times 40\text{ m}$.
- mode *Wave* (WV) : les données sont acquises en bandes dans de petites scènes de $20\text{ km} \times 20\text{ km}$ appelées vignettes, avec une résolution de $5\text{ m} \times 5\text{ m}$, tous les 100 km . WV est le mode opérationnel de Sentinel-1 au-dessus de l’océan. On utilise surtout ce mode pour l’étude des vents marins.

Grâce à une chaîne d’émission et deux chaînes de réception parallèles pour les polarisations horizontales (H) et verticales (V), les instruments RSO en bande C de Sentinel-1 fonctionnent en simple polarisation (HH ou VV) et double polarisation (HH+HV ou VV+VH). Les produits acquis en mode SM, IW et EW sont disponibles en polarisation simple ou double tandis que les produits acquis en WV sont uniquement disponibles en simple polarisation.

3.1.3 Types de produits

Les images Sentinel sont mises systématiquement et gratuitement à la disposition de tous les utilisateurs, y compris le grand public et les utilisateurs scientifiques et commerciaux. Tous les produits sont distribués dans le format SAFE (Sentinel Standard Archive Format for Europe). Pour chaque mode, les produits sont disponibles selon trois niveaux (Level), correspondant généralement au niveau de traitement : niveau-0, niveau-1 et niveau-2.

3.1.3.1 Produits de Niveau-0

Les produits de Niveau-0 (Level-0) correspondent aux données brutes RSO non focalisées, compressées par la méthode FDBAQ (*Flexible Dynamic Block Adaptive Quantization*) [113] et dotées d’informations supplémentaires pour faciliter le traitement. Ces données doivent être décompressées et traitées au préalable afin de pouvoir être utilisables. Tous les autres produits sont obtenus par le traitement de ceux de Niveau-0. Chaque mode peut potentiellement générer des produits *Single Look Complex* (SLC) de Niveau-1, *Ground Range Detected* (GRD) de Niveau-1 et *Ocean* de Niveau-2. Pour le mode WV en particulier, les produits de Niveau-0 ne sont pas distribués systématiquement.

3.1.3.2 Produits de Niveau-1

Les produits utilisés la plupart du temps dans les applications courantes sont de Niveau-1. Ils constituent les données de base à partir desquelles les produits de Niveau-2 sont dérivés. On distingue deux types de produits pour ce niveau, les produits de Niveau-1 SLC et les produits de Niveau-1 GRD.

Les données SLC de Niveau-1 sont des données RSO focalisées et chaque pixel d'une image y est représenté par des valeurs complexes (amplitude et phase). Ces produits sont disponibles selon tous les modes d'acquisition de données.

Les produits GRD de Niveau-1 quant à eux ne gardent pas l'information de phase, mais uniquement les amplitudes. L'image qui en résulte a des pixels de résolution spatiale approximativement carrée (même valeurs en x et en y), avec un chatoiement réduit suite à de multiples traitements. Les produits GRD sont également caractérisés par leurs types de résolutions qui peuvent être *Full Resolution* (FR), *High Resolution* (HR) ou *Medium Resolution* (MR). Les produits sont disponibles en HR et MR pour les modes IW et EW, en MR pour le mode WV et en FR, HR et MR pour le mode SM. Dans le cadre de nos travaux de thèse, les données HR de Niveau-1 GRD seront utilisées.

3.1.3.3 Produits de Niveau-2

Les produits de Level-2, de type *Ocean* (OCN), sont des produits géolocalisés obtenus à partir de ceux de Niveau-1. Ils peuvent contenir les informations géophysiques de l'océan telles que le champ de vent océanique ou *Ocean Wind field* (OWI) en anglais, la gamme spectrale de la houle océanique ou *Ocean Swell spectra* (OSW) et la vitesse radiale de surface ou *Ocean Swell spectra* (RVL). La disponibilité de chacun de ces paramètres dépend fortement du mode d'acquisition. Les données de Niveau-2 n'étant pas utilisées dans le cadre de cette, plus de détail dans les explications ne seront pas données dans ce manuscrit.

3.2 Sites d'études

Dans cette section, les zones d'études correspondant aux images utilisées pour constituer les jeux de données d'entraînement sont présentées. Les zones couvrant la Réserve de Faune de Togodo (RFT), le Parc National de Bouba-Ndjida (PNB) et la réserve de Biosphère du Dja (RBD) seront décrites. La RFT est située au Togo tandis que le PNB et la RBD sont localisés au Cameroun.

3.2.1 Réserve de Faune de Togodo

La RFT, localisée au sud-est du Togo, un pays de l'Afrique de l'Ouest, est située entre $6^{\circ}23'$ et 7° de latitude nord, et entre $1^{\circ}23'$ et $1^{\circ}34'$ de longitude ouest [114]. Avec une superficie estimée à 30 000 *ha*, elle est délimitée par le fleuve Mono à l'Est et par les rivières Akpaka et Afan, ainsi que les fermes Kpové et Tsafé à l'Ouest. Au Sud, la RFT est limitée par les villages de

Tomety-Kondji, Sédomé, Adodomé, Gboto zévé et Zouvi, tandis qu'au Nord on retrouve la forêt classée et la rivière d'Asrama [41].

Le climat dans cette zone est subéquatorial, comprenant deux saisons de pluies, une longue saison et une autre plus courte. La longue saison s'étale de mars à juillet, avec un pic des précipitations enregistré en juin. Tandis que la courte saison de pluie a généralement lieu de septembre à octobre, le maximum de précipitations étant enregistré en octobre. Les deux saisons des pluies sont entrecoupées d'une longue saison sèche et d'une courte saison sèche, ce qui donne un régime pluviométrique bimodal qui se succèdent, permettant aux agriculteurs de réaliser deux cultures par an sur la même parcelle [41]. La figure 3.2 représente la courbe ombrothermique dans la zone de la RFT. La courbe ombrothermique est une représentation graphique combinant les données mensuelles des températures moyennes (représentées par la ligne rouge sur la figure 3.2) et des précipitations (représentées par les histogrammes en bleu sur la figure 3.2) d'un lieu donné.

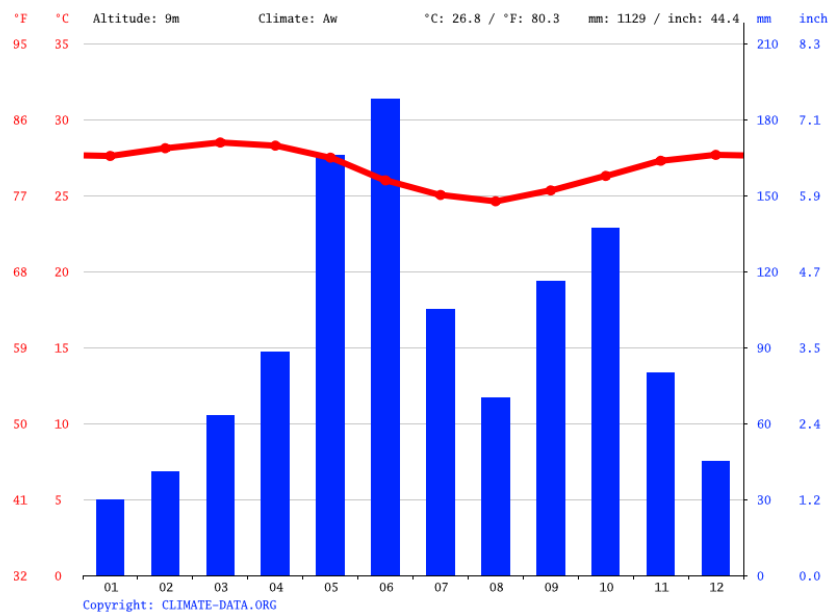


FIGURE 3.2 – Courbe ombrothermique du climat dans la zone de la RFT [115].

L'occupation humaine autour de la RFT est assez ancienne et comprend divers groupes ethniques installés dans la région depuis plus de trois siècles [116]. Le quatrième Recensement Général de la population et de l'habitat (RGPH4) de 2010 estime la densité de la population entre 100 et 150 habitants/ km^2 . Cette population à l'évolution démographique rapide, est principalement agricole et pratique le brûlage dirigé ainsi que l'agriculture extensive des cultures. En ce qui concerne le couvert végétal, on y retrouve de la forêt dense, des forêts galeries, des forêts claires, de la savane boisée, arbustive et herbeuse, des zones de culture et de jachère. La végétation hydrophile est également très riche floristiquement, avec environ 210 espèces [41]. En outre, la plantation de palmiers huile de palme est particulièrement développée dans la zone, considérée comme une source de déforestation. La figure 3.3 présente la carte de la RFT.

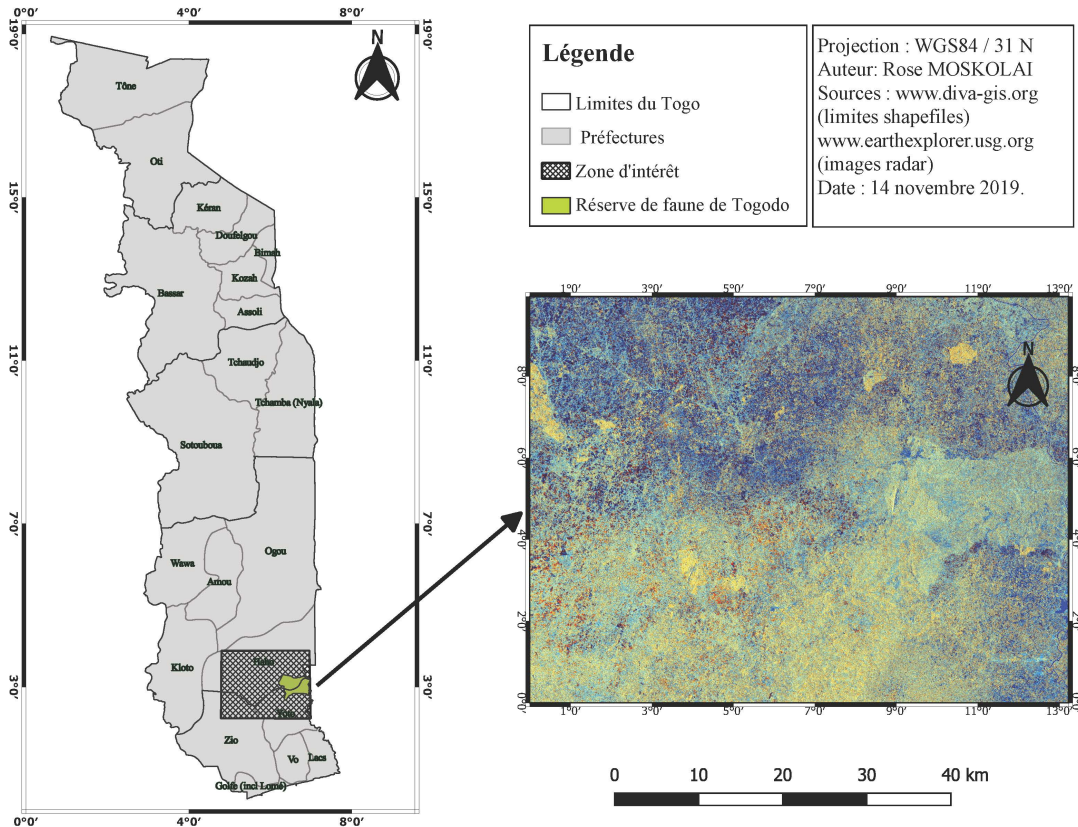


FIGURE 3.3 – Localisation de la réserve de faune de Togodo.

3.2.2 Parc National de Bouba-Ndjida

La deuxième zone d'étude sélectionnée dans le cadre de cette thèse est le Parc National de Bouba-Ndjida en abrégé PNB. Il s'agit de l'un des parcs nationaux du Cameroun, un pays d'Afrique centrale. Il a été créé en 1968 et se situe dans la Région du nord, à la frontière du Tchad. Ce parc est traversé par de nombreuses rivières saisonnières et par une rivière permanente, le mayo Sénaroua. Géographiquement, le PNB est situé entre $08^{\circ}21'$ et 09° de latitude Nord, et entre $14^{\circ}25'$ et $14^{\circ}55'$ de longitude ouest [117]. Même s'il s'agit du plus grand parc national du pays, avec sa superficie totale d'environ 220 000 *ha*, c'est aussi le plus isolé. Depuis août 2011, un accord de partenariat a été signé entre le Tchad et le Cameroun pour créer le complexe BSB (Binationale Séna-Oura - Bouba-Ndjida) de Yamoussa. Le complexe BSB Yamoussa est un complexe écologique composé de deux principales aires protégées, notamment le parc national de Séna-Oura au Tchad et le PNB au Cameroun [118].

Le PNB est le dernier refuge où évoluent toutes les espèces d'animaux sauvages de la savane arborée spécifique à la région. Sa végétation est principalement constituée de savane arbustive de type soudano-guinéen. Le climat est soudano-sahélien avec une saison sèche et une saison des pluies. Les mois les plus pluvieux étant de juillet à septembre, comme le montre la courbe ombrothermique de la zone (figure 3.4).

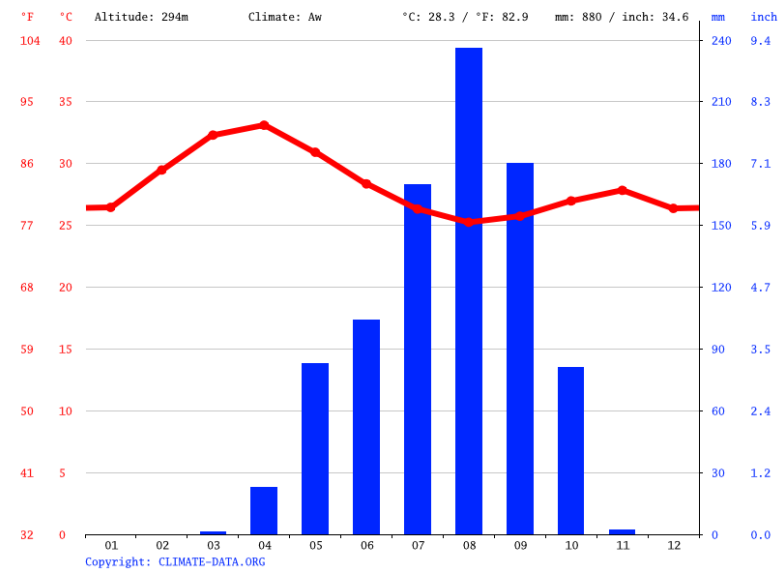


FIGURE 3.4 – Courbe ombrothermique du climat dans la zone du PNB [115].

Bien que le PNB bénéficie le statut d'aire protégée, son paysage est soumis à une forte pression des activités humaines et des phénomènes naturels. Les principales causes de déforestation dans cette zone sont l'agriculture, le pâturage, l'abattage des arbres, les feux de brousse, la sécheresse, le défrichement incontrôlé, l'exploitation du bois de chauffage, etc.

La figure 3.5 illustre la zone d'étude choisie autour du parc national de Bouba-Ndjida.

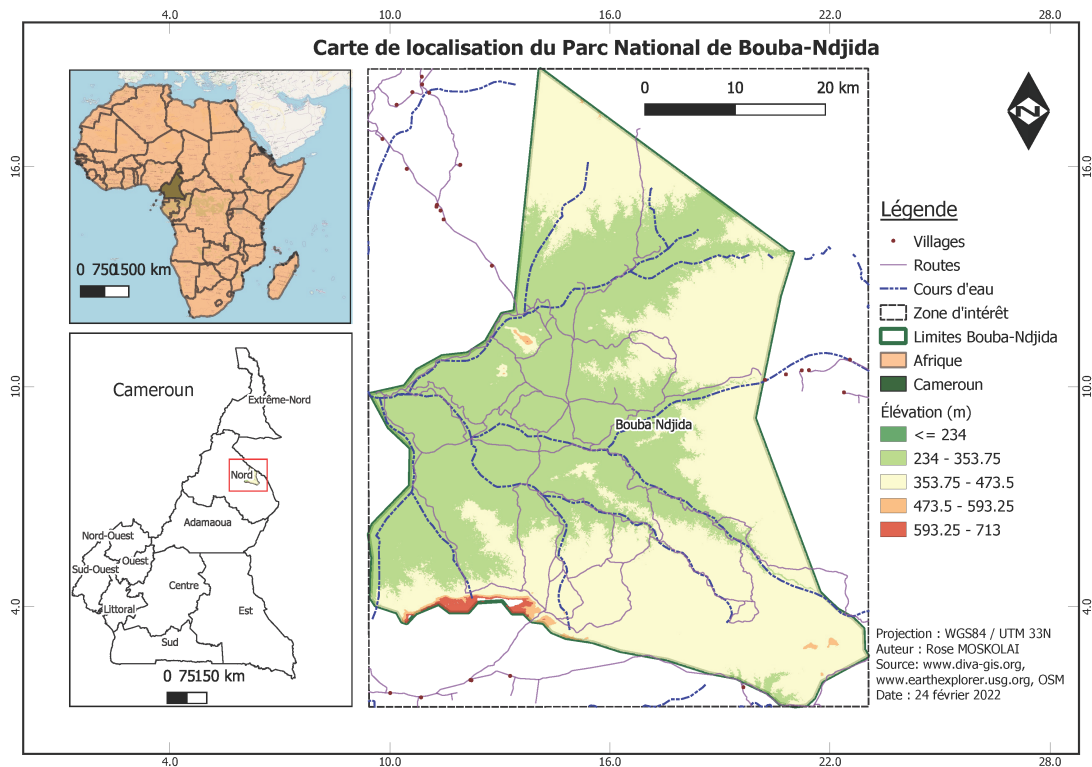


FIGURE 3.5 – Localisation du Parc National de Bouba-Ndjida.

3.2.3 Réserve de Biosphère du Dja

La RBD tient son nom du fleuve Dja qui l'entoure. Elle est située entre 02°40' et 03°23' de latitude Nord et 12°25' et 13°35' de longitude Est. Cette réserve est située au sud-est de Yaoundé, la capitale politique du Cameroun. Administrativement, la RBD appartient à deux régions, à savoir les régions de l'Est et du Sud. Le fleuve Dja l'encercle presque complètement, constituant sa principale frontière naturelle. Le terrain est assez plat dans la RBD, le paysage étant composé d'une multitude de petites buttes. Cependant, dans le sud, la topographie est plus accidentée avec des falaises, des rapides et des chutes d'eau.

Selon le recensement de la population camerounaise de 2005, le nombre de personnes vivant dans les localités voisines était de 26 153 habitants [119]. Les différents groupes ethniques y vivent en parfaite collaboration, aussi bien à l'intérieur qu'à l'extérieur de la réserve.

Il y a quatre saisons dans la zone d'étude : une longue saison des pluies, allant d'août à novembre, une longue saison sèche de novembre à mars, une courte saison des pluies de mars à juin et une courte saison sèche de juin à août [119]. Les précipitations annuelles moyennes sont de 1 563 *mm* et la température moyenne varie de 19,8 °C à 27 °C. La figure 3.6 présente la répartition des précipitations et des températures dans la zone.

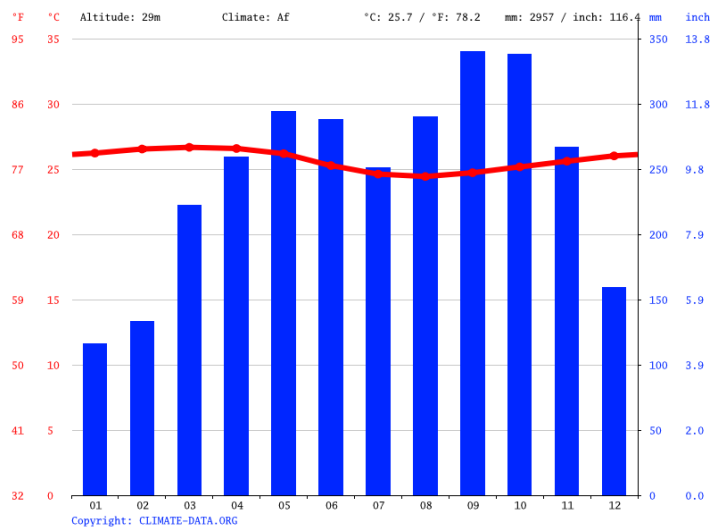


FIGURE 3.6 – Courbe ombrothermique du climat dans la zone de la RBD [115].

La réserve du Dja couvre 526 000 *ha* de surface et représente l'une des forêts d'Afrique les plus vastes. Composée d'une forêt primaire restée largement épargnée par les actions déstabilisantes de l'exploitation forestière qui s'est développée à sa périphérie immédiate depuis les années 1970, la RBD est soumise aujourd'hui à une pression croissante des opérateurs économiques de la filière bois. La détérioration du contexte économique au cours de la dernière décennie a encore renforcé les menaces liées aux activités humaines qui plainaient déjà sur la conservation de cette forêt.

La figure 3.7 présente la troisième zone d'étude située autour de la RBD.

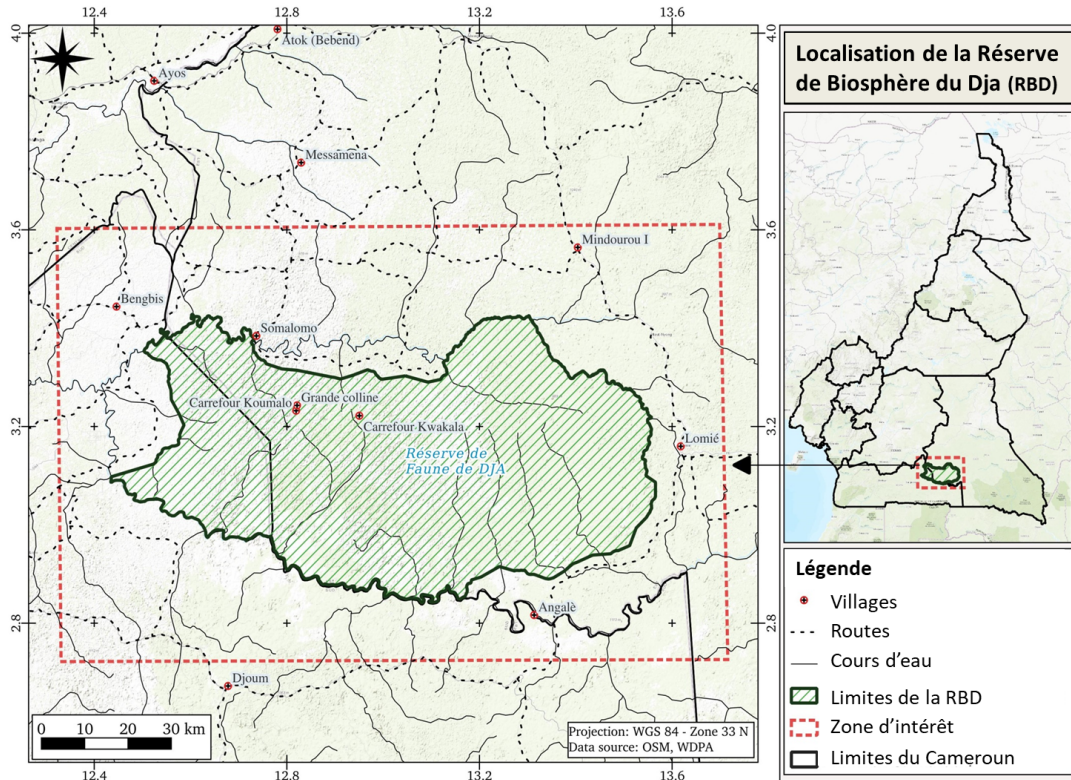


FIGURE 3.7 – Localisation de la Réserve de Biosphère du Dja.

3.3 Collecte et prétraitement

Les images radar S1A, dépourvues de nuages, ont été choisies pour l'ensemble des sites d'étude. Les satellites Sentinel-1A et Sentinel-1B sont en orbite autour de la Terre à 180° et revisitent chaque emplacement après seulement six jours. Cependant, les directions orbitales (ascendante ou descendante) et les angles de prise de vue des deux satellites sont différents et ont un impact sur les valeurs des pixels des images. Ainsi, seules les images S1A ont été considérées.

3.3.1 Collecte

Les différents types d'images S1A peuvent être téléchargés gratuitement à partir de plusieurs plateformes, chacune ayant ses particularités. Parmi les plus utilisées, on peut citer :

- le [Copernicus Open Access Center](#) : il offre aux utilisateurs un accès complet aux produits Sentinel. Les images sont disponibles en ligne dès que les images satellites sont reçues. Cependant, les images plus anciennes ne sont généralement pas accessibles immédiatement et doivent faire l'objet d'une requête. En effet, environ six mois après l'enregistrement d'une image, celle-ci est mise en mode hors ligne sur le site officiel de Copernicus, et l'utilisateur doit envoyer une demande pour la mise en ligne du produit, pendant une période déterminée.

- le [Copernicus Dedicated Access Center](#) : il est destiné à des services spécifiques tels que la mise à disposition d'archives d'images qui ne sont plus disponibles sur le serveur principal.
- la [Plateforme Vertex de la NASA](#) ou Alaska Satellite Facility (ASF) : elle met à la disposition du public la série chronologique complète d'images d'une zone. En d'autres termes, il n'y a pas de produits hors ligne dans ce centre de données. Cette plateforme a l'avantage de faciliter le téléchargement des images en spécifiant la zone d'intérêt par un fichier *shapefile* ou par les coordonnées géographiques au format *Well Known Text* (WKT).
- La plateforme d'observation de la Terre [EO Browser](#) : elle permet aux utilisateurs de parcourir et de comparer des images provenant de toutes les collections de données fournies.

Dans la plupart des plateformes, le téléchargement des images se fait une par une, bien que plusieurs téléchargements puissent être lancés simultanément. Pour pallier cette lourdeur, des exemples de scripts de téléchargement par lot (`dhusget`¹), basés sur les programmes de téléchargement en ligne de commande `cURL` et `Wget`, sont proposés par le centre de données Sentinel. L'utilisation de `dhusget` peut également se faire par une requête en ligne de commande ayant la structure ci-dessous.

```
dhusget.sh [CONNEXION] [RECHERCHE] [AFFICHAGE DES RESULTATS] [TELECHARGEMENT]
```

Dans les paramètres de connexions, l'URL du service du centre de données à interroger, l'identifiant et le mot de passe de l'utilisateur doivent être fournis. Dans les critères de recherche, les informations telles que le nom de la mission Sentinel, la date d'acquisition du produit, les coordonnées de la zone d'intérêt, etc. peuvent être utilisées comme options de recherche. Les paramètres d'affichage permettent quant eux de configurer l'organisation des résultats de recherche à l'écran et les options de téléchargement permettent de spécifier ce qui doit être téléchargé. Plus de détails sur les différents paramètres de `dhusget` sont disponibles à la page dédiée aux utilisateurs du centre d'accès de données Copernicus².

Une autre façon simple de télécharger automatiquement les images S1 est d'utiliser `aria2c`, qui est le gestionnaire de téléchargement sur le site officiel de Copernicus. Avec cet outil, l'utilisateur spécifie les critères de recherche liés à la zone d'intérêt, et la validation des critères de recherche génère un fichier nommé `products.meta4` qui contient la liste de tous les produits à télécharger. La commande utilisée pour lancer le téléchargement par lots est indiquée ci-dessous. Il est à noter qu'un compte utilisateur valide du centre d'accès de Copernicus Open Access Center est nécessaire pour exécuter ces lignes de commande.

```
aria2c --http-user='username' --http-passwd='password'  
--check-certificate=false  
--max-concurrent-downloads=2  
-M products.meta4
```

L'utilisation des commandes `dhusget` et `aria2c` ne permettent pas de télécharger immédiatement les données lorsqu'elles sont hors-ligne. Cette particularité constitue un inconvénient

1. Disponible sur le site web du centre de données Sentinel à l'adresse <https://scihub.copernicus.eu>

2. En ligne. [Disponible] : <https://scihub.copernicus.eu/userguide/BatchScripting>

majeur pour ces commandes. Toutefois, il est possible de configurer le script de téléchargement pour qu'il puisse envoyer une requête au serveur lorsqu'un produit est hors ligne. La disponibilité du produit est alors vérifiée après un intervalle de temps et envoyée dans la file de téléchargement une fois que son statut a changé.

Pour les données utilisées dans cette thèse, les images ont été téléchargées depuis la plateforme ASF. Toutes les images S1A, de Niveau 1-GRD, acquises en mode IW, direction ascendante, correspondant aux trois zones d'étude, du 15 juin 2014 au 3 juillet 2022 ont été sélectionnées. Les scripts Python de téléchargement ont été exécutés en lignes de commande sur une machine locale pour l'enregistrement de toutes les images. Plus d'informations sur la collecte en masse des images Sentinel-1 à partir de la plateforme Vertex sont disponibles sur le site Web d'ASF.³

Les données brutes S1 sont téléchargées sous la forme d'archives portant l'extension `.SAFE` et sont nommées selon une convention de nommage spécifiant les principales caractéristiques de chaque donnée. La figure 3.8 présente les différentes caractéristiques d'une image Sentinel identifiables à partir de son nom de fichier.



FIGURE 3.8 – Convention de nommage des produits Sentinel-1 [112].

Considérons par exemple une image Sentinel dont le nom de fichier est donné ci-dessous.

`S1A_IW_GRDH_1SDV_20150409T171237_20150409T171302_005410_006E19_7760.zip`

Dans cet exemple, les trois premiers caractères du nom de l'image indiquent la mission dont est issue l'image (Sentinel-1A). Ensuite les deux caractères suivants indiquent le mode d'acquisition qui est IW.

