

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE APPLIQUÉES

PAR
MASSE BA

ESTIMATION DE LA PRODUCTION DES DÉCHETS POUR LA
PRODUCTION D'ÉNERGIE PAR LE MODÈLE ARIMA ET LE LSTM

DÉCEMBRE 2023

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire, de cette thèse ou de cet essai a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire, de sa thèse ou de son essai.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire, cette thèse ou cet essai. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire, de cette thèse et de son essai requiert son autorisation.

Résumé

Les déchets solides comprennent tous les matériaux non fluides mis au rebut. Selon Compétivert¹, un système de gestion efficient de ces déchets peut contribuer à atteindre les Objectifs de Développement Durable, en particulier l'ODD7 (énergie renouvelable) et l'ODD12 (consommation et production responsable). Les décideurs préconisent donc le passage de l'économie linéaire à l'économie circulaire, qui vise à réutiliser et recycler les déchets pour une utilisation ultérieure. Notre étude se concentre sur l'estimation des déchets solides de bois pour la production d'énergie. Pour ce faire, nous avons utilisé un ensemble de données de Statistique Canada, qui fournit la quantité de déchets produits annuellement de 1995 à 2021. Nous avons ensuite développé deux modèles, ARIMA et LSTM, pour estimer la quantité de déchets produits chaque année. Afin d'évaluer la précision de nos modèles, nous avons utilisé la technique de validation croisée. Nos analyses indiquent que le modèle ARIMA donne de meilleurs résultats, avec une erreur quadratique moyenne (*RMSE*) de 0,0141. Nous suggérons d'inclure la quantité de déchets générés avant 1995 et, si possible, de collecter des données sur une base trimestrielle, voire mensuelle, pour obtenir de meilleures prévisions.

Abstract

Solid waste includes all non-fluid materials that are discarded. According to Compétivert ², an efficient waste management system can contribute to achieving the Sustainable Development Goals, in particular SDG7 (renewable energy) and SDG12 (responsible consumption and production). Decision-makers therefore advocate the transition from a linear economy to a circular economy, which aims to reuse and recycle waste for further use. Our study focuses on estimating solid wood waste for energy production. To do this, we used a dataset from Statistics Canada, which provides the amount of waste produced annually from 1995 to 2021. We then developed two models, ARIMA and LSTM, to estimate the amount of waste produced each year. To evaluate the accuracy of our models, we used the cross-validation technique. Our analyses indicate that the ARIMA model gives better results, with a root mean square error (*RMSE*) of 0.0141. We suggest including the amount of waste generated before 1995 and, if possible, collecting data on a quarterly, or even monthly, basis to obtain better forecasts.

2. <https://www.competivert.com>

Remerciements

Je tiens à remercier tout d'abord ma directrice de recherche, Professeure Amina Lamghari, pour tout le soutien et la confiance qu'elle m'a accordés durant tout mon parcours à l'UQTR. Son esprit critique et sa rigueur m'ont permis de bonifier ce document. Je lui serai éternellement reconnaissant pour toutes les connaissances et expériences qu'elle a partagées avec moi tout au long de ma maîtrise ainsi que de son soutien financier.

Mes remerciements vont aussi à l'endroit de mon co-directeur de recherche, Professeur Mhamed Mesfioui, pour son soutien pédagogique et financier. Je remercie tout le personnel de l'UQTR, particulièrement celui du département de mathématiques et d'informatique pour tous les services rendus, leur réactivité et leur orientation qui ont beaucoup facilité mon intégration.

Je ne saurais terminer sans adresser mes remerciements à tous les étudiants avec qui j'ai eu à collaborer et, plus précisément, à Mamadou Mbodj pour son esprit de fraternité, de solidarité et son brillantissime esprit critique.

Je remercie ma famille au Sénégal, mes amis et toute la communauté sénégalaise au Canada. J'adresse spécialement un grand merci à mon père Abdourahmane Ba, à mon oncle Waly Faye et sa famille au Sénégal ainsi qu'à mes frères et soeurs pour leur amour, leur soutien moral et leur conseil.

Je dédie ce mémoire, aux deux dames valeureuses qui ont eu un grand impact dans ma vie, précisément, ma mère Marième Diop et ma grand-mère Mame Sokhna

Tandiné.

Elles m'ont soutenu dans mes activités scolaires tant durant ma période de scolarisation qu'après l'école.

Merci pour tout ce que vous avez fait pour moi.

Ce mémoire témoigne toute l'affection que j'ai pour vous, et me permettra de me souvenir tout au long de mon parcours les moments d'intense joie et de bonheur que vous m'avez apporté.

*« Adouna loko diay mou dieund
ko... » Mon oncle Abdoulaye Fofana
Seck*

Table des matières

Résumé	ii
Abstract	iii
Remerciement	iv
Table des matières	vi
Liste des tableaux	ix
Table des figures	xi
Introduction	1
1 Revue de la littérature	4
1.1 Revue sur l'économie circulaire	4
1.2 Revue sur l'estimation des déchets	7
2 Méthodologie utilisée	13
2.1 Modèle des séries chronologiques	13
2.1.1 Tendance	14
2.1.2 Stationnarité	15
2.1.3 Saisonnalité (S)	15
2.2 Mémoire à court et long terme (LSTM)	20
2.2.1 Réseaux de neurones	20
2.2.2 L'idée de base derrière les LSTM	22

2.2.3	Les Étapes du modèle LSTM	23
2.2.4	Méthode d'apprentissage	27
2.2.5	Algorithme du gradient	28
2.2.6	Algorithme d'ADAM	29
2.3	Choix du modèle ARIMA et LSTM	30
2.3.1	Avantages des modèles ARIMA et LSTM	31
2.3.2	Limites des modèle ARIMA et LSTM	32
2.4	Méthode expérimentale	32
2.4.1	Vraisemblance (L)	33
2.4.2	Critère d'information d'Akaike (AIC)	34
2.4.3	Critère d'information bayésien (BIC)	34
2.4.4	Erreur quadratique moyenne ($RMSE$)	34
2.4.5	Coefficient de détermination (R^2)	35
3	Résultats des modèles ARIMA et LSTM	36
3.1	Familiarisation avec les données	36
3.1.1	Source de données	37
3.1.2	Traitements sur les données	38
3.2	Résultat des modèles ARIMA : Approche Box-Jenkins	42
3.2.1	Spécification des modèles	43
3.2.2	Estimation des paramètres	44
3.2.3	Validation par tests	46
3.2.4	Prévision	48
3.3	Résultats des modèles LSTM	50
3.4	Recommandations	53
	Conclusion et perspectives	55
	Bibliographie	57
	A Code du modèle ARIMA sur R	60

B Code du modèle LSTM sur \mathbb{R} **71**

Liste des tableaux

1.1	Méthodes et modèles utilisés pour estimer la production des déchets solides municipaux	8
3.1	Tests de stationnarité sur les séries transformées	40
3.2	Récapitulatif des différents modèles testés et leurs pouvoirs explicatifs .	43
3.3	Récapitulatif des différents modèles testés et leurs pouvoirs explicatifs .	44
3.4	Valeurs estimées des paramètres	47
3.5	Prévisions de la production des déchets sur un horizon de 5 ans	49
3.6	Prévisions et valeurs réelles	50
3.7	Récapitulatif des différents modèles testés et leurs RMSE	51

Table des figures

1.1	Schéma de l'économie circulaire	7
1.2	Comparaison entre les modèle ARIMA, ANN, et LSTM	12
2.1	Module répétitif dans un RNN standard contenant une seule couche. . .	21
2.2	Module répétitif dans un LSTM contenant quatre couches en interaction.	21
2.3	Notation des opérateurs et des couches	22
2.4	Module répétitif dans un LSTM contenant quatre couches en interaction.	22
2.5	État de la cellule LSTM.	23
2.6	Étape 1 du LSTM.	24
2.7	Étape 2 du LSTM.	25
2.8	Étape 3 du LSTM.	26
2.9	Étape 4 du LSTM.	27
3.1	Représentation des déchets de bois solide durant la période de 1995 à 2021	37
3.2	Boxplot de la série de la base de données des déchets de bois solides . .	39
3.3	Quelques transformations sur la série brute	40
3.4	Représentation de la série corrigée ($\ll Y_t \gg$)	41
3.5	Représentation AIC, BIC et de la vraisemblance	45
3.6	Représentation des RMSE des modèles ARIMA associés aux jeux de données	45
3.7	Graphe quantile-quantile (QQ-Plot) pour test de normalité des résidus avec la statistique de Ljung-box	47
3.8	Prévision des déchets de bois pour la production d'énergie	49

3.9 Valeurs prédites comparativement aux valeurs réelles	50
3.10 Valeurs prédites comparativement aux valeurs réelles	52

Introduction

Le Canada accorde une grande importance à la prévention des déchets, que ce soit au niveau des organisations que des individus. Le réacheminement des déchets vers des usages alternatifs plutôt que leur enfouissement est également une préoccupation majeure, non seulement au Canada, mais partout dans le monde. Les déchets, lorsqu'ils sont valorisés, recyclés et réutilisés, peuvent générer des bénéfices économiques [1]. Nous nous intéressons aux déchets solides qui englobent tous les matériaux non fluides mis au rebut. Le recyclage ou la réutilisation de ces déchets est un élément clé de l'économie circulaire, qui vise à réduire, réutiliser, recycler et récupérer les déchets pour une utilisation ultérieure. L'objectif est de transformer la structure de l'économie en un système plus résilient et durable, en contraste avec l'économie linéaire qui tend à jeter et à ne pas réutiliser les matériaux ou produits déjà utilisés. Ainsi, les décideurs politiques mènent des actions pour faciliter la transition de l'économie linéaire à l'économie circulaire.

L'objectif de ce mémoire est d'estimer la production des déchets solides de bois pour la production d'énergie, et ce en utilisant des techniques statistiques classiques et d'apprentissage machine, notamment la mémoire à court et long terme (LSTM). Cette dernière est la technique la plus utilisée lorsqu'on a des données chronologiques sur une période relativement courte [2]. De plus, elle donne de meilleurs résultats avec des erreurs de prévision faibles [3]. Notre but est donc de développer le meilleur modèle pour estimer la quantité annuelle de déchets. À cette fin, nous testons plusieurs

types de modèles en faisant varier leurs paramètres, et nous utilisons la technique de validation croisée pour évaluer leur précision. Spécifiquement, nous varions la taille de notre jeu de données en retirant un certain nombre d'années, selon le modèle, et nous effectuons une estimation du modèle avec le reste des données. Nous calculons ensuite l'erreur de prévision sur les lignes de données qui ont été retirées et les critères de performance du modèle. Ainsi, nous avons obtenu un ensemble de 16 modèles, dont 8 de types ARIMA et 8 de type LSTM. La différence entre ces modèles, réside dans les jeux de données sur lesquels ils ont été entraînés et, par conséquent, les paramètres des modèles qui dépendent des données d'entraînement. Notre objectif est de comprendre le comportement de chaque type de modèle lorsque la taille du jeu de données d'entraînement augmente ou diminue. En parallèle, nous cherchons à déterminer la meilleure façon de construire un modèle qui permet d'obtenir des prévisions précises de la quantité de déchets produits annuellement. À partir des résultats obtenus, nous souhaitons identifier les politiques à adopter pour une meilleure gestion des déchets, tenant compte de l'évolution de la quantité de déchets et de sa tendance.

Le reste du mémoire est organisé comme suit. Le premier chapitre est composé de trois sections : une revue théorique sur l'économie circulaire, une présentation des méthodes statistiques utilisées dans la littérature pour estimer la quantité des déchets produits et une description de la méthode expérimentale utilisée dans ce mémoire. Le second chapitre pose le cadre méthodologique en présentant les sources de données, les techniques utilisées pour les analyses préliminaires, et les spécifications des modèles de prévision utilisés. Nous y présentons également les algorithmes d'estimation. Nous avons fait appel à des modèles classiques tels que les approches autorégressives (AR) et les moyennes mobiles (MA), mais aussi à des techniques d'apprentissage profond comme les réseaux de neurones récurrents. Le troisième chapitre porte sur les résultats des modèles susmentionnés. D'abord, nous discutons des résultats des modèles classiques testés. Ensuite, nous évaluons les résultats obtenus avec l'apprentissage profond, notamment en termes de la qualité des prévisions et la capacité à s'ajuster aux

données. Enfin, nous présentons quelques recommandations pour une meilleure gestion des déchets solides de bois. Quelques politiques visant une utilisation plus efficace de ces déchets sont présentées. Dans le dernier chapitre, nous résumons nos principaux résultats de recherche et proposons quelques perspectives pour des travaux futurs.

Chapitre 1

Revue de la littérature

Dans ce chapitre, nous passons en revue les travaux existants sur l'estimation de la production des déchets ainsi que le cadre de l'économie linéaire et de l'économie circulaire. Nous nous concentrons particulièrement sur les études portant sur la production des déchets solides et les liens multiples connexes avec notre thématique ; l'objectif étant de contextualiser notre sujet, de le positionner et de définir notre problématique.

1.1 Revue sur l'économie circulaire

Depuis longtemps, l'économie linéaire a servi de référence et a été utilisée comme modèle d'apprentissage dans plusieurs domaines en logistique. L'économie linéaire, étant assimilée à un système consistant à prendre, fabriquer et jeter, peut mener à une surconsommation, et parfois même influencer négativement sur les conditions de développement durable. L'économie circulaire se différencie de l'économie linéaire par sa réutilisation et sur le recyclage de produits et matériaux déjà existants, pour pallier les insuffisances de l'économie linéaire. L'économie circulaire est un modèle qui

intéresse les décideurs politiques.

D'après Geissdoerfer et al. [4], l'économie circulaire est susceptible d'apporter une solution aux inconvénients de l'économie linéaire. Elle permet, d'une part, de répondre aux exigences actuelles sans compromettre celles des générations futures (ex : le recyclage...). D'autre part, elle consiste à réutiliser des produits ou des matériaux. C'est ainsi que différents auteurs ont essayé d'estimer la production des déchets afin de répondre à la problématique de réutilisation de certains produits ou matériaux. Aussi, il convient de remarquer qu'à travers des processus écosystémiques, l'économie circulaire tente de durabiliser les matériaux tout en essayant de les maintenir à leur plus haut niveau [4]. Elle a pris un envol majeur avec la commission économique de l'Union Européenne, mais aussi, et surtout à travers le programme de la Chine sur la transition vers l'économie circulaire. Selon les différentes approches adoptées par les pays afin de donner le but de l'économie circulaire, Yuan et al. [5] l'assimilent à un modèle industriel. Ghisellini et al. [6] suggèrent d'employer un terme commercial pour définir l'économie circulaire. Toutefois, la définition de Korhonen et al. [7] semble plus adaptée car la définissant comme un système économique s'authentifiant à un « modèle de flux de matières et d'Énergie ». Schroder et al. [8] rapportent que l'économie circulaire contribue à la réalisation de plusieurs Objectifs de Développement Durable (ODD) à savoir :

- l'ODD6 (eau propre et assainissement),
- l'ODD7 (énergie propre et abordable),
- l'ODD8 (travail décent et croissance économique),
- l'ODD12 (consommation et production responsable),
- Et, l'ODD15 (vie sur terre).