Le tableau 3.1 présente les coordonnées en WKT des polygones délimitant les zones d'intérêt à télécharger.

3. En ligne. [Disponible] : <https://asf.alaska.edu/how-to/data-tools/data-tools/>

TABEAU 3.1 – Paramètres de recherche.

Zone	Coordonnées de la zone d’intérêt (WKT)
BNP	POLYGON((14,6088 8,7979; 14,6793 8,7979; 14,6793 8,824; 14,6088 8,824; 14,6088 8,7979))
DBR	POLYGON((12,1466 2,9639; 12,8304 2,9639; 12,8304 3,2727; 12,1466 3,2727; 12,1466 2,9639))
WRT	POLYGON((12,1466 2,9639; 12,8304 2,9639; 12,8304 3,2727; 12,1466 3,2727; 12,1466 2,9639))

3.3.2 Prétraitement

3.3.2.1 Logiciel SNAP

L’objectif du prétraitement des images est d’améliorer la qualité des données en supprimant les bruits et les distorsions indésirables, de définir la zone d’intérêt de travail et de faire ressortir certaines caractéristiques des images importantes pour les applications ultérieures. Les opérations de prétraitement des données SAR sont diverses et variées, et sont utilisées en fonction des besoins spécifiques. Deux outils ont principalement été utilisés pour l’ensemble des opérations de prétraitement des images. Il s’agit des logiciels gratuits et open sources *SentiNel Application Platform* (SNAP) et, QGIS appelé autrefois Quantum GIS.

SNAP est un logiciel de l’Agence Spatiale Européenne développé conjointement par Brockmann Consult, SkyWatch et C-S pour la visualisation, le traitement et l’analyse des données d’observation de la Terre en général, et en particulier les produits Sentinel. La figure 3.9 représente l’architecture de SNAP qui utilise plusieurs technologies telles que NetBeans platform, Install4J, GeoTolls, GDAL, ou Java Advanced Imaging (JAI). Les traitements peuvent se faire soit à partir de l’interface graphique *SNAP-Desktop*, soit en ligne de commande grâce à la partie *SNAP-Engine* [120].

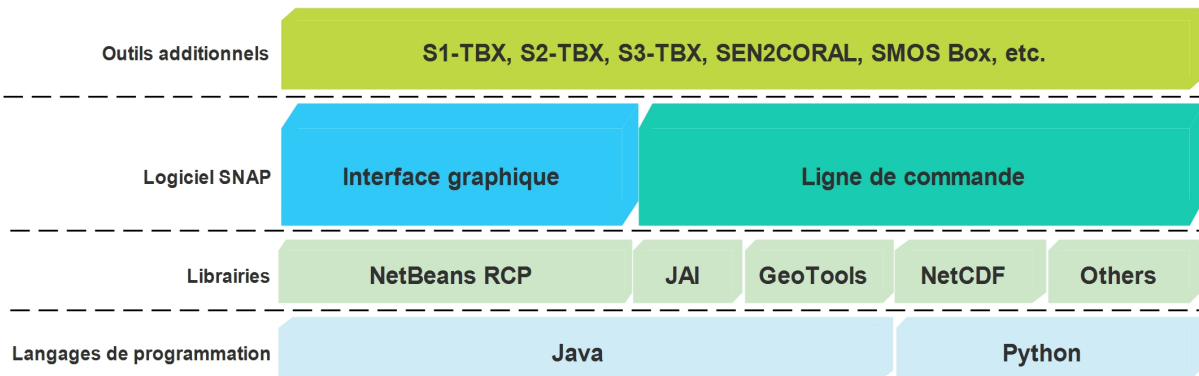


FIGURE 3.9 – Architecture générale du logiciel SNAP.

Un autre atout qui a justifié l’utilisation de SNAP pour le traitement de nos données est la disponibilité dans ce logiciel, d’un framework de traitement flexible, appelé *Graph Processing Framework* (GPF) [121]. Le GPF est basé sur JAI et permet aux utilisateurs de mettre en place des chaînes de traitement par lot personnalisées. La figure 3.10 présente le mode de fonctionnement général du GPF, qui utilise des graphes assemblés à partir d’une liste d’opérateurs disponibles et permet de connecter les nœuds des opérateurs à leurs sources. Il s’agit en

fait d'un graphe acyclique dirigé où les nœuds représentent les étapes de traitement appelées opérateurs, et les arêtes indiquent la direction dans laquelle les données sont transmises entre les nœuds. Les données sources sont les images reçues en entrée par l'opérateur de lecture (*Read*) et la sortie peut être soit une image enregistrée par l'opérateur d'écriture (*Write*), soit une image affichée. Chaque fois qu'une donnée passe par un opérateur, celui-ci transforme l'image, et elle est transmise au nœud suivant jusqu'à ce qu'elle atteigne la sortie. Aucun fichier intermédiaire n'est créé au cours du processus, à moins qu'une opération d'écriture n'ait été volontairement introduite dans le graphe.

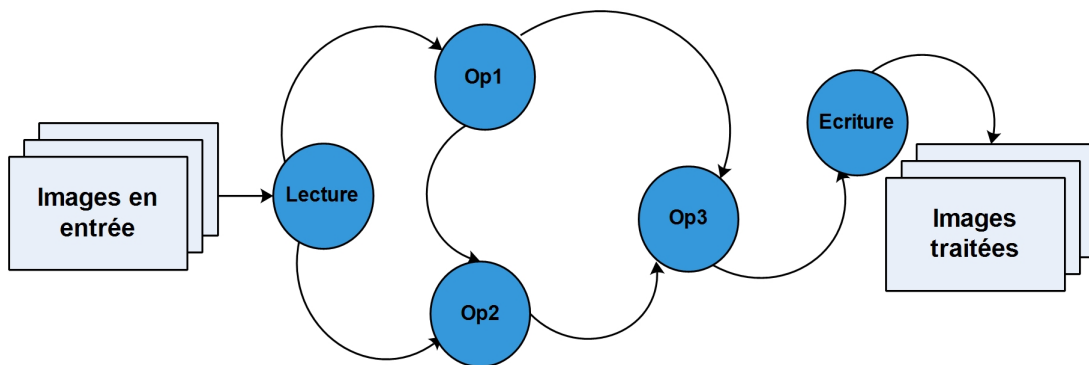


FIGURE 3.10 – Graphe de traitement des données à partir du GPF [121].

En somme, l'architecture du GPF offre les avantages suivants :

- l'absence de surcharge des opérations d'entrée/sortie ;
- une bonne utilisation de la mémoire de stockage (pas d'écriture de fichiers intermédiaires) ;
- une réutilisabilité des chaînes de traitement ;
- la possibilité de réutiliser les configurations des opérateurs ;
- le traitement en parallèle des graphes en fonction du nombre de cœurs d'unités de calcul disponibles.

3.3.2.2 Workflow de prétraitement

Pour faciliter le traitement et optimiser l'utilisation des ressources matérielles au cours de cette thèse, un script basé sur le GPF a été développé pour le traitement par lot des séries d'images téléchargées. L'ensemble des opérations appliquées sur les données d'entrées correspondant aux trois zones d'étude est schématisé par le graphe représenté par la figure 3.11. Les différentes opérations sont décrites dans les paragraphes qui suivent.

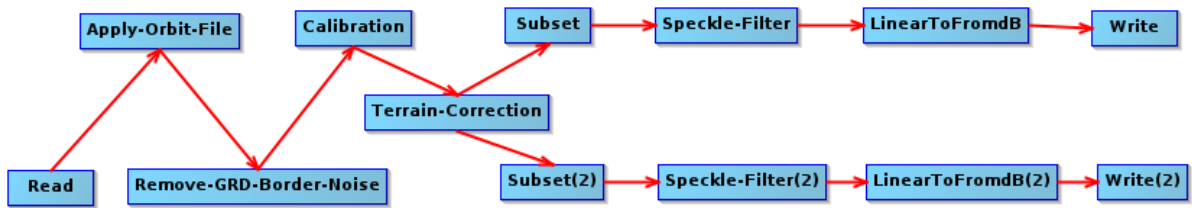


FIGURE 3.11 – Workflow des opérations de prétraitement par lot.

Read (lecture) : l'opérateur *Read* permet de charger les données. L'ensemble des données est passé en entrée du graphe. Pour des zones représentées sur plusieurs images comme c'est le cas pour la RBD, une entrée supplémentaire de lecture est ajoutée.

Apply orbit File (correction de la position du satellite) : les métadonnées fournies par défaut lors du téléchargement des données SAR ne sont généralement pas très précises. Dans le logiciel SNAP, le fichier d'orbite mis à jour, avec des informations plus précises pour aider à améliorer le géocodage et d'autres résultats de traitement, est automatiquement téléchargé par l'opérateur *Apply orbit File*. Cette opération permettant d'obtenir la position précise des images Sentinel-1 doit prioritairement être effectuée avant toutes les autres étapes de prétraitement pour escompter de meilleurs résultats.

Border noise removal (suppression du bruit de bord) : des déformations peuvent apparaître sur les images de niveau-1 lors de leur génération. L'opération correspondant à l'algorithme *Border noise removal* dans SNAP, permet de corriger et supprimer le bruit présent sur la bordure des images.

Calibrate (calibration) : l'étape de calibration a pour objectif la correction de l'intensité du signal en fonction des caractéristiques du capteur et de l'angle d'incidence local. Les valeurs numériques des pixels des images représentant l'intensité du signal sont converties en coefficients de rétrodiffusion radar, par une opération de correction radiométrique (normalisation). Les métadonnées des produits en entrée permettent au logiciel SNAP de déterminer automatiquement quelles corrections appliquer. Dans le cadre de ces travaux, l'opération est effectuée pour les deux polarisations VV et VH.

Terrain correction (correction de terrain) : les images S1 téléchargées initialement sont dépourvues de coordonnées géographiques, ce qui les rend inexploitable à ce stade pour la plupart des applications. L'étape de d'ortho-rectification est réalisée non seulement dans le but de géoréférencer les images pour qu'elles puissent être projetées dans un système de référence terrestre cartographique (UTM) ou géographique (WGS84), mais aussi pour corriger les effets de distorsion survenus lors de l'acquisition (superposition, ombrage, etc). L'opération de correction de terrain se fait à partir d'un modèle numérique de terrain téléchargé automatiquement par SNAP ou fourni par l'utilisateur. L'approche *Range-Doppler* a été utilisée dans nos travaux, avec l'option de téléchargement automatique du MNT. A cette étape, les images correspondant aux bandes VH et VV sont traitées séparément afin d'être enregistrées plus tard dans des fichiers différents. Cette séparation permet de faciliter les analyses et les traitements des séries temporelles par bande.

Subset (découpage de l'image) : Chaque image sentinel-1 téléchargée couvre une superficie de $250 \text{ km} \times 250 \text{ km}$ (fauchée). Dans certains cas, une zone d'étude assez large peut nécessiter l'utilisation de deux images assemblées (mosaïque), c'est le cas par exemple de la zone de RBD. Par contre, pour des zones plus petites, il peut arriver qu'il ne soit pas nécessaire d'utiliser l'ensemble d'une image, mais juste d'en extraire une région (PNB, RFT). Dans tous les cas, pour réduire la consommation de la mémoire, du temps de traitement et de stockage, il est important de ne garder que la zone d'étude considérée pour la suite des traitements.

Speckle filter (filtrage) : Les images radar présentent des bruits sur les images, appelés speckle, qui les dégradent et les rendent inutilisables dans la plupart des cas. Pour améliorer la qualité des images et leur analyse, il est utile, dans certains cas, de réduire l'effet du speckle par un filtrage. L'opération *Speckle filter* permet de supprimer autant que possible l'effet poivre et sel et d'améliorer la lisibilité de l'image. Il existe plusieurs algorithmes et méthodes de filtrage tels que le filtrage spatial simple, le filtrage spatio-temporel ou le filtrage multi-temporel. Dans cette étude, l'algorithme du filtre amélioré de Lee a été utilisé.

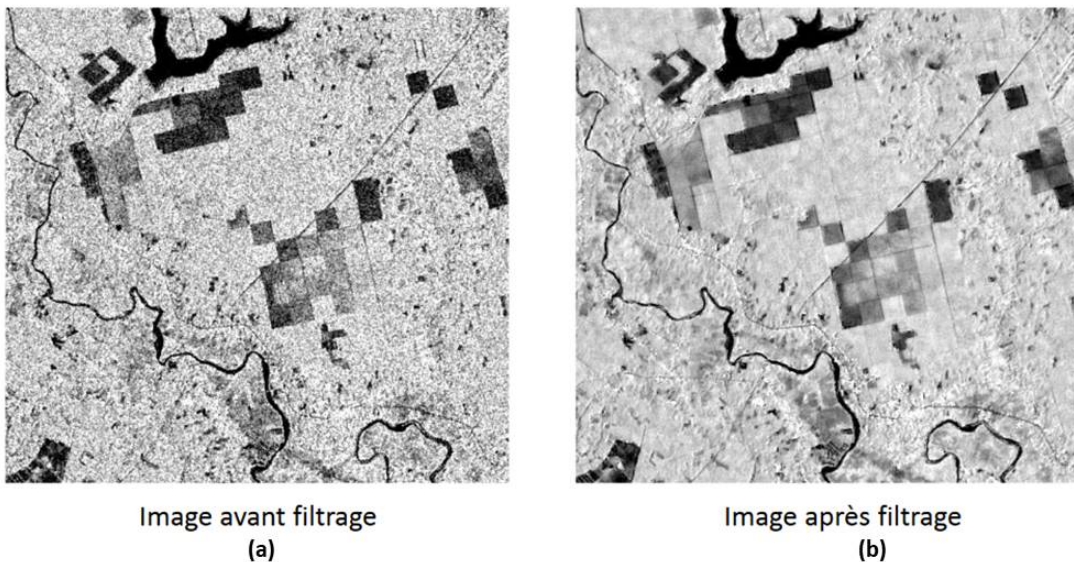


FIGURE 3.12 – Effets d'un filtre appliqué sur une image radar : (a) avant filtrage ; (b) après filtrage.

Linear to dB (conversion des valeurs en décibels) : avant l'écriture des données sur le disque, la dernière étape du processus consiste à convertir les valeurs initiales des pixels en décibels (dB). Généralement, cela se fait grâce à une transformation logarithmique qui change les coefficients de rétrodiffusion en valeurs en décibels (dB) [122].

Write (écriture) : A la fin de tous les traitements, chaque image de la série est transformée et enregistrée sur le disque grâce à l'opérateur *Write*. Les données obtenues sont sous la forme de fichiers GeoTiff.

Après les opérations de base réalisées grâce aux scripts de prétraitement par lot, une étape supplémentaire consistant en la création des compositions colorées a été faite à partir de QGIS. La composition colorée d'une image permet une restitution visible de ses valeurs numériques

et une meilleure interprétation. Elle consiste à attribuer à chacune des trois couleurs primaires rouge, vert et bleu (RVB), trois bandes spectrales d'un capteur satellite. Pour les expériences réalisées dans nos travaux, les images en polarisation VV, VH et VH/VV ont été affectées respectivement aux canaux Rouge, Vert et Bleu. Cette composition généralement notée composition VV, VH, VH/VV, permet de mettre en évidence la végétation sur les images radar. En effet, il a été montré qu'il existe une forte corrélation entre la mesure VH/VV et le NDVI issu de capteurs optiques [123].

Les paramètres correspondant aux opérations du workflow de prétraitement sont présentés dans le tableau 3.2.

3.3.3 Prétraitement automatisé par lot avec GPT

Chaque workflow conçu avec le constructeur de graphiques SNAP peut être exporté sous forme de fichier XML pour automatiser le processus. Même s'il est possible d'exécuter un traitement par lots à partir du logiciel SNAP, il est recommandé d'exécuter les opérateurs via l'outil GPT de SNAP pour une plus grande flexibilité. L'outil GPT est l'interface de ligne de commande de SNAP qui est utilisée pour exécuter les opérateurs de données matricielles en mode *batch*. Il s'agit d'une autre façon d'effectuer un traitement par lots sans utiliser le mode d'interface utilisateur graphique du logiciel SNAP. Pour utiliser la commande `gpt`, les opérations peuvent être traitées individuellement, ou elles peuvent être traitées dans un workflow défini.

Au lieu de spécifier les paramètres de l'opérateur séparément dans le flux de traitement, il est possible d'utiliser le fichier encodé en XML comme paramètre de la commande `gpt`. La syntaxe générale utilisée pour exécuter la commande pour une image est proposée ci-dessous.

```
gpt <GraphFile.xml> [options] [<sourcefile1> <sourcefile2> ...]
```

Pour exécuter le traitement de toutes les images brutes, un script a été proposé. Les lignes de code ont été écrites pour être exécutées dans un environnement Unix. Ce script parcourt l'ensemble du répertoire et applique la chaîne de traitement à chaque image trouvée. L'avantage d'avoir chaque nom de fichier écrit selon une convention de nommage particulière⁴ permet d'extraire la date d'acquisition de chaque image et de renommer l'image traitée. Par exemple, considérons une image dont le nom de fichier initial est :

```
S1A_IW_GRDH_1SDV_20150409T171237_20150409T171302_005410_006E19_7760.zip
```

Cette image est renommée après traitement en `20150409_VH.dim` pour la bande VH et `20150409_VV.dim` pour la bande VV. Ce changement facilite l'utilisation et la manipulation ultérieures des images traitées.

4. En ligne. [Disponible] : <https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-1-sar/>

TABLEAU 3.2 – Valeur des paramètres utilisés pour chaque opérateur.

Opérateur	Paramètre	Valeur
Read	Data format	Any format
Apply orbit file	Orbit state vectors Polynomial degree	Sentinel precise (Auto download) 3
Border noise removal	Polarizations	500
	Threshold	0,5
	Polarizations	VH, VV
Calibration	Polarizations	VH, VV
	Output sigma0 band	True
Terrain correction	Source bands band DEM	Sigma0_VH, Sigma0_VV SRTM 3 sec (Auto download)
	DEM resampling method	Bilinear interpolation
	Image resampling method	Bilinear interpolation
	Pixel spacing	10 m
	Map projection	MP1 for BNP and DBR; MP2 for WRT
	Selected source bands	True
Subset	Source bands	Sigma0_VH
	Copy metadata	True
	Geographic coordinates	WKT1 (BNP), WKT2 (DBR) and WKT3 (WRT)
Subset(2)	Source bands	Sigma0_VV
	Copy metadata	True
	Geographic coordinates	WKT1 (BNP), WKT2 (DBR) and WKT3 (WRT)
Speckle filtering	Source bands	VH
	Filter	Refined Lee
Speckle filtering (2)	Source bands	VV
	Filter	Refined Lee
Linear to dB	Source bands	VH
Linear to dB(2)	Source bands	VV
Write	Save as	BEAM-DIMAP
Write(2)	Save as	BEAM-DIMAP

MP1 = EPSG : 32633-WGS84/UTM 33N; MP2 = EPSG :32631-WGS84/UTM 31N; WKT1 = POLYGON ((14,411 8,480; 14,914 8,480; 14,914 8,996; 14,411 8,996; 14,411 8,480; 14,411 8,480)); WKT2 = POLYGON ((12,480 2,904; 12,480 3,350; 13,369 3,350; 13,369 2,904; 12,480 2,904)); WKT3 = POLYGON ((1,398 6,761; 1,611 6,759; 1,613 6,930; 1,400 6,932; 1,398 6,761)).

3.4 Résultats d'implémentation du workflow

3.4.1 Outils utilisés

Les outils logiciels et les programmes suivants ont été utilisés dans le cadre de nos travaux :

- SNAP 8.0 : un outil utilisé pour concevoir et mettre en œuvre le workflow de traitement automatique par lots. Ce logiciel a également permis d'afficher et d'analyser les données téléchargées.
- QGIS 3.16.0-Hannover : un programme utilisé pour manipuler les fichiers vectoriels (shapefiles) et les informations géographiques. Cet outil a également été utilisé pour analyser les séries temporelles d'images obtenues.
- Python 3 : un langage de programmation principalement utilisé pour les scripts de téléchargement.

3.4.2 Description des datasets constitués

Après l'implémentation du workflow présenté, une base de données contenant trois séries temporelles d'images S1 prétraitées a été constituée. Les STIS proviennent des zones d'étude sélectionnées, à savoir le PNB, la RBD et la RFT. Pendant l'exécution de la chaîne de prétraitement, aucun fichier intermédiaire n'est sauvegardé. Seuls les résultats finaux sont sauvegardés sur le disque. Comme la taille des images brutes est très grande, cela évite une utilisation inutile de la mémoire de stockage.

Au total, 628 images sont collectées. Chaque image brute, d'une taille moyenne de 1 *Go*, est téléchargée sous forme de fichier compressé (zip), avec un débit moyen égal à 10 *Mo/s*. Il a fallu environ 18 heures (*h*) pour télécharger toutes les données, d'une taille totale d'environ 654,9 *Go*. L'essentiel des informations de téléchargement est donné dans le tableau 3.3.

TABLEAU 3.3 – Informations de téléchargement.

Zone	Nombre de fichiers	Taille totale
PNB	204	198,1 <i>Go</i>
RBD	207	247,0 <i>Go</i>
RFT	217	209,8 <i>Go</i>

L'analyse des images collectées montre que certaines données sont manquantes dans les séries temporelles. En effet, il y a environ 6% de données manquantes pour les zones du PNB et de la RBD, contre seulement 2% pour la RFT. Le tableau 3.4 donne la proportion des données manquantes pour chaque zone d'étude.

TABLEAU 3.4 – Proportions des données manquantes.

Zone d'étude	Images manquantes	Pourcentage
PNB	15	6,85%
DBR	14	6,33%
WRT	5	2,25%

Les jeux de données constitués, ainsi que tous les scripts de collecte et de prétraitement sont disponibles sur <http://w-abdou.fr/sits/> et peuvent être téléchargés gratuitement. Un ensemble de 1 256 images prétraitées en polarisation VV et VH sont disponibles : 408 pour le PNB, 414 pour la RBD et 434 pour la RFT. Le tableau 3.5 donne les caractéristiques de chaque jeu de données. Toutes les images sont de type Float32 et enregistrées sous forme de fichiers DIMAP.

TABLEAU 3.5 – Principales caractéristiques des données prétraitées.

Zone	Taille des images	Nombre d'images	Taille totale des données
PNB	5 553 × 3 980	408	39,2 Go
RBD	9 899 × 4 948	414	84,7 Go
RFT	2 380 × 1 869	434	11,1 Go

La Figure 3.13 donne un aperçu des trois STIS constituées. Les images sont séparées en bandes VH et VV pour chacune des zones.

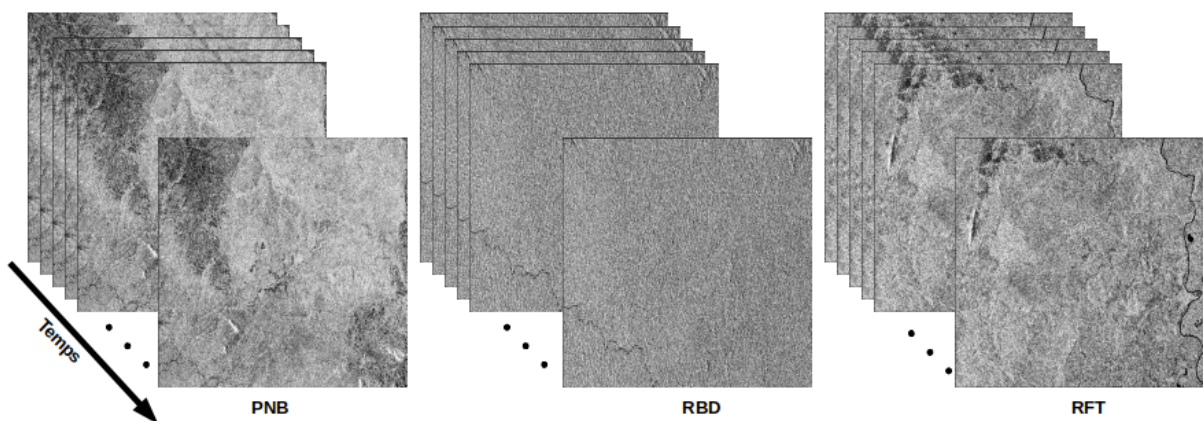


FIGURE 3.13 – Illustration des STIS constituées en bande VH pour chacune des trois zones d'étude (PNB, RBD et RFT).

Nous avons présenté dans ce chapitre une méthode permettant de faciliter la collecte et le prétraitement des images satellites en vue de leur utilisation par des algorithmes DL. Au final, trois bases de données constituées des séries d'images S1A ont été obtenues. Ces jeux de données seront utilisés pour les expérimentations présentées dans le chapitre 4. Ces expérimentations visent à évaluer les architectures LSTM pour la prédiction des STIS.

Chapitre 4

Evaluation des architectures LSTM pour la prédiction des séries temporelles d'images satellites

Objectifs

Ce chapitre étudie des méthodes deep learning particulières, notamment les architectures LSTM (Long Short-Term Memory) pour la prédiction des STIS. Les résultats de deux expérimentations qui ont permis la comparaison de ces architectures sont présentés et discutés. La première comparaison est faite entre trois types d'algorithmes LSTM utilisant les données du PNB et de la RFT. La deuxième comparaison quant à elle est réalisée entre les modèles DL et les modèles utilisant les automates cellulaires intégrés aux chaînes de Markov (CA-Markov).

Sommaire

4.1	Description des architectures LSTM	70
4.1.1	Réseaux LSTM empilés	71
4.1.2	Réseaux ConvLSTM	72
4.1.3	Réseaux CNN-LSTM	73
4.2	Évaluation des architectures LSTM pour la prédiction des STIS	74
4.2.1	Formulation du problème et méthodologie	74
4.2.2	Structure des modèles et paramètres utilisés	75
4.2.3	Résultats obtenus	76
4.3	Comparaison des méthodes DL et CA-Markov	83
4.3.1	Formulation du problème et description des méthodes	83
4.3.2	Structure des modèles et hyperparamètres utilisés	85
4.3.3	Résultats obtenus	86

4.1 Description des architectures LSTM

Pour éviter les problèmes « d’explosion » de gradient qui surviennent généralement lors de l’entraînement des RNN, les architectures LSTM ont été introduites par Hochreiter et al. dans [58]. Les LSTM sont des réseaux récurrents avec des unités de mémoire appelées « cellules ». Ces cellules constituent un élément clé du LSTM et permettent au réseau de prendre en compte les dépendances temporelles à long terme des données dans une séquence univariée. Plusieurs tâches complexes insolubles par les algorithmes RNN classiques ont bien été traitées grâce aux réseaux LSTM [124]. A titre d’exemple de domaine d’application, on peut citer le traitement du langage naturel, la reconnaissance vocale, la traduction automatique, la prédiction des séries temporelles, l’analyse des audio et vidéo, la reconnaissance et la génération de l’écriture manuscrite, etc.[32, 125, 59].

L’architecture LSTM décrite par l’équation (4.1) comporte trois portes (la porte d’entrée, la porte de sortie et la porte d’oubli).

$$\begin{aligned}
 c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
 h_t &= o_t \odot \sigma_c(c_t) \\
 i_t &= \sigma_g(W_i x_t + R_i h_{t-1} + b_i) \\
 f_t &= \sigma_g(W_f x_t + R_f h_{t-1} + b_f) \\
 g_t &= \sigma_c(W_g x_t + R_g h_{t-1} + b_g) \\
 o_t &= \sigma_g(W_o x_t + R_o h_{t-1} + b_o)
 \end{aligned} \tag{4.1}$$

où :

c_t et h_t représentent respectivement les cellules *cell state* et *hidden state* au temps t ,

i_t , f_t , et o_t sont respectivement les portes d’entrée, d’oubli et de sortie au temps t ,

g_t désigne la cellule *cell candidate* au temps t ,

x_t est le vecteur d’entrée aux unités du LSTM,

W représente les poids pour chaque composante,

R désigne les poids récurrents pour chaque composante,

b représente les paramètres de biais,

σ_c et σ_g correspondent respectivement aux fonctions d’activation (la fonction sigmoïde par défaut) et d’état (tanh par défaut),

et \odot est le produit Hadamard.

La figure 4.1 donne une représentation d’un LSTM simple. De façon générale, un workflow LSTM est constitué de trois étapes principales qui se déroulent comme décrit ci-dessous [126] :

- Au cours de la première étape, la couche sigmoïde, également connue sous le nom de « porte d’oubli », choisit quelle information déduire de l’état de la cellule. Elle examine les informations au temps précédent (h_{t-1} et x_t), et produit une sortie entre 0 et 1.
- Ensuite, lors de la deuxième étape, les nouvelles informations à stocker dans l’état de la cellule sont définies. Une couche sigmoïde appelée « porte de sortie » décide d’abord quelles

valeurs seront mises à jour, puis une couche *tanh* crée un nouveau vecteur de valeurs candidates c_t . Par la suite, l'ancien état de la cellule c_{t-1} est mis à jour par le nouvel état de la cellule c_t .

- La dernière étape consiste à décider de la sortie du système. La sortie est basée sur l'état de la cellule, mais il s'agit d'une version filtrée. Tout d'abord, la couche sigmoïde décide quelles parties de l'état de la cellule seront présentées en sortie. L'état de la cellule est ensuite converti en une valeur comprise entre -1 et 1 par la fonction *tanh*. Cette valeur est finalement multipliée par la couche sigmoïde pour n'envoyer en sortie que les informations souhaitées.

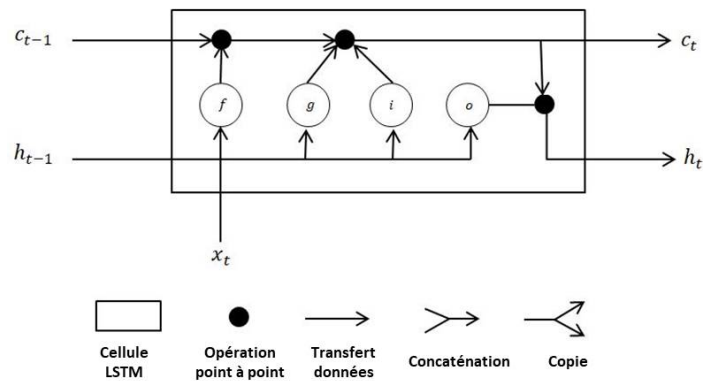


FIGURE 4.1 – Structure générale d'un simple bloc LSTM.

Il existe plusieurs façons d'utiliser les réseaux LSTM. Ceux-ci peuvent être empilés, fusionnés ou combinés à d'autres types d'architectures. Dans le cadre de nos expérimentations, trois variantes des réseaux LSTM sont sélectionnées et évaluées pour déterminer la meilleure approche à utiliser pour les problèmes de prédiction dans les STIS.

4.1.1 Réseaux LSTM empilés

L'architecture LSTM la plus simple et la plus utilisée est appelée Vanilla LSTM [125]. Elle est constituée d'une seule couche cachée mais permet néanmoins d'obtenir des résultats raisonnables. Pour des problèmes plus complexes, plusieurs couches LSTM sont empilées dans le réseau pour en augmenter la profondeur. Cette extension du Vanilla LSTM ainsi obtenue est appelée *Stacked LSTM* (Stack-LSTM) et est représentée par la figure 4.2.

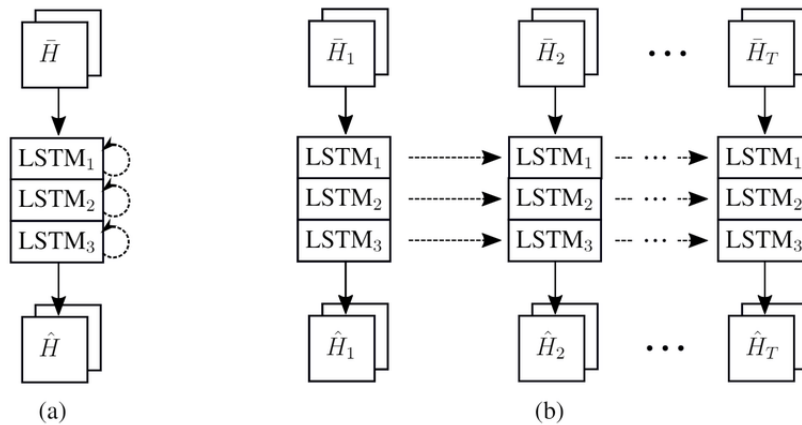


FIGURE 4.2 – Architecture d’un réseau Stack-LSTM : (a) réseau avec 3 couches empilées et les connexions récurrentes ; (b) version dépliée du réseau sur T pas de temps. [127].

Les réseaux Stack-LSTM ont été introduits par Graves et al. dans [128] lors de la résolution d’un problème de reconnaissance de la parole. Grâce à ces travaux, les auteurs ont pu montrer que pour des applications complexes, la profondeur du réseau Stack-LSTM permet à celui-ci d’obtenir de meilleurs résultats comparativement à des réseaux LSTM simples, ayant une seule couche cachée constituée de nombreuses cellules. Depuis lors, plusieurs travaux ont conduit à des résultats similaires faisant désormais des réseaux LSTM empilés une technique standard pour des tâches difficiles [128].

Dans le cadre de nos travaux, nous utilisons des séries temporelles d’images pour faire de la prédiction. Les réseaux Stack-LSTM ayant la capacité de modéliser des dépendances à long terme, nous sélectionnons premièrement cette architecture pour être évaluée.

4.1.2 Réseaux ConvLSTM

La deuxième architecture sélectionnée pour les expérimentations dans cette thèse est basée sur les réseaux ConvLSTM. En effet, les modèles LSTM décrits dans la section précédente ont été conçus initialement pour fonctionner avec des données unidimensionnelles. Pour les données en plusieurs dimensions comme les images ou les vidéos, la meilleure approche conseillée est généralement l’utilisation des modèles basés sur les CNN [25]. En effet, dans un CNN, les caractéristiques importantes des données sont d’abord extraites à partir des couches de convolution, avant d’être connectées à un réseau dense entièrement connecté. Ainsi, pour atteindre de meilleures performances, les capacités des réseaux CNN et des réseaux LSTM ont été fusionnées pour créer un réseau LSTM convolutif appelé ConvLSTM.

Le modèle ConvLSTM a été utilisé pour la première fois dans [76]. Il est l’un des modèles DL les plus utilisés pour la prédiction des données en 2D ou 3D [32]. Dans cette architecture, la multiplication de la matrice à chaque porte du LSTM classique est remplacée par une opération de convolution qui permet de capturer les corrélations spatio-temporelles locales sous-jacentes. De plus, la dimension d’entrée des données est conservée dans la couche de sortie au lieu d’être un simple vecteur unidimensionnel. Cette configuration particulière a permis aux modèles basés sur le réseau ConvLSTM d’obtenir de très bonnes performances de prédiction dans de nombreux do-

maines tels que la vision par ordinateur, le traitement automatique du langage, la météorologie, les finances, la médecine, pour ne citer que ceux-là [129, 130, 131, 132, 59].

La figure 4.3 montre la représentation d'un simple bloc mémoire de l'architecture ConvLSTM. Par comparaison au bloc LSTM représenté par la figure 4.1, l'opération de convolution est introduite dans chaque entrée. Dans l'architecture ConvLSTM, tous les attributs X_t , C_t , h_t , et les portes sont des tenseurs 3D [76].

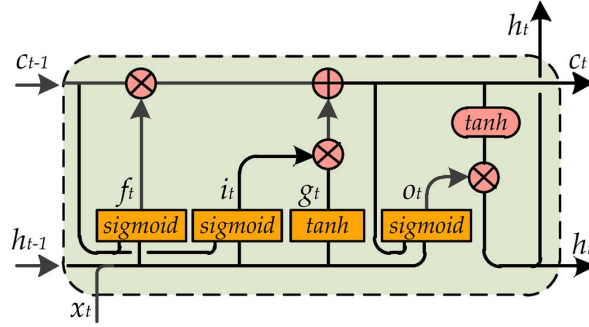


FIGURE 4.3 – Bloc mémoire ConvLSTM [133].

Les principales opérations qui définissent un ConvLSTM sont présentées par l'équation (4.2), où $*$ désigne l'opérateur de convolution et \odot le produit Hadamard.

$$\begin{aligned}
 i_t &= \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \odot C_{t-1} + b_i) \\
 f_t &= \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \odot C_{t-1} + b_f) \\
 C_t &= f_t \odot C_{t-1} + i_t \odot \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \\
 o_t &= \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \odot C_t + b_o) \\
 H_t &= o_t \odot \tanh(C_t)
 \end{aligned} \tag{4.2}$$

où :

- C_t et H_t représentent respectivement les cellules *cell state* et *hidden state* au temps t ,
- i_t , f_t , et o_t sont respectivement les portes d'entrée, d'oubli, et de sortie au temps t ,
- X_t est le vecteur d'entrée aux unités du LSTM,
- W représente les poids pour chaque composante,
- b désigne les paramètres de biais et σ , \tanh sont les fonctions d'activation.

4.1.3 Réseaux CNN-LSTM

Le CNN-LSTM est une architecture hybride dans laquelle les réseaux CNN et LSTM s'enchaînent [66]. La figure 4.4 présente la structure générale d'un réseau CNN-LSTM. Les données passent d'abord par les couches de convolution qui extraient les caractéristiques importantes. Ensuite la sortie du modèle CNN est transmise aux couches LSTM sous forme de vecteur (couche dense¹).

1. Couches où tous les neurones sont connectés aux neurones de la couche précédente.

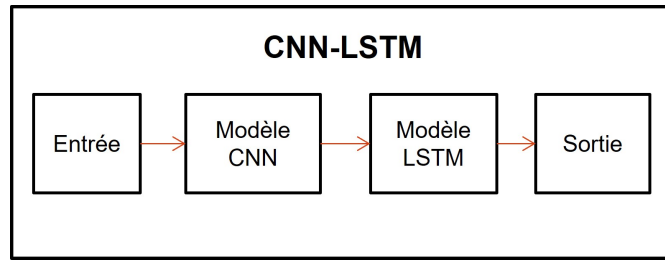


FIGURE 4.4 – Structure générale d’un réseau CNN-LSTM.

Les données ayant une structure spatio-temporelle ne sont pas facilement modélisées par les LSTM standards. L’idée sur laquelle repose le CNN-LSTM est d’utiliser d’abord les capacités du CNN à analyser les images pour capter les informations les plus pertinentes, et ensuite rajouter un réseau LSTM pour prendre en compte les dépendances temporelles entre les données.

Le CNN-LSTM est une architecture qui est à la fois spatialement et temporellement profonde, et a la flexibilité d’être appliquée à une variété de tâches impliquant des entrées et des sorties séquentielles. Tout comme le ConvLSTM, cette architecture est idéale pour des tâches telles que la génération de descriptions textuelles d’images, la reconnaissance vocale, le traitement du langage naturel, etc. [134, 135]. De façon générale, on peut dire que les réseaux CNN-LSTM sont appropriés pour les problèmes qui ont une structure spatiale (images, textes, vidéos, etc.) et temporelle (ordre des images dans une vidéo, ordre des mots dans un texte, série temporelle, etc.) dans leur entrée, comme dans la prédiction des STIS. Ainsi, l’architecture CNN-LSTM est également sélectionnée pour la phase expérimentale.

4.2 Évaluation des architectures LSTM pour la prédiction des STIS

Grâce à la disponibilité des données et de la puissance de calcul des ordinateurs, les algorithmes DL parviennent de nos jours à produire de très bons résultats. Cependant, la particularité de cette technique réside dans le fait que pour chaque type de problème ou de données, les architectures et les paramètres à utiliser varient. En effet, une architecture utilisée pour résoudre un problème de prédiction peut produire d’excellents résultats sur un type de données, et réagir mal face à un autre type de données. Avant de se lancer dans la résolution d’une tâche précise, il est primordial de déterminer quel type d’architecture est le plus approprié, en tenant compte également du type de données. Dans le cadre de nos travaux, trois variantes des architectures LSTM, à savoir les réseaux Stack-LSTM, CNN-LSTM et ConvLSTM, ont été étudiés et évalués de façon empirique. L’objectif de cette comparaison étant de déterminer la configuration de modèle la plus adaptée pour les tâches de prédiction dans les STIS.

4.2.1 Formulation du problème et méthodologie

Pour évaluer les performances des architectures Stack-LSTM, CNN-LSTM et ConvLSTM pour la prédiction future dans les SITS, nous avons testé des modèles basés sur chacune des architectures et entraînés sur les mêmes jeux de données pour la résolution d’un problème de prédiction à une étape, dans une série univariée.

De façon spécifique, étant données $X = X_1, X_2, \dots, X_n$ une STIS utilisé pour entraîner les modèles DL, $Z = Z_1, Z_2, \dots, Z_t$ une séquence considérée comme entrée pour la prédiction, l'objectif du problème considéré est de prédire $Y = f(Z)$, la sortie du modèle prédictif au temps $t + 1$, telle que $Y = Z_{t+1}$.

Pour la phase expérimentale, les séries d'images S1A du PNB et de la RFT ont été utilisées pour entraîner les modèles (se référer au chapitre 3 pour plus de détails sur les données utilisées). En plus des prétraitements déjà effectués sur ces données, les compositions colorées VV, VH, VH/VV sont d'abord réalisées pour chaque image avant d'être convertie en format JPG. Par la suite, toutes les images sont transformées en niveaux de gris, redimensionnées et les valeurs des pixels normalisées entre 0 et 1.

Pour la phase d'entraînement, 80% de l'ensemble des données ont été sélectionnées et 20% utilisées pour les tests. L'ordre chronologique des images est conservé dans chaque donnée puisqu'il s'agit de la résolution d'une tâche de prévision.

Avant la phase d'entraînement, la série temporelle est transformée pour un problème d'apprentissage supervisé en générant un tableau constitué des vecteurs `X_train` et `Y_train` comme indiqué dans le tableau 4.1. Le vecteur `X_train` correspond à la donnée en entrée et `Y_train` à la sortie. Dans la pratique, la taille de `X_train` correspond à la variable `timestep` et celle de `Y_train` au nombre de variables à prédire (`features`). Dans l'exemple du tableau 4.1, `timestep=5` et `features=1`. En d'autres termes, le modèle est entraîné pour recevoir 5 images en entrée et de prédire la sixième image en sortie.

TABLEAU 4.1 – Une série temporelle univariée organisée pour un apprentissage supervisé.

<code>X_train</code>	<code>Y_train</code>
$[X_1, X_2, X_3, X_4, X_5]$	X_6
$[X_2, X_3, X_4, X_5, X_6]$	X_7
$[X_3, X_4, X_5, X_6, X_7]$	X_8
...	...
$[X_{t-5}, X_{t-4}, \dots, X_{t-1}]$	X_t

4.2.2 Structure des modèles et paramètres utilisés

Pour chaque architecture, plusieurs configurations et paramétrages ont été testées avant d'adopter la structure adéquate.

4.2.2.1 Configuration des architectures

Pour les architectures ConvLSTM et Stack-LSTM, trois couches ConvLSTM et trois couches LSTM respectivement sont empilées, associées à des couches `batchnormalization` et `dropout`. La couche `batchnormalization` a essentiellement pour rôle de normaliser les valeurs provenant de la couche précédente tandis que la couche `dropout` permet d'éviter les phénomènes de surapprentissage. Pour finir, une couche `dense` est introduite à la sortie du modèle.

Pour le modèle CNN-LSTM, une couche de convolution à deux dimensions (Conv2D) a été d'abord été insérée suivie d'une couche `maxpooling2D` et d'une couche `flatten` qui permet d'obtenir un vecteur à une dimension dont les éléments sont transmis aux couches LSTM. Ensuite, deux couches LSTM ont été empilées et enfin une couche `dense` a été ajoutée pour la sortie. Les images étant des objets non linéaires, la fonction `Relu` a été utilisée comme fonction d'activation dans toutes les couches convolutionnelles afin d'ajouter de la linéarité aux images de sortie.

4.2.2.2 Hyperparamètres utilisés

Fonction d'optimisation : parmi les fonctions d'optimisation les plus utilisées dans la littérature, les algorithmes RMSProp, Adam, ou Momentum sont mentionnés. Pour l'étape d'apprentissage, les fonctions RMSProp et Adam ont été testées comme fonctions d'optimisation. En effet, la fonction RMSProp est généralement un bon choix pour les réseaux récurrents et la fonction Adam a permis d'obtenir des résultats satisfaisant dans de nombreuses études [66]. Pour nos travaux, la fonction Adam s'est avérée plus efficace et a été sélectionnée pour la suite des expérimentations.

Fonction de coût : les deux fonctions suivantes ont été testées comme fonction de perte : l'erreur quadratique moyenne ou *Root Mean Square Error* (RMSE) et l'erreur absolue moyenne ou *Mean Absolut Error* (MAE) exprimées par les équations (4.3) et (4.4) respectivement. L'objectif étant de déterminer laquelle des fonctions est la plus adaptée.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (p_i - o_i)^2}{n}} \quad (4.3)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |p_i - o_i| \quad (4.4)$$

où n représente la taille de l'échantillon, p_i représente les valeurs prédites et o_i les valeurs observées.

L'étape d'apprentissage a été réalisée avec un nombre de 100 epochs grâce à un GPU² accessible via un environnement virtuel. Pour les implémentations, la bibliothèque Keras utilisant le back-end Tensorflow a été utilisée.

4.2.3 Résultats obtenus

Dans cette section, les performances de prédiction des STIS à partir de trois modèles basés sur les architectures ConvLSTM, Stack-LSTM et CNN-LSTM sont comparées. L'étude est basée sur des séries temporelles d'images Sentinel-1 de la réserve de Faune de Togodo (RFT) et du Parc National de Bouba-Ndjida (PNB).

2. www.colab.research.google.com

4.2.3.1 Métriques d'évaluation

Dans l'apprentissage supervisé, l'objectif principal est de produire des modèles capables de faire de bonnes prédictions sur de nouvelles données. De faibles valeurs de perte d'apprentissage indiquent généralement que le modèle apprend bien. Cependant, un modèle avec de faibles valeurs ne signifie pas automatiquement qu'il est efficace. De plus, des valeurs trop proches de zéro indiquent parfois que le modèle est sur-ajusté (*overfitting*) et qu'il n'est donc pas en mesure d'obtenir de bonnes performances sur de nouvelles données. Il est donc important de considérer d'autres paramètres tels que la valeur de la fonction de perte avec les données de validation, le temps mis pour la phase d'apprentissage ou l'indice de similarité structurelle SSIM [136] entre l'image originale et l'image prédite. L'indice SSIM représente la donnée qui permet de mesurer la similarité structurelle entre les images prédites et les images réelles. Les valeurs proches de 1 indiquent une bonne similarité.

Au final, en plus de l'évolution des fonctions de perte d'apprentissage et de validation, les valeurs des métriques suivantes ont été considérées pour évaluer les modèles : RMSE, MAE, SSIM et le temps mis pour l'apprentissage (*training time*).

4.2.3.2 Discussions

Pour étudier l'évolution des fonctions de pertes selon le `timestep`, les courbes des valeurs obtenues pendant la phase d'apprentissage sont tracées. Le `timestep` varie de 5 à 10. Les valeurs moyennes de la perte et de l'écart-type sont également calculées. Les variations calculées sur les deux zones d'études étant sensiblement similaires, seuls les graphes représentant les résultats obtenus sur la zone du RFT sont présentés.

La figure 4.5 présente l'évolution du MAE d'apprentissage pendant l'entraînement du modèle. Avec le jeu de données considéré, nous observons qu'il n'y a pas de changement significatif lorsque les valeurs de `timestep` varient pour les architectures CNN-LSTM et Stack-LSTM (figure 4.5.c et figure 4.5.d). Cependant, comme le montre la figure 4.5.a, plus le `timestep` est élevé, plus les valeurs du MAE sont élevées avec le modèle ConvLSTM. On en déduit donc que l'utilisation du modèle ConvLSTM avec de longues séquences en entrée peut avoir des limites.

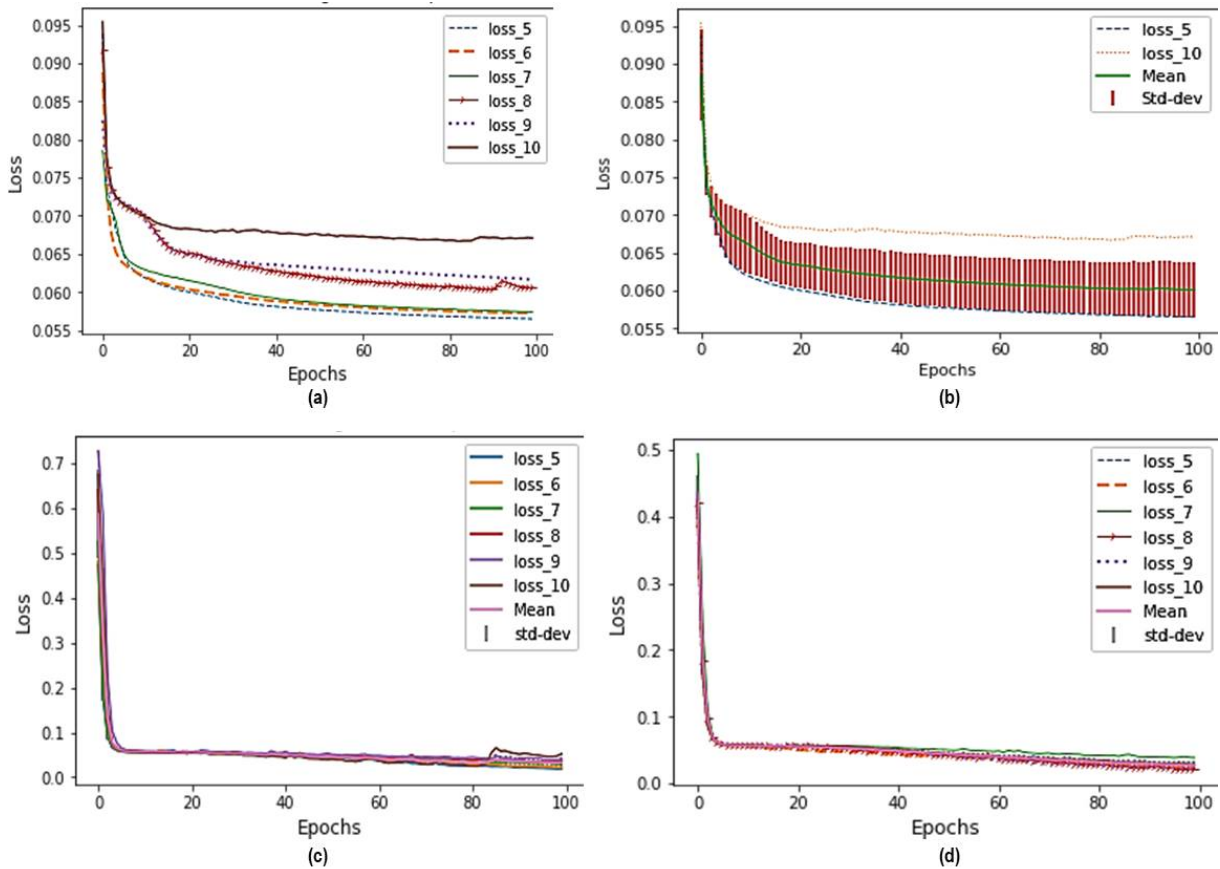


FIGURE 4.5 – Évolution de la perte d'apprentissage (MAE) au fil des epochs en fonction du `timestep` (5 à 10) : (a) perte avec ConvLSTM ; (b) moyenne et écart-type avec ConvLSTM ; (c) perte avec Stack-LSTM ; (d) perte avec CNN-LSTM.

La figure 4.6 présente l'évolution du MAE de validation pendant l'entraînement du modèle. Sur la figure 4.6.e, on peut observer qu'après une cinquantaine d'itérations, les valeurs de la perte de validation augmentent avec le modèle CNN-LSTM quel que soit la valeur de `timestep`. Cette situation pourrait signifier que l'entraînement devrait être arrêté à ce stade (*early stop*) ou que les paramètres devraient être davantage optimisés pour éviter l'*overfitting*.

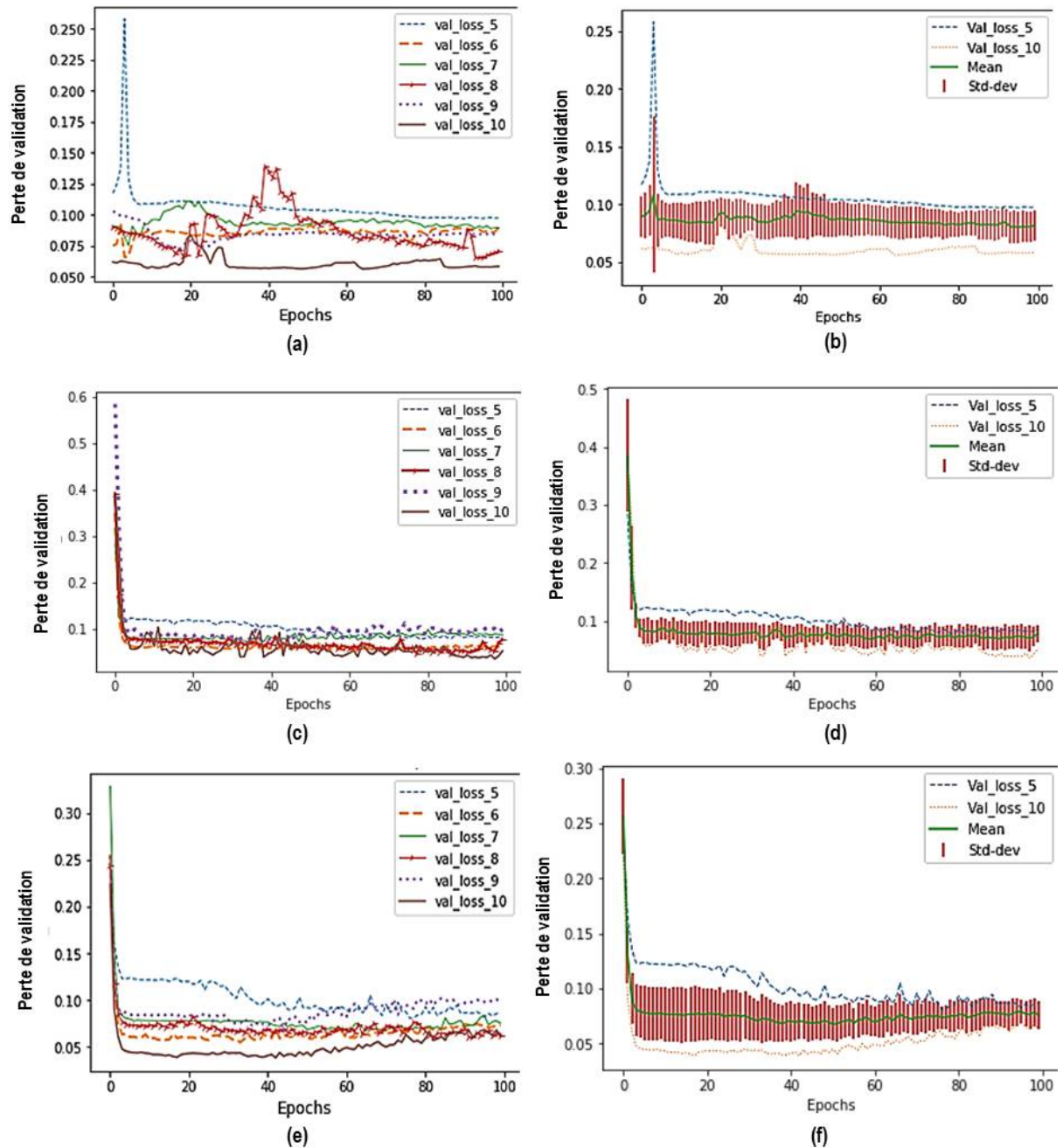


FIGURE 4.6 – Évolutions de la perte de validation (MAE) au fil des époques en fonction du `timestep` (5 à 10) : (a) perte avec ConvLSTM; (b) moyenne et écart-type avec ConvLSTM; (c) perte avec Stack-LSTM; (d) moyenne et écart-type avec Stack-LSTM; (e) perte avec CNN-LSTM; (f) moyenne et écart-type avec CNN-LSTM.

La figure 4.7 permet de comparer l'évolution du MAE d'apprentissage et de validation pour le modèle ConvLSTM. On constate que les valeurs ne suivent pas la même évolution pour les pertes d'apprentissage et les pertes de validation. Par exemple, lorsque le `timestep=10`, la perte de validation devient plus faible que la perte de d'apprentissage. Généralement, cette différence se produit lorsqu'un modèle est sous-entraîné (*under-fitting*). Pour surmonter le problème de sous-apprentissage, le modèle pourrait avoir besoin de plus de données pour la formation ou

certaines paramètres devraient être optimisés.

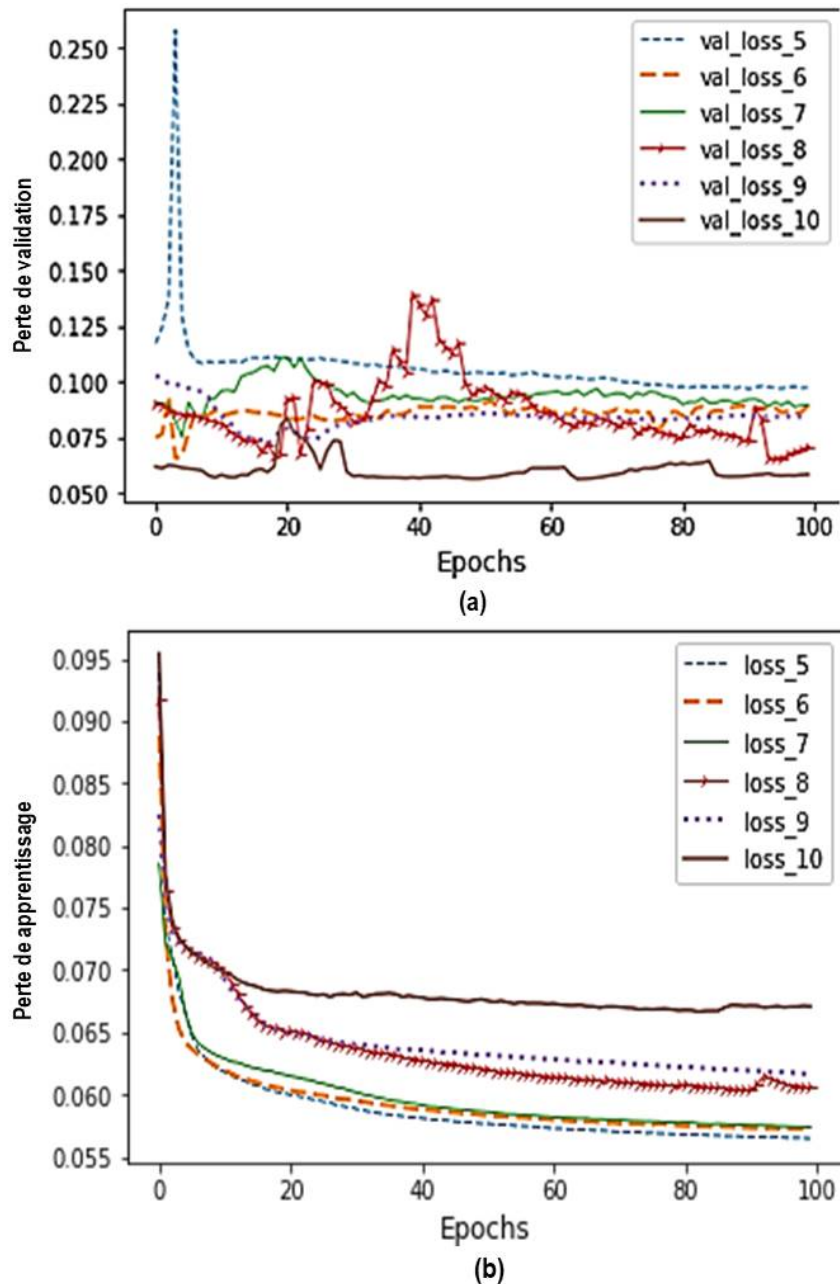


FIGURE 4.7 – Évolution de la fonction MAE au fil des époques en fonction du timestep (5 à 10) pour le modèle ConvLSTM : (a) perte validation ; (b) perte apprentissage.