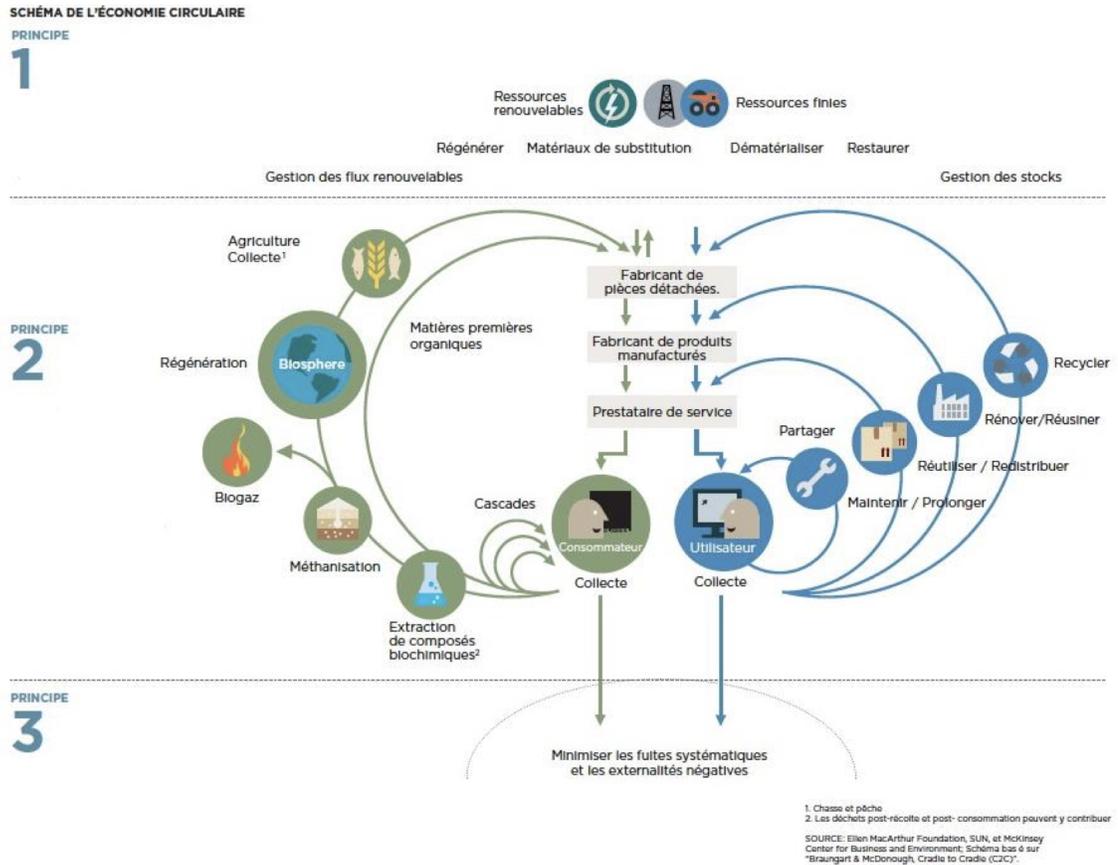
Korhonen et al. [7] définissent l'économie circulaire comme une collection d'idées vagues et séparées provenant de plusieurs domaines et de concepts semi scientifiques et suggèrent des études plus poussées pour évaluer l'impact de l'économie circulaire dans la poursuite du développement durable. En sus de cela, il n'existe pas de garantie assurant si l'allongement de la durée de vie d'un produit affecterait positivement l'en-

vironnement. Ce faisant, il n’y a pas de preuves confirmant que l’économie circulaire réduirait à bon escient l’extraction de matières premières. Néanmoins, l’économie circulaire a un effet moins accentué que l’économie linéaire sur le processus de durabilité des produits ou des matières premières.

Pour une réduction des déchets, il est important de faire son estimation pour connaître la quantité de déchets ainsi que leur provenance. C’est d’ailleurs, dans cette perspective que nous développons dans la suite les techniques d’estimation et les solutions qui ont été proposées dans la littérature, et sur ce, pour une bonne gestion de la production des déchets. Ce qui constitue une bonne démarche à adopter pour atteindre les Objectifs de Développement Durable et par conséquent, ceux de l’économie circulaire.

La figure 1.1 décrit les trois principes de l’économie circulaire et le système favorisant vers sa transition. La lecture du schéma se fait en trois étapes liées verticalement. En effet, nous avons d’abord le principe 1 qui décrit la façon dont les ressources renouvelables et les ressources finies sont utilisées. Parmi elles, certaines sont utilisées pour des matériaux de substitution ou régénérer à une utilisation agricole, de biogaz, de méthanisation ou d’extraction de composés biochimiques. Ces composantes sont réutilisés en cascade du fait que ces ressources sont renouvelables. D’autres sont dématérialisées et restaurées pour la réutilisation et le recyclage par d’autres utilisateurs. Ensuite, il y a le principe 2 qui stipule le processus de fabrication des pièces détachées, de produits manufacturés et de prestataire de service qui sont ensuite exploités pour la collecte qui sera enfin utilisée pour le principe 3. Ce dernier sert à minimiser les fuites systématiques et les externalités négatives.

FIGURE 1.1 – Schéma de l'économie circulaire



Source : Ellen MacArthur Foundation, SUN, et McKinsey Center for Business and Environment ;
Schéma basé sur « Braungart McDonough, Cradle to Cradle (C2C) » [9]

1.2 Revue sur l'estimation des déchets

Dans cette section, nous parcourons brièvement les travaux des auteurs sur les méthodes statistiques d'estimation des déchets. C'est ainsi que Oguz-Ekim [10] essaie de restituer les meilleurs modèles d'estimation de la production des déchets à partir de son ouvrage sur les approches d'apprentissage automatique pour la prévision de la production de déchets solides municipaux. À la sortie de ces résultats, il en ressort que le réseau de neurones artificiels (« ANN ») est la technique la plus utilisée pour

la prévision des déchets solides municipaux (tableau 1.1).

TABLE 1.1 – Méthodes et modèles utilisés pour estimer la production des déchets solides municipaux

Modèle utilisé	Nombre de publication
Méthode conventionnelle	
Méthodes/modèle de base de données	7
Méthode d'enquête par sondage	12
Régression linéaire	13
Modèles économétriques	4
Dynamique du système	15
Analyse des séries chronologiques	5
Analyse factorielle	9
Systèmes d'information géographique (SIG)	5
Méthode non conventionnelle	
Réseaux neuronaux artificiels (ANN)	16
Méthode floue	8
Système d'inférence floue basé sur un réseau adaptatif (ANFIS)	4
Méthodes grises	5
Machine à vecteur de support (SVM)	3
Nombre total de publications	106

Source : Oguz-Ekim [10]

Abassi et Hanandeh [11] évaluent la puissance des modèles d'apprentissage automatique (*machine learning*) à savoir le système d'inférence neuro-adaptatif (*neuro-adaptive inference system (ANFIS)*), la machine à vecteurs de support (*support vector machine (SVM)*), les K-plus proche voisins et les réseaux de neurones artificiels (ANN). Ces modèles sont testés dans la ville de Logan située au sud-est du Queensland en Australie pour voir leur capacité à prédire la production des déchets

solides, et surtout avec des données temporelles mensuelles. Au terme de leur analyse, le modèle ANFIS a réussi à prédire les pics de production des déchets, et ce, avec précision. Mais aussi, la méthode KNN a bien estimé les valeurs moyennes mensuelles de la production de déchets. Ces résultats ont aussi permis de prédire une production de déchets solides municipaux de $9,4 \times 10^7$ kg d'ici 2020 alors que le pic mensuel est évalué à $9,37 \times 10^6$ kg.

En se référant aux données démographiques et socioéconomiques de 220 municipalités de la province d'Ontario en 2017, Kannangara et al. [12] proposent dans leur étude une approche de prétraitement pour la modélisation des données publiques.

Oguz-Ekim [10], quant à lui, prédit la production de déchets solides municipaux à travers le produit intérieur brut (PIB), la consommation intermédiaire des ressources et la productivité des ressources. De ce fait, il utilise trois algorithmes de machine learning tels que le réseau de neurones à propagation arrière (*Back propagation neural Network (BPNN)*), le SVM et le réseau de neurones à régression générale (*General Regression Neural Network (GRNN)*). En utilisant le pourcentage d'erreur relative et le coefficient de détermination (R^2), le BPNN a obtenu de meilleurs résultats de prévision sur les années 2010, 2014, 2015 et 2016 .

Pour évaluer la production des déchets en Inde, Soni et al. [13] comparent les différents modèles tels que le système d'inférences neuro-adaptatives (ANFIS), la théorie des ondelettes discrètes-réseau de neurones artificiels (*discrete wavelet theory artificial neural network (DWT-ANN)*), le système d'inférence neuro-floue adaptatif à la théorie des ondelettes discrètes (DWT-ANFIS), l'algorithme génétique-réseau de neurones artificiels (GA-ANN) et le système d'inférence neuro-floue adaptatif par algorithme génétique (*genetic algorithm-adaptive neuro-fuzzy inference system (GA-ANFIS)*). Le meilleur modèle est choisi sur la base du coefficient de détermination (R^2), de l'erreur quadratique ($RMSE$) et de l'indice d'accord (IA). Après l'évaluation de tous ces critères, le GA-ANN a obtenu le meilleur pouvoir prédictif ($AUC = 87\%$)

avec un RMSE plus faible.

He et al. [14], dans leur étude, proposent une base de connaissance globale pour la gestion des déchets solides municipaux. Dans leurs expériences, les auteurs ont travaillé sur des bases de données existantes et les rapports gouvernementaux. Pour ce faire, ils ont compilé 1720 enregistrements sur la génération, la composition, les pratiques de gestion des déchets solides et les contextes socio-économiques pour 219 pays et 410 villes. Ils ont également implémenté des modèles de régression linéaire multivariée (RL) et additifs (AM) pour relier la génération, la composition et les taux de valorisation des déchets solides municipaux aux données démographiques, au développement économique et aux tendances climatiques des villes et des régions. Les modèles de taux de valorisation des déchets solides municipaux présentent également des valeurs intermédiaires de R-carré ajusté (0,46 pour le modèle LR et 0,58 pour le modèle AM). À partir de leurs résultats, les auteurs concluent que les sociétés ayant un PIB par habitant plus élevé, une population vieillissante, une densité de population plus élevée et un niveau d'éducation plus élevé ont tendance à avoir des taux de valorisation des déchets solides municipaux plus élevés. Ils concluent aussi qu'une mise en place d'une réglementation sur les déchets est également un facteur significatif, soutenant l'efficacité de stimuler la valorisation des déchets.

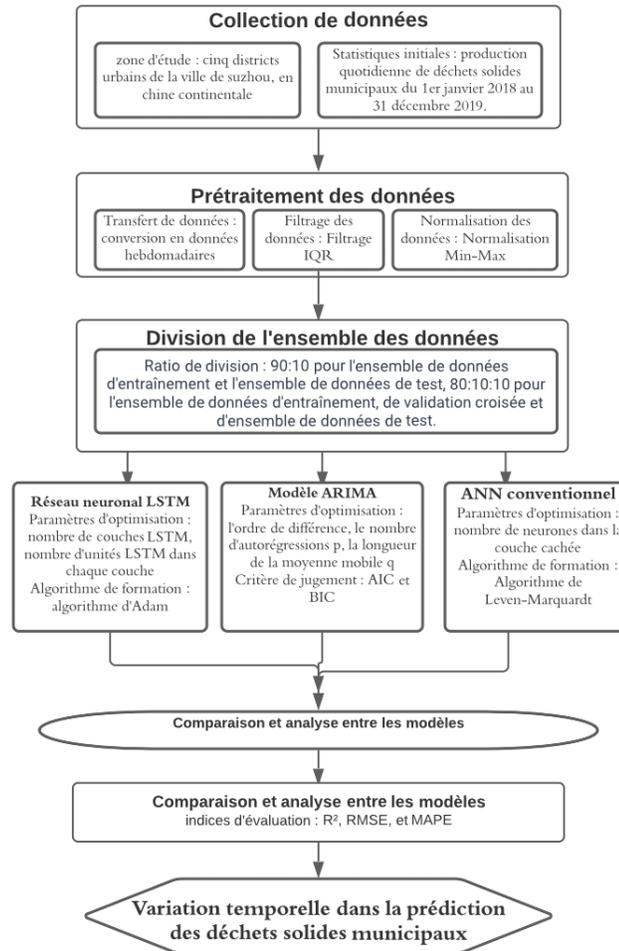
Drici [15] utilise le LSTM dans la prévision sur l'utilisation d'une méthode guidée par les données pour le pronostic des défauts. Dans un contexte d'insuffisance des données et en utilisant plusieurs jeux de données, le meilleur résultat est obtenu avec le jeu de données nommé « FD001 » qui fournit un RMSE et un MAPE respectivement égal à 17,12 et 30,59. En faisant une comparaison des résultats sur le même jeu de données mais augmenté, le modèle obtient un RMSE et un MAPE respectifs de 14,20 et 20,52. Ce qui pousse l'auteur à conclure qu'avec augmentation de données, les chercheurs pourront obtenir de meilleurs résultats lorsqu'ils l'appliquent sur leur modèle.

Dongjie et al. [3] comparent le modèle ARIMA, les modèles de réseaux de neurones conventionnels et le LSTM. La figure 1.2 donne une meilleure idée des différentes étapes depuis la collecte des données jusqu'aux modèles ARIMA, ANN, et LSTM. Sur ce, elle décrit les étapes de la collecte passant par 5 zones d'études sur des déchets solides municipaux du 1er janvier au 31 décembre 2019. Après cela, en prétraitement, les données sont transférées et converties en données hebdomadaires selon l'écart interquartile (¹) qui contient la moitié des données. Aussi, une normalisation a été faite sur les données. Ensuite les données ont été divisées en 90% de leur jeu de données en entraînement et 10% en jeu de données test. Une comparaison entre les modèles ARIMA (on l'évalue avec l'AIC² et le BIC³), LSTM et ANN sont faites avec trois indices d'évaluation à savoir le R^2 , le $RMSE$ et le $MAPE$ ⁴ aux fins de choisir le meilleur modèle de prévision des déchets solides municipaux.

Le choix du meilleur modèle est effectué par le coefficient de détermination (R^2), l'erreur quadratique ($RMSE$) et l'erreur absolue moyenne ($MAPE$). Sur la base de ces éléments, le LSTM a donné de plus bons résultats avec un R^2 de 92% , un $RMSE$ de 0,2116 et un $MAPE$ de 0,74 [3]. Par ailleurs, cette étude a permis de révéler l'effet plus ou moins long terme de la prévision de la production des déchets sur des données temporelles. Des résultats montrent que le LSTM est amélioré de 28% lorsqu'on prend en compte des données agrégées (par exemple de données multi-sites). Cubillos [16] note que le LSTM est plus puissant que le modèle ARIMA avec un pouvoir prédictif de 85%. En utilisant l'arbre de décision et le réseau de neurones, il trouve que le réseau de neurones(« ANN ») a une meilleure performance avec un pouvoir prédictif de 70%.

-
1. IQR : Écart Interquartile
 2. AIC : Critère d'information d'Akaike
 3. Critère d'information bayésien
 4. MAPE : Erreur moyenne absolue en pourcentage

FIGURE 1.2 – Comparaison entre les modèle ARIMA, ANN, et LSTM



Source : Dongjie et al. [3]

Chapitre 2

Méthodologie utilisée

Dans ce chapitre, nous présentons, en première partie les détails du modèle classique de séries chronologiques (AR, MA, ARIMA, etc. . .) qui s'adapte avec nos données à partir de quelques analyses préliminaires. Dans la seconde partie, il s'agira de présenter le réseau de neurones récurrents, plus précisément, la mémoire à court et long terme (« *Long Short Term Memory* » (LSTM)). Leur architecture nous permettra de mieux comprendre et d'étayer les résultats des chapitres dans la suite du travail.

2.1 Modèle des séries chronologiques

Il existe plusieurs paramètres et hypothèses à prendre en compte dans les modèles de séries chronologiques. Parmi ceux-ci, les plus essentiels et généralement contrôlés sont la tendance, la stationnarité, la saisonnalité. Pour ce faire, nous décrivons dans les sections qui suivent le mode de fonctionnement du modèle ARIMA ainsi que leur paramétrisation pour une bonne prévision.

2.1.1 Tendances

La tendance correspond à l'évolution à long terme de la série, encore appelée évolution fondamentale de la série [17].

Soit $X_t : t \in \mathbb{Z}$ une série chronologique. Les modèles de séries chronologiques avec tendance sont de la forme :

$$X_t = m_t + e_t \quad \text{pour tout } t;$$

où m_t représente la partie déterministe changeant lentement dans le temps et e_t la partie aléatoire de moyenne nulle. Aussi, il faudrait noter que la partie aléatoire n'est pas toujours indépendante.

Il existe plusieurs types de tendances dans les modèles de séries chronologiques. Parmi celles-ci, nous pouvons citer les tendances linéaires, les tendances exponentielles, les tendances polynomiales, etc.

Cependant, nous effectuons souvent des différenciations d'ordre d , pour obtenir une série éliminée de la tendance, obtenues à partir de l'opérateur retard noté B , défini par :

$$BX_t = X_{t-1} \quad \text{pour tout } t.$$

Parallèlement, l'opérateur retard d'ordre $m \geq 2$ est donné par :

$$B^m X_t = X_{t-m} \quad \text{pour tout } t.$$

L'opérateur retard d'ordre 1 (B) permet d'éliminer la tendance linéaire et celui d'ordre m (B^m) élimine les tendances polynômiales d'ordre m .

2.1.2 Stationnarité

La stationnarité se définit par une non affectation dans le temps de la moyenne, de la variance et de la covariance d'une série chronologique. En d'autres termes, la moyenne, la variance et la covariance de la série chronologique sont inchangées dans le temps [18]. Pour la stationnarité des séries chronologiques, les tests les plus utilisés sont ceux de racine unitaire. Ainsi, les tests de stationnarité fonctionnent comme suit :

- **Test de stationnarité** : Pour un test de stationnarité, on a : H_0 : la série est stationnaire vs H_1 : la série n'est pas stationnaire.
- **Test de non stationnarité** : Pour un test de non stationnarité, on a : H_0 : la série n'est pas stationnaire vs H_1 : la série est stationnaire.

Sachant que le rejet de l'hypothèse nulle n'est pas toujours équivalent à l'acceptation de l'hypothèse alternative, nous combinons différents tests statistiques afin de trancher sur l'évolution de notre série, c'est à dire si dans le temps, la moyenne et la variance varient. Autrement dit, il s'agit de voir si l'espérance et la covariance sont une fonction constante du temps.

2.1.3 Saisonnalité (S)

Les variations saisonnières sont des fluctuations périodiques à l'intérieur d'une année, et qui se reproduisent de façon plus ou moins permanente d'une année sur l'autre [17]. Soit X_t une série chronologique. X_t est saisonnière de période s si :

$$X_t = S_t + e_t,$$

où $S_{t+s} = S_t$ pour tout t représentant la saisonnalité, et, e_t représentant la partie résiduelle.

Lorsqu'une série chronologique admet une composante saisonnière variable, il incombe de la transformer pour obtenir une série pour laquelle le terme saisonnier est constant. Ceci dans le but d'éliminer les variations régulières, périodiques, et prévisibles, et se répétant à des intervalles de temps fixes. Nous effectuons des corrections pour mieux comprendre la série étudiée, et ce, en l'absence de cycle saisonnier.

Il n'y a pas d'unicité sur les transformations des séries chronologiques mais les plus fréquentes et qui marchent souvent sont :

— les transformations exponentielles de la forme :

$$X_t^* = X_t^\lambda \quad \text{pour tout } \lambda \text{ valeur réelle,}$$

— les transformations logarithmiques de la forme :

$$X_t^* = \log X_t,$$

— les transformations racines carrées de la forme :

$$X_t^* = \sqrt{X_t},$$

— les transformations inverses de la forme :

$$X_t^* = \frac{1}{X_t},$$

où X_t^* représente la nouvelle série corrigée des effets saisonniers.

La composante saisonnière se doit de satisfaire la condition suivante :

$$S_{t+s} = S_t,$$

où s est la composante saisonnière.

À cet effet, nous pouvons prendre en compte l'opérateur $\nabla_s = 1 - B^s$ avec $B^s X_t = X_{t-s}$, et il s'ensuit après élimination de l'effet saisonnier que $\nabla_s \times S_t = 0$ pour tout t .

La compréhension de la saisonnalité passe par deux paramètres importants à savoir l'autocovariance et l'autocorrélation. Leur procédure d'estimation est fournie ci-après.

— Fonction d'autocovariance et d'autocorrélation

Supposons que nous avons une série chronologique de suite d'observation X_1, \dots, X_n .

La fonction d'autocovariance d'ordre k , $cov(X_{t+k}, X_t)$, est estimée par :

$$\hat{\gamma}(k) = c_k = \sum_{t=1}^{n-k} (X_t - \bar{X})(X_{t+k} - \bar{X}).$$

Pareillement, la fonction d'autocorrélation d'ordre k , (X_{t+h}, X_t) est approximée de la façon suivante :

$$\hat{\rho}(k) = r_k = \frac{\sum_{t=1}^{n-k} (X_t - \bar{X})(X_{t+k} - \bar{X})}{\sum_{t=1}^n (X_t - \bar{X})^2} = \frac{c_k}{c_0},$$

où $\hat{\rho}(k)$ est la corrélation empirique entre X_t et X_{t+k} .

Afin d'identifier si une série chronologique peut être modélisée par un modèle autorégressif noté $AR(p)$, nous avons besoin de définir la notion de fonction d'autocorrélation partielle.

Soit P_k , $k \geq 1$ la matrice des autocorrélations, c'est-à-dire :

$$P_k = (\rho(i - j)), 1 \geq i, j \leq k.$$

Pour tout $k \geq 1$, nous définissons les nombres $\Phi_{k,1}, \dots, \Phi_{k,k}$ comme solutions du système :

$$\begin{pmatrix} \Phi_{k,1} \\ \vdots \\ \Phi_{k,k} \end{pmatrix} = \begin{pmatrix} \rho(1) \\ \vdots \\ \rho(k) \end{pmatrix}.$$

Le nombre $\Phi_{k,k}$ est noté autocorrélation partielle d'ordre k et ρ_k les autocorrélations.

Dans la partie ci-après, nous y expliquons le corrélogramme et la façon de l'interpréter .

— Corrélogramme et son interprétation

Le corrélogramme est une représentation de r_k en fonction de k . Le corrélogramme permet, d'une part, de détecter la structure du modèle à partir des ordres de différenciation des modèles autorégressifs et moyennes mobiles nommés par p et q . Mais, aussi, il donne la possibilité de savoir si une série est complètement aléatoire ou non.

Une série complètement aléatoire est appelée "Bruit Blanc" et est notée par $r_k \approx 0$.

En plus, la loi de r_k est approximativement normale, et, nous écrivons :

$$r_k \approx N\left(0, \frac{1}{n}\right).$$

Pour ce faire, nous pouvons obtenir une intervalle de confiance de niveau $1 - \alpha$, de r_k , définie par :

$$-\frac{z_{\frac{\alpha}{2}}}{n} \leq r_k \leq \frac{z_{\frac{\alpha}{2}}}{n}.$$

À partir de cette équation, nous pouvons déduire la disposition du corrélogramme ainsi que la nature des séries étudiées. En effet,

- Pour une série saisonnière, le corrélogramme oscille avec la même fréquence.
- Pour une corrélation à court terme, $r_k \approx 0$, pour k grand.
- Pour une série non stationnaire, r_k tend lentement vers 0.
- Pour une série alternante, le corrélogramme est décrit comme suit : $r_1 \leq 0, r_2 \leq 0, r_3 \leq 0, r_4 \leq 0, \dots$

Le corrélogramme nous aide à identifier l'ordre q d'un processus moyenne mobile nommé $MA(q)$. En effet, puisque les autocorrélations théoriques $\rho(k)$ s'annulent pour $k > q$. Donc, en observant le corrélogramme, nous pouvons repérer l'ordre q tel que les r_k sont petits pour $k > q$. Nous tentons alors de modéliser la série par une moyenne mobile d'ordre q .

— Prévisions

Étant donné une suite d'observation X_1, \dots, X_n . Nous notons la prévision d'une observation future X_{n+k} dénotée par $\hat{X}_n(k)$. Pour effectuer une prévision, nous utilisons l'espérance conditionnelle comme valeur de $\hat{X}_n(k)$. Algébriquement, nous avons :

$$\hat{X}_n(k) = E(X_{n+k} | X_1, \dots, X_n).$$

Nous estimons les valeurs futures, mais il est important de les définir à un seuil de confiance, et dans nos standard, nous l'évaluons le plus souvent à 5%.

— Erreur de prévisions

L'erreur de prévisions mesure l'écart entre les valeurs réelles et les valeurs prédites

par le modèle. Elle est donnée par :

$$e_n(k) = X_{n+k} - \hat{X}_n(k).$$

L'intervalle de prévision de X_{n+k} au niveau de $1 - \alpha$ où α est choisi à 5% est définie par :

$$\left[\hat{X}_n(k) - 2\sigma(e_n(k)); \hat{X}_n(k) + 2\sigma(e_n(k)) \right].$$

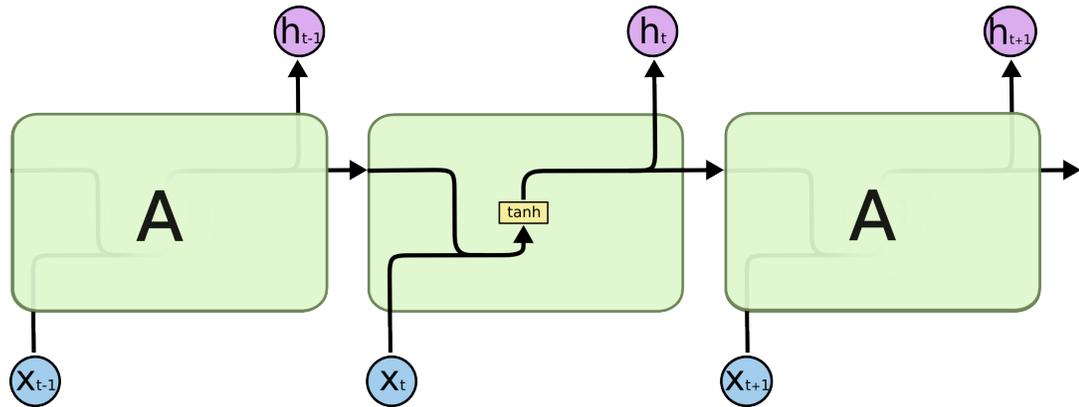
2.2 Mémoire à court et long terme (LSTM)

Dans cette partie, nous nous inspirons de l'article de Oinkina et Hakyll [19]. La mémoire à court et long terme (LSTM) est un modèle statistique permettant de faire des estimations sur des données chronologiques. Il est inventé en 1997 par Hochreiter et Schmidhuber et fonctionne sur une grande variété de problème. Le LSTM surpasse les limites du problème de dépendance à long terme. Il a comme capacité de se souvenir des informations pendant de longues périodes, et apprend facilement sur les jeux de données.

2.2.1 Réseaux de neurones

Tous les réseaux neuronaux récurrents (RNN)[19] ont la forme d'une chaîne de modules répétitifs de réseau de neurones. Dans nos RNN standards, ce module répétitif aura une structure très simple, telle qu'une seule couche tanh (Voir figure 2.1).

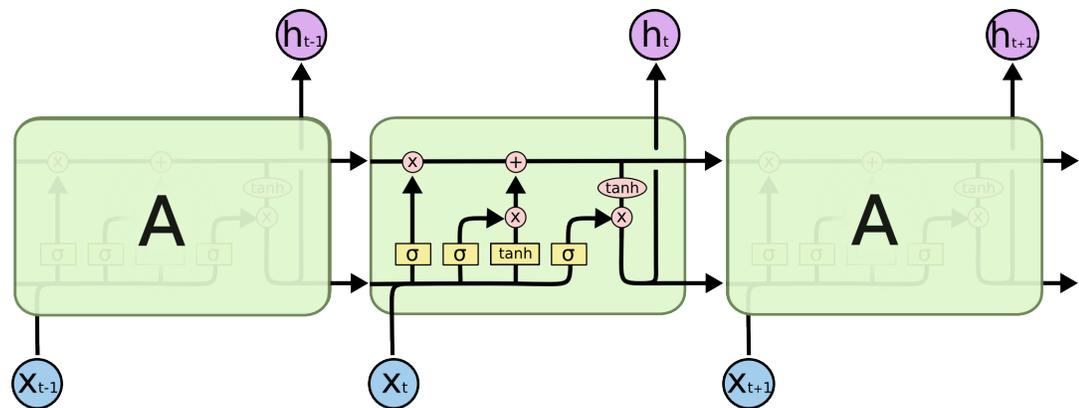
FIGURE 2.1 – Module répétitif dans un RNN standard contenant une seule couche.



Source : Oinkina et Hakyll [19]

Les LSTM ont également cette structure de chaîne, mais le module répétitif a une structure différente. Nous distinguons quatre couches de réseau de neurones qui se connectent les unes entre les autres, comme nous pouvons le remarquer sur la figure 2.2. En effet, il existe trois couches sigmoïdes et une couche \tanh qui se connectent les unes contre les autres.

FIGURE 2.2 – Module répétitif dans un LSTM contenant quatre couches en interaction.

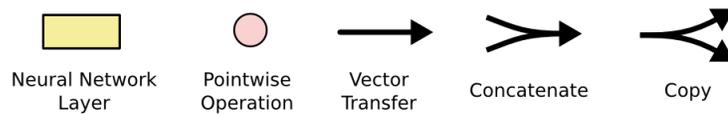


Source : Oinkina et Hakyll [19]

La figure 2.3 donne une représentation des certaines notations utilisées dans la structure du modèle LSTM. Chaque ligne porte un vecteur entier, de la sortie d'un

nœud aux entrées des autres. Les opérateurs $+$, et \times (cercles roses) représentent des opérations ponctuelles, comme l'addition ou la multiplication vectorielle, tandis que les cases jaunes sont des couches de réseau de neurones apprises (comme décrite dans les représentations précédentes). La fusion des lignes indique la concaténation, tandis qu'une séparation de lignes indique que son contenu est copié et que les copies vont à différents endroits.

FIGURE 2.3 – Notation des opérateurs et des couches

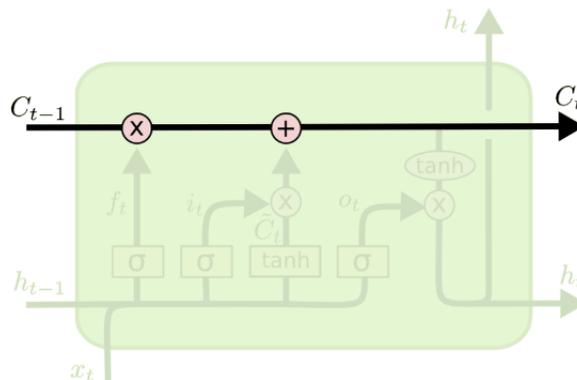


Source : Oinkina et Hakyll [19]

2.2.2 L'idée de base derrière les LSTM

L'idée principale du LSTM est l'état de la cellule C_{t-1} , C_t , décrit par la ligne horizontale de la figure 2.4. Il est comme une bande transporteuse et s'étend tout droit sur toute la chaîne, avec seulement quelques interactions linéaires mineures. Il est très facile pour l'information de s'écouler sans changement.

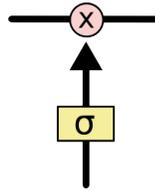
FIGURE 2.4 – Module répétitif dans un LSTM contenant quatre couches en interaction.



Source : Oinkina et Hakyll [19]

Le LSTM dispose des portes (voir figure 2.5) qui ont la capacité d'ajouter ou de supprimer des informations à l'état de la cellule. Ces portes sont composées de couche de filet neuronal sigmoïde et d'une opération de multiplication ponctuelle. La couche sigmoïde a des valeurs comprises entre zéro et un, décrivant la quantité de chaque composant à laisser passer l'information. Une valeur de zéro signifie « ne rien laisser passer », tandis qu'une valeur de un signifie « tout laisser passer ! ».

FIGURE 2.5 – État de la cellule LSTM.



Source : Oinkina et Hakyll [19]

Le LSTM dispose de trois portes pour rendre adaptable (permet de passer l'information facilement) et contrôler l'état de la cellule.

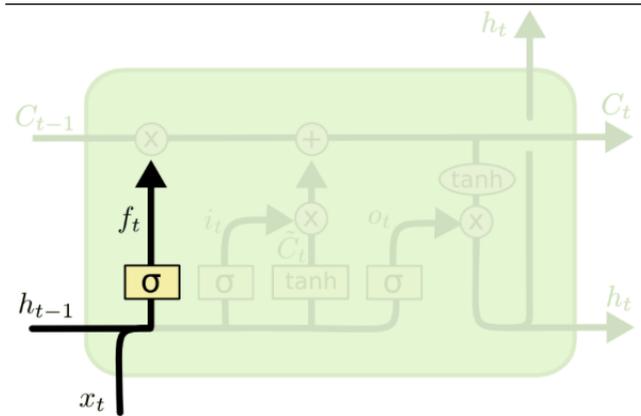
2.2.3 Les Étapes du modèle LSTM

La première étape du modèle LSTM est de décider les informations à entrer dans l'état de la cellule. Cette décision est prise par une couche sigmoïde appelée « couche de porte d'oubli ». Cette dernière utilise h_{t-1} et x_t , pour produire un nombre entre 0 et 1 pour chaque numéro de l'état de la cellule C_{t-1} . L'état 0 correspond à « garder complètement l'information » alors que l'état 1 représente « Se débarrasser totalement de l'information ». Cette étape est formalisée par l'équation suivante :

$$f_t = \sigma(w_f \times [h_{t-1}, x_t] + b_f). \quad (2.1)$$

Avec σ représentant la fonction sigmoïde, h_{t-1} la sortie du bloc LSTM précédent et b_f le biais pour la porte f , w_f le poids pour les neurones de la porte f , et x_t les données d'entrée.