Le tableau 4.2 et le tableau 4.3 résument les différentes valeurs obtenues par les métriques d'évaluation sur les images de la RFT et du PNB respectivement. La fonction de coût utilisée est le MAE et la fonction Adam pour l'optimisation. L'entraînement s'est fait sur 100 epochs à partir de la plateforme collaborative Google Colab qui permet d'utiliser un GPU virtuel. La taille de la séquence d'entrée timestep est fixée à 5. On constate de façon générale que les résultats sont pratiquement similaires pour les deux zones d'étude choisies.

TABLEAU 4.2 – Valeurs des métriques d’évaluation pour la zone RFT.
A=Apprentissage ; V=Validation.

Résolution	Modèle	Perte A.	Perte V.	Temps A.	SSIM
128 × 128	ConvLSTM	0,0574	0,0952	1933	0,83
	Stack-LSTM	0,0193	0,0824	304	0,60
	CNN-LSTM	0,0083	0,0860	315	0,67
64 × 64	ConvLSTM	0,0562	0,0975	605	0,64
	Stack-LSTM	0,0071	0,1161	305	0,53
	CNN-LSTM	0,0152	0,0847	301	0,72

TABLEAU 4.3 – Valeurs des métriques d’évaluation la zone PNB.
A=Apprentissage ; V=Validation.

Résolution	Modèle	Perte A.	Perte V.	Temps A.	SSIM
128 × 128	ConvLSTM	0,0635	0,0959	1927	0,79
	Stack-LSTM	0,0542	0,0823	302	0,71
	CNN-LSTM	0,0068	0,0633	305	0,78
64 × 64	ConvLSTM	0,0637	0,1005	612	0,69
	Stack-LSTM	0,0075	0,1223	302	0,63
	CNN-LSTM	0,0133	0,1002	303	0,78

Pour voir l’impact de la résolution des images sur la précision de la prédiction, les images originales ont été redimensionnées à (128 × 128) et (64 × 64) avant l’apprentissage des modèles. Comme le montrent les résultats, les valeurs minimums du *training loss* et du *validation loss* sont différentes en fonction des résolutions et selon l’architecture. Nous constatons que le *training loss* du CNN-LSTM est la plus faible avec des images de (128 × 128) alors que le Stack-LSTM produit la meilleure valeur en utilisant des images de (64 × 64).

En ce qui concerne le temps d’apprentissage, nous remarquons qu’il augmente de manière significative lorsque les résolutions d’image sont grandes pour le modèle ConvLSTM, comme présenté dans les tableaux 4.2 et 4.3. Cependant, la différence n’est pas très importante avec les modèles Stack-LSTM et CNN-LSTM. Le temps d’apprentissage est un facteur important à prendre en compte lors de l’évaluation des modèles et particulièrement lorsqu’on travaille sur des données de télédétection (images d’observation de la terre) qui sont très grandes. De plus, la qualité de la sortie en termes de résolution est très importante pour une meilleure interprétation. Nous pouvons observer également que l’architecture ConvLSTM prend trop de temps pour l’apprentissage et consomme plus de mémoire. Avec les paramètres de base de 100 epochs et `timstep=5`, le temps total de formation avec cette architecture est environ quatre fois plus long que le temps pris par les modèles Stack-LSTM et CNN-LSTM. Pour des résolutions supérieures à (128 × 128), la quantité de mémoire offerte par la plateforme Google collab devient insuffisante

pour l'entraînement du modèle ConvLSTM. Cependant, les valeurs de SSIM obtenues pour les images (128×128) prédites avec le ConvLSTM sont les meilleures comparées à celles obtenues avec le Stack-LSTM et le CNN-LSTM. Cela indique une bonne similarité entre les images réelles et celles prédites par le ConvLSTM.

Nous observons également que dans toutes les situations, le Stack-LSTM fournit les valeurs les plus faibles, comparativement au ConvLSTM et CNN-LSTM.

Les figures 4.8 et 4.9 présentent les résultats des prédictions effectuées par les trois modèles étudiés. La fonction de coût utilisée est la fonction MAE. Les figures montrent la séquence d'entrée, l'image réelle existante et celles prédites respectivement par les modèles ConvLSTM, Stack-LSTM et CNN-LSTM. Sur la figure 4.8, le `timestep=5` alors que `timestep=10` sur la figure 4.9. On constate que le ConvLSTM permet de prédire une image de meilleure qualité pour `timestep=5`, tandis que pour `timestep=10`, c'est le CNN-LSTM qui produit la meilleure prédiction.

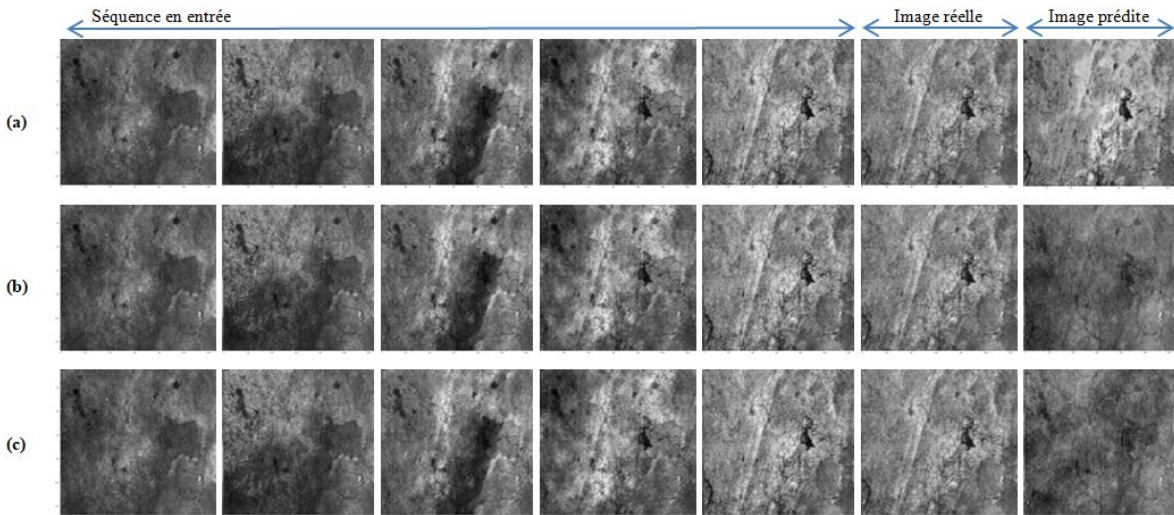


FIGURE 4.8 – Résultats de prédiction sur des images (128×128). `timestep=5`. Prédiction à partir des modèles : (a) ConvLSTM ; (b) Stack-LSTM ; (c) CNN-LSTM.

Après analyse des résultats, on peut conclure premièrement que l'utilisation de l'architecture ConvLSTM pour la prédiction des STIS peut être une bonne approche si l'on dispose de ressources suffisantes. En effet, lorsque la taille des images et la longueur des séquences deviennent plus élevées, l'architecture ConvLSTM n'est pas très performante et les résultats ne sont pas assez satisfaisants. De plus, à cause des opérations de convolution, le temps de traitement est significativement plus élevé avec le ConvLSTM qu'avec le CNN-LSTM et le Stack-LSTM.

Deuxièmement, bien que les valeurs de la fonction de perte d'apprentissage soient les plus faibles avec l'architecture Stack-LSTM dans certains cas, on observe par contre que les valeurs de SSIM sont les moins bonnes dans toutes les situations avec cette architecture. Ce qui signifie que pour le modèle Stack-LSTM, les images prédites et les images réelles présentent le moins de similarité comparativement aux prédictions faites par les réseaux CNN-LSTM et ConvLSTM.

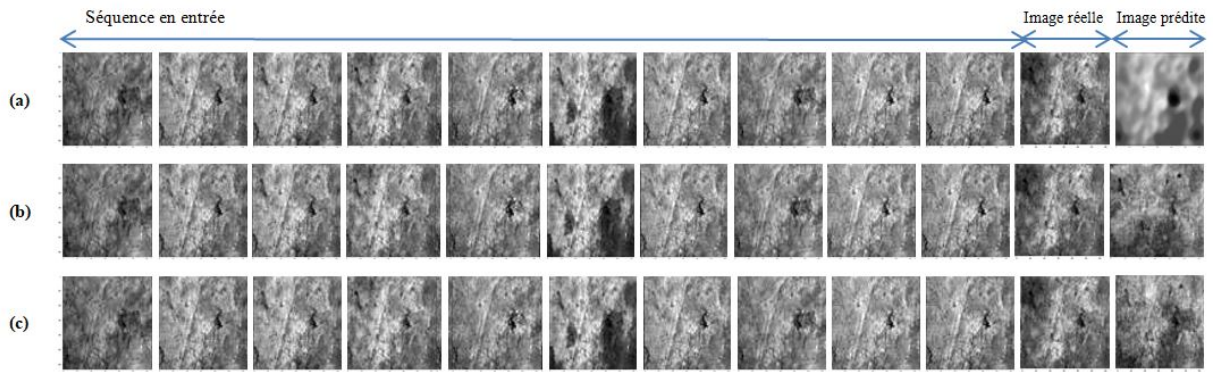


FIGURE 4.9 – Résultats de prédiction sur des images (128×128). $\text{timestep}=10$. Prédiction à partir des modèles : (a) ConvLSTM ; (b) Stack-LSTM ; (c) CNN-LSTM.

Enfin, en considérant toutes les métriques d'évaluation, le CNN-LSTM présente les meilleures valeurs dans plus de cas que le ConvLSTM et le Stack-LSTM.

Pour toutes ces raisons, nous déduisons que les architectures CNN-LSTM et ConvLSTM sont recommandées pour les tâches de prévision utilisant des séries chronologiques d'images d'observation de la Terre, à conditions d'avoir des ressources mémoires suffisantes.

4.3 Comparaison des méthodes DL et CA-Markov

4.3.1 Formulation du problème et description des méthodes

L'objectif de cette section est de comparer les méthodes CA-Markov et DL pour la prédiction des changements de classes d'occupation de sol, plus connu par le terme *Land Use and Land Cover Change prediction*. Pour cela, étant donné un ensemble d'images $X = \{X_1, X_2, \dots, X_t\}$, nous appliquerons chaque méthode pour prédire l'image X_{t+1} au temps $t + 1$.

Chacune de ces méthodes est utilisée tour à tour sur des images radar Sentinel-1A du PNB, afin de prédire l'état futur des classes d'occupation du sol. Les images prédites par chaque méthode sont ensuite comparées aux images réelles pour l'évaluation des modèles à l'aide des métriques décrites dans la section 4.3.3.1.

La figure 4.10 décrit le workflow général des tâches exécutées pour la réalisation de cette étude. Trois grandes phases sont considérées :

- (I) toutes les images de la zone d'intérêt sont collectées et prétraitées pour former la base de données générale ;
- (II) les cartes thématiques (TM) sont produites et les données auxiliaires collectées pour former le modèle prédictif
- (III) les images sont préparées et entraînées par les réseaux CNN-LSTM et ConvLSTM pour la prédiction. Pour finir les résultats obtenus en (I) et (II) sont évalués et comparés.

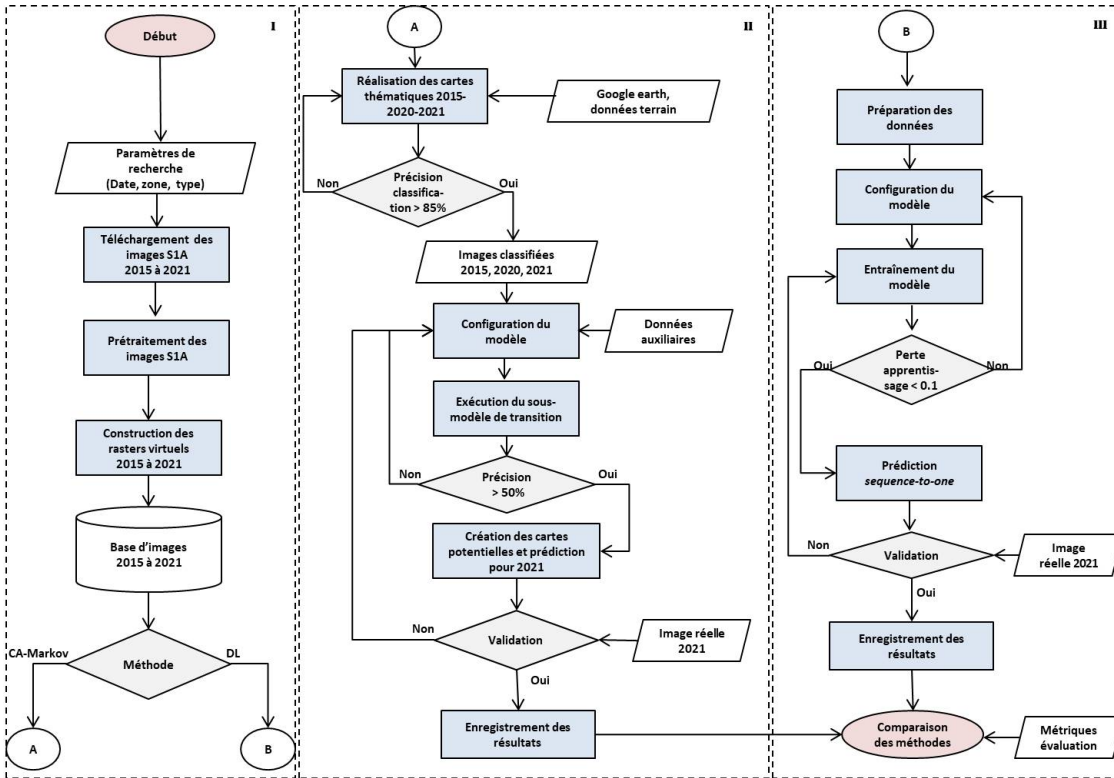


FIGURE 4.10 – Workflow général de comparaison des méthodes CA-Markov et DL.

4.3.1.1 Méthode CA-Markov

Le modèle CA-Markov est une méthode hybride qui combine le modèle CA et les chaînes de Markov. Il a été largement utilisé pour la prédiction des changements de sol et a donné des résultats satisfaisants : [137, 138, 139, 140]. Dans un modèle CA-Markov, la partie chaîne de Markov traite des changements temporels entre les classes d'utilisation des sols. En effet, un modèle de chaîne de Markov est un processus stochastique qui analyse la probabilité de passer d'un état à un autre, c'est-à-dire que l'état d'un système au temps t_2 est prédit à partir de l'état au temps t_1 [141]. Ainsi, l'état à un temps particulier t dépend exclusivement de l'état au temps $t - 1$ et non des autres états précédents, c'est-à-dire aux temps $t - 2$, $t - 3$ et ainsi de suite [11]. L'équation de probabilité de transition est définie par (4.5).

$$P(X_t = a_t | X_{t-1} = a_{t-1}; X_0 = a_0, \dots, X_{t-1} = a_{t-1}) = P(X_t = a_t | X_{t-1} = a_{t-1}) \quad (4.5)$$

Où $P(X_t = a_t | X_{t-1} = a_{t-1})$ représente la probabilité de transition de l'état a_{i-1} à l'état a_i en un pas de temps [11].

Les chaînes de Markov ne prennent pas en compte la distribution géographique des pixels lors des prédictions. La partie CA d'un modèle CA-Markov, contrôlée par des règles déterminées localement, traite de la dynamique spatiale [70]. En effet, dans un modèle CA, les changements d'état d'un pixel ne dépendent pas seulement de son état précédent, mais aussi de l'état de ses cellules adjacentes [11]. Les modèles CA sont adaptés à la modélisation des phénomènes

liés à l'extension spatiale. Ils sont régis par des règles de transition spéciales qui permettent de déterminer la direction des données pendant le changement. Ainsi, la combinaison du modèle CA avec les chaînes de Markov permet d'améliorer significativement la précision des prédictions.

La figure 4.11 illustre l'architecture globale d'un modèle CA-Markov.

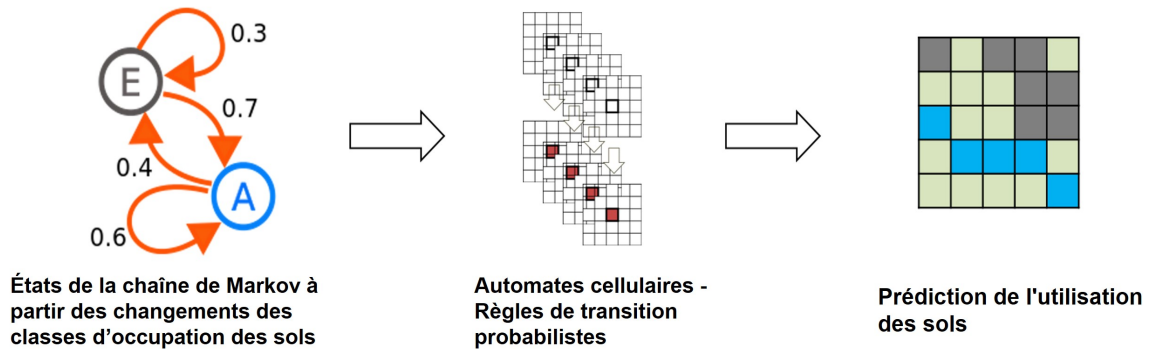


FIGURE 4.11 – Architecture d'un automate cellulaire intégré à la chaîne de Markov.

4.3.1.2 Méthode DL

En ce qui concerne la partie DL, les architectures CNN-LSTM et ConvLSTM sont utilisées pour faire des prédictions. Pour implémenter la méthode DL, la bibliothèque open source de Google, Tensorflow, dédiée à l'apprentissage automatique est utilisée. L'implémentation des algorithmes DL se fait à partir de la version professionnelle de l'environnement Google Colaboratory (Colab pro), permettant un accès cloud privilégié aux unités de traitement graphique (GPU) les plus rapides.

Toutes les images acquises de 2015 à 2021 sont utilisées : 70% pour l'étape d'entraînement, 15% pour la validation et 15% pour les tests. Pour manipuler les images au format TIFF sans perdre les informations géographiques à partir de Python, la bibliothèque Rasterio a été utilisée. Il s'agit d'un nouveau projet open source de l'équipe satellite de Mapbox³, basé sur GDAL et Numpy, conçue pour faciliter la manipulation des données matricielles géo-spatiales. Le modèle a été entraîné pour recevoir cinq images en entrée et en prédire une en sortie (prédiction *sequence-to-one*). Après l'entraînement du modèle, les cinq dernières images de l'année 2020 sont utilisées pour prédire celle de 2021. Des classifications sur les deux images sont ensuite réalisées afin d'évaluer les taux d'erreur de prédiction pour chaque classe qui servent de base pour le calcul de la précision globale, comme cela est fait avec la méthode CA-Markov.

4.3.2 Structure des modèles et hyperparamètres utilisés

4.3.2.1 Modèle CA-Markov

Pour la méthode CA-Markov, trois cartes thématiques correspondant aux années 2015, 2020 et 2021, respectivement, sont produites à l'aide des algorithmes de classification Random Forest.

3. Société spécialisée dans la cartographie en ligne

La classification est réalisée avec une précision d'environ 87%, ce qui est suffisant pour obtenir de bons résultats. Notons que la qualité des images classées influe sur la qualité des résultats finaux. Il est donc essentiel d'avoir des images classées avec une bonne précision.

Dans cette étude, cinq classes sont considérées pour toutes les images comme décrit dans le tableau 4.4. Toutes les valeurs de pixel représentent la même classe dans toutes les données.

TABLEAU 4.4 – Principales classes d'occupation de sol considérées.

Valeur	Classe	Description
1	Forêt	Écosystèmes composés des arbres à larges feuilles
2	Savane	Zones avec des arbres de petites tailles dispersés
3	Agriculture	Surfaces cultivées
4	Sol nu	Zones dénuées de végétation
5	Eau	Ensemble des cours d'eau, rivières, lacs ou fleuves

Ensuite, les images de 2015 et de 2020 sont utilisées pour prédire l'image attendue en 2021. Cette image prédite est ensuite comparée à l'image réelle de 2021 pour validation. Au cours du processus, des données auxiliaires telles que le modèle numérique d'élévation ou *Digital Elevation Model* (DEM) et la carte routière sont ajoutées au modèle pour améliorer les résultats de la prédiction.

4.3.2.2 Modèles DL

Les valeurs des hyperparamètres listées dans le tableau 4.5 sont utilisées pour entraîner les modèles DL.

TABLEAU 4.5 – Valeurs des hyperparamètres - Méthode DL.

Hyperparamètres	CNN-LSTM	ConvLSTM
Fonction d'optimisation	Adam	Adam
Fonction de perte	RMSE	RMSE
Nombre de filtres	8	16
Taille du noyau	3×3	3×3
Nombre d'époques	100	100

4.3.3 Résultats obtenus

L'objectif de cette étude est de comparer les méthodes DL et CA-Markov pour les prédictions de changement des classes d'occupation des sols. Il est en effet question de déterminer quelle méthode réalise la meilleure prédiction en considérant les valeurs des métriques d'évaluation. Ainsi, les mesures permettant de comparer des images ont été utilisées afin d'évaluer les deux méthodes. Ces métriques sont décrites dans la section 4.3.3.1.

4.3.3.1 Métriques d'évaluation

Les mesures suivantes, utilisées dans les tâches de vision par ordinateur, ont été utilisées pour évaluer les méthodes CA-Markov et DL :

- MSE : le MSE est considéré comme la mesure la plus répandue pour l'évaluation de la qualité des images. Il s'agit d'une métrique de référence et les valeurs les plus proches de zéro indiquent généralement les meilleures similarités [104]. Pour deux images x et y de taille $r \times c$, le MSE est défini selon l'équation (4.6).

$$MSE = \frac{1}{r \times c} \sum_{i=0}^{r-1} \sum_{j=0}^{c-1} (x(i, j) - y(i, j))^2 \quad (4.6)$$

- PSNR : il calcule le rapport signal/bruit maximal, en décibels, entre deux images. Son but est d'évaluer la qualité entre l'image originale et l'image reconstruite par la valeur du rapport. Le PSNR est la technique la plus couramment utilisée pour mesurer la qualité de reconstruction des images compressées [104].
- SSIM : l'indice SSIM mesure la similarité structurelle entre une image prédite et l'image réelle. Les valeurs varient entre 0 et 1 et celles proches de 1 indiquent généralement de bonnes similarités entre les deux images comparées.
- Le coefficient de corrélation r : encore appelé le coefficient de corrélation produit-moment de Pearson (CCPM), cette métrique est couramment utilisée dans les domaines des statistiques, de la reconnaissance des formes et du traitement des images. Le Coefficient r est utilisé pour effectuer des comparaisons entre deux images, pour mesurer la corrélation linéaire entre des données, ou pour quantifier la disparité, etc. [142]. Cette métrique exprimée par (4.7), renvoie une valeur comprise entre -1 et 1. La valeur absolue du coefficient indique le niveau de corrélation entre les données comparées. Les valeurs proches de 1 indiquent une forte relation positive, tandis que la valeur 0 n'indique aucune relation. Lorsque r est négatif, il s'agit d'une relation inverse, par opposition à une relation directe dans le cas de valeurs positives.

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (4.7)$$

où x_i et y_i représentent l'intensité du i^{ieme} pixel pour la première et la seconde image respectivement. \bar{x} et \bar{y} sont l'intensité moyenne de la première et de la deuxième image respectivement, et n la taille de l'échantillon.

- Prediction accuracy (PA) : Pour valider les modèles prédictifs avec la méthode CA-Markov, l'indice de kappa, défini comme dans [143], est adopté comme mesure standard pour la validation des modèles. La précision de prédiction pour chaque classe d'occupation de sol est calculée à partir des images prédites et réelles. Ensuite, la précision globale du modèle en est déduite. Dans notre étude, cette technique sera utilisée pour comparer les prédictions faites par les méthodes DL et CA-Markov. La métrique notée ici PA , est définie par (4.8).

Plus les valeurs de PA sont proches de 100%, plus la concordance entre deux cartes est forte.

$$PA = \frac{1}{n} \sum_{i=1}^n (100 - |\frac{y_i - x_i}{x_i} \times 100|) \quad (4.8)$$

où y_i est la superficie de la i^e classe d'utilisation du sol dans l'image prédite, x_i est la superficie de la i^e classe d'utilisation du sol dans l'image réelle et n est le nombre de classes.

- Nombre de pixels proches de zéro (NPCZ) : Pour mesurer la similarité entre l'image prédite et l'image réelle (pixel par pixel), une simple opération de soustraction matricielle peut être effectuée. L'image résultante est généralement appelée « image de différence » ou *Difference Image* (DI). Une DI sombre avec des valeurs proches de 0 indique généralement que la prédiction est bonne. Afin de quantifier cette mesure, nous définissons dans cette étude une variable notée NPCZ, qui correspond au pourcentage de pixels proches de 0 dans l'image DI. Une constante ϵ est définie afin de déterminer les limites supérieure et inférieure des valeurs de pixels à considérer.

NPCZ est défini par l'équation (4.9), où $r \times c$ est le nombre total de pixels dans l'image DI, et $d_{i,j}$ sont les valeurs des pixels de l'image DI.

$$NPCZ = \frac{1}{r \times c} \sum_{i=1}^r \sum_{j=1}^c f(i, j) \quad (4.9)$$

Où

$$f(i, j) = \begin{cases} 1, & \text{si } -\epsilon < d_{i,j} < \epsilon \\ 0, & \text{sinon.} \end{cases}$$

4.3.3.2 Résultats

La figure 4.12 illustre le résultat de prédiction obtenu avec l'outil Land change Modeler (LCM) du logiciel Terset IDRISI. Les cartes d'occupation de sol de 2015 et de 2020 ont été utilisées en entrée du modèle, et l'image attendue en 2021 prédite.

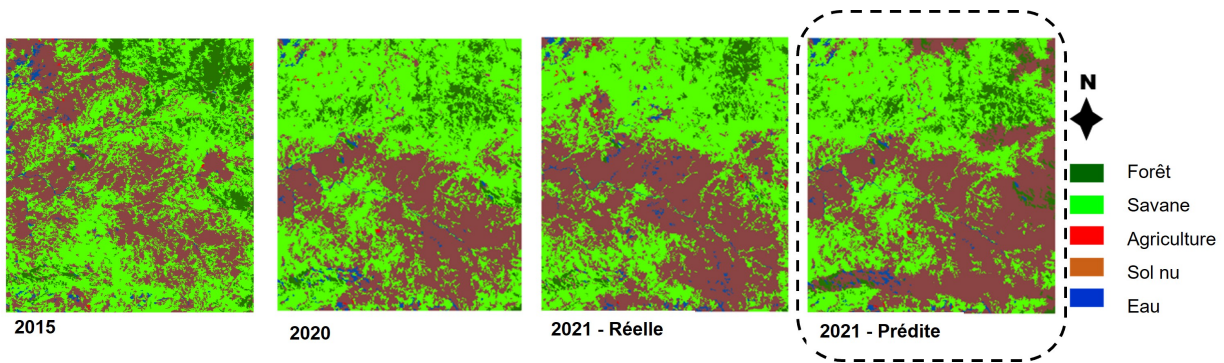


FIGURE 4.12 – Résultat de la prédiction avec la méthode CA-Markov.

Après la prédiction de l'image 2021, les surfaces des classes d'occupation du sol prédites sont comparées à celles de la carte de référence. Les précisions obtenues par classe sont ensuite utilisées pour calculer la précision globale de prédiction (PA). Le tableau 4.6 montre les surfaces en km^2 de chaque classe thématique dans l'image réelle et dans l'image prédite, ainsi que la précision de prédiction pour chaque classe.

Le tableau 4.7 présente les valeurs des paramètres d'évaluation considérés pour comparer les méthodes. L'observation des métriques d'évaluation données dans le tableau 4.7 permet de déduire que le modèle basé sur ConvLstm est le plus performant pour tous les critères sauf le PSNR qui mesure la qualité de l'image reconstruite. La valeur élevée du PSNR obtenue par CA-Markov signifie que la qualité de l'image qu'il prédit est meilleure que celle des images prédites par les autres modèles. Les performances inférieures en termes de qualité enregistrées par les algorithmes DL sont dues au redimensionnement effectué sur les images avant leur traitement.

TABLEAU 4.6 – Surfaces des classes d'occupation de sol.