FIGURE 2.6 – Étape 1 du LSTM.



Source : Oinkina et Hakyll [19]

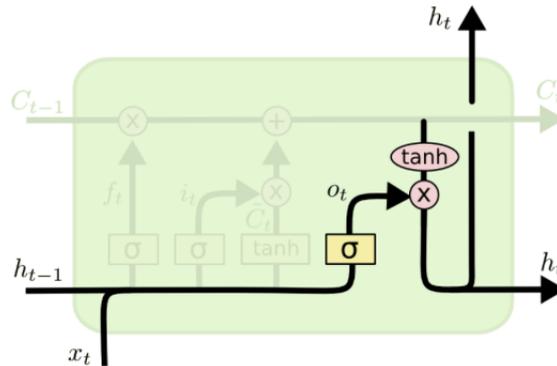
La deuxième étape consiste à décider quelles nouvelles informations nous allons stocker dans l'état cellulaire, et celle-ci se décline en deux grandes lignes. Premièrement, une couche sigmoïde appelée « couche de porte d'entrée » décide quelles valeurs nous allons mettre à jour. Ensuite, une couche tanh crée un vecteur de nouvelles valeurs candidates, \tilde{C}_t , qui pourrait être ajouté à l'état. À l'étape suivante, nous combinons ces équations (2.2) et (2.3) pour créer une mise à jour de l'état.

$$i_t = \sigma(w_i \times [h_{t-1}, x_t] + b_i); \quad (2.2)$$

$$\tilde{C}_t = \tanh(w_c \times [h_{t-1}, x_t] + b_c). \quad (2.3)$$

Avec i_t représentant la porte d'entrée, h_{t-1} la sortie du bloc LSTM précédent et b_x le biais pour la porte $x = i, c$, w_f le poids pour les neurones de la porte f , et x_t les données d'entrée.

FIGURE 2.7 – Étape 2 du LSTM.

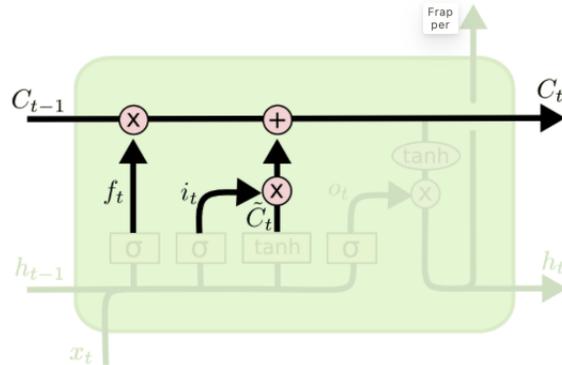


Source : Oinkina et Hakyll (2018) [19]

Dans la troisième étape, le modèle met à jour l'ancien état de la cellule, C_{t-1} , dans le nouvel état C_t . Pour ce faire, nous multiplions l'ancien état par f_t , en oubliant les informations moins utiles. On ajoute aussi $i_t \times \tilde{C}_t$. Il s'agit des nouvelles valeurs candidates, mises à l'échelle en fonction de la mesure dans laquelle nous avons décidé de mettre à jour chaque valeur d'état. Ainsi, nous formalisons algébriquement cette étape par :

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t. \quad (2.4)$$

FIGURE 2.8 – Étape 3 du LSTM.



Source : Oinkina et Hakyll [19]

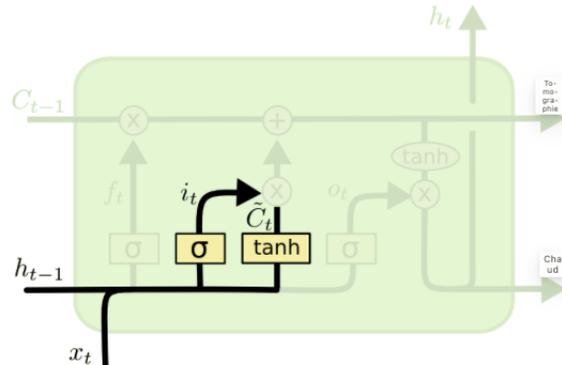
À l'étape 4, nous décidons de ce que nous allons produire. Cette sortie sera basée sur l'état de notre cellule, mais sera une version filtrée. Premièrement, nous exécutons une couche sigmoïde qui décide quelles parties de l'état de la cellule nous allons produire. Ensuite, nous avons fait passer l'état de la cellule \tanh (les valeurs comprises également entre $[-1, 1]$) et le multiplier par la sortie de la porte sigmoïde, de sorte que nous ne produisons que les parties que nous avons décidé de faire. Les équations (2.5) et (2.6) ci-après formalisent l'écriture mathématique de la quatrième étape (voir figure 2.9).

$$o_t = \tanh(w_o \times [h_{t-1}, x_t] + b_o); \quad (2.5)$$

$$h_t = o_t \times \tanh(C_t). \quad (2.6)$$

Avec o_t représentant la porte de sortie et h_t la sortie du bloc LSTM à l'instant t .

FIGURE 2.9 – Étape 4 du LSTM.



Source : Oinkina et Hakyll (2018) [19]

En résumé le LSTM fonctionne en oubliant les informations inutiles f_t , stocke la mémoire existante à travers le produit $\tilde{C} \times i_t$ afin de produire des sorties qui permettront d’avoir une qualité d’information et de prévision.

Les méthodes d’apprentissage sont abordées dans les parties suivantes car elles sont d’une importance capitale pour améliorer la capacité de convergence des modèles d’apprentissage profond.

2.2.4 Méthode d’apprentissage

Dans cette partie, nous nous sommes inspirés en grande partie des travaux de Francis Grégoire [20] pour expliquer les méthodes d’apprentissage de nos modèles. Nous y abordons les plus populaires et qui sont utilisées dans la suite à savoir l’algorithme de gradient et l’algorithme d’ADAM.

2.2.5 Algorithme du gradient

Comme algorithme d'optimisation, nous utilisons la descente de gradient afin d'estimer les paramètres θ minimisant la fonction coût nommée $J(\theta)$. La log-vraisemblance négative est la plus couramment utilisée comme fonction perte sur notre jeu de données d'entraînement qui représente 70% de notre quantité de données. L'algorithme du gradient est une méthode d'optimisation itérative qui modifie l'ensemble des paramètres successivement jusqu'à ce que la fonction de coût sur les données d'entraînement converge vers la solution optimale [20]. Nous optimisons pour que $J(\theta)$ se dirige vers un minimum local afin qu'à chaque pas de temps, nous avançons d'un pas de temps dans la direction opposée au gradient de la fonction coût.

$$\theta_{t+1} = \theta_t - \alpha \nabla J(\theta)$$

où α nommé taux d'apprentissage (*learning rate*), est un hyperparamètre crucial permettant de faire un bon apprentissage. L'ajustement du taux d'apprentissage α est très important, et ce, durant l'initialisation et l'optimisation des paramètres. En effet, si α est trop faible, nous concluons que la fonction coût converge lentement vers un minimum local. Aussi, si α est trop grand, il s'avère être difficile d'atteindre un minimum local et par conséquent l'optimisation risque de faire un pas de distance trop éloignée. Avec la croissance de la taille des ensembles de données d'entraînement utilisés pour faire l'apprentissage des modèles, il est devenu de plus en plus computationnellement et exigeant de calculer $\nabla J(\theta)$ lorsque nous utilisons un grand ensemble de jeu de données, et ceci, dans le but d'amoindrir le coût de calcul de chaque itération afin d'optimiser l'algorithme du gradient stochastique (*stochastic gradient descent*). Rumelhart et al.[21] ont introduit l'algorithme de rétropropagation du gradient qui est un moyen efficace de calculer le gradient des fonctions. La rétropropagation ajuste les paramètres d'un réseau de neurones afin de minimiser les erreurs de prévision des modèles d'apprentissage automatique.

2.2.6 Algorithme d'ADAM

Beaucoup d'améliorations ont été constatées avec les systèmes de gestion de données (SGD). La méthode d'ADAM est développée dans notre algorithme d'apprentissage à savoir le LSTM afin d'évaluer sa capacité à s'entraîner et à fournir de bon résultat de prévisions sur nos modèles. Nous nous inspirons des travaux de Kingma et Ba (2014) [22] sur le calcul des taux d'apprentissages adaptifs pour les paramètres estimés. La mise à jour des paramètres est donnée par les équations suivantes [20] :

$$\begin{aligned}
 g_t &= \nabla J\theta(t-1); \\
 m_t &= \beta_1 m_{t-1} + (1 - \beta_1)g_t; \\
 v_t &= \beta_2 v_{t-1} + (1 - \beta_1)g_t^2; \\
 \hat{m}_t &= \frac{m_t}{1 - \beta_1^t}; \\
 \hat{v}_t &= \frac{v_t}{1 - \beta_2^t}; \\
 \alpha_t &= \alpha \frac{\sqrt{(1 - \beta_2^t)}}{1 - \beta_1^t}; \\
 \theta_t &= \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \varepsilon}};
 \end{aligned}$$

où :

- α : le taux d'apprentissage initial,
- β_i , $i = 1, 2$: hyperparamètres appartenant à $[0, 1]$ contrôlant le taux de décroissance exponentielle des moyennes mobiles du gradient m_t et du carré du gradient v_t .

D'après Grégoire [20], ces moyennes mobiles sont des estimations de la moyenne et de la variance non centrée du gradient. Aussi, pour certains auteurs les hyperparamètres sont initialisés à : $\alpha = 0.0001$; $\beta_1 = 0.9$; $\beta_2 = 0.999$ et $\varepsilon = 10^{-8}$.

Dans ce mémoire, nous testons l'algorithme de gradient et d'ADAM pour la mise

en œuvre et l'apprentissage du modèle LSTM.

2.3 Choix du modèle ARIMA et LSTM

Dans la théorie de la modélisation de séries chronologiques univariées suivant la méthode de Box Jenkins, le modèle ARIMA est le modèle le plus généralement utilisé pour modéliser une variable Y_t . Les modèles AR, MA et ARMA sont des cas particuliers d'ARIMA avec respectivement les paramètres (p, d, q) qui sont égaux à $(p, 0, 0)$, $(0, 0, q)$ et $(p, 0, q)$. Il est alors judicieux, dans le cas d'une modélisation de série chronologique univariée, de partir sur un modèle ARIMA et déterminer les paramètres p , d et q qui nous conduiront à un type de processus qui fonctionne au mieux avec nos données. D'ailleurs dans la littérature, Abadi et Meslier [23] utilise le modèle ARIMA pour la prévision à court terme de l'énergie journalière produite par l'électricité de France. Il l'expérimente avec la méthode Box-Jenkins pour la description des principaux résultats. Également pour la prévision à court terme des accidents de la route, Ahmed et Abdelmounaime [24] utilisent des données de circulations, mensuellement, de 2010 à 2018 afin de faire des prévisions à partir de 2019. Les résultats obtenus par le modèle comparativement aux données du premier semestre de l'année 2019, montrent que les prévisions et les valeurs réelles sont en parfaite concordance. Ainsi, nous prenons ce modèle du fait de son approche à court terme et de sa capacité d'interprétabilité avec les séries temporelles univariées.

Concernant le modèle LSTM, il est le plus utilisé dans la littérature 1 pour l'estimation de la production des déchets. En effet, il a une mémoire à court et long terme, ce qui veut dire que si nous avons peu de données, le modèle a la capacité de mémorisation des informations à court et long terme afin de donner de bons résultats de prévision [25]. Parallèlement, il est un modèle d'apprentissage automatique, et l'enjeu qui se pose est de le comparer avec le modèle classique de base qui est dans notre

cas le modèle ARIMA, afin de voir lequel entre les deux modèles donnent de meilleurs résultats, et ce, en optimisant leurs paramètres, et en variant notre jeu de données.

2.3.1 Avantages des modèles ARIMA et LSTM

Le modèle LSTM et ARIMA sont deux structures utilisées pour la prévision des séries temporelles. Chacun d'eux dispose des avantages distincts. Le modèle ARIMA modélise des tendances temporelles en prenant en compte les composantes autorégressives (AR), intégrées (I) et moyennes mobiles (MA). Sa mise en œuvre ne nécessite pas la connaissance de la distribution des données. Ce qui le rend facile à utiliser face à certains types de bruit et de variations dans les données. Aussi, disposant de peu de données, il peut être utilisé pour la prévision à court et moyen terme, tenant compte du fait que la structure des données n'est pas complexe.

Quant au modèle LSTM, il a la capacité de capturer les dépendances à long et court terme des séries temporelles. Cela veut dire que lorsque nous disposons de peu de données, il permet de mémoriser les informations précédentes pour faire des prévisions pertinentes dans le temps. En même temps, il est approprié pour effectuer la modélisation des processus non linéaires et complexes dans les séquences de données. Aussi, il permet d'effectuer la modélisation multivariée et ce, en procédant par des processus d'apprentissage automatique. Le LSTM a une capacité de prévision à long terme du fait que leur mémoire permet de mémoriser des informations aussi bien sur de courtes périodes mais aussi sur des périodes de longues durées.

2.3.2 Limites des modèle ARIMA et LSTM

Tous les deux modèles présentent des limites. En effet, le modèle ARIMA suppose une relation linéaire entre les données passées et futures. De ce fait, il devient difficile à mettre en œuvre lorsque nous avons des phénomènes complexes. La détermination des ordres p (autorégressif), d (intégratif), q (moyenne mobile) peut nécessiter des connaissances approfondies et des essais erreurs. Ce qui peut être très difficile si nous voulons modéliser plusieurs séries de données avec cette approches. Le modèle ARIMA est moins performant ou ne donne pas de bon résultats lorsque nous avons des données irrégulières ou manquantes. Parallèlement, sur une très longue période, le modèle ARIMA peut être moins performant à faire des prévisions fiables à cause de leur capacité limitée à capturer des tendances complexes sur de grandes périodes.

Pour ce qui est du modèle LSTM, il est souvent complexe à mettre en œuvre et se surajuste souvent lorsque nous avons très peu de données. Pour des données petites ou limitées, des techniques de régularisation et d'ajustement des hyperparamètres sont faites par essais erreurs. Le temps de calcul du modèle LSTM peut être coûteux si nous disposons de grandes séquences temporelles ou avec des réseaux profonds. Son interprétabilité n'est pas simple car il se présente sous forme de boîte noire en raison de sa complexité. Aussi, en raison de la disparition du gradient, il peut arriver au modèle LSTM d'avoir des difficultés pour modéliser des séquences temporelles sur une périodes de temps très longues.

2.4 Méthode expérimentale

Nous procédons par plusieurs tests de différents types de modèles en faisant varier les paramètres. Pour cela, nous nous inspirons de la technique de validations croisées

souvent utilisées pour évaluer la capacité de prévision des modèles. En effet, nous faisons varier la taille de notre jeu de données en retirant un certain nombre d'années, selon le modèle, et en effectuant une estimation du modèle avec le restant des données pour ensuite calculer l'erreur de prédiction sur les lignes de données qui ont été retirées et les critères de performance du modèle. Ainsi, nous obtenons un ensemble de 16 modèles dont 8 de types ARIMA et 8 autres de type LSTM. La différence entre ceux-ci, comme mentionnée, reste les jeux de données sur lesquels ils ont été entraînés et, par ricochet, les paramètres des modèles qui dépendent des données d'entraînement. Ensuite, nous définissons quatre types d'indicateurs d'évaluation statistique à savoir le critère d'information d'Akaike (AIC), le critère d'information bayésien (BIC), l'erreur quadratique moyenne (RMSE) et le coefficient de détermination (R^2) afin d'évaluer les performances des modèles pour une meilleure comparaison entre eux mais également pour faire le choix du meilleur modèle.

2.4.1 Vraisemblance (L)

La vraisemblance (L) d'un modèle mesure l'ajustement des données vis-à-vis au modèle. Autrement dit, elle quantifie la quantité d'information apportée par le modèle. La fonction de vraisemblance est définie par $L_X(\theta) = f_X(\theta)$. Disposant d'un échantillon (X_1, \dots, X_n) , la vraisemblance est calculée par :

$$L_{X_1, \dots, X_n}(\theta) = \prod_{i=1}^n f_{\theta}(X_i)$$

avec θ représentant le vecteur des paramètres du modèle, $f_{\theta}(X_i)$ la fonction de densité de probabilité (ou de masse de probabilité) sous le modèle avec les paramètres θ .

2.4.2 Critère d'information d'Akaike (*AIC*)

Le critère d'information d'Akaike noté *AIC* évalue la quantité d'information que le modèle restitue en termes de vraisemblance. Plus l'*AIC* est faible, plus le modèle apporte une quantité d'information plus importante. Il est calculé par la formule suivante :

$$AIC = -2 \frac{\log(L)}{n} + 2 \frac{p}{n}$$

avec p le nombre de variable explicative, n la taille de l'échantillon, L la vraisemblance du modèle.

2.4.3 Critère d'information bayésien (*BIC*)

Le critère d'information bayésien noté *BIC* est un indicateur d'évaluation d'un modèle et s'interprète analogiquement que l'*AIC*. Il est calculé par la formule suivante :

$$BIC = -2 \frac{\log(L)}{n} + \frac{p \log(n)}{n}$$

avec p le nombre de variable explicative, n la taille de l'échantillon, L la vraisemblance du modèle.

2.4.4 Erreur quadratique moyenne (*RMSE*)

L'erreur quadratique moyenne est un critère d'évaluation d'un modèle nommé *RMSE* encore appelé la racine de l'erreur moyenne quadratique mesure l'écart entre les valeurs réelles et les valeurs estimées par le modèle. Plus le *RMSE* est faible et proche de 0, plus le modèle est performant et s'ajuste bien avec les données d'étude.

Cet indicateur est calculé en utilisant la formule suivante :

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

avec y_i et \hat{y}_i représentant respectivement les valeurs réelles et prédites par le modèle pour l'individu i et n le nombre d'individus utilisés pour calculer le RMSE.

2.4.5 Coefficient de détermination (R^2)

Le coefficient de détermination donne l'ajustement d'un modèle par rapport aux données observées. Il est souvent nommé par R^2 dans un modèle statistique et mesure la proportion de la variance dans la variable dépendante qui est prédictible à partir des variables indépendantes. Il est calculé par la formule suivante :

$$R^2 = 1 - \frac{Var(\text{résidus})}{Var(\text{dépendante})}$$

avec $Var(\text{résidus})$ la variance des résidus du modèle, c'est-à-dire la variance des différences entre les valeurs observées et les valeurs prédites par le modèle et $Var(\text{dépendante})$ la variance totale de la variable dépendante.

R^2 est compris entre 0 et 1 et plus il est proche de 1, meilleur le modèle est.

Chapitre 3

Résultats des modèles ARIMA et LSTM

Ce chapitre s'ouvre par la familiarisation avec les données afin de comprendre leur structure et d'effectuer les traitements appropriés avant la modélisation. Nous présentons et discutons, dans la suite, les résultats des modèles ARIMA et LSTM. En dernière partie, nous analysons les caractéristiques et les performances des modèles à faire de bonnes prévisions dans le temps.

3.1 Familiarisation avec les données

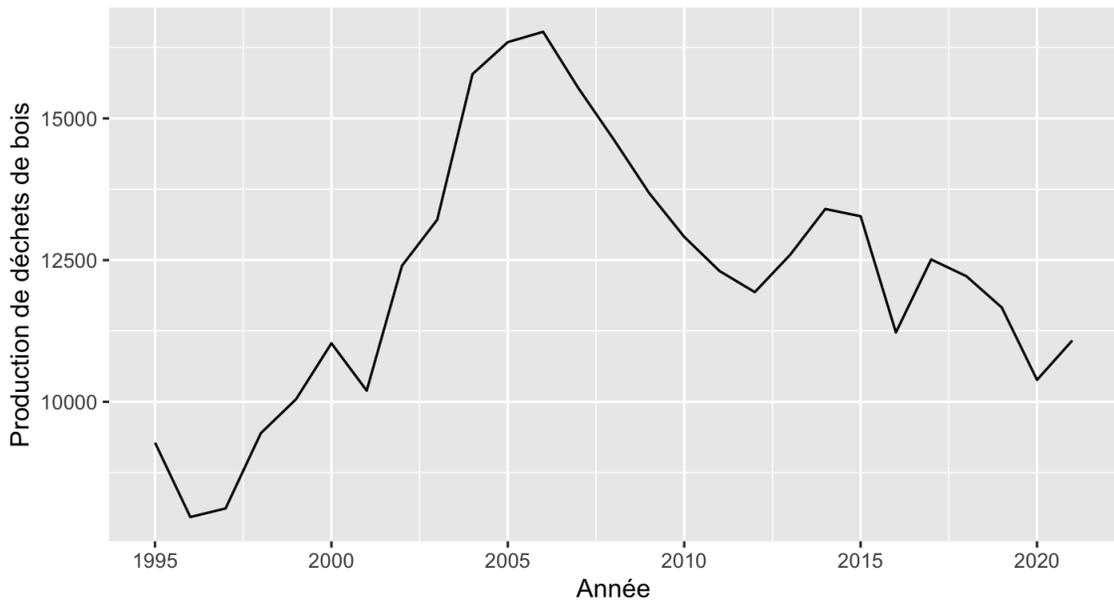
Dans cette partie, nous présentons les données dont on dispose. Nous faisons aussi quelques vérifications sur nos données ainsi que quelques transformations sur la série brute afin d'avoir une série mieux adaptée à expérimenter.

3.1.1 Source de données

Nous travaillons avec les données de séries chronologiques dans l'industrie de la construction, de la rénovation et de la démolition (CRD) de 1995 à 2021. Ces données, exprimées en kilotonnes, proviennent de Statistique Canada et sont issues du programme d'enquêtes statistiques sur la consommation industrielle d'énergie. La date de publication des données actualisées est le 06 décembre 2022.

Ainsi, la base de données contient 27 observations sur la quantité de déchets de bois solides en tonnes répertoriées sur les années 1995 à 2021. la figure 3.1 donne une visualisation des données de production de bois de 1995 à 2021.

FIGURE 3.1 – Représentation des déchets de bois solide durant la période de 1995 à 2021



Source : Base Statistique Canada, calcul de l'auteur

Après observation de la série (figure 3.1), nous remarquons qu'il y a une tendance entre les années 2002 à 2012. Elle est due à une forte augmentation de la production

des déchets de bois (tendance à la hausse). De ce fait, nous procédons aux traitements sur les données afin d'avoir une série plus homogène.

3.1.2 Traitements sur les données

Pour le traitement des données, nous faisons des vérifications sur les données pour savoir s'il y a des données aberrantes et aussi faire quelques transformations pour obtenir une série éliminée de la tendance, non saisonnière et stationnaire.

— Recherche de valeurs aberrantes

Nous procédons par une étape de vérification de valeurs aberrantes à partir d'un boxplot. En effet, nous disons qu'il y a des valeurs aberrantes (*outliers*) si l'on observe des valeurs supérieures à la moustache gauche ou à la moustache droite. Les moustaches gauches et droites ont été calculées en utilisant respectivement les équations (7) et (8) :

$$MG = Q_{25\%} - 1,5(Q_{75\%} - Q_{25\%}), \quad (3.1)$$

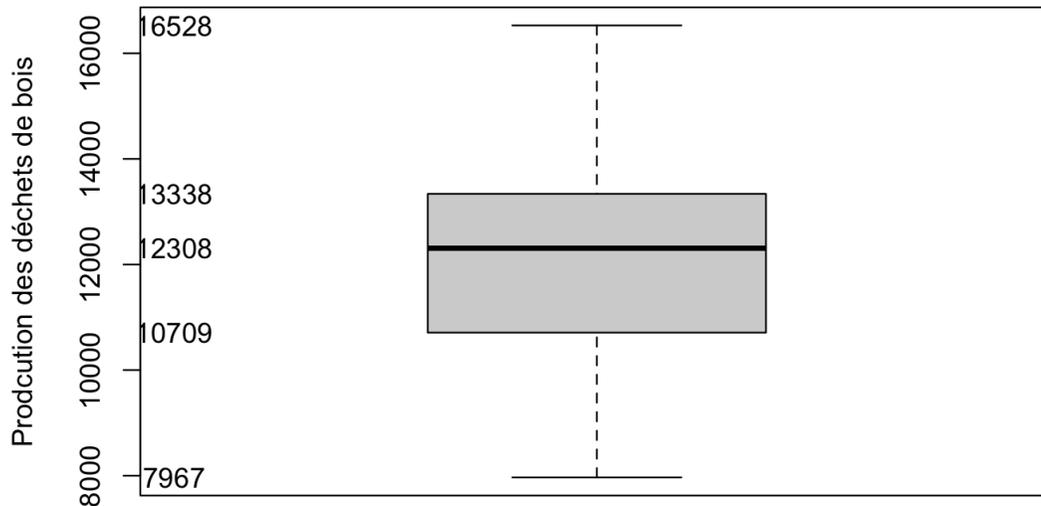
$$MD = Q_{75\%} + 1,5(Q_{75\%} - Q_{25\%}), \quad (3.2)$$

où :

- Q_i est le quantile à $i\%$,
- MG représentant la moustache gauche,
- MD représentant la moustache droite.

Selon la figure 3.2 ci-après, et au vu de la définition d'une observation aberrante, nous pouvons constater que la série ne présente pas de valeurs aberrantes

FIGURE 3.2 – Boxplot de la série de la base de données des déchets de bois solides



— Quelques transformations

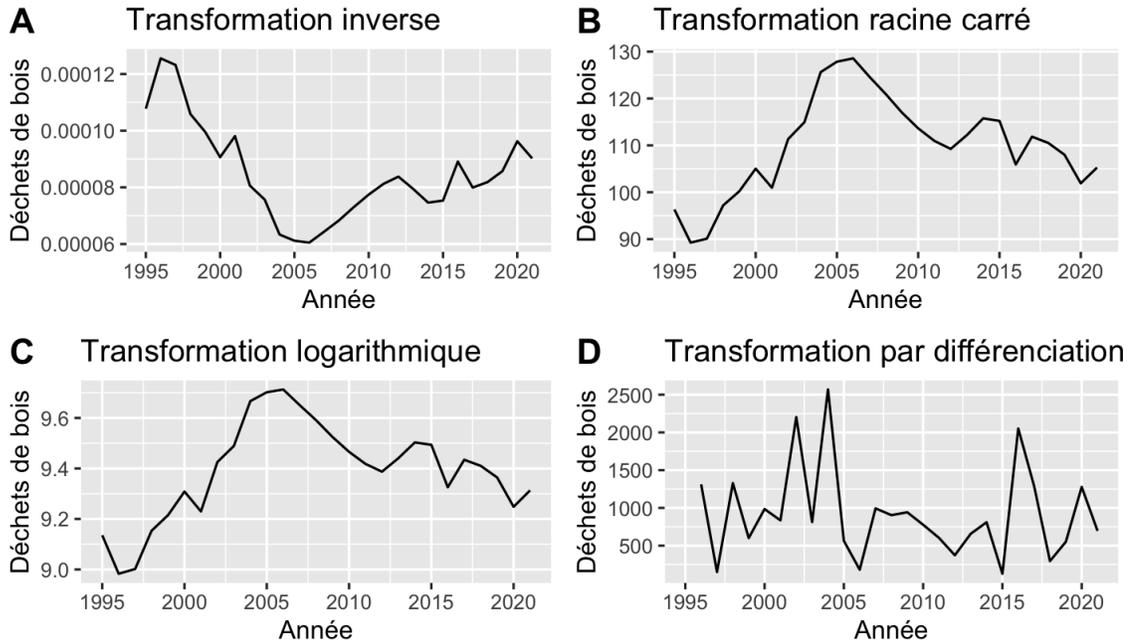
Au vu des données dont on dispose (peu de données), nous travaillons au seuil de significativité de 10% pour ne pas rendre les résultats des tests très stricts.

Il y a lieu de préciser que des tests primaires de saisonnalité et de stationnarité ont été effectués sur la série brute. Les résultats de ces tests, précisément de Kwiatkowski-Phillips-Schmidt-Shin (KPSS) indiquent une p-valeur égale à 0,092. Ce qui mène à la conclusion selon laquelle la série n'est pas stationnaire au seuil de 10%. Cela nous amène alors à effectuer une transformation sur notre série brute.

Dans ces conditions, il est commode de faire des transformations sur les

séries chronologiques pour enlever la tendance et la saisonnalité afin de mieux modéliser la série étudiée. Mesfioui [26] donne quelques transformations parmi lesquelles la transformation logarithmique, racine carrée, inverse, et la série différenciée 1 fois. Ainsi, nous représentons toutes ces transformations pour voir celle qui a un meilleur comportement.

FIGURE 3.3 – Quelques transformations sur la série brute



Source : Base Statistique Canada [27], calcul de l'auteur

TABLE 3.1 – Tests de stationnarité sur les séries transformées

Test de KPSS	Hypothèse nulle H0	P-value
Transformation inverse	Stationnarité	0,05
Transformation racine carré	Stationnarité	0,08
Transformation logarithmique	Stationnarité	0,07
Transformation par différenciation	Stationnarité	0,09

À partir des transformations effectuées, nous constatons que les p-valeurs des séries sont inférieures à 0,1 donc reste toujours non significatives au seuil de 10%. Ce qui conduit à rejeter l'hypothèse nulle de la stationnarité de la série.

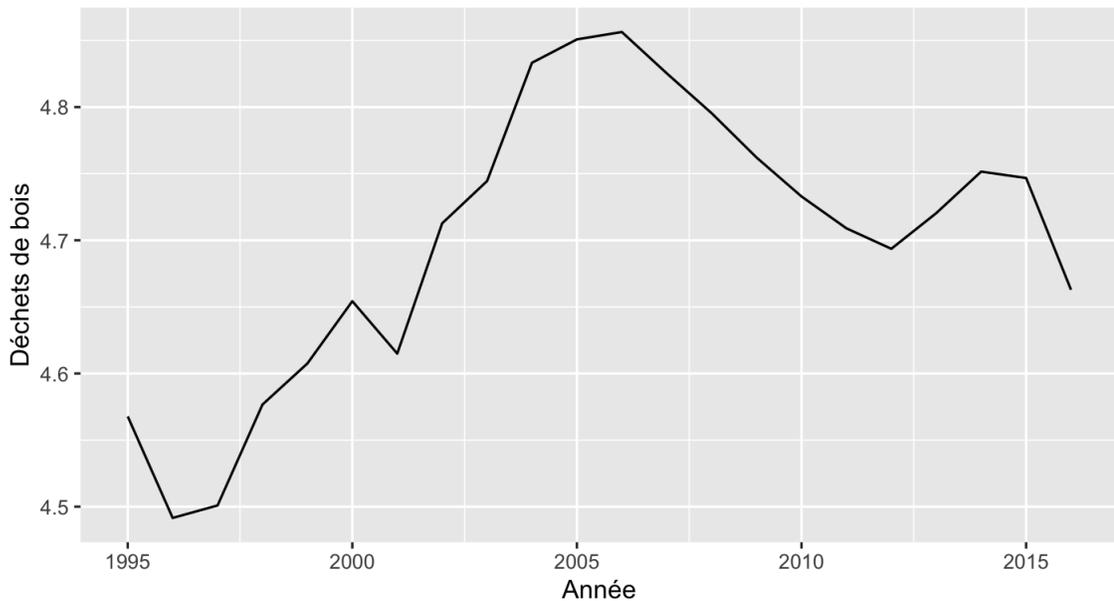
À cet égard, nous avons fait une transformation combinée de la série logarithmique et racine carré de la forme :

$$Y_t = \log(\sqrt{X_t}).$$

Y_t est la série transformée et X_t la série initiale avant traitement.

FIGURE 3.4 – Représentation de la série corrigée (« Y_t »)

Série transformée par le logarithme et la racine carré



Après transformation, nous voyons que la série corrigée est plus homogène et ne présente pas une tendance particulière ou un effet saisonnier (voir figure 3.4). En conséquence, nous travaillerons avec cette série afin de déterminer le modèle adapté à nos données dans le sens de faire une bonne prévision sur les déchets de bois à des fins de valorisation énergétique.