Classe	2021-Réelle	2021-Prédite	% Précision
Forêt	162,258	258,223	40,86
Savanne	1587,51	1476,28	92,99
Agriculture	3,264	3,0246	92,65
Sol nu	1513,51	1534,12	98,63
Eau	48,304	43,205	89,44

Précision globale de prédiction (PA) = 83%

TABLEAU 4.7 – Valeurs des métriques utilisées pour comparer les modèles CA-Markov et DL.

Méthode	MSE	PSNR	SSIM	r	PA	NPCZ
CA-Markov	0,64	30,02	0,67	0,42	83%	47%
CNN-LSTM	0,69	28,27	0,75	0,78	91%	43%
ConvLSTM	0,58	28,52	0,96	0,98	98%	97%

Le DI présenté par la figure 4.13 a permis de calculer les valeurs de *NPCZ*. Premièrement, on peut remarquer que l'image DI avec ConvLSTM est plus sombre que les autres. Cela indique que la prédiction spatio-temporelle a été faite correctement avec ConvLSTM. La valeur de *NPCZ* calculée à partir du DI confirme cette hypothèse. 97% pour ConvLSTM contre seulement 47% et 43% pour les autres méthodes.

La figure 4.14 présente les résultats des prédictions obtenus par les modèles ConvLSTM et CNN-LSTM.

Le but de cette étude était de comparer les performances des méthodes DL et CA-Markov pour prédire les classes d'occupation du sol. Les résultats obtenus montrent que toutes les méthodes produisent des précisions de prédiction globales acceptables. De façon générale, les



FIGURE 4.13 – Images de la différence entre image prédite et image réelle (DI).

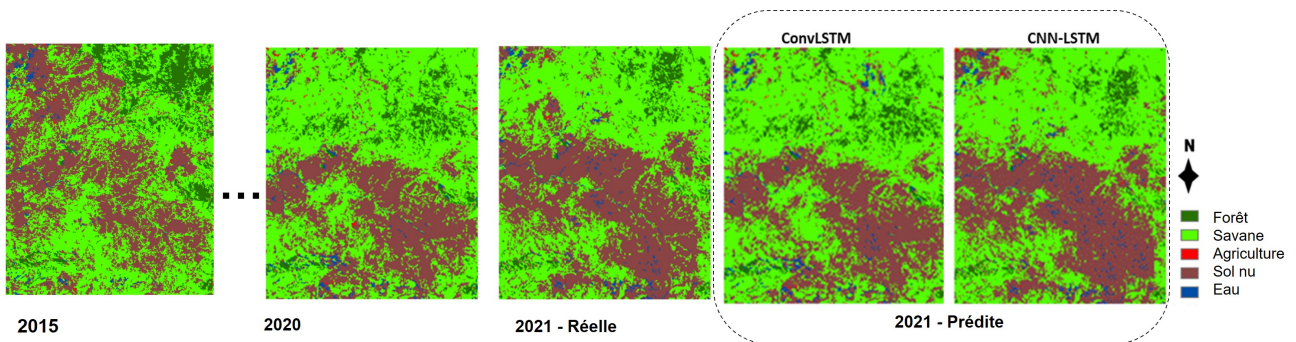


FIGURE 4.14 – Résultat des prédictions à partir des modèles ConvLSTM et CNN-LSTM.

résultats de ces expérimentations permettent de déduire que les modèles basés sur les architectures CNN-LSTM et ConvLSTM peuvent permettre d’atteindre de meilleurs résultats pour la prédiction des classes d’occupation de sol qu’un modèle CA-Markov. En plus, la méthode CA-Markov requiert une étape supervisée importante d’extraction de caractéristiques et l’utilisation des données auxiliaires qui influencent la qualité des résultats finaux.

Les comparaisons des différentes méthodes faites dans ce chapitre ont permis de conclure que les réseaux CNN-LSTM et ConvLSTM sont plus adaptés pour la prédiction des STIS que les Stack-LSTM et les automates cellulaires intégrés aux chaînes de Markov. Cependant, déterminer quelle architecture DL est la plus adéquate pour résoudre un problème ne suffit pas pour avoir des résultats satisfaisants. Il est nécessaire d’optimiser les modèles pour améliorer ses performances. Dans le chapitre 5, nous proposerons un modèle basé sur l’architecture CNN-LSTM pour la prédiction de la déforestation. Ce modèle sera amélioré par des techniques d’augmentation de données et d’optimisation des hyperparamètres.

Chapitre 5

Cas d'usage d'un modèle DL pour la prédiction des changements du couvert forestier : application à la réserve de Biosphère du Dja au Cameroun

Objectifs

Nous proposons dans ce chapitre un modèle d'apprentissage profond basé sur l'architecture CNN-LSTM pour la prédiction de la déforestation autour de la réserve de Biosphère du Dja en 2025. Les méthodes utilisées pour améliorer les performances du modèle, notamment l'augmentation de données et l'optimisation des hyperparamètres sont décrites. De même, les résultats de prédiction multi-step obtenus sont présentés et discutés.

Sommaire

5.1	Introduction	92
5.2	Outils et méthodes	94
5.2.1	Données	94
5.2.2	Modèle	98
5.3	Résultats et discussions	101
5.3.1	Outils utilisés	101
5.3.2	Paramètres utilisés	102
5.3.3	Prédiction one-step et validation du modèle	104
5.3.4	Prédiction multi-step pour l'année 2025	108
5.4	Conclusion	109

5.1 Introduction

La déforestation est définie comme la réduction des zones forestières. Elle se produit lorsque des zones forestières sont perdues au profit d’autres utilisations telles que l’agriculture, l’urbanisation ou les activités minières. La déforestation a de nombreuses conséquences telles que la menace pour la biodiversité, la fragilisation des sols, le changement climatique. Malgré le statut d’aire protégée attribué à certaines zones, elles sont toujours confrontées au phénomène de déforestation. C’est le cas de la Réserve de Faune du Dja plus connue sous le nom de Réserve de Biosphère du Dja (RBD) au Cameroun [119].

Malgré son statut d’aire protégée, la RBD fait face continuellement au problème d’exploitation illégale de la forêt ; ce qui explique d’ailleurs les nombreuses actions entreprises dans le cadre de la lutte contre les changements climatiques par les défenseurs de l’environnement sur la stratégie Redd+ (Réduction des émissions des gaz dues à la dégradation et à la déforestation). Parmi les principales causes de déforestation évoquées par l’Organisation Non Gouvernementale (ONG) [Greenpeace Afrique](#), agissant pour la protection de l’environnement et de la planète, on note le développement de la vaste plantation d’hévéas d’une entreprise à quelques centaines de mètres de la frontière de la RBD. La figure 5.1 représente la carte réalisée en 2016 par Green Peace pour illustrer les surfaces de nouvelles plantations près de la RBD depuis 2011. Au total, 5.930 ha de terres ont été dévastés au profit de la culture de l’hévéa, dont 2220 ha entre février 2015 et janvier 2016 ; et 310 ha entre janvier 2016 et avril 2016.

Afin d’assurer un meilleur suivi des surfaces ayant subi la déforestation (et d’anticiper sur la prise de décision), il est important, en plus des solutions déjà existantes, de concevoir et d’améliorer les systèmes de prévision de la déforestation dans cette zone.

L’objectif général de ce chapitre est de proposer un modèle DL pour la prédiction de la déforestation autour de la réserve de Biosphère du Dja. Les images Sentinel-1A de la zone d’étude sont collectées et prétraitées grâce au workflow décrit dans le chapitre 3.

Pour la conception du modèle, l’architecture CNN-LSTM définie dans le chapitre 4 est utilisée. En effet, l’évaluation des architectures présentée dans le chapitre 4 a montré que les réseaux CNN-LSTM et ConvLSTM sont adéquats pour la prédiction des STIS. Cependant, il a également été noté que l’architecture ConvLSTM nécessite beaucoup de ressources mémoire à cause des opérations de convolutions dans les cellules LSTM. Compte tenu de la taille des données et de la longueur de la série temporelle d’images utilisés pour ces travaux, nous utiliserons uniquement l’architecture CNN-LSTM.

Les performances du modèle CNN-LSTM présenté dans le chapitre 3 seront améliorées par :

- la configuration des couches du modèle ;
- une technique d’augmentation de données que nous proposons ;
- la régularisation du modèle ;
- l’optimisation des hyperparamètres par l’approche hyperbande.

Pour prédire la déforestation en 2025, le résultat de la prédiction pour 2022 sera d’abord comparé à l’image réelle de 2022 pour la validation, puis un processus de prédiction *multi-step* défini dans la section 1.3.1 sera appliqué pour obtenir la prédiction de 2025.

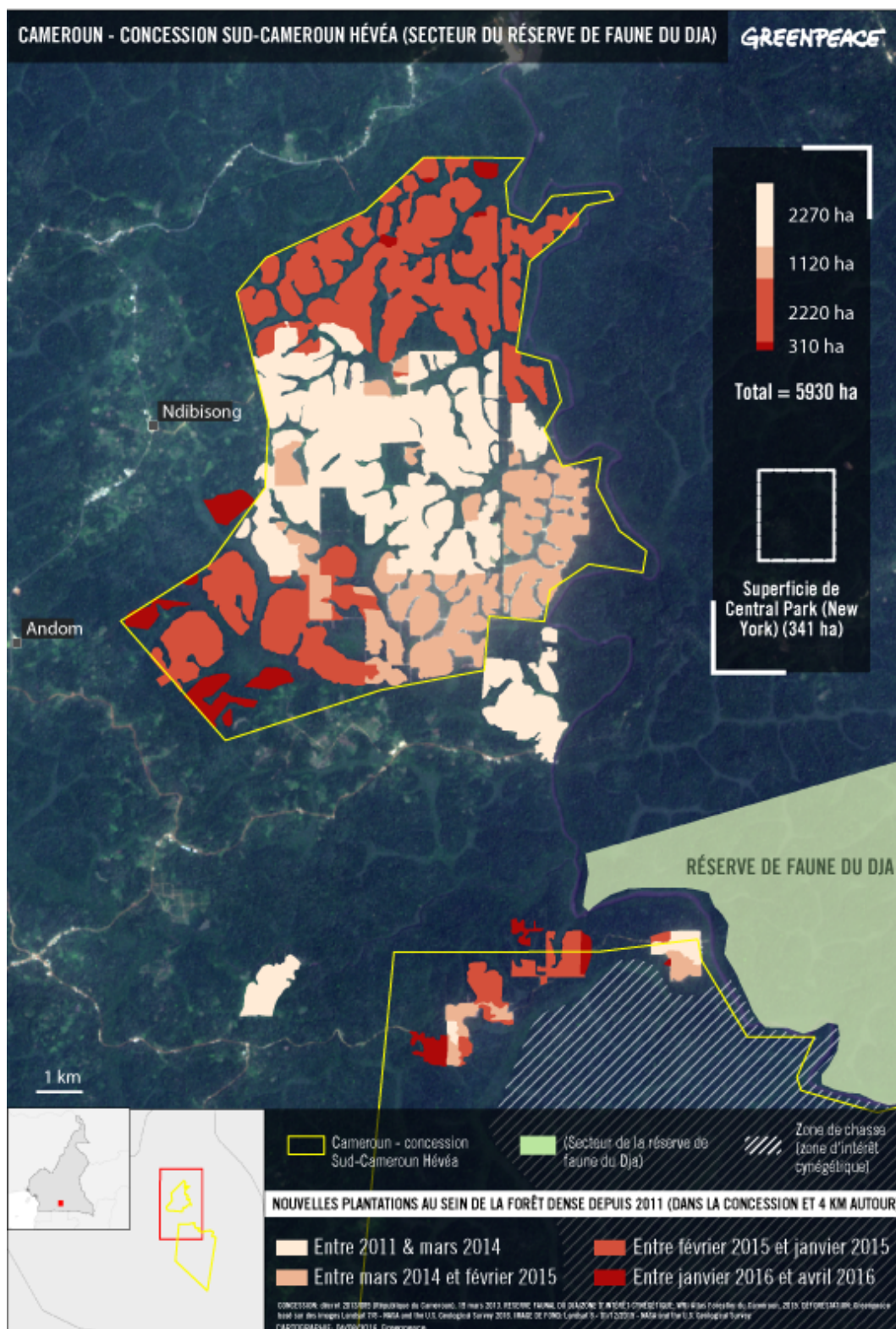


FIGURE 5.1 – Répartition des plantations d’hévéas près de la RBD [144].

5.2 Outils et méthodes

5.2.1 Données

Pour prédire la déforestation autour de la RBD, une série d'images Sentinel-1A a été utilisée comme Dataset pour l'entraînement des modèles. En plus des prétraitements de base effectués sur les images brutes tels que décrits dans le chapitre 3, les données ont été préparées afin d'alimenter les réseaux d'apprentissage profonds. Au total, une série de 195 images acquises entre le 07 avril 2015 et le 24 janvier 2022 a été utilisée.

5.2.1.1 Description de la zone d'étude

La zone d'intérêt choisie pour les expérimentations est une région d'environ 98 000 ha, située à la périphérie Ouest de la RBD, où est implantée la société de plantation d'hévéas et le barrage hydroélectrique de Mékin¹.

La figure 5.2 permet d'avoir un aperçu sur le type de végétation dans la zone d'étude. Les surfaces déforestées sont présentées par une couleur vert clair sur l'image optique (figure 5.2.a); tandis que sur l'image radar (figure 5.2.b), elles présentent une texture lisse, contrairement à la zone de forêt où elles ont un aspect plus rugueux.

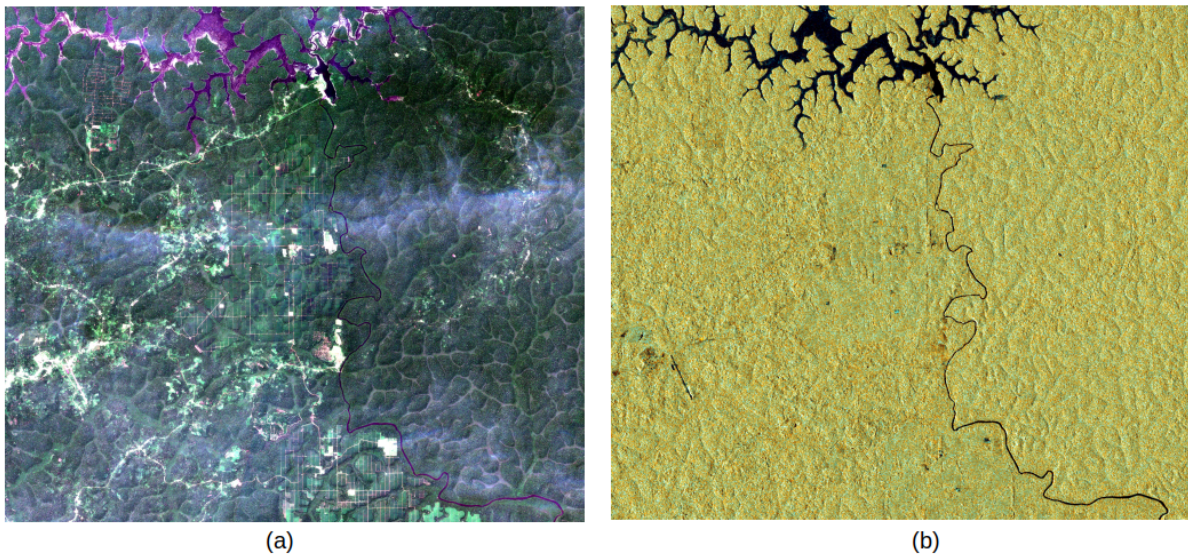


FIGURE 5.2 – Représentation de la zone d'étude : (a) Composition colorée RVB (B4,B3,B2) de l'image Sentinel-2B du 04 janvier 2020; (b) Composition colorée VV, VH, VH/VV de l'image Sentinel-1A du 31 décembre 2021.

Les valeurs des pixels sur les zones déforestées sont inférieures à celles des surfaces de forêt. Les valeurs des zones claires varient approximativement entre -20 et -15 dB et celles des zones de forêts denses entre -15 et -10 dB. La figure 5.3 montre un exemple de profil temporel de deux pixels issus de deux types de surface. La figure 5.3.a décrit les variations des valeurs du

1. Village situé dans la zone d'intérêt

pixel de coordonnées (208 172 , 347 557)² qui se trouve dans une région déforestée. On constate en effet une baisse brusque des valeurs du pixel aux environs de la cinquantième occurrence ; et ces valeurs recommencent à croître progressivement peu de temps après. Par contre, le pixel de coordonnées (222 322 , 348 593) sur la figure 5.3.b se trouve dans une zone forestière intacte : les valeurs du pixel sont donc quasi constantes au cours du temps.

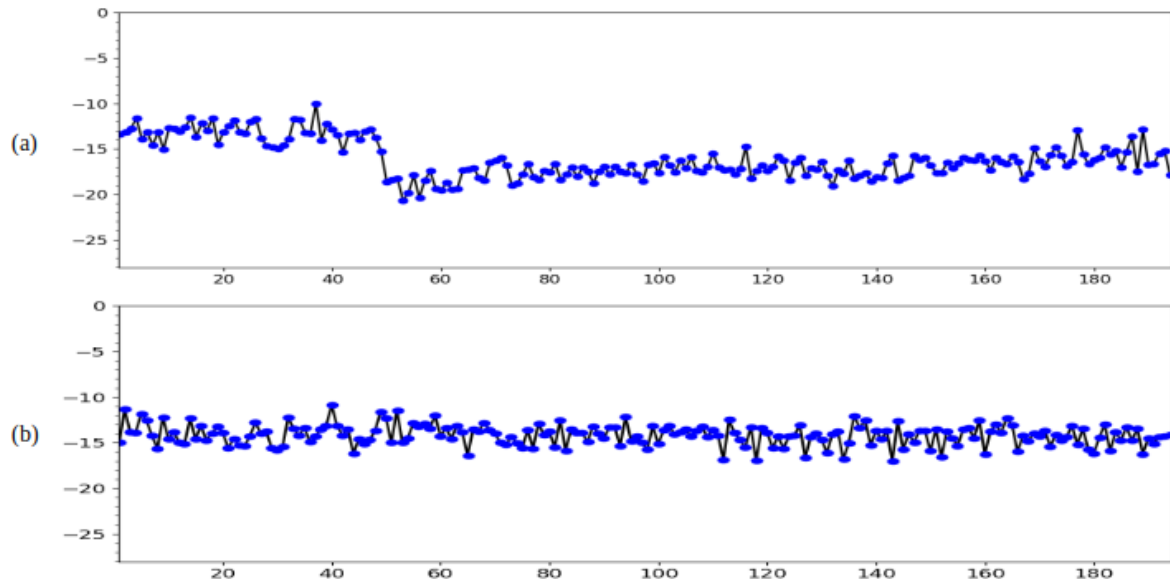


FIGURE 5.3 – Courbes représentant les variations des valeurs des pixels au cours du temps. En abscisse : pas temporel (12 jours). En ordonnées : valeurs des pixels en décibels : (a) pixel appartenant à une zone déforestée ; (b) pixel issu d’une zone forestière intacte.

5.2.1.2 Préparation des données

Après la collecte des données, l’étape suivante consiste à préparer ces données afin de les rendre utilisables par des algorithmes d’apprentissage supervisés. Dans un problème d’apprentissage supervisé univarié, il y a des variables d’entrée (X), des variables de sortie (Y), et un modèle qui utilise un algorithme pour apprendre la fonction de correspondance entre l’entrée et la sortie : $Y = f(X)$.

Pour les expérimentations présentées dans ce chapitre, la méthode permettant de transformer les séries temporelles univariées pour la résolution des problèmes de prévision de type *sequence-to-one* a été utilisée.

Pour simplifier la phase expérimentale, quelques traitements ont été faits. Tout d’abord, toutes les images ont été découpées à des tailles plus petites et transformées en niveaux de gris avant l’étape de préparation. Chaque image de la série est manipulée comme un tenseur de la forme (W, H, N) , où W , H et N désignent respectivement le nombre de lignes, de colonnes et de canaux pour chaque image. L’ordre chronologique des images dans le dataset a été conservé

2. Coordonnées prises dans le système de coordonnées de référence (SCR) EPSG : 32633.

pour permettre aux algorithmes de mieux apprendre les changements qui surviennent au cours du temps. 80% des données ont été sélectionnées pour l'entraînement du modèle (training set), et les 20% restants étant destinés aux tests (test set).

Pour la phase d'apprentissage du modèle DL, des exemples de données en entrée et de la sortie attendue doivent lui être fournis. Le modèle attend donc des données sous forme d'une collection d'échantillons, où chaque échantillon a une composante d'entrée (X) et une composante de sortie (Y). Cette transformation permet de savoir ce que le modèle va apprendre et comment le modèle peut être utilisé pour faire des prédictions.

Pour les données bidimensionnelles utilisant des architectures DL tels que les CNN ou les RNN (LSTM), des transformations supplémentaires sont nécessaires pour préparer les données avant l'entraînement des modèles. Ainsi, les données sont transformées sous la forme de (`samples`, `timestep`, `W`, `H`, `features`), où `samples` représente les séquences en entrée, `timestep` est le nombre d'éléments dans chaque séquence, `W` et `H` le nombre de lignes et de colonnes de l'image respectivement, et `features` correspond au nombre de variables à prédire. Dans le cas des prédictions *sequence-to-one*, `features=1`.

En résumé, pour préparer les séries temporelles avant l'ajustement des modèles DL, les actions suivantes sont nécessaires :

- redimensionner les données ;
- séparer les données en ensemble d'apprentissage et ensemble de test (80% et 20%) ;
- organiser la séquence univariée sous forme de séquences d'entrée et de sortie (`X_train` et `Y_train`) ;
- organiser les données sous la forme de (`samples`, `timestep`, `W`, `H`, `features`).

5.2.1.3 Augmentation des données

Le phénomène de surapprentissage ou d'overfitting survient généralement dans les modèles de classification d'images, lorsque le modèle prend en compte des paramètres très spécifiques des données d'entraînement et ne parvient pas à faire de bonnes prédictions sur de nouvelles données. Typiquement, si l'erreur sur les données d'entraînement est beaucoup plus faible que celle sur les données de test, il y a généralement surapprentissage tel que montré sur la figure 5.4. Parmi les causes possibles d'un surapprentissage, on peut citer le fait qu'un modèle soit trop complexe, le nombre élevé d'hyperparamètres, ou une insuffisance du jeu de données d'entraînement. Pour cette dernière cause, une solution proposée pour éviter le surapprentissage est l'augmentation de données.

L'augmentation de données est une approche qui consiste à créer des données artificielles qui sont intégrées dans le jeu de données initial. Les techniques généralement utilisées pour les images sont le redimensionnement, la rotation, le zoom, l'inversion gauche-droite, etc. Dans le cadre de classification des séries temporelles, les techniques d'augmentation des données sont classées en deux grands groupes [62] à savoir les techniques de base et les techniques avancées. La première catégorie utilise des techniques basées sur le domaine du temps et de la fréquence, tandis que les techniques avancées sont basées sur la décomposition des séries temporelles ou

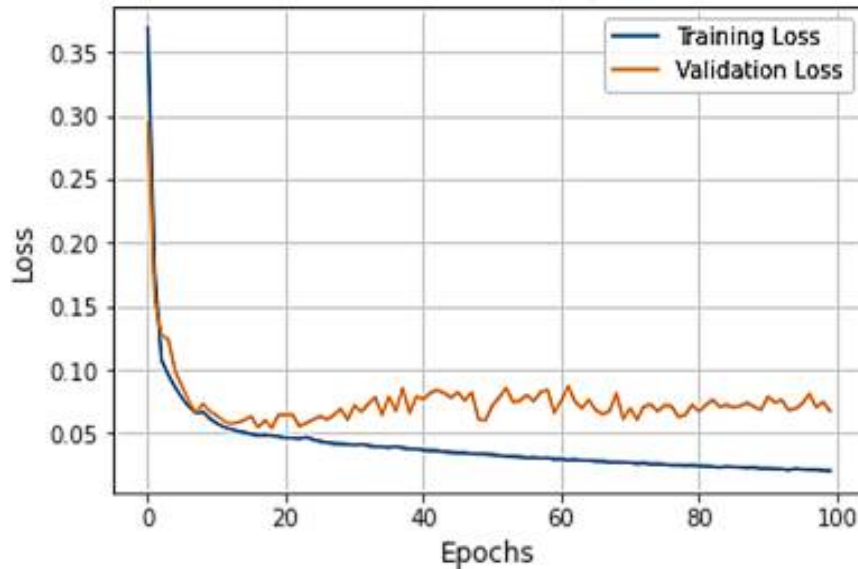


FIGURE 5.4 – Évolution de la fonction de coût avec le jeu de données d'entraînement et de validation dans un modèle qui fait du surapprentissage.

sur des méthodes d'apprentissage automatique. En ce qui concerne l'augmentation des données pour le cas particulier de la prédiction des séries temporelles à court terme, les auteurs dans [145] indiquent qu'aucun travail d'augmentation des données n'avait encore été fait. Ils ont alors proposé une technique inspirée de certains travaux étudiant l'augmentation des données pour la classification des séries temporelles. La méthode proposée dans [145] est une technique hybride basée sur la déformation temporelle et le *jittering*. La partie déformation temporelle a permis d'augmenter le nombre d'éléments dans la série, tandis que la partie de *jittering* a permis d'éviter la linéarité dans les données ajoutées.

Dans le cadre de nos travaux, une méthode inspirée des travaux décrits dans le paragraphe précédent a été proposée. Elle consiste à introduire entre chaque paire d'images de la série temporelle du training set, des images synthétiques qui seront rajoutées dans l'ordre chronologique dans la série finale, tel que présenté sur la figure 5.5.

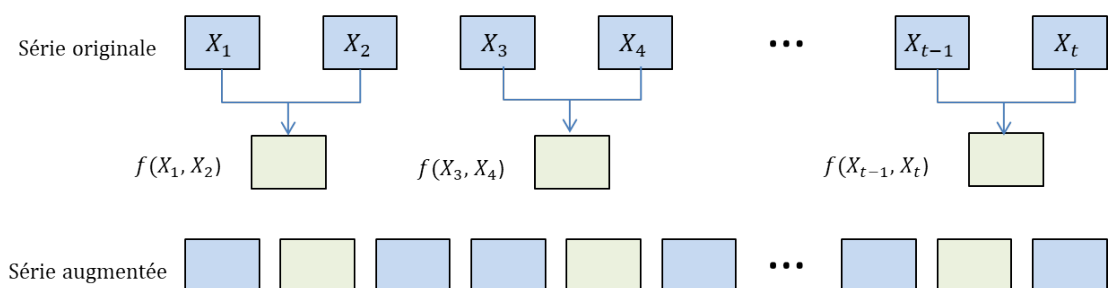


FIGURE 5.5 – Augmentation des STIS par fusion d'images.

Deux méthodes de génération de données ont été testées. La première consiste à effectuer une opération mathématique entre les pixels de chaque paire d'images. La deuxième méthode d'augmentation consiste à considérer chaque paire d'images de la série et à ajouter un bruit gaussien sur la première image de la paire. Cette image générée est alors considérée comme l'image intermédiaire entre la paire d'images considérée.

En considérant X_i et X_{i+1} , deux images consécutives de la série temporelle d'entraînement, Y l'image synthétique générée à insérer entre les images X_i et X_{i+1} , les deux techniques d'augmentation testées pour l'implémentation des modèles de prédiction des STIS sont définies par les équations (5.1) et (5.2).

$$Y = f(X_i, X_{i+1}) = \text{Min}(X_i, X_{i+1}) + \text{Random}(0, (\text{Max}(X_i, X_{i+1}) - \text{Min}(X_i, X_{i+1}))) \quad (5.1)$$

$$Y = f(X_i, X_{i+1}) = X_i + \text{bruit} \quad (5.2)$$

Ainsi, si N est la taille initiale de la série, la série résultante est augmentée de $N/2$ éléments ; la taille de la série passe alors de N à $N + N/2$ éléments.

5.2.2 Modèle

Le modèle pour prédire la déforestation autour de la réserve du Dja a été construit à partir du CNN-LSTM. Pour prévenir le phénomène d'overfitting, le jeu de données d'entraînement a été augmenté avec des données synthétiques et les méthodes de régularisation de modèle DL ont été appliquées.

5.2.2.1 Architecture des couches

La figure 5.6 présente la configuration générale de l'architecture CNN-LSTM utilisée pour la prédiction de la déforestaion. Elle utilise une STIS comme jeu de données d'apprentissage et le modèle est entraîné pour retourner en sortie une image correspondant à la prochaine occurrence de la séquence fournie en entrée. Pour améliorer les performances du modèle présenté dans le chapitre 4, des couches ont été ajoutées dans ce nouveau modèle. Il est ainsi constitué de deux couches convolutionnelles 2D suivies chacune d'une couche de Pooling ainsi que de deux couches LSTM. Deux couches Dense sont également utilisées pour connecter tous les neurones à la couche précédente.

Le tableau 5.1 présente le nombre de paramètres utilisés par le modèle ainsi que la disposition des couches. La taille des séquences en entrée a été fixée à 5 (`timstep=5`). Le modèle est donc entraîné pour recevoir en entrée une séquence de cinq images et prédire la sixième.