Ainsi, nous concluons que la série n'est pas saisonnière.

En se basant sur la p-valeur, nous pouvons s'apercevoir qu'elle est égale à

0,1, donc supérieure ou égale au seuil de 10%. Ce qui permet de dire qu'on ne peut pas rejeter l'hypothèse nulle qu'il y a stationnarité dans notre série corrigée. Ainsi, nous concluons à partir des résultats du test de KPSS que la série transformée par la combinaison du logarithme et de la racine carré est stationnaire. Au vu des données dont on dispose et des résultats du test de KPSS, nous formulons comme hypothèses que cette transformation est corrigée de la tendance, de la saisonnalité, et est stationnaire. À cet effet, pour récupérer les nouvelles valeurs prédites et voir si nos prévisions sont satisfaisantes, nous avons travaillé avec les données de 1995 à 2018, et faire des prévisions sur les valeurs à partir de 2019, et ce, sur un horizon temporel de 6 ans (entre 2019-2025).

3.2 Résultat des modèles ARIMA : Approche Box-Jenkins

Pour un modèle de séries chronologiques, les étapes phares de la validation passent par l'élimination de la tendance, la correction de la saisonnalité et de la stationnarité afin d'avoir une bonne estimation des paramètres et de pouvoir faire une bonne prévision. Nous adaptons l'approche Box-Jenkins qui est une technique d'identification, d'estimation et de prévision des moyennes mobiles intégrées autorégressives (ARIMA). Le choix du modèle ARIMA est établi par l'approche Box-Jenkins. Cette dernière propose 6 étapes pour une bonne estimation des données temporelles, telles que :

- la familiarisation avec les données,
- l'analyse préliminaire,
- l'identification du modèle ou la spécification du modèle,
- l'estimation des paramètres,
- la validation par tests ou le test d'adéquation du modèle,

— et enfin, la prévision.

Ces différentes étapes, à l'exception des deux premières qui ont été déjà effectuées dans la section 3.1, sont décrites dans la suite des sections pour diagnostiquer et optimiser le modèle choisi.

3.2.1 Spécification des modèles

Nous nous référons à la fonction d'autocorrélation (ACF) et à la fonction d'autocorrélation partielle (PACF) pour déterminer les ordres p et q des $ARIMA(p, d, q)$. En effet, les séries n'étant pas dessaisonnalisées ne vont pas comporter un terme d , donc d est égal à 0. Ainsi, les modèles finaux que nous allons obtenir vont être des $ARIMA(p, 0, q)$ ou $ARMA(p, q)$. Il s'agira seulement d'identifier les différents modèles $ARIMA(p, 0, q)$, c'est-à-dire de trouver les paramètres p et q raisonnables, ce qui revient à estimer les modèles correspondant et à les valider en examinant les résidus (i.e vérifier si les résidus sont des bruits blancs). De ces graphiques, nous déterminons les paramètres p et q des différents modèles à partir du nombre d'ordre de retard (*lag*) où l'ACF et le PACF croisent la zone de signification. Le tableau 3.2 suivant donne les différents ordres de retard p et q selon la variation des jeux de données.

TABLE 3.2 – Récapitulatif des différents modèles testés et leurs pouvoirs explicatifs

Jeux de données	ordre de retard p	ordre de retard q
19 observations (J(1))	4	1
20 observations (J(2))	3	1
21 observations (J(3))	3	2
22 observations (J(4))	2	1
23 observations (J(5))	3	1
24 observations (J(6))	4	1
25 observations (J(7))	4	1
26 observations (J(8))	3	1

3.2.2 Estimation des paramètres

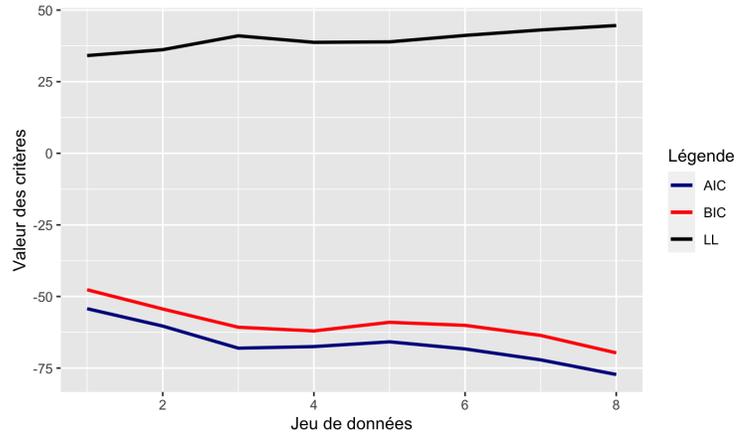
À présent, nous procédons à l'estimation des paramètres des différents modèles de chaque jeu de données nommés $J(i)$ avec i indice des différents jeux de données ($i = 1, \dots, 8$) selon leurs observations. Les estimations sont basées sur la méthode du maximum de vraisemblance. L'intérêt de cette méthode est qu'elle utilise la totalité de l'information sur le comportement des séries. Ainsi, plusieurs modèles ont été testés malgré la conclusion effectuée dans la spécification du modèle. L'objectif de ceci est de voir lequel de ces modèles obtient le plus grand pouvoir explicatif et s'ajuste au mieux sur les données. Les résultats ont ainsi montré, comme indiqué dans le tableau 3.3 ci-après, que le modèle $ARIMA(3, 0, 1)$ du jeu de données $J(8)$ est celui avec le plus petit AIC (critère d'information d'Akaike). Or plus l'AIC est petit, plus le modèle est performant. Ce qui suggère le choix de ce modèle. Cependant, ce choix ne peut être définitif qu'après la phase de validation des différents modèles.

TABLE 3.3 – Récapitulatif des différents modèles testés et leurs pouvoirs explicatifs

Jeux de données	Modèle	AIC	BIC	LL	RMSE
J(1)	ARIMA(4,0,1)	-54,25	-47,64	34,12	0,046
J(2)	ARIMA(3,0,1)	-60,33	-54,36	36,17	0,048
J(3)	ARIMA(3,0,2)	-68,04	-60,73	41,02	0,027
J(4)	ARIMA(2,0,1)	-67,49	-62,03	38,74	0,027
J(5)	ARIMA(3,0,1)	-65,82	-59,01	38,91	0,033
J(6)	ARIMA(4,0,1)	-68,31	-60,07	41,16	0,037
J(7)	ARIMA(4,0,1)	-72,12	-63,58	43,06	0,026
J(8)	ARIMA(3,0,1)	-77,24	-69,68	44,62	0,014

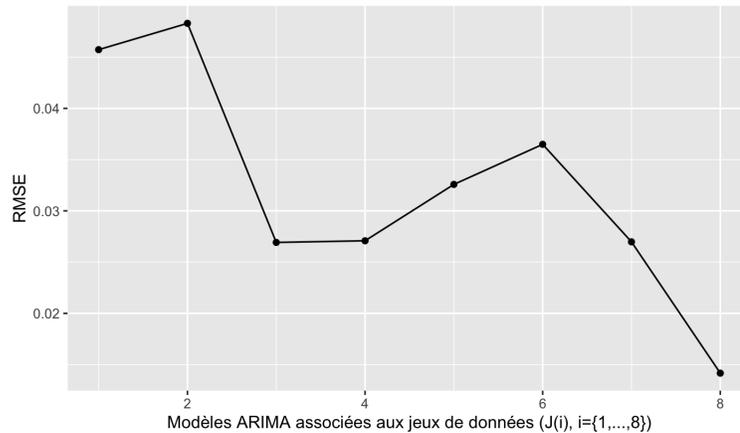
La figure 3.5 donne une meilleure visualisation de la variation des valeurs des critères AIC, BIC et de la vraisemblance. De plus, le jeu de données $J(8)$ associé au modèle $ARIMA(3, 0, 1)$ obtient l'AIC et le BIC le plus faible, et une vraisemblance(LL) plus élevée en comparaison avec les autres jeux de données.

FIGURE 3.5 – Représentation AIC, BIC et de la vraisemblance



La figure 3.6 indique que le RMSE a tendance à diminuer, suite à l'ajout d'une observation sur le jeu de données de départ nommé $J(1)$. Le modèle $ARIMA(3, 0, 1)$ associé au jeu de données $J(8)$ obtient un RMSE plus faible (0,0142) comparativement aux autres jeux de données.

FIGURE 3.6 – Représentation des RMSE des modèles ARIMA associés aux jeux de données



3.2.3 Validation par tests

Pour la validation, nous allons procéder d'abord par faire un test de nullité des paramètres et finir par un test de non autocorrélation des erreurs afin de statuer sur le modèle à retenir.

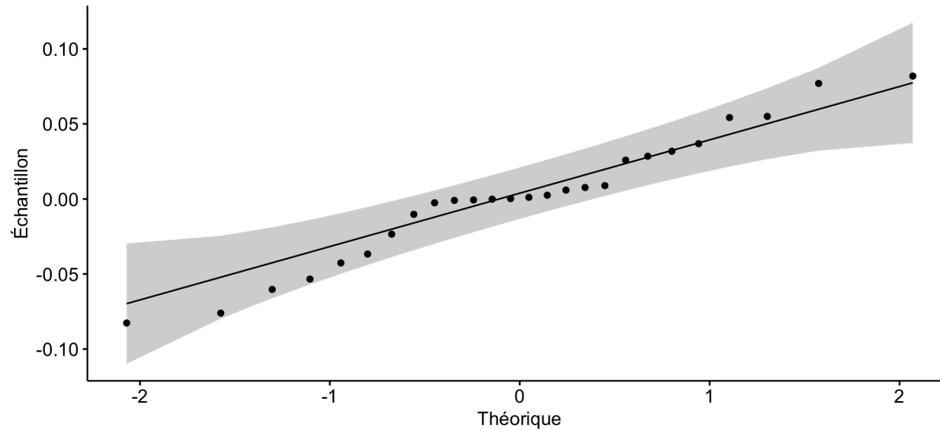
— **Test de nullité des paramètres**

Nous comparons si la valeur absolue du paramètre est supérieure à $1,96 \times \text{écart-type}$, auquel cas nous rejetons l'hypothèse de nullité du paramètre en question. Le modèle $ARIMA(3, 0, 1)$ a plus de coefficients significatifs (*True*) que les autres modèles. Vu qu'il a plus de paramètres non nuls et un AIC plus faible alors, il est retenu pour la validation dans la suite.

— **Test de non autocorrélation des erreurs**

Nous faisons le test de non autocorrélation des erreurs avec notre modèle $ARIMA(3, 0, 1)$ à l'aide du logiciel **R** en utilisant la fonction « `Box.test()` » qui est incluse dans la base du logiciel **R**. L'évaluation du test est faite par la p-valeur. En effet, si la p-valeur est inférieure à un certain seuil de signification, en général 5%, alors le modèle ARIMA retenu ne sera pas approprié pour effectuer de bonnes prévisions. Ainsi, l'idéal serait d'avoir une p-valeur supérieure à 5% pour que le modèle soit approprié. Après le test sur le logiciel **R**, nous obtenons une p-valeur de 0,06824 et qui est supérieure à 5% d'où la pertinence du modèle $ARIMA(3, 0, 1)$.

FIGURE 3.7 – Graphe quantile-quantile (QQ-Plot) pour test de normalité des résidus avec la statistique de Ljung-box



Au niveau de la figure 3.7, nous représentons respectivement en abscisse et en ordonnées, les quantiles théoriques de la loi normale et celles de notre échantillon afin de comparer les deux distributions. Nous pouvons nous apercevoir que le nuage de point s’ajuste assez bien sur la droite, ce qui permet de conclure que les résidus suivent une distribution normale. Ceci atteste que notre modèle $ARIMA(3,0,1)$ s’adapte bien aux données d’étude et est adéquat pour une effectuer de bonne prévision sur nos données.

Le modèle retenu se formalise comme suit :

$$X_t = \mu + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \phi_3 X_{t-3} + \varepsilon_1 X_{t-1}.$$

Les paramètres du modèle sont décrites dans le tableau 3.4 ci-dessous.

TABLE 3.4 – Valeurs estimées des paramètres

Paramètres	ϕ_1	ϕ_2	ϕ_3	ε_1	μ
Valeurs estimées	0,613	0,524	-0,423	0,48	4,689

Au terme de ces analyses, le modèle ARIMA(3,0,1) a un σ^2 estimé égal à 0,0017. Elle représente la variance des erreurs résiduelles non expliquées par le modèle. Cette valeur de σ^2 , quasiment nulle, indique la qualité du modèle à s'ajuster sur les données et à faire de bonne prévision dans le temps. En prenant aussi comme mesure de performance l'erreur quadratique moyenne (RMSE), le modèle ARIMA(3,0,1) a un RMSE égal à 0,0141. Ce qui prouve que l'erreur quadratique moyenne entre les valeurs prédites par le modèle et les valeurs de la série est très faible. D'où, la performance du modèle à faire de bonne prévision dans le temps.

3.2.4 Prévision

La forme canonique d'un modèle ARIMA peut s'écrire comme suit :

$$\Phi(B)X_t = \Theta(B)\varepsilon_t.$$

Avec cette formulation, on peut prédire la valeur de X_t à l'horizon h par la formule suivante :

$$\hat{X}_{t+h} = E(X_{t+h}|X_{t-1}, X_{t-2}, \dots).$$

Nous avons utilisé la sortie des résultats du logiciel **R** pour trouver ces valeurs futures. La figure 3.8 en est une illustration car représentant un ensemble de valeurs prévues pour les 5 prochaines années, à un seuil de 5%.

Dans le tableau 3.5, nous observons les valeurs prédites avant transformations de la série, et ceci, pour constater la quantité de déchets à des fins de valorisation énergétique et leur ajustement par rapport aux données réelles.

Les valeurs réelles ont été reconstituées en faisant une transformation de la forme :

$$X_t = (\exp \{Y_t\})^2$$

FIGURE 3.8 – Préviation des déchets de bois pour la production d'énergie

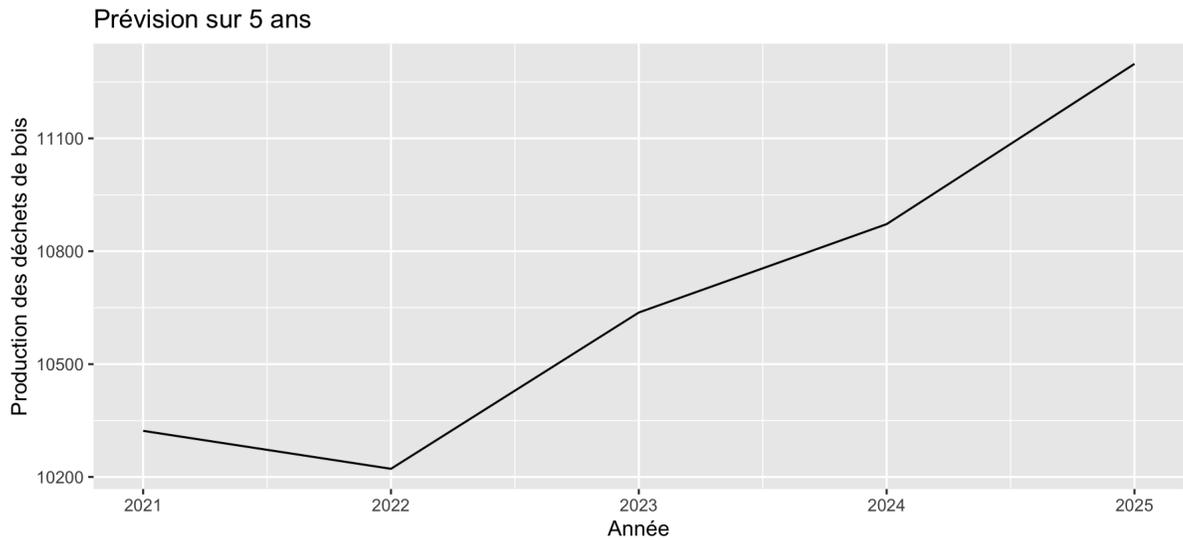


TABLE 3.5 – Préviation de la production des déchets sur un horizon de 5 ans

Période	Valeurs prédites	Valeurs réelles
2021	4,621052	4,527902
2022	4,616135	–
2023	4,636055	–
2024	4,646979	–
2025	4,666194	–

$$X_t = \exp \{2Y_t\}$$

Cette transformation nous permet de capter les valeurs prédites comparativement aux valeurs réelles. Tout bien considéré, nous remarquons que les valeurs prédites sont proches des valeurs réelles. Ceci est un fait qui prouve que notre modèle $ARIMA(3, 0, 1)$ est judicieux et adapté à nos données d'étude.