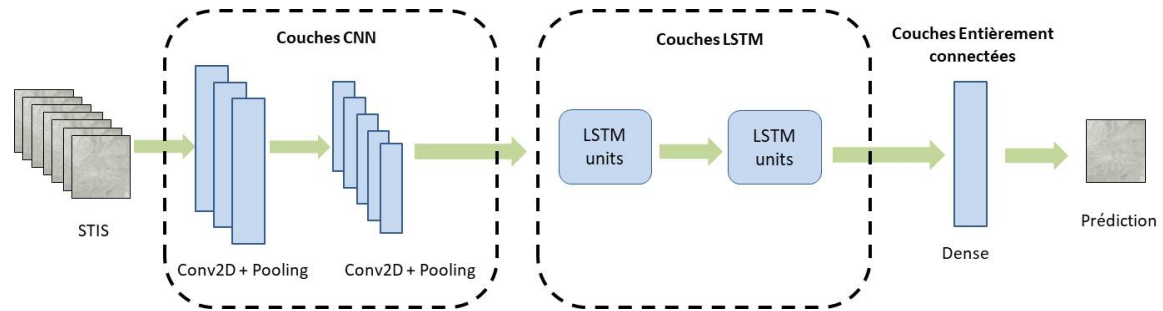


FIGURE 5.6 – Architecture hybride CNN-LSTM proposée.

TABLEAU 5.1 – Description des couches et paramètres du réseau CNN-LSTM.

Couches	Forme de sortie	Paramètres
Conv2D	(None ; 5 ; 510 ; 510 ; 32)	320
Maxpooling	(None ; 5 ; 170 ; 170 ; 32)	0
Conv2D	(None ; 5 ; 168 ; 168 ; 32)	9248
Maxpooling	(None ; 5 ; 56 ; 56 ; 32)	0
Flatten	(None ; 5 ; 100 352)	0
LSTM	(None ; 5 ; 512)	206 571 520
LSTM	(None ; 512)	2 099 200
Dense	(None ; 262 144)	134 479 872
Reshape	(None ; 512 ; 512)	0
Dense	(None ; 512 ; 512)	262 656

Paramètres total = 343 422 816
 Paramètres entraînaables = 343 422 816
 Paramètres non-entraînables = 0

5.2.2.2 Régularisation

Le terme régularisation a plusieurs définitions et est interprété différemment selon le contexte d'utilisation [146]. Par exemple, dans les années 90, l'expression faisait uniquement référence aux notions de pénalité dans la fonction de perte [147]. Dernièrement, le terme a acquis un sens beaucoup plus large. Par exemple, Goodfellow dans [25] le définit comme toute modification apportée à un algorithme d'apprentissage et qui est destinée à réduire son erreur de test mais pas son erreur d'apprentissage. Les auteurs dans [148] généralisent encore plus le terme et le définissent comme l'ensemble de techniques supplémentaires qui visent à faire en sorte que le modèle généralise mieux, c'est-à-dire qu'il produise de meilleurs résultats sur l'ensemble de test ; en d'autres termes, une méthode développée pour lutter contre le surapprentissage.

En considérant cette définition, on pourrait donc également affirmer que l'augmentation de données est une méthode de régularisation des modèles DL. Cependant des techniques telles le dropout ou l'arrêt précoce (*early stopping*) sont les plus mentionnées quand on parle de

régularisation. Dans le cadre de notre étude, nous considérons le dropout comme méthode de régularisation pour le modèle DL.

Le dropout est une technique de régularisation proposée dans [149], où des neurones sélectionnés aléatoirement sont ignorés pendant la formation. Cela signifie que leur contribution à l'activation des neurones en aval est temporairement supprimée lors du passage en avant, et que les mises à jour de poids ne sont pas appliquées au neurone lors de la rétropropagation.

La figure 5.7 présente un exemple de modèle de RNA où le dropout est appliqué. Sur l'image de droite, les neurones barrés représentent les unités supprimées. Pendant la formation, le dropout supprime les nœuds de façon aléatoire à chaque itération. Mais pendant la phase de prédiction sur un ensemble de données de test, le réseau entier doit être utilisé sans dropout.

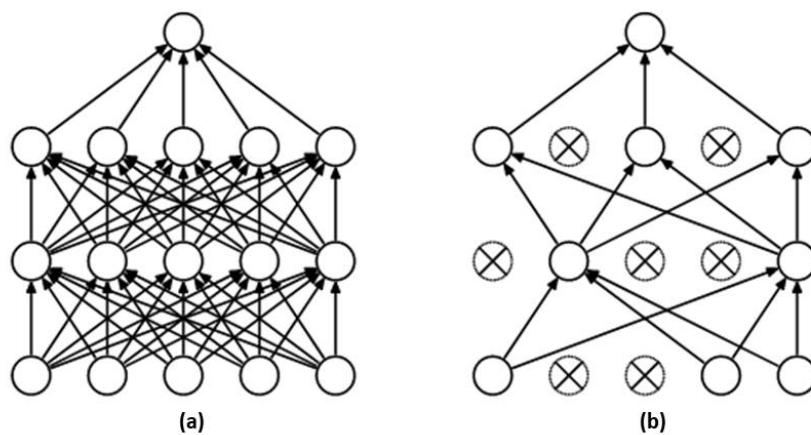


FIGURE 5.7 – Modèle de réseau de neurones dropout. Gauche : un réseau neuronal standard ayant deux couches cachées. Droite : exemple de réseau avec le nombre de connexions réduit après l'application du dropout au réseau de gauche.[149]

5.2.2.3 Optimisation des hyperparamètres

Un hyperparamètre est un paramètre dont la valeur peut être ajustée par le *datascientist* avant la phase d'apprentissage pour contrôler son processus de formation. Les hyperparamètres sont très importants pour les performances, la vitesse et la qualité des modèles DL, il est donc crucial de les optimiser. L'optimisation des hyperparamètres consiste à choisir un ensemble d'hyperparamètres qui permet au modèle de fournir les meilleures performances [150].

Le taux d'apprentissage (*learning rate*) est probablement l'un des plus importants hyperparamètres à choisir lors de la formation de réseaux de neurones. Si la valeur du taux d'apprentissage est trop faible, la formation sera très lente, mais si elle est trop élevée, la formation ne convergera pas. Les optimiseurs plus avancés comme Adam par exemple tentent de compenser ce défaut en variant efficacement le taux d'apprentissage dynamiquement, mais la valeur initiale est toujours importante.

Plusieurs méthodes d'optimisation des hyperparamètres existent. Dans le cadre de cette étude, nous avons utilisé l'approche d'optimisation hyperbande décrite dans [151]. L'algorithme 3 décrit l'approche d'optimisation hyperbande implémentée.

Algorithme 3 : Algorithme HYPERBAND pour l'optimisation des hyperparamètres [151]. R représente la quantité maximale de ressources qui peut être allouée à une seule configuration ; η est une entrée qui contrôle la proportion de configurations rejetées à chaque itération.

Entrées : R, η ($\eta = 3$ par défaut)

```

1 Initialisation :  $S_{max} = \lfloor \log_n(R) \rfloor, B = (s_{max} + 1)R;$ 
2 pour  $s \in \{s_{max}, s_{max}-1, \dots, 0\}$  faire
3    $n = \lceil \frac{B}{R} \times \frac{\eta^s}{S+1} \rceil;$ 
4    $r = R\eta^{-s};$ 
5   // Début des divisions successives
6    $T = \text{get\_hyperparameter\_configuration}(n);$ 
7   pour  $i \in \{0, \dots, s\}$  faire
8      $n_i = \lfloor n\eta^{-i} \rfloor;$ 
9      $r_i = r\eta^i;$ 
10     $L = \{ \text{run\_then\_return\_val\_loss}(t, r_i) : t \in T \};$ 
11     $T = \text{top\_k}(T, L, \lfloor n_i/\eta \rfloor)$ 
12 retourner configuration permettant d'obtenir les plus petites valeurs de perte.
```

5.3 Résultats et discussions

5.3.1 Outils utilisés

Pour réaliser les différentes expérimentations, des outils logiciels libres d'accès ont été utilisés. Ils peuvent être séparés en deux principaux groupes à savoir les outils utilisés pour la partie traitement des images brutes et la partie pour l'implémentation des méthodes DL. Les principaux outils utilisés sont :

SNAP : le logiciel SNAP ou plate-forme d'application Sentinel réunit les boîtes à outils Sentinel. Cette plate-forme est développée pour pouvoir exploiter les données du programme Copernicus. Cet outil a été utilisé pour visualiser, analyser et prétraiter les images Sentinel-1A.

OrfeoToolBox : ORFEO ToolBox (OTB) est une librairie open-source de traitement d'image. Elle est construite à partir de la librairie de traitement d'image ITK et propose des fonctionnalités adaptées à la télédétection et particulièrement aux images de très haute résolution spatiale. Cette librairie a été utilisée pour les classifications et l'analyse des images Sentinel-1.

QGIS : QGIS est un logiciel SIG libre et multiplateforme publié sous licence GPL. Il permet d'exploiter les images sous formes de raster ou de vecteurs. Plusieurs extensions de téléchargement, de traitement et d'analyse des images satellites sont prises en charge par QGIS. Il permet notamment la réalisation des cartes d'occupation des sols. Dans le cadre de nos travaux, QGIS a été le principal outil de base par lequel les traitements et analyses ont été faits.

GDAL : GDAL est une bibliothèque de traduction pour les formats de données géospatiales matricielles et vectorielles. Elle est publiée sous une licence Open Source de type MIT par l'*Open Source Geospatial Foundation*. Elle est fournie avec une variété d'utilitaires utilisables en ligne de commande pour la traduction et le traitement des données. Le passage des images du format TIFF au format JPG (et inversement) a été rendu possible grâce à GDAL.

Rasterio : les systèmes d'information géographiques utilisent GeoTIFF et d'autres formats pour organiser et stocker des ensembles de données matricielles maillées, comme des images satellites et des modèles de terrain. Rasterio lit et écrit ces formats et fournit une API Python basée sur des tableaux Numpy à N dimensions et GeoJSON. Rasterio a été très utile pour faciliter l'exploitation des jeux de données à partir de Python et la conservation des informations géospatiales.

Tensorflow : TensorFlow est une plateforme open source de bout en bout pour l'apprentissage automatique. Elle dispose d'un écosystème complet et flexible d'outils, de bibliothèques et de ressources communautaires qui permet aux chercheurs de faire progresser l'état de l'art de l'apprentissage automatique et aux développeurs de créer et de déployer facilement des applications basées sur l'apprentissage automatique.

Keras : Keras est un framework d'apprentissage profond très utilisé. Il facilite l'exécution de nouvelles expériences et fournit des messages d'erreur clairs et exploitables pour des utilisateurs non expérimentés. Keras dispose également d'une documentation complète et de guides de développement.

KerasTuner : la bibliothèque Keras permet d'implémenter l'optimisation hyperbande à travers le framework KerasTuner [152]. Il s'agit d'un framework évolutif qui résout les problèmes liés à la recherche d'hyperparamètres. Plusieurs algorithmes d'optimisation sont possibles avec KerasTuner tels que l'approche bayésienne ou de recherche aléatoire. Il a été conçu pour être facilement amélioré par les chercheurs afin de proposer de nouveaux algorithmes.

Python 3 : Python est un langage de programmation interprété. Il est à la fois simple et puissant grâce à de nombreuses bibliothèques qui permettent de travailler sur des projets très ambitieux. Les codes utilisés dans le cadre de nos travaux ont été écrits en python.

5.3.2 Paramètres utilisés

Avant de fixer les paramètres à utiliser pour la construction du modèle prédictif, plusieurs combinaisons ont été testées pour déterminer les fonctions les plus adaptées et produisant les meilleurs résultats. L'algorithme Hyperband pour la détermination des meilleures valeurs des hyperparamètres a également été implémenté.

La figure 5.8.a présente l'évolution de quelques fonctions de coût pendant l'apprentissage du modèle avec différentes fonctions d'optimisation. Six combinaisons de fonctions de coûts et de fonctions d'optimisation ont été testées :

- ADAM+RMSE ;
- ADAM+ MAE ;
- SGD+RMSE ;

- SGD+MAE;
- RMSProp+MAE;
- RMSProp+RMSE.

Nous constatons que la fonction Adam permet d'atteindre les valeurs les plus basses quelle que soit la fonction de coût. Comme c'est le cas avec d'autres types de données, nous remarquons sur la figure 5.8.a que la fonction SGD ne permet pas au modèle de converger aussi rapidement que les deux autres fonctions de coût au bout de 100 epochs.

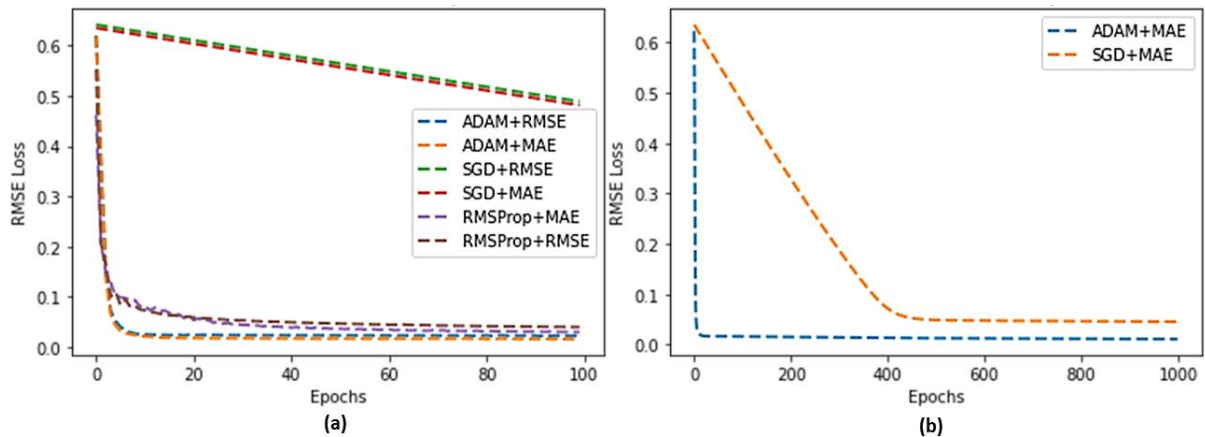


FIGURE 5.8 – Comparaison des valeurs pertes entre : (a) six combinaisons de fonctions de coût+fonction d'optimisation; (b) combinaison ADAM+MAE et SGD+MAE.

Cependant, il peut arriver que la fonction SGD permette d'atteindre des faibles valeurs de perte et donc présenter une bonne capacité de généralisation. Pour vérifier cette capacité des fonctions SGD, le nombre d'epochs pour l'entraînement du modèle a été augmenté à 1000 pour comparer les valeurs obtenues avec les fonctions SGD et Adam. La figure 5.8.b permet de déduire que la fonction Adam est finalement celle qui produit les valeurs les plus basses. La fonction Adam est donc sélectionnée comme fonction d'optimisation pour le reste des expérimentations.

En ce qui concerne les valeurs des hyperparamètres, la fonction Hyperband a été utilisée. Les éléments pris en compte pour faire les tests de sélection sont :

- le nombre de filtres pour les couches de convolution : nombre qui varie entre 16 et 128;
- les fonctions d'activation : choix entre les fonctions Relu, sigmoïde et tanh;
- la valeur du dropout : entre 0,1 et 0,6;
- la valeur du *learning rate* (taux d'apprentissage) pour la fonction d'optimisation Adam prend les valeurs parmi 0,01 , 0,001 et 0,0001.

Après l'exécution de l'algorithme d'optimisation des hyperparamètres, les meilleures valeurs ont été obtenues. Les valeurs présentées dans le tableau 5.2 ont permis d'obtenir un modèle avec la meilleure précision.

TABLEAU 5.2 – Valeurs des hyperparamètres.

Hyper paramètres	Filtres	Noyau	Activation	Dropout	Taux apprentissage
Valeurs	32	3×3	Relu	0,3	0,001

5.3.3 Prédiction one-step et validation du modèle

Avant de passer à la prédiction multi-step, le modèle est entraîné et testé sur des données de test pour la validation à partir des données existantes. La figure 5.9 montre l'évolution des valeurs de la fonction de coût obtenues après 100 epochs sur les données d'entraînement. Avec le jeu de données sans augmentation, on constate que les valeurs calculées par le modèle sont moins bonnes que lorsque les techniques d'augmentation sont appliquées. La première méthode d'augmentation définie par l'équation (5.1), qui présente les valeurs les plus basses, est adoptée pour la suite des expérimentations.

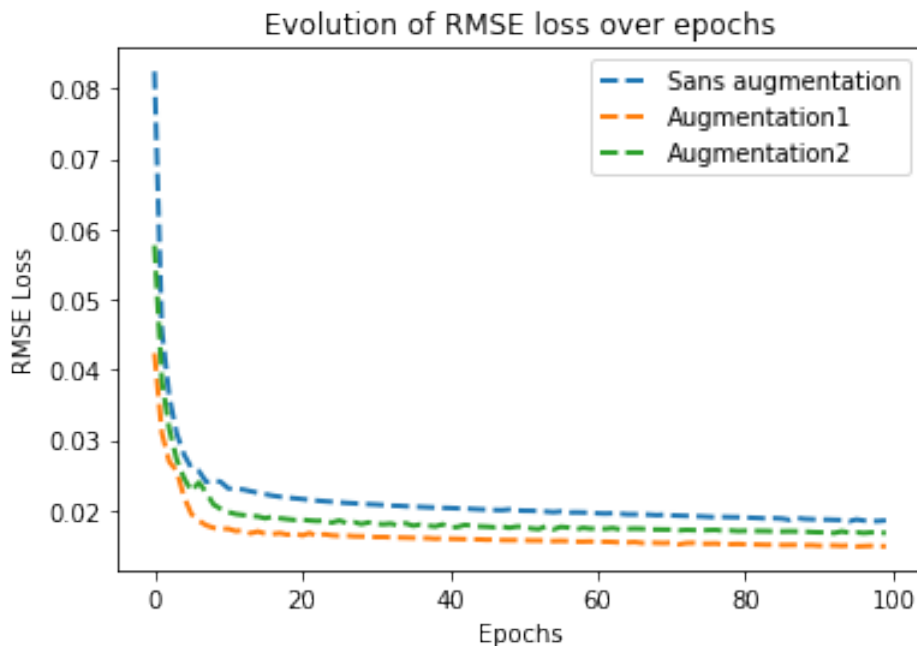


FIGURE 5.9 – Évolution de l'erreur pendant l'apprentissage du modèle, avec et sans augmentation de données.

Pour la validation des modèles, deux types de métriques ont été utilisées. Les métriques liées au modèle telles que l'évolution des valeurs du RMSE d'entraînement et de validation ainsi que la précision. Ensuite, les métriques utilisées en vision par ordinateur pour comparer des images sont également utilisées pour évaluer les capacités de prédiction du modèle. Il s'agit du PSNR, MSE, SSIM, coefficient de corrélation r , de la précision et du RMSE d'entraînement. Ces métriques sont décrites dans le chapitre 2 et le chapitre 4.

La figure 5.10 montre l'évolution de la fonction de coût RMSE pendant l'apprentissage (Training loss et validation loss). Les deux courbes convergent de manière similaire, ce qui nous permet de supposer qu'il n'y a pas de problème d'overfitting, et que le modèle a une bonne capacité de généralisation.

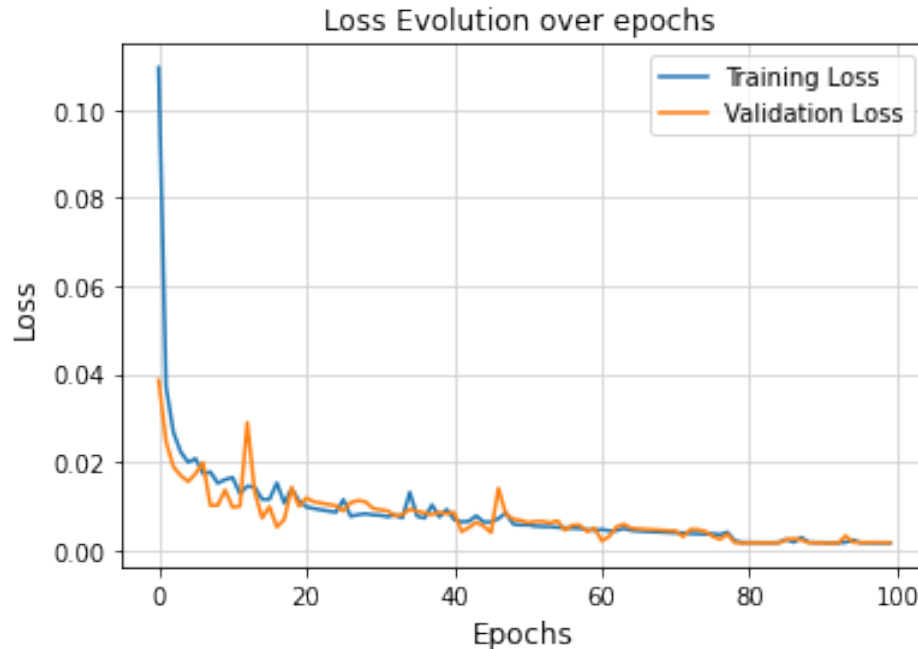


FIGURE 5.10 – Évolution de la fonction de coût RMSE (apprentissage et validation) pendant l'entraînement du modèle.

Le tableau 5.3 présente l'ensemble des valeurs des métriques calculées pour valider le modèle. Les valeurs sont calculées à partir de l'image prédite par le modèle et l'image réelle attendue. Une valeur de perte minimum de 0,01 est obtenue à la fin du processus d'apprentissage.

TABLEAU 5.3 – Valeurs des métriques d'évaluation. ACC=Précision.

Métriques	MSE	PSNR	SSIM	r	ACC	RMSE
Valeurs	0,64	30,02	0,67	0,42	97,2%	0,01

La figure 5.11 donne un aperçu de la nature du sol dans la zone d'étude au fil des années. Nous constatons la présence d'une zone sombre sur la figure 5.11.a qui correspond à une zone de forêt rasée pour la culture de l'hévéa. Au fil des années, on constate que cette surface s'étend à d'autres zones (figure 5.11.b) et puis a tendance à disparaître sur les figures suivantes (figure 5.11.c à figure 5.11.f). Cette situation peut s'expliquer par la présence des arbres d'hévéa. Il est donc difficile de discriminer visuellement les zones de forêts des cultures d'hévéa sans traitements supplémentaires. On constate également à partir de la figure 5.11.c des changements au niveau de la rivière Dja qui s'élargit. Ce changement pourrait être causé par l'aménagement du barrage

hydroélectrique de Mékin [153]. Les histogrammes des images de 2016 à 2019 (figures 5.11.a à 5.11.b) ont été égalisés pour mieux voir les changements au fil des années.

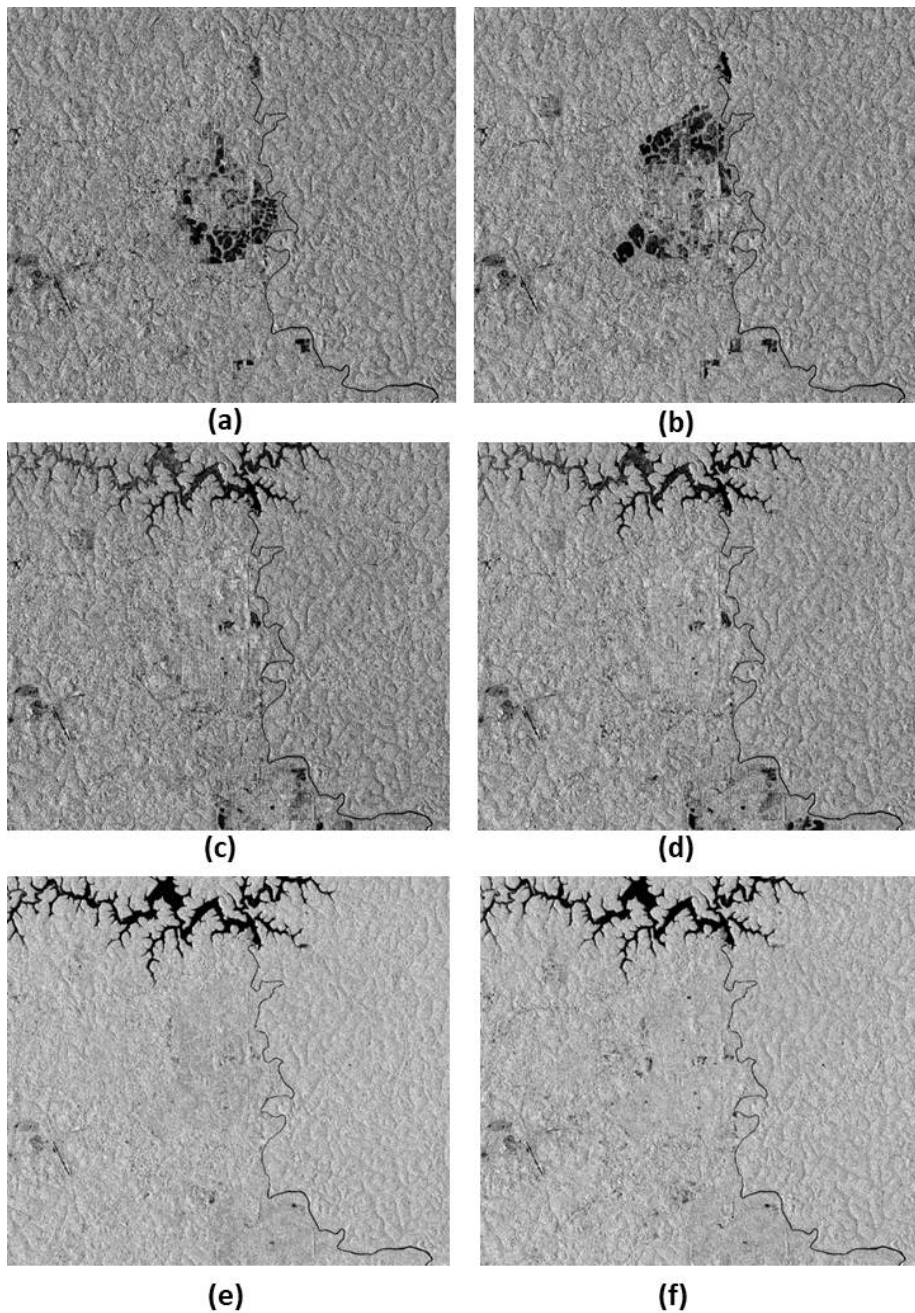


FIGURE 5.11 – Images des classes d’occupation des sols dans la zone d’étude en : (a) 2016; (b) 2017; (c) 2018; (d) 2019; (e) 2020; (f) 2021.

Pour faire la prédiction one-step (pour l’année 2022), les cinq dernières images de 2021 sont utilisées en entrée du modèle. La figure 5.12 présente la séquence d’entrée et le résultat de la prédiction est représenté sur la figure 5.13. On observe que l’aspect général de l’image prédite est en accord avec celle de l’image réelle, malgré des zones sombres qui apparaissent sur l’image

prédite (figure 5.13.b). Sur l'image représentant la différence entre l'image prédite et l'image réelle (figure 5.13.c), les zones sombres indiquent que les valeurs des pixels sont en concordance sur les deux images comparées, tandis que les zones où il y a du blanc indiquent une forte différence.

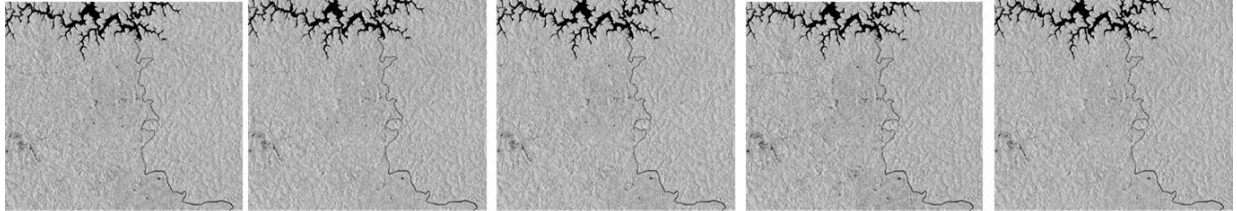


FIGURE 5.12 – Séquence d'images utilisée comme entrée pour la prédiction de l'image de 2022.

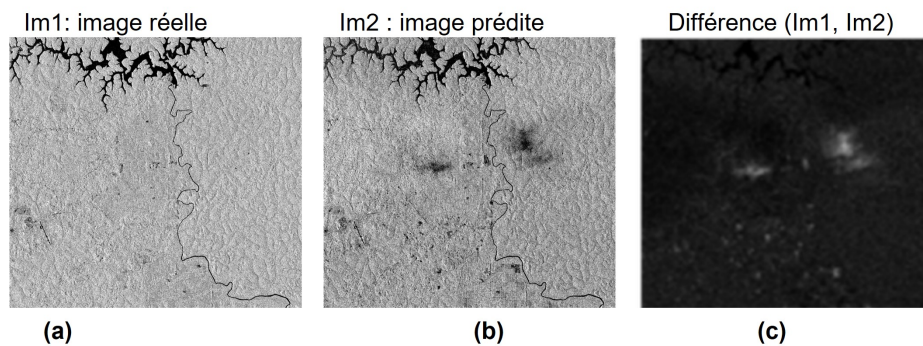


FIGURE 5.13 – Résultat de la prédiction du modèle. (a) image de janvier 2022 attendue; (b) image prédite par le modèle; (c) différence entre image réelle et image prédite.

Après l'étape de prédiction, les cartes d'occupation des sols sont réalisées pour chacune des images pour calculer les surfaces des zones forêts, non forêts et surface d'eau. L'algorithme de classification utilisé est celui du Random forest, intégré à l'outil Orfeo Toolbox. Le tableau 5.4 présente la répartition des surfaces en hectares pour les trois classes d'occupation des sols. La différence entre les valeurs réelles et prédites permet de calculer la précision de prédiction qui est évaluée à environ 92%. Cette valeur est supérieure aux valeurs généralement considérées pour la validation des prédictions. Ce qui permet l'utilisation du modèle pour la prédiction multi-step.