Le tableau 3.6 en est une parfaite illustration de l'évolution de la série prédite, et montre que les prévisions sont à la hausse, et ce, à partir de 2023.

FIGURE 3.9 – Valeurs prédites comparativement aux valeurs réelles

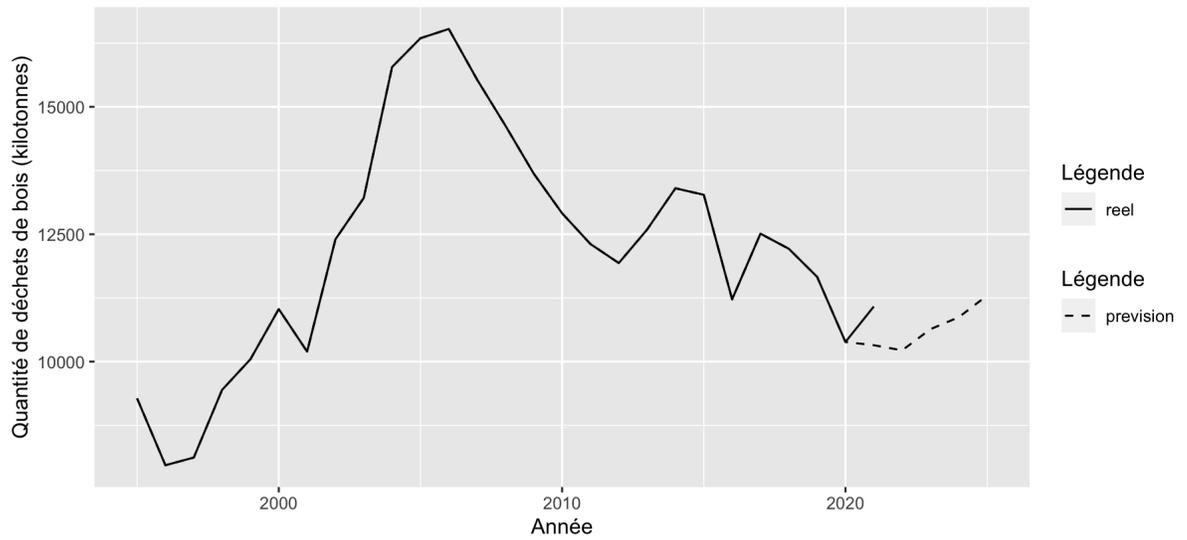


TABLE 3.6 – Prévisions et valeurs réelles

Période	Valeurs prédites	Valeurs réelles
2021	10322	11082
2022	10221	–
2023	10637	–
2024	10872	–
2025	11298	–

3.3 Résultats des modèles LSTM

Les étapes de la modélisation du LSTM se formalisent comme suit. D’abord, nous chargeons les données pour la phase de préparation et de prétraitement. De ce fait, nous divisons les données en entraînement et test. En effet, nous répartissons les données en 8 sous bases pour effectuer 8 modèles. Ensuite, nous normalisons les données d’entrées en créant des séquences d’entrée sur les données d’entraînement et les données test.

Après cela, il s’ensuit la phase de construction des modèles LSTM. Nous faisons

TABLE 3.7 – Récapitulatif des différents modèles testés et leurs RMSE

Jeux de données	RMSE
19 observations (J(1))	1257
20 observations (J(2))	1206
21 observations (J(3))	1349
22 observations (J(4))	1111
23 observations (J(5))	1138
24 observations (J(6))	1521
25 observations (J(7))	1373
26 observations (J(8))	483

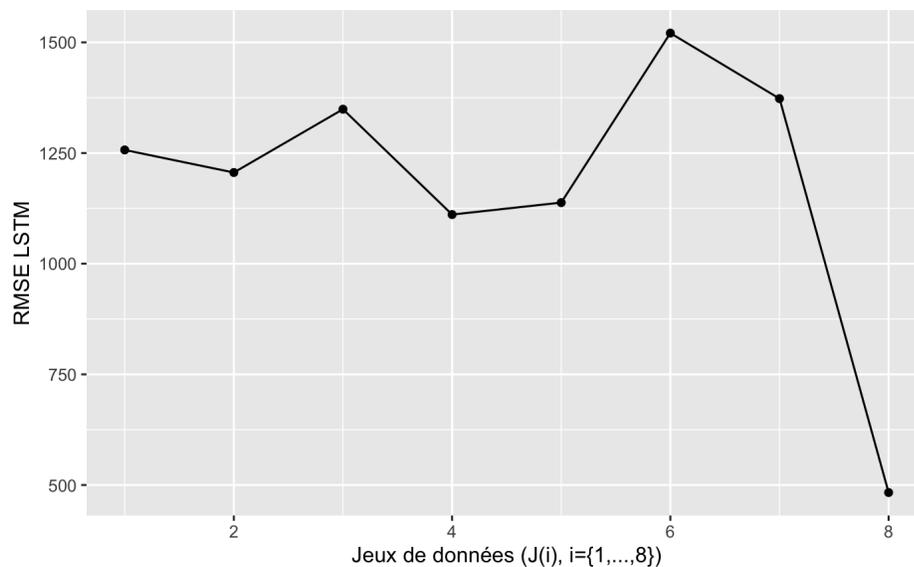
le choix de modèles LSTM à 3 couches du fait que nous disposons de peu de données et nous testons plusieurs modèles simples par souci d’adaptabilité à nos données. En procédant par essai-erreur, nous mettons en place une spécification de la forme de l’entrée du modèle d’un tenseur 3D¹ avec une taille de *batch*² de 20 (20 exemples de séquence), une longueur de séquence de 1 (chaque exemple de séquence est une valeur numérique) et une seule caractéristique par valeur numérique (il n’y a qu’une seule variable d’entrée). Pour chaque modèle, nous utilisons une couche dense à 1 neurone et la fonction d’unité linéaire rectifiée, comme fonction d’activation. L’argument 1 signifie que la couche dense a une seule sortie. La fonction de perte (*RMSE*) et l’optimiseur Adam [28] sont les paramètres à utiliser pour l’entraînement du modèle. Enfin, nous terminons par faire une évaluation des modèles sur leurs données test, pour voir si l’erreur de prévision est moins grande afin de faire la validation du modèle retenu. En effet, le modèle associé au jeu de données *J(8)* donne de meilleur résultat et ces prévisions sont assez proches des valeurs réelles avec un *RMSE* égal à 483. Cette valeur, étant relativement petite comparée à la grandeur d’échelle de notre variable réponse, atteste la précision du modèle à faire de bonnes prévisions.

1. Tenseur 3D : objet mathématique tridimensionnel qui généralise le concept de vecteur et de matrice

2. hyperparamètre utilisé dans les méthodes d’optimisation par descente de gradient stochastique et définissant le nombre d’échantillons d’apprentissage utilisés dans une itération d’entraînement

À partir du tableau 3.7, nous remarquons que le RMSE fluctue relativement lorsque notre jeu de données augmente. Le RMSE enregistre sa valeur la plus faible au jeu de données $J(8)$. En effet, ce jeu de données a plus d'observations, mais aussi il faudrait remarquer qu'entre deux jeux de données successifs $J(i)$ et $J(i + 1)$, le RMSE a presque tendance à diminuer suite à l'ajout d'une observation. Ceci est constaté lors du passage de $J(1)$ à $J(2)$, de $J(3)$ à $J(4)$, de $J(6)$ à $J(7)$, et de $J(7)$ à $J(8)$. Le graphique 3.10 apporte plus de clarté sur le comportement de la série.

FIGURE 3.10 – Valeurs prédites comparativement aux valeurs réelles



Nous constatons que les deux modèles $ARIMA(3, 0, 1)$ et LSTM à 3 couches sont plus performants lorsque nous passons à un jeu de données comportant plus d'observations. Suivant les métriques utilisées (RMSE et AIC), nous concluons que le modèle $ARIMA(3, 0, 1)$ est choisi pour la prévision des déchets solides de bois. Il est plus performant car ayant un AIC faible ainsi qu'un RMSE relativement égal à 0. Somme toute, les prévisions des modèles $ARIMA(3, 0, 1)$ et LSTM à 3 couches sont à la hausse, et ce, pour un horizon temporel de 6 ans. Pour ce faire, les politiques économiques à mettre en place se baseront sur ces différents résultats, mais aussi et surtout, sur les politiques envisageables et en lien avec les perspectives d'utilisation

des déchets solides à bon escient.

3.4 Recommandations

À l'issue de nos résultats de prévision, nous voyons que les deux modèles ont quasiment les mêmes performances de prévisions et deviennent plus performants lorsque nous augmentons la taille de notre base de données d'étude. Parallèlement, la quantité de déchets prédite augmente progressivement sur un horizon de 5 ans. Ceci nous amène à penser à des actions politiques et/ou des recommandations, et celles-ci tirées bien évidemment des fluctuations futures. Comme recommandations majeures, en lien avec les résultats obtenues, nous pouvons suggérer de remonter dans le temps pour trouver la quantité de déchets obtenus avant 1995 et, dans la mesure du possible collecter les données mensuellement, voir trimestriellement, afin de faire des prévisions beaucoup plus exactes. Ce qui est envisageable est de mettre en place des centres de collecte au niveau des centre de triage ou de dépôt des déchets à partir des questionnaires d'enquête ou des applications de collectes pour un suivi du processus de collecte des données. La multiplication des centres de triage demeure aussi un enjeu majeur pour évaluer le processus de génération des déchets de bois. Les politiques publiques sont bien établies lorsque nous avons une bonne estimation sur la production des déchets, mais aussi, elles permettraient de mettre en place des actions multipliant, améliorant et encourageant le triage du bois au niveau des centres de triage et un meilleur compostage plutôt que de les mettre dans les centres d'enfouissement. Nous pouvons envisager aussi de déterminer la taille des centres de tri par la quantité de déchets à travers les résultats de prévisions. Comme perspectives d'étude, nous nous demandons s'il ne serait pas possible de faire une approximation de la quantité de déchets réutilisés ou recyclés à partir des déchets éliminés. En même temps, il serait intéressant d'obtenir la quantité de déchets pour chaque province, voire pour chaque région ou municipalité afin de procéder à des études comparatives et de pou-

voir estimer la quantité de déchets par des approches plus efficaces. Ce qui est un défi important du fait que ces données ne sont pas disponibles dans le secteur de la construction, de la rénovation et de la démolition. La connaissance de la fluctuation des déchets peut permettre de planifier les ressources nécessaires et dans le futur à développer des stratégies de réductions efficaces.

Conclusion et perspectives

En conclusion, rappelons que l'objectif de ce mémoire est d'estimer la production de déchets de bois pour une utilisation énergétique en utilisant le modèle ARIMA et un modèle d'apprentissage profond, la mémoire à court et long terme (LSTM). Le but est de déterminer, à partir des données disponibles dans le secteur de la construction, de la rénovation et de la démolition (CRD), le modèle qui s'ajuste le mieux à nos données.

Dans ce contexte, nous avons fait une brève contextualisation du champ d'étude, en travaillant avec les données de bois pour la production d'énergie issues de la base de données de Statistique Canada [27]. Nous avons défini les enjeux de l'économie circulaire et réalisé une revue de littérature sur les approches statistiques utilisées dans le domaine de prévision de la quantité des déchets solides. À travers cette revue et compte tenu des données dont nous disposons, nous avons développé un modèle classique de base, le modèle ARIMA, qui s'adapte le mieux à nos données. Nous avons également développé un modèle d'apprentissage automatique LSTM. Ce modèle est souvent utilisé pour sa capacité à mémoriser les informations à court et à long terme et à fournir de bonnes prévisions.

Le modèle ARIMA a fourni de très bonnes prévisions pour l'ensemble des 8 jeux de données. Le meilleur résultat a été obtenu avec le jeu de données $J(8)$ de 26 observations avec une $RMSE$ de 0,0141 (calculée sur la variable transformée). Pour

le même jeu de données, le modèle LSTM a également permis d'obtenir la plus faible *RMSE* de 483. Cela conduit à conclure que le modèle ARIMA et le modèle LSTM ont de faibles *RMSE* et s'ajustent bien aux données étudiées. En outre, plus les données sont nombreuses, meilleur est le modèle, et plus faible est la *RMSE*. L'étude réalisée nous a permis de formuler quelques recommandations pour une bonne gestion des déchets solides de bois à des fins de valorisation énergétique. Il serait bénéfique de multiplier les centres de tri et de renforcer les matériaux en bois pour une meilleure durabilité.

Comme toute recherche scientifique, cette étude présente certaines limites, notamment l'accès limité à certaines données pour des raisons de confidentialité et le manque de données historiques. Seules les données annuelles de 1995 à 2021 étaient disponibles. Un volume de données plus important aurait permis d'améliorer le processus d'apprentissage de nos modèles et la qualité des résultats.

Bibliographie

- [1] K. ENVIRONMENTAL, « Preliminary resource recovery report card and gaps assessment for canada », *Kelleher Environmental : Toronto, ON, Canada*, 2020.
- [2] Y. MESSAOUD, *LSTM, Intelligence artificielle sur des données chronologiques*. 22 November 2018.
- [3] D. NIU, F. WU, S. DAI, S. HE et B. WU, « Detection of long-term effect in forecasting municipal solid waste using a long short-term memory neural network », *Journal of Cleaner Production*, vol. 290, p. 125187, 2021.
- [4] N. M. B. E. J. H. MARTIN GEISSDOERFER, Paulo Savaget, *L'économie circulaire - Un nouveau paradigme de durabilité ?*, vol. 143. 1er février 2017.
- [5] Z. YUAN, J. BI et Y. MORIGUICHI, « The circular economy : A new development strategy in china », *Journal of industrial ecology*, vol. 10, no. 1-2, p. 4-8, 2006.
- [6] P. GHISELLINI, C. CIALANI et S. ULGIATI, « A review on circular economy : the expected transition to a balanced interplay of environmental and economic systems », *Journal of Cleaner production*, vol. 114, p. 11-32, 2016.
- [7] J. KORHONEN, A. HONKASALO et J. SEPPÄLÄ, « Circular economy : the concept and its limitations », *Ecological economics*, vol. 143, p. 37-46, 2018.
- [8] P. SCHRÖDER, M. BENGTTSSON, M. COHEN, P. DEWICK, J. HOFSTETTER et J. SARKIS, « Degrowth within-aligning circular economy and strong sustainability narratives », *Resources, Conservation and Recycling*, vol. 146, p. 190-191, 2019.
- [9] E. MACARTHUR, *Le cercle vertueux*, vol. 7. European Investment Bank, 2019.

- [10] P. OGUZ-EKIM, « Machine learning approaches for municipal solid waste generation forecasting », *Environmental Engineering Science*, vol. 38, no. 6, p. 489–499, 2021.
- [11] M. ABBASI et A. EL HANANDEH, « Forecasting municipal solid waste generation using artificial intelligence modelling approaches », *Waste management*, vol. 56, p. 13–22, 2016.
- [12] L. A. F. B. MIYURU KANNANGARA, Rahul Dua, *Modeling and prediction of regional municipal solid waste generation and diversion in Canada using machine learning approaches*, vol. 74. April 2018.
- [13] U. SONI, A. ROY, A. VERMA et V. JAIN, « Forecasting municipal solid waste generation using artificial intelligence models—a case study in india », *SN Applied Sciences*, vol. 1, p. 1–10, 2019.
- [14] R. HE, M. SANDOVAL-REYES, I. SCOTT, R. SEMEANO, P. FERRAO, S. MATTHEWS et M. J. SMALL, « Global knowledge base for municipal solid waste management : Framework development and application in waste generation prediction », *Journal of Cleaner Production*, vol. 377, p. 134501, 2022.
- [15] D. DRICI, *Contribution au pronostic de défaut dans les systèmes complexes par les techniques intelligentes*. Thèse doctorat, Université Mohamed Khider Biskra, 2023.
- [16] M. CUBILLOS, *Multi-site household waste generation forecasting using a deep learning approach*, vol. 115. September 2020.
- [17] F. NICOLAU, *Chapitre 1 : Généralités sur les séries chronologiques*. 2005-2006.
- [18] E. F. Z. et ZOUIRI SARA, *Analyse de la stationnarité d’une série chronologique : ADF Test*. 24 février 2016.
- [19] O. et HAKYLL, *Comprendre les réseaux LSTM*. 27 août 2015.
- [20] F. GRÉGOIRE, « Extraction de phrases parallèles à partir d’un corpus comparable avec des réseaux de neurones récurrents bidirectionnels », 2018.
- [21] D. E. RUMELHART, G. E. HINTON et R. J. WILLIAMS, « Learning representations by back-propagating errors », *nature*, vol. 323, no. 6088, p. 533–536, 1986.