TABLEAU 5.4 – Surface des classes d'occupation des sols en 2022.

Classe	Surfaces en hectares		
	Image originale	Image prédite	% Précision
Forêt	71 320	64 257	90,1
Non Forêt	24 860	22 958	92,35
Eau	1 405	1302	92,67

Précision globale de prédiction (PA) = 91,7%.

5.3.4 Prédiction multi-step pour l'année 2025

Lorsque le modèle est testé et validé par comparaison entre les données prédites et les données existantes, une prédiction multi-étape (multi-step) peut alors être envisagée pour prédire les prochaines images. Si des erreurs sont commises dans les données du passées, celles-ci se répercuteront dans les valeurs futures et créeront un problème d'accumulation d'erreurs.

Étant donnée $X_t = [X_1, X_2, \dots, X_t]$, une séquence d'image de taille t fournie en entrée au modèle, le type de prévision généralement rencontré consiste à prédire la donnée X_{t+1} . Dans le cas des prévisions multi-étapes, il est question de prédire des valeurs futures plus éloignées dans le temps, mais en parcourant la série pas à pas, avec les avaleurs prédites au fur et à mesure.

La figure 5.14 illustre le principe de fonctionnement de la prédiction multi-étapes appliquée.

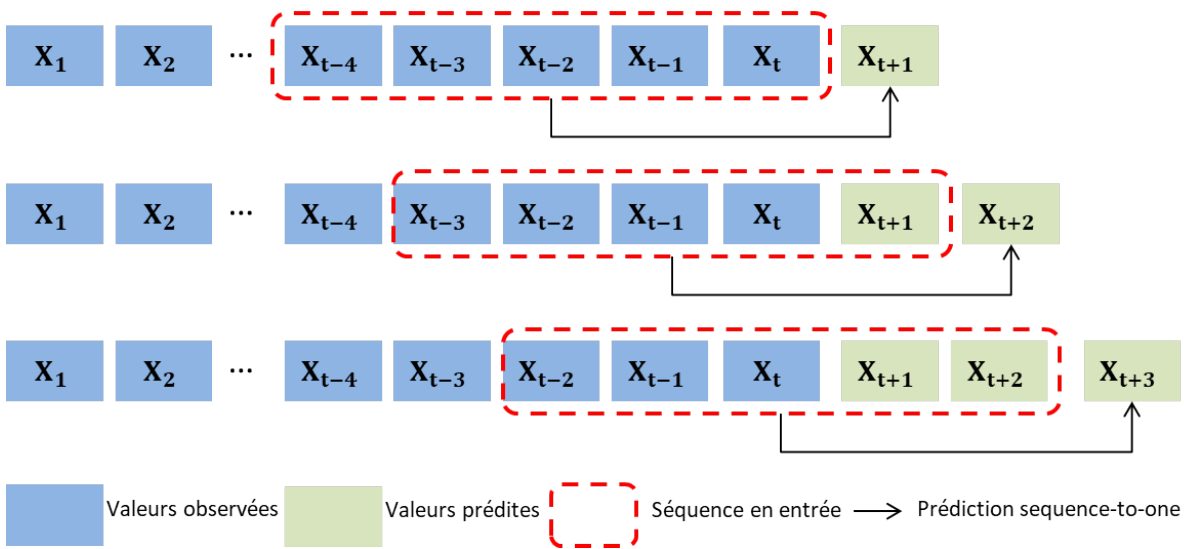


FIGURE 5.14 – Principe de la prédiction multi-étapes.

Il existe plusieurs stratégies de prévision multi-étapes [154]. Les plus utilisées dans la littérature sont :

- la stratégie de prévision directe à plusieurs étapes qui consiste à développer un modèle distinct pour chaque étape de la prédiction ;
- la stratégie de prévision multi-étapes récursive qui implique l'utilisation multiple d'un modèle à une étape où la prédiction pour l'étape temporelle précédente est utilisée comme une entrée pour faire une prévision sur l'étape temporelle suivante ;
- la stratégie de prévision hybride directe-récursive à plusieurs étapes qui combine les stratégies directes et récursives afin d'exploiter les avantages des deux méthodes ;
- la stratégie de prévision à sorties multiples qui implique le développement d'un modèle capable de prédire la séquence entière de prévisions en une seule fois.

Pour prédire la déforestation autour de la RBD en 2025, nous avons choisi d'utiliser le principe de prédiction multi-étapes récursive. Il permet de prédire les valeurs futures progressivement à

partir d'un seul modèle. Sachant que le pas de temps entre deux images de la série est de 12 jours, il faut environ 90 pas pour prédire la déforestation en 2025 (si l'on part de l'année 2022).

La figure 5.15 présente les résultats de la prédiction multi-step pour les années 2023, 2024 et 2025. En termes de changement de surfaces du couvert forestier (Forêt – Non Forêt), le modèle prévoit une hausse approximative de 6% comparativement à 2022, passant d'environ 7 400 ha à 7 900 ha de surface déforestée.

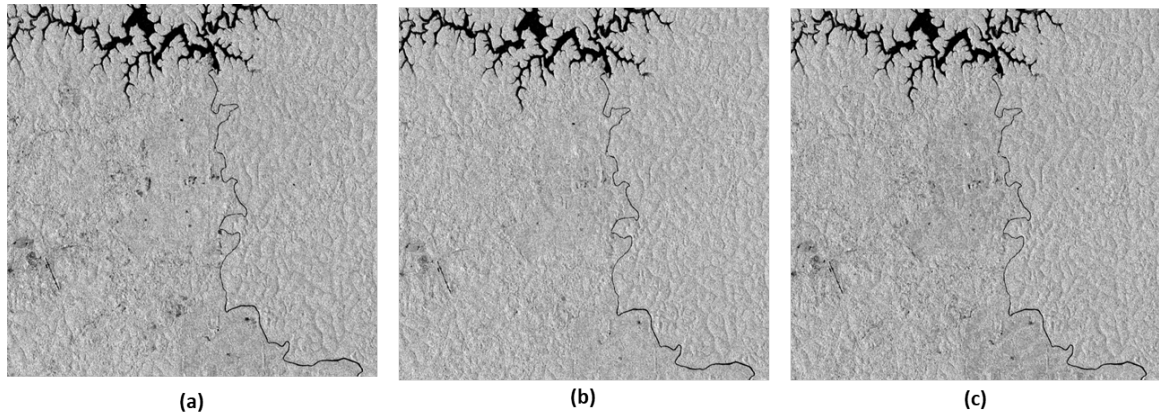


FIGURE 5.15 – Prédiction multi-step des images de : (a) janvier 2023 ; (b) janvier 2024 ; (c) janvier 2025.

Bien que le modèle présenté soit apte à prédire les changements des classes d'occupation des sols, celui-ci peut être amélioré en réalisant les actions suivantes :

- utiliser des modèles multivariés : disposer d'autres types de données pourrait permettre de croiser les données de différentes sources et améliorer les prédictions. Les valeurs numériques des surfaces déforestées au fil du temps pourraient par exemple être rajoutées comme variables du modèle.
- optimiser le modèle : proposer de nouvelles architectures ou des fonctions d'optimisation et de perte adaptées au problème traité pourrait significativement améliorer les résultats.

5.4 Conclusion

Un modèle d'apprentissage hybride a été proposé pour la prédiction de la déforestation autour de la RBD. Une zone d'étude d'environ 98 000 ha, où est implantée une entreprise pratiquant la culture d'hévéa a été choisie. Un réseau CNN combiné à un réseau LSTM a été utilisé avec une série temporelle d'images Sentinel-1A de la zone d'étude pour l'entraînement du modèle. Trois types de fonction d'optimisation ont été testés, puis la fonction Adam a été sélectionnée pour l'apprentissage du modèle. En ce qui concerne les valeurs des hyperparamètres, une recherche par l'approche hyperbande a été implémentée pour déterminer les valeurs permettant au modèle d'avoir la meilleure précision.

Pour éviter le phénomène surapprentissage, un processus d'augmentation de données a été

proposé et les résultats ont montré que le modèle produisait de meilleurs résultats avec l'augmentation des données.

Avant de prédire le changement du couvert forestier en 2025, le résultat de la prédiction pour l'année 2022 a d'abord été comparé à l'image réelle de 2022. Une précision globale de prédiction de 91,7% a été obtenue. La valeur de la précision de prédiction étant satisfaisante, le même modèle a été utilisé dans un processus de prédiction multi-step pour générer les images futures.

Au terme du processus de prédiction, les images attendues en 2023, 2024 et 2025 ont été produites. Le modèle a permis de prédire une surface déforestée d'environ 7 900 ha, contre environ 7 400 en 2022; soit une augmentation de 6%. Ces résultats pourraient être réévaluées au bout de trois ans avec les futures images; ce qui permettrait une mise à jour et amélioration du modèle. Le modèle s'est avéré utile pour l'atteinte des objectifs fixés. Cependant, la fusion des données optiques-radar, la conception des modèles DL pour la classification, l'introduction de nouvelles architectures et/ou variables dans le modèle, l'application du *transfer learning* ou du *few shot learning* sont quelques pistes de réflexions possibles pour les futurs travaux afin d'augmenter la précision des prédictions.

Conclusion et perspectives

Grâce à la disponibilité des données de télédétection ainsi qu’au développement des méthodes d’apprentissage profond de plus en plus performantes, il est aujourd’hui possible de créer des modèles capables de prédire les phénomènes de changement des classes d’occupation de sol, tels que la déforestation. Cependant, exploiter le potentiel des algorithmes DL sur les STIS n’est pas une tâche triviale. En effet, plusieurs contraintes telles que la taille, l’hétérogénéité et la complexité des images satellites (informations géographiques, méta-data), les séries d’images incomplètes, ou encore la nature spatio-temporelle des prédictions, rendent les tâches de conception et d’exécution des modèles prédictifs assez difficiles. Dans le cadre de cette thèse, l’objectif général était d’étudier les approches de prédiction de l’évolution de l’occupation des sols ainsi que les outils et modèles d’apprentissage profond utilisant des images satellites à haute résolution, dans le but de prédire l’évolution du couvert forestier. Plus précisément, il était premièrement question de résoudre le problème de manque de données d’entraînement. Deuxièmement, nous avons étudié des architectures DL afin d’identifier celle(s) qui s’adapte(nt) le mieux à la prédiction des STIS. Si les approches DL sont populaires aujourd’hui, il est intéressant d’évaluer leur efficacité vis-à-vis des méthodes classiques de la littérature. À cette fin, une étude comparative entre deux architectures DL et la méthode CA-Markov est réalisée. Pour finir, un modèle DL a été proposé et amélioré pour prédire la déforestation dans une zone forestière.

Pour atteindre les objectifs visés, quatre principales contributions correspondant aux objectifs spécifiques cités ont été proposées. Premièrement, en ce qui concerne le manque de données d’entraînement, un workflow de collecte et de prétraitement par lot des images Sentinel-1A a été proposé. Il a permis la mise en œuvre de scripts s’appuyant sur les outils ASF, GPF et SNAP. Les codes sources des programmes mis en place ainsi qu’une base d’images constituée des trois STIS sur les zones du PNB, de la RBD et de la RFT ont été rendus publics et mis à la disposition de la communauté scientifique.

Deuxièmement, dans le but de déterminer l’architecture DL la mieux adaptée au problème de prédiction dans les STIS, trois modèles basés sur les architectures Stack-LSTM, CNN-LSTM et ConvLSTM ont été entraînés sur des images Sentinel-1A de la RFT et du PNB. L’objectif de l’apprentissage était la prédiction de la prochaine occurrence dans une séquence de cinq images fournies en entrée. Plusieurs configurations ont été testées et plusieurs mesures d’évaluation utilisées pour déterminer l’architecture la plus adaptée au problème traité. Au terme de l’étude, il a été montré que même si tous les trois modèles pouvaient être utilisés pour la prédiction des

STIS, les architectures CNN-LSTM et ConvLSTM sont les plus appropriées. Cependant, lorsqu'il s'agit de travailler avec des données de grandes tailles et de longues séquences en entrée, des limites ont été notées pour le modèle ConvLSTM.

Ensuite, pour valider les méthodes DL pour la prédiction des changements de classes d'occupation de sol, comparativement à la méthode utilisant les automates cellulaires intégrés aux chaînes de Markov (CA-Markov), une étude a été menée en utilisant les images Sentinel-1A du PNB. De façon générale, les résultats de ces expérimentations ont montré que les modèles DL pouvaient atteindre de meilleurs résultats pour la prédiction des classes d'occupation des sols qu'un modèle CA-Markov. Il a également été noté que les résultats des prédictions avec les méthodes CA-Markov dépendaient fortement des données auxiliaires utilisées ainsi que des précisions obtenues pour la production des cartes d'occupation des sols utilisées comme données d'entraînement du modèle.

Pour finir, un modèle basé sur une architecture CNN-LSTM a été proposé pour prédire la déforestation aux périphéries de la RBD en 2025. Pour déterminer les hyperparamètres permettant d'obtenir les meilleurs résultats, une recherche par l'approche Hyperband a été implémentée. Afin d'améliorer la qualité des prédictions et pour éviter le phénomène surapprentissage, un processus d'augmentation de données a été proposé et les résultats ont montré que le modèle produisait de meilleurs résultats avec l'augmentation des données. Avant de prédire le changement du couvert forestier en 2025, le résultat de la prédiction pour l'année 2022 a d'abord été comparé à l'image réelle de 2022 existante. Une précision de prédiction satisfaisante de 91,7% a été obtenue et le modèle a été utilisé dans un processus de prédiction multi-step pour générer les images futures. Au terme du processus de prédiction, les images attendues en 2023, 2024 et 2025 ont été produites. Le modèle a permis de prédire une surface déforestée d'environ 7 900 *ha* en 2025, contre environ 7 400 en 2022 ; soit une augmentation d'environ 6%.

Bien que les objectifs de cette thèse aient été atteints et que les solutions proposées aient produit de bons résultats, des extensions et améliorations de nos travaux sont envisagées.

Il a été montré au chapitre 4 que les architectures ConvLSTM sont capables d'atteindre de très bonnes précisions de prédiction. Cependant, elles nécessitent des temps de calcul élevés et des ressources importantes à cause des opérations de convolution. Pour des problèmes à grande échelle utilisant des données telles que les images satellites, il est ainsi difficile d'utiliser les réseaux ConvLSTM. De plus, le modèle de prédiction de déforestation autour de la RBD proposé dans le chapitre 5 ne peut être utilisé que sur des données de la RBD. Pour utiliser le modèle sur une autre zone, celui-ci doit être ré-entraîné totalement sur les images de la nouvelle zone.

Pour améliorer les capacités de généralisation de notre modèle et pour nous permettre d'exploiter les avantages qu'offrent les architectures ConvLSTM, nous prévoyons appliquer les techniques telles que le *few-shot learning* (FSL), le *meta-learning* ou le *transfer-learning* pour la prédiction des STIS. En effet, le FSL est une technique en apprentissage automatique permettant d'obtenir des résultats satisfaisants à partir d'un ensemble de données limité. Le FSL est surtout utilisé pour des problèmes de vision par ordinateur et peut être considéré comme un

problème de *meta-learning* où le modèle apprend comment apprendre à résoudre un problème donné. En ce qui concerne le *transfer-learning*, il s'agit d'un ensemble de techniques permettant de transférer les connaissances acquises par un modèle pour traiter un autre problème similaire. Cette méthode de transfert d'apprentissage permettra de réutiliser le modèle de prédiction de déforestation sur d'autres zones, sans avoir à ré-entraîner totalement le modèle.

Ensuite, il serait intéressant d'intégrer d'autres architectures DL qui ne s'appuient pas sur LSTM. Nous envisageons pour la suite d'introduire de nouvelles méthodes basées sur la combinaison de plusieurs approches, permettant d'atteindre de meilleures précisions. En effet, des architectures d'apprentissage non supervisé telles que les réseaux GAN qui permettent de générer des images, ou les réseaux AE qui réduisent la taille des données, combinés aux réseaux LSTM ou GRU pourraient être une bonne solution pour la prédiction des STIS.

Pour finir, nous projetons fusionner les images radar et optiques pour une meilleure discrimination des surfaces végétales. En effet, les images radar sont bien adaptées pour le suivi des surfaces forestières. Cependant, une confusion entre les classes d'occupation des sols « forêt » et « non-forêt » est souvent possible et altère la qualité des classifications. Une fusion des images optiques et radar permettra de mieux différencier les zones de forêts des autres types de sols et d'améliorer la robustesse du modèle de prédiction.

Liste des publications

Revue internationale avec comité de lecture

Waytehad Rose Moskoläi, Wahabou Abdou, Albert Dipanda, Kolyang. "*A Workflow for Collecting and Preprocessing Sentinel-1 Images for Time Series Prediction Suitable for Deep Learning Algorithms*". Geomatics 2022 - MDPI.

DOI: <https://doi.org/10.3390/geomatics2040024>

Waytehad Rose Moskoläi, Wahabou Abdou, Albert Dipanda, Kolyang. "*Application of Deep Learning Architectures for Satellite Image Time Series Prediction : A Review*". Remote Sensing 2021 - MDPI.

DOI: <https://doi.org/10.3390/rs13234822>

Conférences internationales avec comité de lecture

Waytehad Rose Moskoläi, Wahabou Abdou, Albert Dipanda, Kolyang. "*A Comparative Study of Deep Learning and CA-Markov Methods for Land Use / Land Cover Change Prediction*". International Conference on Signal Image Technology & Internet based Systems, Dijon (France), 2022.

Waytehad Rose Moskoläi, Wahabou Abdou, Albert Dipanda, Kolyang. "*Application of LSTM architectures for next frame forecasting in Sentinel-1 images time series*". Colloque Africain sur la Recherche en Informatique et en Mathématiques Appliquées, Thies (Sénégal), 2020.

Articles en préparation pour une revue internationale avec comité de lecture

Waytehad Rose Moskoläi, Wahabou Abdou, Albert Dipanda, Kolyang. "*Using sentinel-1 images and deep learning architectures for deforestation prediction : a case study in the Dja Biosphere Reserve, Cameroon*". ISPRS Journal of Photogrammetry and Remote Sensing.

Bibliographie

- [1] Peter POTAPOV, Matthew C HANSEN, Lars LAESTADIUS, Svetlana TURUBANOVA, Alexey YAROSHENKO, Christoph THIES et al. « The last frontiers of wilderness : Tracking loss of intact forest landscapes from 2000 to 2013 ». *Science advances* 3.1 (2017), e1600821 (cf. p. 1).
- [2] FAO. *The State of the World's Forests 2022. Forest pathways for green recovery and building inclusive, resilient and sustainable economies*. FAO, 2022 (cf. p. 1).
- [3] Joël BOULIER et Laurent SIMON. *Atlas des forêts dans le monde*. Autrement, 2022 (cf. p. 1).
- [4] Eric SIEGEL. *Predictive analytics : The power to predict who will click, buy, lie, or die*. John Wiley & Sons, 2013 (cf. p. 2).
- [5] Eric XU et Orien ZENG. *Predicting Amazon Deforestation with Satellite Images*. 2017 (cf. p. 2).
- [6] David DOMINGUEZ, Luis de Juan del VILLAR, Odette PANTOJA et Mario GONZÁLEZ-RODRIGUEZ. « Forecasting Amazon Rain-Forest Deforestation Using a Hybrid Machine Learning Model ». *Sustainability* 14.2 (2022), p. 691 (cf. p. 2).
- [7] Matthew LINKIE, Robert J SMITH et Nigel LEADER-WILLIAMS. « Mapping and predicting deforestation patterns in the lowlands of Sumatra ». *Biodiversity & Conservation* 13.10 (2004), p. 1809-1818 (cf. p. 2).
- [8] Mark BROICH, Stephen V STEHMAN, Matthew C HANSEN, Peter POTAPOV et Yosio E SHIMABUKURO. « A comparison of sampling designs for estimating deforestation from Landsat imagery : A case study of the Brazilian Legal Amazon ». *Remote Sensing of Environment* 113.11 (2009), p. 2448-2454 (cf. p. 2).
- [9] Clovis GRINAND, Fety RAKOTOMALALA, Valéry GOND, Romuald VAUDRY, Martial BERNOUX et Ghislain VIEILLEDENT. « Estimating deforestation in tropical humid and dry forests in Madagascar from 2000 to 2010 using multi-date Landsat satellite images and the random forests classifier ». *Remote Sensing of Environment* 139 (2013), p. 68-80 (cf. p. 2).
- [10] Vahid AHMADI. « Using GIS and Artificial Neural Network for Deforestation Prediction » (2018) (cf. p. 2).
- [11] Pramit GHOSH, Anirban MUKHOPADHYAY, Abhra CHANDA, Parimal MONDAL, Anirban AKHAND, Sandip MUKHERJEE et al. « Application of Cellular automata and Markov-chain model in geospatial environmental modeling-A review ». *Remote Sensing Applications : Society and Environment* 5 (2017), p. 64-77 (cf. p. 2, 27, 28, 84).
- [12] Semegnew TADESE, Teshome SOROMESSA et Tesefaye BEKELE. « Analysis of the current and future prediction of land use/land cover change using remote sensing and the CA-markov model in Majang forest biosphere reserves of Gambella, southwestern Ethiopia ». *The scientific world journal* 2021 (2021) (cf. p. 2).
- [13] Anne GHARAIBEH, Abdulrazzaq SHAAMALA, Rasha OBEIDAT et Salman AL-KOFAHI. « Improving land-use change modeling by integrating ANN with Cellular Automata-Markov Chain model ». *Heliyon* 6.9 (2020), e05092 (cf. p. 2).
- [14] Devanatham ABIJITH et Subbarayan SARAVANAN. « Assessment of land use and land cover change detection and prediction using remote sensing and CA Markov in the northern coastal districts of Tamil Nadu, India ». *Environmental Science and Pollution Research* (2021), p. 1-13 (cf. p. 2).
- [15] Chen LIPING, Sun YUJUN et Sajjad SAEED. « Monitoring and predicting land use and land cover changes using remote sensing and GIS techniques—A case study of a hilly area, Jiangle, China ». *PloS one* 13.7 (2018), e0200493 (cf. p. 2).
- [16] James George Clifford BALL, Katerina PETROVA, David COOMES et Seth FLAXMAN. « Using deep convolutional neural networks to forecast spatial patterns of Amazonian deforestation ». *bioRxiv* (2021) (cf. p. 2).

- [17] N Thulasi CHITRA, R ANUSHA, Sheetha Hemanth KUMAR, D Siri CHANDANA, Ch HARIKA et V Uday KUMAR. « Satellite Imagery for Deforestation Prediction using Deep Learning ». *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE. 2021, p. 522-525 (cf. p. 2).
- [18] Robert N MASOLELE, Veronique DE SY, Martin HEROLD, Diego MARCOS, Jan VERBESSELT, Fabian GIESEKE et al. « Spatial and temporal deep learning methods for deriving land-use following deforestation : A pan-tropical case study using Landsat time series ». *Remote Sensing of Environment* 264 (2021), p. 112600 (cf. p. 2).
- [19] Jason BROWNLIE. *Deep Learning for Time Series Forecasting*. 2018 (cf. p. 2, 25, 26).
- [20] Laith ALZUBAIDI, Jinglan ZHANG, Amjad J HUMAIDI, Ayad AL-DUJAILI, Ye DUAN, Omran AL-SHAMMA et al. « Review of deep learning : Concepts, CNN architectures, challenges, applications, future directions ». *Journal of big Data* 8.1 (2021), p. 1-74 (cf. p. 2, 17).
- [21] Asifullah KHAN, Anabia SOHAIL, Umme ZAHOORA et Aqsa Saeed QURESHI. « A survey of the recent architectures of deep convolutional neural networks ». *Artificial Intelligence Review* 53.8 (2020), p. 5455-5516 (cf. p. 2).
- [22] Lei MA, Yu LIU, Xueliang ZHANG, Yuanxin YE, Gaofei YIN et Brian Alan JOHNSON. « Deep learning in remote sensing applications : A meta-analysis and review ». *ISPRS journal of photogrammetry and remote sensing* 152 (2019), p. 166-177 (cf. p. 2, 17).
- [23] Li DENG et Dong YU. « Deep learning : methods and applications ». *Foundations and trends in signal processing* 7.3-4 (2014), p. 197-387 (cf. p. 2).
- [24] Maryam M NAJAFABADI, Flavio VILLANUSTRE, Taghi M KHOSHGOFTAAR, Naeem SELIYA, Randall WALD et Edin MUHAREMAGIC. « Deep learning applications and challenges in big data analytics ». *Journal of big data* 2.1 (2015), p. 1-21 (cf. p. 2).
- [25] Ian GOODFELLOW, Yoshua BENGIO et Aaron COURVILLE. *Deep learning*. MIT press, 2016 (cf. p. 2, 16, 18, 20, 22, 72, 99).
- [26] Yann LECUN, Yoshua BENGIO et Geoffrey HINTON. « Deep learning ». *nature* 521.7553 (2015), p. 436-444 (cf. p. 2, 16).
- [27] Josh PATTERSON et Adam GIBSON. *Deep learning : A practitioner's approach*. " O'Reilly Media, Inc.", 2017 (cf. p. 2, 18, 19, 39).
- [28] Waytehad Rose MOSKOLAÏ, Wahabou ABDOU, Albert DIPANDA et KOLYANG. « A Workflow for Collecting and Preprocessing Sentinel-1 Images for Time Series Prediction Suitable for Deep Learning Algorithms ». *Geomatics* 2.4 (2022), p. 435-456. URL : <https://www.mdpi.com/2673-7418/2/4/24> (cf. p. 2).
- [29] P Rhyma PURNAMASAYANGSUKASIH, K NORIZAH, Adnan AM ISMAIL et I SHAMSUDIN. « A review of uses of satellite imagery in monitoring mangrove forests ». *IOP Conference Series : Earth and Environmental Science*. T. 37. 1. IOP Publishing. 2016, p. 012034 (cf. p. 3).
- [30] John Cristian Borges GAMBOA. « Deep learning for time-series analysis ». *arXiv preprint arXiv :1701.01887* (2017) (cf. p. 3).
- [31] Yanbo HUANG, Zhong-xin CHEN, YU TAO, Xiang-zhi HUANG et Xing-fa GU. « Agricultural remote sensing big data : Management and applications ». *Journal of Integrative Agriculture* 17.9 (2018), p. 1915-1931 (cf. p. 3).
- [32] Waytehad Rose MOSKOLAI, Wahabou ABDOU, Albert DIPANDA et al. « Application of Deep Learning Architectures for Satellite Image Time Series Prediction : A Review ». *Remote Sensing* 13.23 (2021), p. 4822 (cf. p. 3, 16, 20, 27, 30, 46, 47, 70, 72).
- [33] Ray HARRIS. « Satellite remote sensing. An introduction » (1987) (cf. p. 8).
- [34] W EMERY et A CAMPS. « Optical Imaging Systems ». *Introduction to Satellite Remote Sensing*. Elsevier (2017), p. 85-130 (cf. p. 8, 10).
- [35] CNES et ONERA. *Imagerie spatiale*. Disponible en ligne : <https://www.recovery-observatory.org/drupal/fr/system/files/public/News/20180511%20-%20Introduction%20Imagerie%20Spatiale.pdf> (visité le 25 mai 2022). 2015 (cf. p. 9, 10).
- [36] Faivre ROBIN. *Introduction à la réalisation de spatio-cartes*. Disponible en ligne : <https://www.recovery-observatory.org/drupal/fr/system/files/public/News/20180511%20-%20Intro%20Realisation%20Spatiocarte.pdf> (visité le 15 mars 2021) (cf. p. 10, 14).
- [37] Laurette PASTORÉ. « Imagerie radar par synthèse d'ouverture en basse fréquence ». Thèse de doct. Paris 10, 2003 (cf. p. 10).