- [22] D. P. KINGMA et J. BA, « Adam : A method for stochastic optimization », *arXiv preprint arXiv :1412.6980*, 2014.
- [23] J. ABADIE et F. MESLIER, « Etude de l'utilisation des modèles arima pour la prévision à très court terme de l'énergie journalière produite par electricité de france », *RAIRO-Operations Research*, vol. 13, no. 1, p. 37–54, 1979.
- [24] A. CHERIET et A. DERDICHE, « Etude prévisionnelle à court terme de l'évolution des accidents de la route en algérie (2010-2018) », 2021.
- [25] Y. FANCHETTE, *Combinaison d'un modèle régional climatique et de modèles à réseaux de neurones pour la prévision du productible photovoltaïque : application au cas de l'île de La Réunion*. Thèse doctorat, Université de la Réunion, 2023.
- [26] M. MESFIOUI, *Notes de cours de séries chronologiques*. 2022.
- [27] *Consommation de déchets de bois et de lessive de pâte épuisée pour la production d'énergie, annuel*. 2022-12-06.
- [28] S. RUDER, « An overview of gradient descent optimization algorithms », *arXiv preprint arXiv :1609.04747*, 2016.

Annexe A

Code du modèle ARIMA sur R

```
---  
title: "Modèle ARIMA sur nos données chronologiques  
avec les 24 premières observations"  
author: "Masse BA"  
date: "2022-10-04"  
output: html_document  
editor_options:  
  chunk_output_type: console  
---  
  
```${r setup, include=FALSE}  
knitr::opts_chunk$set(echo = TRUE)
```${r  
  
## 1. Importation de la base
```

```
```${r}
set.seed(1234)
base1 <- read.csv2("~/Documents/uqtr/Memoire/Data/dechet_bois.csv", sep=";")

names(base1)=c("Annee", "Dechet")

base=base1[1:22,]

...

2. Exploration et familiarisation avec la base
```${r}
library(dplyr)
#base$annee=seq(1995,2016,1)
base1 <- base1 %>% mutate(annee=seq(1995,2021,1))
...

### 2.0 Boxplot sur R
```${r}
boxplots <- ggplot(data = base1) +
 geom_boxplot(mapping = aes(x = annee, y = Dechet)) +
 labs(
 x = "année",
 y = "Production des déchets"
)

boxplots
...

2.1. Quelques transformations de la série brute
```

```

```{r}
base1 <- base1 %>% mutate(base_inverse=1/base1$Dechet)
base1 <- base1 %>% mutate(base_sqrt=sqrt(base1$Dechet))
base1 <- base1 %>% mutate(base_log=log(base1$Dechet))
#MA<-diff(base1$Dechet, lag = 1)
#MA<-abs(MA)
base1<-base1 %>% mutate(base_diff=c(NA, abs(diff(base1$Dechet, lag = 1))))

```

```

### ### 2.1. Représentation des séries transformées

```

```{r}
library(ggplot2)

g1=ggplot(data = base1, aes(x = annee, y = base_inverse)) +
  geom_line() +
  labs(x = "Année",
       y = "Déchets de bois",
       title = "Transformation inverse")

g2=ggplot(data = base1, aes(x = annee, y = base_sqrt)) +
  geom_line() +
  labs(x = "Année",
       y = "Déchets de bois",
       title = "Transformation racine carré")

g3=ggplot(data = base1, aes(x = annee, y = base_log)) +
  geom_line() +
  labs(x = "Année",

```

```

    y = "Déchets de bois",
    title = "Transformation logarithmique")

g4=ggplot(data = base1, aes(x = annee, y = base_diff)) +
  geom_line() +
  labs(x = "Année",
       y = "Déchets de bois",
       title = "Transformation par différenciation")

...

### 2.3 Regroupement des graphes dans une seule graphique
```{r}
library("cowplot")
div_perform_plot_default <- ggdraw() +
 draw_plot(g1, x = 0, y = .5, width = .5, height = .5) +
 draw_plot(g2, x = .5, y = .5, width = .5, height = .5) +
 draw_plot(g3, x = 0, y = 0, width = 0.5, height = 0.5) +
 draw_plot(g4, x = 0.5, y = 0, width = 0.5, height = 0.5) +
 draw_plot_label(label = c("A", "B", "C", "D"), size = 15,
 x = c(0, 0.5, 0, 0.5), y = c(1, 1, 0.5, 0.5))
div_perform_plot_default
...

```{r}
library(ggplot2)
#Représentation avec l'axe des abscisses
ggplot(data = base1, aes(x = annee, y = Dechet)) +
  geom_line() +
  labs(x = "Année",

```

```

    y = "Production de déchets de bois",
    title = "Production des déchets de 1995 à 2020")
  ...

## 3. Tendance, saisonnalité et stationnarité

### 3.1 Série corrigée de la tendance, de la saisonnalité et de la stationnarité

```{r}
MA<-sqrt(base$Dechet) #lag=5
MA<-log(MA)
MA<-abs(MA)

base<-base%>% mutate(MA)

ggplot(data = base, aes(x = annee, y = MA)) +
 geom_line() +
 labs(x = "Année",
 y = "Déchets de bois",
 title = "Série transformée par le logarithme et la racine carré")
...

3.2 Étude de la tendance et de la saisonnalité

```{r}
library("Kendall")
## Test de tendance : MannKendal
MannKendall(MA)

```

```

## Test de saisonnalité : SeasonalMannKendal
#SeasonalMannKendall(MA)
...

### 3.3 Test de stationnarité (Dickey-Fuller Simple, Dickey-Fuller augmenté, Phill

```{r}
library(tseries)
 #Test de non stationnarité
 adf.test(MA, k=0)
 adf.test(MA, k=1)
 pp.test(MA)
 kpss.test(MA)
 #Test de stationnarité
 kpss.test(MA)
...

3.4 Stationnarité

```{r}
library(tseries)
## Test de Dickey Fuller
adf.test(MA)
## Vu qu'on a moins de 30 observation, on utilise
## le test non paramétrique de PHILLIP PERRON
## Test de PHILLIP-Perron
pp.test(MA,alternative = c("stationary", "explosive"),
        type = c("Z(alpha)", "Z(t_alpha)"), lshort = TRUE)
...

```

```
##4. Visualisation de ACF et PACF
```

```
```{r}
layout(matrix(1:2, nrow=1,ncol=2))
acf(MA, main=" ")
pacf(MA, main=" ")
```
```

```
##5. Estimation des modèles proposés
```

```
```{r}
On applique le modèle ARIMA sur les données brutes en y mettant l'ordre de diffé
modele1<-arima(MA, order = c(1,0,1)) #ARIMA(1,6,0)
modele2<-arima(MA, order = c(2,0,1)) #ARIMA(0,6,1)
modele3<-arima(MA, order = c(2,0,0)) #ARIMA(1,6,1)
modele4<-arima(MA, order = c(2,0,0)) #ARIMA(1,6,1)
modele5<-arima(MA, order = c(2,0,0)) #ARIMA(1,6,1)

Validation du modèle-----
abs(modele1$coef)>1.96*(diag(modele1$var.coef))
abs(modele2$coef)>1.96*(diag(modele2$var.coef))
abs(modele3$coef)>1.96*(diag(modele3$var.coef))
abs(modele4$coef)>1.96*(diag(modele4$var.coef))
abs(modele5$coef)>1.96*(diag(modele5$var.coef))

Choix du modèle à prendre à partir du critère AIC
modele1$aic
modele2$aic
modele3$aic
```

```
modele4$aic
modele5$aic

residu1= residuals(modele1)
residu2= residuals(modele2)
residu3= residuals(modele3)
residu4= residuals(modele4)
residu5= residuals(modele5)

Choix du modèle à prendre à partir du critère BIC
BIC1 <- log(var(residu1))+2*(log(length(residu1)/length(residu1)))
BIC2 <- log(var(residu2))+2*(log(length(residu2)/length(residu2)))
BIC3 <- log(var(residu3))+2*(log(length(residu3)/length(residu3)))
BIC4 <- log(var(residu4))+2*(log(length(residu4)/length(residu4)))
BIC5 <- log(var(residu5))+2*(log(length(residu5)/length(residu5)))

BIC2

...

##6. Test de non corrélation des erreurs

```{r}

#install.packages('ggpubr')
library(ggplot2)
library(ggpubr)
#layout(matrix(1:2, nrow = 1, ncol = 2))
```

```
hist(residu2, main = " ", xlab = "Résidu2", ylab = "Fréquence", col = "blue")
ggqqplot(residu2)
tsdiag(modele2)
## Test de Shapiro Wilk
shapiro.test(residu2)
## Test parametrique de Durbin Watson
#install.packages("car")
#install.packages("car")
library(tidyverse)
library(ggpubr)
library(rstatix)
library(car)
durbinWatsonTest(c(residu2))

hist(residu2)
ggdensity(residu2, fill = "lightgray", xlab='x', ylab='densité')
ggqqplot(residu2, xlab='Théorique', ylab='Échantillon')

residu2=modele2$residuals
plot(residu2, col="light blue", xlab='Théorique', ylab='Échantillon')
shapiro.test(residu2)

# Effectuer le test de Ljung-Box
Box.test(residu2, lag = 5, type = "Ljung-Box")
```


##7. Prévision


```
```{r}
library(forecast)
```


```

```

prevision<- forecast(modele2, h=10, level=90)
prevision
plot(prevision,main= " prévision des déchets sur 5 ans")

prev=predict(modele2, n.ahead=10)$pred
prev
...

##8. Retrouver les bonnes valeurs de prédictions

```{r}
a1=(exp(prev))^2

Le RMSE du modèle ARIMA(2,0,1)
accuracy(modele2)

data <- data.frame(annee=seq(2017,2026,1), prevision=a1)

ggplot(data = data, aes(x = annee, y = prevision)) +
 geom_line() +
 labs(x = "Année",
 y = "Production des déchets de bois",
 title = "Prévision sur 10 ans")
...

##9. Représentation de la série de base réelle et prédite.
```{r}
data_new <- data.frame(annee=seq(1995,2021,1), reel=base1$Dechet, prevision=c(base

ggplot(data_new, aes(x=annee)) +

```

```
geom_line(aes(y = prevision, linetype="prevision"))+
geom_line(aes(y = reel, color = "reel")) +
labs(x = "Année", y = "Quantité de déchets de bois")+
scale_linetype_manual(name = "", values = c("prevision" = "dashed", "reel" = "so
scale_color_manual(name = "Légende", values = c("reel" = "black"))

ggplot(data_new, aes(x=annee)) +
  geom_line(aes(y = prevision, linetype="prevision"))+
  geom_line(aes(y = reel, color = "reel")) +
  labs(x = "Année", y = "Quantité de déchets de bois")+
  scale_linetype_manual(name = "Légende", values = c("prevision" = "dashed", "reel
  scale_color_manual(name = "Légende", values = c( "reel" = "black"))
...

```

Annexe B

Code du modèle LSTM sur R

```
library("lubridate")
library(dplyr)
library(readr)
library(keras)
library(tensorflow)
dfm <- read_delim("~/Documents/uqtr/Memoire/Data/dechet_bois.csv",
                  delim = ";", escape_double = FALSE, trim_ws = TRUE)

mymodel_lstm <- function(df, n_pred){
  df <- df |> data.frame()
  df$Dchet_bois <- as.numeric(df$Dchet_bois)
  #Standardisation
  scale_factors <- c(mean(df$Dchet_bois), sd(df$Dchet_bois))
  scale_factors

  library(dplyr)
  scaled_train <- df |>
```

```
select(Dchet_bois) |>
mutate(Dchet_bois = (Dchet_bois - scale_factors[1]) / scale_factors[2])

prediction <- n_pred
lag <- prediction

scaled_train <- as.matrix(scaled_train)

# Préparation des données d'entrainement : décalage de 11
x_train_data <- t(sapply(
  1:(length(scaled_train) - lag - prediction + 1),
  function(x) scaled_train[x:(x + lag - 1), 1]
))

# Transformation en format 3D : array
x_train_arr <- array(
  data = as.numeric(unlist(x_train_data)),
  dim = c(
    nrow(x_train_data),
    lag,
    1
  )
)

#Préparation de la variable réponse : décalage et transformation 3D
y_train_data <- t(sapply(
  (1 + lag):(length(scaled_train) - prediction + 1),
  function(x) scaled_train[x:(x + prediction - 1)]
))
```

```
y_train_arr <- array(  
  data = as.numeric(unlist(y_train_data)),  
  dim = c(  
    nrow(y_train_data),  
    prediction,  
    1  
  )  
)  
  
x_test <- df$Dchet_bois[(nrow(scaled_train) - prediction + 1):nrow(scaled_train)]  
  
x_test_scaled <- (x_test - scale_factors[1]) / scale_factors[2]  
  
# Préparation des données test  
x_pred_arr <- array(  
  data = x_test_scaled,  
  dim = c(  
    1,  
    lag,  
    1  
  )  
)  
  
lstm_model <- keras_model_sequential()  
  
lstm_model %>%  
  layer_lstm(units = 3, # size of the layer  
             batch_input_shape = c(1, prediction, 1), # batch size, timesteps, f  
             return_sequences = TRUE,  
             stateful = TRUE) %>%
```

```
# fraction of the units to drop for the linear transformation of the inputs
layer_dropout(rate = 0.5) %>%
layer_lstm(units = 3,
           return_sequences = TRUE,
           stateful = TRUE) %>%
layer_dropout(rate = 0.5) %>%
time_distributed(keras::layer_dense(units = 1))

lstm_model %>%
  compile(loss = 'mae', optimizer = 'adam', metrics = 'accuracy')

summary(lstm_model)

lstm_model %>% fit(
  x = x_train_arr,
  y = y_train_arr,
  batch_size = 1,
  epochs = 20,
  verbose = 0,
  shuffle = FALSE
)

lstm_forecast <- lstm_model %>%
  predict(x_pred_arr, batch_size = 1) %>%
  .[, , 1]

# On fait revenir à l'échelle normale pour enlever la standardisation
lstm_forecast <- lstm_forecast * scale_factors[2] + scale_factors[1]
lstm_forecast #prédiction
```

```
true_values <- x_pred_arr* scale_factors[2] + scale_factors[1] #vraie valeur
true_values <- as.vector(true_values)
diffs <- true_values-lstm_forecast
df3 <- data.frame(lstm_forecast, true_values, diffs)
return(df3)
}

out <- mymodel_lstm(dfm, 5)

## Consigne : Appliquer la fonction mymodel_lstm() en précisant toujours en argument
#           le dataframe dfm et le nombre de ligne qui doivent être prédites.

#Modèle 1
out1 <- mymodel_lstm(dfm, 1)
#RMSE
out1$diffs^2 |> mean() |> sqrt()

#Modèles 2
out2 <- mymodel_lstm(dfm, 2)
#RMSE
out2$diffs^2 |> mean() |> sqrt()

#Modèles 3
out3 <- mymodel_lstm(dfm, 3)
#RMSE
out3$diffs^2 |> mean() |> sqrt()
```

```
#Modèles 4
out4 <- mymodel_lstm(dfm, 4)
#RMSE
out4$diffs^2 |> mean() |> sqrt()
```

```
#Modèles 5
out5 <- mymodel_lstm(dfm, 5)
#RMSE
out5$diffs^2 |> mean() |> sqrt()
```

```
#Modèle 6
out6 <- mymodel_lstm(dfm, 6)
#RMSE
out6$diffs^2 |> mean() |> sqrt()
```

```
#Modèle 7
out7 <- mymodel_lstm(dfm, 7)
#RMSE
out7$diffs^2 |> mean() |> sqrt()
```

```
#Modèle 8
out8 <- mymodel_lstm(dfm, 8)
#RMSE
out8$diffs^2 |> mean() |> sqrt()
```