- [38] ML TIGHE, D KING, H BALZTER, A BANNARI et H MCNAIRN. « Airborne X-Hh Incidence Angle Impact on Canopy Height Retrieval : Implications for Spaceborne X-Hh Tandem-X Global Canopy Height Model ». *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 39 (2012), p. 91-96 (cf. p. 11).
- [39] Christophe PROISY. « Apport des données radar à synthèse d'ouverture pour l'étude de la dynamique des écosystèmes forestiers ». Thèse de doct. Toulouse 3, 1999 (cf. p. 11, 13).
- [40] Le Toan THUY, Bouvet ALEXANDRE, Phan THI Hoa, Veloso AMANDA, Planells MILENA et Villard MERMOZ Ludovic. *Sentinel-1 pour l'observation de la végétation*. Disponible en ligne : https://www-loa.univ-lille1.fr/workshops/workshopPNTS2016/Documents/pres2016/2_Presentation_4ieme_JT_PNTS_ABouvet.pdf (visité le 25 mai 2022) (cf. p. 12).
- [41] Anoumou KEMAVO. « Potentialités des données de télédétection optique et radar libres d'accès pour l'évaluation et le suivi des écosystèmes forestiers tropicaux : étude de cas au Togo, en République Démocratique du Congo, en Guyane française et en République Dominicaine ». Thèse de doct. Université Paris-Est, 2018 (cf. p. 12, 53).
- [42] Joseph W GOODMAN. « Some fundamental properties of speckle ». *JOSA* 66.11 (1976), p. 1145-1150 (cf. p. 12).
- [43] Jong-Sen LEE, L JURKEVICH, Piet DEWAELE, Patrick WAMBACQ et A OOSTERLINCK. « Speckle filtering of synthetic aperture radar images : A review ». *Remote sensing reviews* 8.4 (1994), p. 313-340 (cf. p. 12).
- [44] S BOUTARFA, Y SMARA, L BOUCHEMAKH et Z HAMADACHE. « Etude comparative de méthodes de filtrage du speckle dans les images radar SAR polarimétriques ». *CGE* 6 (2009), p. 13-14 (cf. p. 12).
- [45] JS LEE et E POTTIER. « Introduction to the polarimetric target decomposition concept ». *Polarimetric Radar Imaging : From Basics to Applications ; CRC Press : Boca Raton, FL, USA* (2009), p. 1-422 (cf. p. 12).
- [46] Jong-Sen LEE. « A simple speckle smoothing algorithm for synthetic aperture radar images ». *IEEE Transactions on Systems, Man, and Cybernetics* 1 (1983), p. 85-89 (cf. p. 12, 13).
- [47] Jong-Sen LEE, Jen-Hung WEN, Thomas L AINSWORTH, Kun-Shan CHEN et Abel J CHEN. « Improved sigma filter for speckle filtering of SAR imagery ». *IEEE Transactions on Geoscience and Remote Sensing* 47.1 (2008), p. 202-213 (cf. p. 13).
- [48] Narcisse Talla TANKAM. « Une nouvelle approche d'analyse automatique de texture d'images : application à l'étude de la dynamique d'occupation spatiale sur le Mont Cameroun ». Thèse de doct. Atelier national de reproduction des thèses, 2009 (cf. p. 15).
- [49] Guillaume SAINT-CIRGUE. « Apprendre le Machine Learning en une semaine ». *Machine Learning, Londres* (2019) (cf. p. 16).
- [50] LOVERETHYM. *Machine learning and AI difference*. Disponible en ligne : loverethymno.com (visité le 25 janvier 2023) (cf. p. 17).
- [51] Jesus LAGO, Karel DE BRABANDERE, Fjo DE RIDDER et Bart DE SCHUTTER. « Short-term forecasting of solar irradiance without local telemetry : A generalized model using satellite data ». *Solar Energy* 173 (2018), p. 566-577 (cf. p. 19).
- [52] Yann LECUN, Yoshua BENGIO et al. « Convolutional networks for images, speech, and time series ». *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995 (cf. p. 20).
- [53] Sumit SAHA. *A Comprehensive Guide to Convolutional Neural Networks - the ELI5 way*. Disponible en ligne : <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53> (visité le 25 mai 2022) (cf. p. 22).
- [54] Ahmed TEALAB. « Time series forecasting using artificial neural networks methodologies : A systematic review ». *Future Computing and Informatics Journal* 3.2 (2018), p. 334-340 (cf. p. 22).
- [55] Tobias SAUTER, Björn WEITZENKAMP et Christoph SCHNEIDER. « Spatio-temporal prediction of snow cover in the Black Forest mountain range using remote sensing and a recurrent neural network ». *International Journal of Climatology* 30.15 (2010), p. 2330-2341 (cf. p. 22).
- [56] Ahmad ALZAHIRANI, Pourya SHAMSI, Cihan DAGLI et Mehdi FERDOWSI. « Solar irradiance forecasting using deep neural networks ». *Procedia Computer Science* 114 (2017), p. 304-313 (cf. p. 22).
- [57] Arthur STEPCHENKO, Jurij CHIZHOV, Ludmila ALEKSEJEVA et Juri TOLUJEW. « Nonlinear, non-stationary and seasonal time series forecasting using different methods coupled with data preprocessing ». *Procedia Computer Science* 104 (2017), p. 578-585 (cf. p. 22).

- [58] Sepp HOCHREITER et Jürgen SCHMIDHUBER. « Long short-term memory ». *Neural computation* 9.8 (1997), p. 1735-1780 (cf. p. 22, 70).
- [59] Margarita KHOKHLOVA, Cyrille MIGNIOT, Alexey MOROZOV, Olga SUSHKOVA et Albert DIPANDA. « Normal and pathological gait classification LSTM model ». *Artificial intelligence in medicine* 94 (2019), p. 54-66 (cf. p. 22, 70, 73).
- [60] Junyoung CHUNG, Caglar GULCEHRE, KyungHyun CHO et Yoshua BENGIO. « Empirical evaluation of gated recurrent neural networks on sequence modeling ». *arXiv preprint arXiv :1412.3555* (2014) (cf. p. 22).
- [61] Olah CHRISTOPHER. *Understanding LSTM Networks*. Disponible en ligne : <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> (visité le 30 juin 2018). 2015 (cf. p. 22).
- [62] Connor SHORTEN et Taghi M KHOSHGOFTAAR. « A survey on image data augmentation for deep learning ». *Journal of big data* 6.1 (2019), p. 1-48 (cf. p. 23, 96).
- [63] Hadrien BERTRAND. « Hyper-parameter optimization in deep learning and transfer learning : applications to medical imaging ». Thèse de doct. Université Paris-Saclay, 2019 (cf. p. 24).
- [64] Ayoosh KATHURIA. *Intro to optimization in deep learning : Momentum, rmsprop and adam*. 2018 (cf. p. 24, 39, 41).
- [65] Youssef HMAOUCHE. « Prédiction des séries temporelles larges ». Thèse de doct. AMU-Aix Marseille Université, 2018 (cf. p. 24).
- [66] Waytehad MOSKOLAI, Wahabou ABDOU, Albert DIPANDA et Dina Taiwe KOLYANG. « Application of LSTM architectures for next frame forecasting in Sentinel-1 images time series ». *arXiv preprint arXiv :2009.00841* (2020) (cf. p. 26, 34, 47, 73, 76).
- [67] Julien JACQUES. « Introduction aux séries temporelles ». *Polytech Lille* (2013) (cf. p. 26).
- [68] Melinda F THIELBAR. *Neural networks for time series forecasting : Practical implications of theoretical results*. North Carolina State University, 2011 (cf. p. 27).
- [69] Billie L TURNER et Jacqueline GEOGHEGAN. « Land-Cover and Land-Use Change (LCLUC) in the Southern Yucatán Peninsular Region (SYPR) ». *People and the Environment*. Springer, 2004, p. 31-60 (cf. p. 27).
- [70] Shrushti JADAWALA SANJAY JADAWALA, Shital H SHUKLA et Poonam S TIWARI. « Cellular Automata and Markov Chain Based Urban Growth Prediction ». *International Journal of Environment and Geoinformatics* 8.3 (), p. 337-343 (cf. p. 27, 84).
- [71] Thomas HOUET et Laurence HUBERT-MOY. « Modeling and projecting land-use and land-cover changes with Cellular Automaton in considering landscape trajectories ». *EARSeL eProceedings* 5.1 (2006), p. 63-76 (cf. p. 27).
- [72] Britaldo Silveira SOARES-FILHO, Gustavo Coutinho CERQUEIRA et Cássio Lopes PENNACHIN. « DINA-MICA—a stochastic cellular automata model designed to simulate the landscape dynamics in an Amazonian colonization frontier ». *Ecological modelling* 154.3 (2002), p. 217-235 (cf. p. 28).
- [73] François PETITJEAN, Jordi INGLADA et Pierre GANÇARSKI. « Satellite image time series analysis under time warping ». *IEEE transactions on geoscience and remote sensing* 50.8 (2012), p. 3081-3095 (cf. p. 30).
- [74] Rolf SIMOES, Gilberto CAMARA, Gilberto QUEIROZ, Felipe SOUZA, Pedro R ANDRADE, Lorena SANTOS et al. « Satellite Image Time Series Analysis for Big Earth Observation Data ». *Remote Sensing* 13.13 (2021), p. 2428 (cf. p. 30, 45).
- [75] Xiaoli REN, Xiaoyong LI, Kaijun REN, Junqiang SONG, Zichen XU, Kefeng DENG et al. « Deep Learning-Based Weather Prediction : A Survey ». *Big Data Research* 23 (2021), p. 100178 (cf. p. 30, 31).
- [76] SHI XINGJIAN, Zhouong CHEN, Hao WANG, Dit-Yan YEUNG, Wai-Kin WONG et Wang-chun WOO. « Convolutional LSTM network : A machine learning approach for precipitation nowcasting ». *Advances in neural information processing systems*. 2015, p. 802-810 (cf. p. 31, 34, 72, 73).
- [77] Xingjian SHI, Zhihan GAO, Leonard LAUSEN, Hao WANG, Dit-Yan YEUNG, Wai-kin WONG et al. « Deep learning for precipitation nowcasting : A benchmark and a new model ». *Advances in neural information processing systems*. 2017, p. 5617-5627 (cf. p. 31, 34).
- [78] Cyril VOYANT, Marc MUSELLI, Christophe PAOLI et Marie-Laure NIVET. « Numerical Weather Prediction (NWP) and hybrid ARMA/ANN model to predict global radiation ». *Energy* 39 (2012) (cf. p. 31).
- [79] Jean-Noël THÉPAUT et Erik ANDERSSON. « The global observing system ». *Data Assimilation*. Springer, 2010, p. 263-281 (cf. p. 31).

-
- [80] Bangqian CHEN, Zhixiang WU, Jikun WANG, Jinwei DONG, Liming GUAN, Junming CHEN et al. « Spatio-temporal prediction of leaf area index of rubber plantation using HJ-1A/1B CCD images and recurrent neural network ». *ISPRS Journal of Photogrammetry and Remote Sensing* 102 (2015), p. 148-160 (cf. p. 32).
- [81] Monidipa DAS et Soumya K GHOSH. « Deep-STEP : A deep learning approach for spatiotemporal prediction of remote sensing data ». *IEEE Geoscience and Remote Sensing Letters* 13.12 (2016), p. 1984-1988 (cf. p. 32, 38, 40, 44).
- [82] Changjiang XIAO, Nengcheng CHEN, Chuli HU, Ke WANG, Zewei XU, Yaping CAI et al. « A spatiotemporal deep learning model for sea surface temperature field prediction using time-series satellite data ». *Environmental Modelling & Software* 120 (2019), p. 104502 (cf. p. 32, 34).
- [83] Qiang ZHANG, Qiangqiang YUAN, Chao ZENG, Xinghua LI et Yancong WEI. « Missing data reconstruction in remote sensing image with a unified spatial-temporal-spectral deep convolutional neural network ». *IEEE Transactions on Geoscience and Remote Sensing* 56.8 (2018), p. 4274-4288 (cf. p. 32, 39, 40, 43, 44).
- [84] Niyazi ARSLAN et Alihsan SEKERTEKIN. « Application of Long Short-Term Memory neural network model for the reconstruction of MODIS Land Surface Temperature images ». *Journal of Atmospheric and Solar-Terrestrial Physics* 194 (2019), p. 105100 (cf. p. 33, 43).
- [85] Mohamed Akram ZAYTAR et Chaker EL AMRANI. « Sequence to sequence weather forecasting with long short-term memory recurrent neural networks ». *International Journal of Computer Applications* 143.11 (2016), p. 7-11 (cf. p. 34, 44).
- [86] Rai A SCHWALBERT, Telmo AMADO, Geomar CORASSA, Luan Pierre POTT, PV Vara PRASAD et Ignacio A CIAMPITTI. « Satellite-based soybean yield forecast : Integrating machine learning and weather data for improving crop yield prediction in southern Brazil ». *Agricultural and Forest Meteorology* 284 (2020), p. 107886 (cf. p. 34, 40).
- [87] Quang-Khai TRAN et Sa-kwang SONG. « Computer Vision in Precipitation Nowcasting : Applying Image Quality Assessment Metrics for Training Deep Neural Networks ». *Atmosphere* 10.5 (2019), p. 244 (cf. p. 34).
- [88] Seongchan KIM, Seungkyun HONG, Minsu JOH et Sa-kwang SONG. « Deeprain : ConvLstm network for precipitation prediction using multichannel radar data ». *arXiv preprint arXiv :1711.02316* (2017) (cf. p. 34).
- [89] Chao TAN, Xin FENG, Jianwu LONG et Li GENG. « FORECAST-CLSTM : a new convolutional LSTM network for cloudage nowcasting ». *2018 IEEE Visual Communications and Image Processing (VCIP)*. IEEE. 2018, p. 1-4 (cf. p. 34, 44).
- [90] Shuai XIE, Wenyan WU, Sebastian MOOSER, QJ WANG, Rory NATHAN et Yuefei HUANG. « Artificial neural network based hybrid modeling approach for flood inundation modeling ». *Journal of Hydrology* 592 (2021), p. 125605 (cf. p. 35).
- [91] Changjiang XIAO, Nengcheng CHEN, Chuli HU, Ke WANG, Jianya GONG et Zeqiang CHEN. « Short and mid-term sea surface temperature prediction using time-series satellite data and LSTM-AdaBoost combination approach ». *Remote Sensing of Environment* 233 (2019), p. 111358 (cf. p. 35, 36).
- [92] Ian GOODFELLOW, Jean POUGET-ABADIE, Mehdi MIRZA, Bing XU, David WARDE-FARLEY, Sherjil OZAIR et al. « Generative adversarial nets ». *Advances in neural information processing systems*. 2014, p. 2672-2680 (cf. p. 36).
- [93] Zhan XU, Jun DU, Jingjing WANG, Chunxiao JIANG et Yong REN. « Satellite Image Prediction Relying on GAN and LSTM Neural Networks ». *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE. 2019, p. 1-6 (cf. p. 36, 37).
- [94] Lin TIAN, Xutao LI, Yunming YE, Pengfei XIE et Yan LI. « A Generative Adversarial Gated Recurrent Unit Model for Precipitation Nowcasting ». *IEEE Geoscience and Remote Sensing Letters* (2019) (cf. p. 37, 40).
- [95] Seungkyun HONG, Seongchan KIM, Minsu JOH et Sa-Kwang SONG. « Pstique : Next sequence prediction of satellite images using a convolutional sequence-to-sequence network ». *arXiv preprint arXiv :1711.10644* (2017) (cf. p. 37).
- [96] Zisis I PETROU et Yingli TIAN. « Prediction of Sea Ice Motion With Convolutional Long Short-Term Memory Networks ». *IEEE Transactions on Geoscience and Remote Sensing* 57.9 (2019), p. 6865-6876 (cf. p. 37, 40, 44).

- [97] Monidipa DAS et Soumya K GHOSH. « A deep-learning-based forecasting ensemble to predict missing data for remote sensing analysis ». *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 10.12 (2017), p. 5228-5236 (cf. p. 38, 43).
- [98] Antonio MAZZA, Massimiliano GARGIULO, Giuseppe SCARPA et Raffaele GAETANO. « Estimating the NDVI from SAR by Convolutional Neural Networks ». *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2018, p. 1954-1957 (cf. p. 38, 43).
- [99] Mykel J KOCHENDERFER et Tim A WHEELER. *Algorithms for optimization*. Mit Press, 2019 (cf. p. 39).
- [100] Diederik P KINGMA et Jimmy BA. « Adam : A method for stochastic optimization ». *arXiv preprint arXiv :1412.6980* (2014) (cf. p. 40, 41).
- [101] Tijmen TIELEMAN et Geoffrey HINTON. « Neural networks for machine learning ». *Coursera (Lecture 65-RMSprop)* (2012) (cf. p. 40).
- [102] Alexei BOTCHKAREV. « Performance metrics (error measures) in machine learning regression, forecasting and prognostics : Properties and typology ». *arXiv preprint arXiv :1809.03006* (2018) (cf. p. 42).
- [103] Alain HORE et Djemel ZIOU. « Image quality metrics : PSNR vs. SSIM ». *2010 20th international conference on pattern recognition*. IEEE. 2010, p. 2366-2369 (cf. p. 43).
- [104] Umme SARA, Morium AKTER et Mohammad Shorif UDDIN. « Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study ». *Journal of Computer and Communications* 7.3 (2019), p. 8-18 (cf. p. 43, 87).
- [105] P JAGALINGAM et Arkal Vittal HEGDE. « A review of quality metrics for fused image ». *Aquatic Procedia* 4 (2015), p. 133-142 (cf. p. 43).
- [106] Dino IENCO, Roberto INTERDONATO, Raffaele GAETANO et Dinh Ho Tong MINH. « Combining Sentinel-1 and Sentinel-2 Satellite Image Time Series for land cover mapping via a multi-source deep learning architecture ». *ISPRS Journal of Photogrammetry and Remote Sensing* 158 (2019), p. 11-22 (cf. p. 44).
- [107] Hanen BALTI, Ali Ben ABBES, Nedra MELLOULI, Imed Riadh FARAH, Yanfang SANG et Myriam LAMOLLE. « A review of drought monitoring with big data : Issues, methods, challenges and research directions ». *Ecological Informatics* 60 (2020), p. 101136 (cf. p. 45).
- [108] Yan MA, Haiping WU, Lizhe WANG, Bormin HUANG, Rajiv RANJAN, Albert ZOMAYA et al. « Remote sensing big data computing : Challenges and opportunities ». *Future Generation Computer Systems* 51 (2015), p. 47-60 (cf. p. 46).
- [109] John E BALL, Derek T ANDERSON et Chee Seng CHAN SR. « Comprehensive survey of deep learning in remote sensing : theories, tools, and challenges for the community ». *Journal of Applied Remote Sensing* 11.4 (2017), p. 042609 (cf. p. 46).
- [110] Lucas Prado OSCO, José Marcato JUNIOR, Ana Paula Marques RAMOS, Lúcio André de Castro JORGE, Sarah Narges FATHOLAH, Jonathan de Andrade SILVA et al. « A review on deep learning in UAV remote sensing ». *arXiv preprint arXiv :2101.10861* (2021) (cf. p. 48).
- [111] Timothy HOSPEDALES, Antreas ANTONIOU, Paul MICAELLI et Amos STORKEY. « Meta-learning in neural networks : A survey ». *arXiv preprint arXiv :2004.05439* (2020) (cf. p. 48).
- [112] Agency EUROPEAN Space. *Couverture géographique de Sentinel-1*. Disponible en ligne : <https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-1/satellite-description/geographical-coverage> (visité le 25 mai 2022) (cf. p. 50, 59).
- [113] Evert ATTEMA, Ciro CAFFORIO, Michael GOTTWALD, Pietro GUCCIONE, Andrea Monti GUARNIERI, Fabio ROCCA et al. « Flexible dynamic block adaptive quantization for Sentinel-1 SAR missions ». *IEEE Geoscience and Remote Sensing Letters* 7.4 (2010), p. 766-770 (cf. p. 51).
- [114] K ADJONOU, A KÉMAVO, JK FONTODJI, W TCHANI, F SODJINO, Ma T SEBASTIÀ et al. « Vegetation dynamics patterns, biodiversity conservation and structure of forest ecosystems in the wildlife reserve of Togodo in Togo, West Africa ». *International Journal of Development Research, 2017, vol. 7, núm. 8, p. 14549-14557* (2017) (cf. p. 52).
- [115] CLIMATE-DATA.ORG. *Les données climatiques pour les villes du monde entier*. Disponible en ligne : <https://fr.climate-data.org/> (visité le 25 mai 2022) (cf. p. 53, 55, 56).
- [116] Tchégnon ABOTCHI. « Dynamisme économique et évolution du milieu rural dans l'est de la région des plateaux au Togo ». Thèse de doct. Bordeaux 3, 1997 (cf. p. 53).

- [117] José Elvire Boukeng DJIONGO, André DESROCHERS, Marie Louise Tiencheu AVANA, Damase KHASA, Louis ZAPFACK, Éric FOTSING et al. « Analysis of spatio-temporal dynamics of land use in the Bouba Ndjidda National Park and its adjacent zone (North Cameroun) ». *Open Journal of Forestry* 10.01 (2019), p. 39 (cf. p. 54).
- [118] Charles DOUMENGE, Florence PALLA, Itsoua MADZOUS et Gervais LUDOVIC. *Aires protégées d'Afrique centrale-État 2020*. Disponible en ligne : <https://www.greenpeace.fr/lunesco-echoue-a-protoger-la-reserve-du-dja-au-cameroun/> (visité le 30 juin 2022) (cf. p. 54).
- [119] Manfred Aimé EPANDA, André Junior Mukam FOTSING, Thomas BACHA, Daniel FRYNTA, Luc LENS, Isaac Roger TCHOUAMO et al. « Linking local people's perception of wildlife and conservation to livelihood and poaching alleviation : A case study of the Dja biosphere reserve, Cameroon ». *Acta Oecologica* 97 (2019), p. 42-48 (cf. p. 56, 92).
- [120] Mora BRICE. *Sentinel Application Platform (SNAP)*. Disponible en ligne : https://gofcgold.org/sites/default/files/2019-04/ESA_SNAP-Toolbox_BriceMora.pdf (visité le 22 janvier 2022) (cf. p. 60).
- [121] L VECL. « SNAP Command Line Tutorial Graph Processing ». ESA. 2016 (cf. p. 60, 61).
- [122] Gaia Vaglio LAURIN, Johannes BALLING, Piermaria CORONA, Walter MATTIOLI, Dario PAPALE, Nicola PULETTI et al. « Above-ground biomass prediction by Sentinel-1 multitemporal data in central Italy with integration of ALOS2 and Sentinel-2 data ». *Journal of Applied Remote Sensing* 12.1 (2018), p. 016008 (cf. p. 63).
- [123] Pauline DUSSEUX. « Exploitation de séries temporelles d'images satellites à haute résolution spatiale pour le suivi des prairies en milieu agricole ». Thèse de doct. Université Rennes 2, 2014 (cf. p. 64).
- [124] Felix A GERS, Jürgen SCHMIDHUBER et Fred CUMMINS. « Learning to forget : Continual prediction with LSTM ». *Neural computation* 12.10 (2000), p. 2451-2471 (cf. p. 70).
- [125] Klaus GREFF, Rupesh K SRIVASTAVA, Jan KOUTNIK, Bas R STEUNEBRINK et Jürgen SCHMIDHUBER. « LSTM : A search space odyssey ». *IEEE transactions on neural networks and learning systems* 28.10 (2016), p. 2222-2232 (cf. p. 70, 71).
- [126] Md Saiful ISLAM et Emam HOSSAIN. « Foreign exchange currency rate prediction using a GRU-LSTM hybrid network ». *Soft Computing Letters* (2020), p. 100009 (cf. p. 70).
- [127] Zhenyu LIU, Mason del ROSARIO et Zhi DING. « A markovian model-driven deep learning framework for massive MIMO CSI feedback ». *IEEE Transactions on Wireless Communications* 21.2 (2021), p. 1214-1228 (cf. p. 72).
- [128] Alex GRAVES, Abdel-rahman MOHAMED et Geoffrey HINTON. « Speech recognition with deep recurrent neural networks ». *2013 IEEE international conference on acoustics, speech and signal processing*. Ieee. 2013, p. 6645-6649 (cf. p. 72).
- [129] Hong LIN, Yunzi HUA, Leiming MA et Lei CHEN. « Application of ConvLSTM network in numerical temperature prediction interpretation ». *Proceedings of the 2019 11th International Conference on Machine Learning and Computing*. 2019, p. 109-113 (cf. p. 73).
- [130] Soonil KWON. « CLSTM : Deep feature-based speech emotion recognition using the hierarchical ConvLSTM network ». *Mathematics* 8.12 (2020), p. 2133 (cf. p. 73).
- [131] Amit KELOTRA et Prateek PANDEY. « Stock market prediction using optimized deep-convlstm model ». *Big Data* 8.1 (2020), p. 5-24 (cf. p. 73).
- [132] Dong YANG, Tao XIONG, Daguang XU, S Kevin ZHOU, Zhoubing XU, Mingqing CHEN et al. « Deep image-to-image recurrent network with shape basis learning for automatic vertebra labeling in large-scale 3D CT volumes ». *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2017, p. 498-506 (cf. p. 73).
- [133] Yang LI, Huahu XU, Minjie BIAN et Junsheng XIAO. « Attention based CNN-ConvLSTM for pedestrian attribute recognition ». *Sensors* 20.3 (2020), p. 811 (cf. p. 73).
- [134] Oriol VINYALS, Alexander TOSHEV, Samy BENGIO et Dumitru ERHAN. « Show and tell : A neural image caption generator ». *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, p. 3156-3164 (cf. p. 74).
- [135] Jeffrey DONAHUE, Lisa ANNE HENDRICKS, Sergio GUADARRAMA et Marcus RÖHRBACH. « Long-term recurrent convolutional networks for visual recognition and description ». *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, p. 2625-2634 (cf. p. 74).

- [136] Zhou WANG, Alan C BOVIK, Hamid R SHEIKH et Eero P SIMONCELLI. « Image quality assessment : from error visibility to structural similarity ». *IEEE transactions on image processing* 13.4 (2004), p. 600-612 (cf. p. 77).
- [137] Rahel HAMAD, Heiko BALZTER et Kamal KOLO. « Predicting land use/land cover changes using a CA-Markov model under two different scenarios ». *Sustainability* 10.10 (2018), p. 3421 (cf. p. 84).
- [138] Isabela Xavier FLOREANO et Luzia Alice Ferreira de MORAES. « Land use/land cover (LULC) analysis (2009–2019) with Google Earth Engine and 2030 prediction using Markov-CA in the Rondônia State, Brazil ». *Environmental Monitoring and Assessment* 193.4 (2021), p. 1-17 (cf. p. 84).
- [139] Aafaf EL JAZOULI, Ahmed BARAKAT, Rida KHELLOUK, Jamila RAIS et Mohamed EL BAGHDADI. « Remote sensing and GIS techniques for prediction of land use land cover change effects on soil erosion in the high basin of the Oum Er Rbia River (Morocco) ». *Remote Sensing Applications : Society and Environment* 13 (2019), p. 361-374 (cf. p. 84).
- [140] Derya OZTURK. « Urban growth simulation of Atakum (Samsun, Turkey) using cellular automata-Markov chain and multi-layer perceptron-Markov chain models ». *Remote Sensing* 7.5 (2015), p. 5918-5950 (cf. p. 84).
- [141] H THOMAS et HM LAURENCE. « Modeling and projecting land-use and land-cover changes with a cellular automaton in considering landscape trajectories : An improvement for simulation of plausible future states ». *EARSel eProc* 5 (2006), p. 63-76 (cf. p. 84).
- [142] A Miranda NETO, A Correa VICTORINO, Isabelle FANTONI, Douglas Eduardo ZAMPIERI, Janito Vaqueiro FERREIRA et Danilo Alves LIMA. « Image processing using Pearson's correlation coefficient : Applications on autonomous robotics ». *2013 13th International Conference on Autonomous Robot Systems*. IEEE. 2013, p. 1-6 (cf. p. 87).
- [143] Sudhir Kumar SINGH, Prosper Basommi LAARI, SK MUSTAK, Prashant K SRIVASTAVA et Szilárd SZABÓ. « Modelling of land use land cover change using earth observation data-sets of Tons River Basin, Madhya Pradesh, India ». *Geocarto international* 33.11 (2018), p. 1202-1222 (cf. p. 87).
- [144] Peace GREE. *L'UNESCO échoue à protéger la Réserve du Dja au Cameroun*. Disponible en ligne : <https://www.recovery-observatory.org/drupal/fr/system/files/public/News/20180511%20-%20Introduction%20Imagerie%20Spatiale.pdf> (visité le 25 mai 2022) (cf. p. 93).
- [145] Anibal FLORES, Hugo TITO-CHURA et Honorio APAZA-ALANOCA. « Data augmentation for short-term time series prediction with deep learning ». *Intelligent Computing*. Springer, 2021, p. 492-506 (cf. p. 97).
- [146] Umberto MICHELUCCI. « Applied Deep Learning with TensorFlow 2 » (2022) (cf. p. 99).
- [147] Christopher M BISHOP et al. *Neural networks for pattern recognition*. Oxford university press, 1995 (cf. p. 99).
- [148] Jan KUKAČKA, Vladimir GOLKOV et Daniel CREMERS. « Regularization for deep learning : A taxonomy ». *arXiv preprint arXiv :1710.10686* (2017) (cf. p. 99).
- [149] Nitish SRIVASTAVA, Geoffrey HINTON, Alex KRIZHEVSKY, Ilya SUTSKEVER et Ruslan SALAKHUTDINOV. « Dropout : a simple way to prevent neural networks from overfitting ». *The journal of machine learning research* 15.1 (2014), p. 1929-1958 (cf. p. 100).
- [150] James BERGSTRA, Rémi BARDENET, Yoshua BENGIO et Balázs KÉGL. « Algorithms for hyper-parameter optimization ». *Advances in neural information processing systems* 24 (2011) (cf. p. 100).
- [151] Lisha LI, Kevin JAMIESON, Giulia DESALVO, Afshin ROSTAMIZADEH et Ameet TALWALKAR. « Hyperband : A novel bandit-based approach to hyperparameter optimization ». *The Journal of Machine Learning Research* 18.1 (2017), p. 6765-6816 (cf. p. 101).
- [152] Tom O'MALLEY, Elie BURSSTEIN, James LONG, François CHOLLET, Haifeng JIN, Luca INVERNIZZI et al. *KerasTuner*. <https://github.com/keras-team/keras-tuner>. 2019 (cf. p. 102).
- [153] Eric Wilson Tegno NGUEKAM, Salomon C Nguemhe FILS, Joachim ETOUNA et Simon Njeudeng TENKU. « Analyse de la déforestation dans la périphérie ouest de la réserve de biosphère du Dja au Cameroun, à partir d'une série multi-annuelle d'images Landsat ». *Revue Française de Photogrammétrie et de Télédétection* 222 (2020), p. 31-41 (cf. p. 106).
- [154] Haibin CHENG, Pang-Ning TAN, Jing GAO et Jerry SCRIPPS. « Multistep-ahead time series prediction ». *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2006, p. 765-774 (cf. p. 108).