

# Rapport Projet BNP

Ben Amor Melek  
El Hannachi Bechir  
Lafaille Nicolas  
Nicaise Maxime

26 mars 2025

# Chapitre 1

## Prédiction des émissions CO<sub>2</sub> avec des modèles de types ARIMA, LSTM et PROPHET

### 1.1 Les modèles

#### 1.1.1 ARIMA

##### Modèle

Le modèle ARIMA (AutoRegressive Integrated Moving Average) est un modèle de séries temporelles largement utilisé pour la prévision. Il repose sur trois composants principaux : l'autorégression (AR), l'intégration (I) et la moyenne mobile (MA).

Un modèle ARIMA est généralement désigné par le triplet  $(p, d, q)$ , où :

- $p$  représente l'ordre du modèle autorégressif (AR), qui indique le nombre de termes passés de la série temporelle utilisés pour prédire la valeur suivante.
- $d$  est le nombre de différenciations nécessaires pour rendre la série temporelle stationnaire
- $q$  est l'ordre du modèle de moyenne mobile (MA), qui indique le nombre de termes passés des erreurs de prévision (ou résidus) utilisés pour prédire la valeur actuelle.

La formulation théorique d'un modèle ARIMA s'écrit :

$$y_t^{(d)} = \sum_{i=1}^p \phi_i y_{t-i}^{(d)} + \sum_{j=1}^q \theta_j \epsilon_{t-j} + \epsilon_t$$

où :

- $y_t^{(d)}$  est la série temporelle différenciée  $d$  fois,
- $\phi_1, \phi_2, \dots, \phi_p$  sont les paramètres du modèle autorégressif,
- $\epsilon_t$  est l'erreur ou le résidu à l'instant  $t$  (souvent supposé être une série de bruits blancs),
- $\theta_1, \theta_2, \dots, \theta_q$  sont les paramètres du modèle de moyenne mobile.

La série temporelle différenciée  $y_t^{(d)}$  suit le modèle ARMA(p,q). Les hypothèses du modèle ARIMA incluent la stationnarité (après les  $d$  différenciations), l'indépendance des erreurs ( $\epsilon_t$ ), ainsi que l'absence de corrélation entre les erreurs et les valeurs passées de la série.

Il est également possible de rajouter des variables exogènes  $z$  pour la prédition du modèle. Cela donne le modèle ARIMAX :

$$y_t^{(d)} = \sum_{i=1}^p \phi_i y_{t-i}^{(d)} + \sum_{j=1}^q \theta_j \epsilon_{t-j} + \langle \beta, z \rangle + \epsilon_t$$

où :

- $\beta$  est le paramètre associé aux variables exogènes.

## Identification des hyperparamètres

Pour choisir les hyperparamètres  $p$  et  $q$  manuellement on se réfère aux graphes ACF (Auto-Correlation Function) et PACF (Partial Auto-Correlation Function) en utilisant les propriétés suivantes : De manière générale, on identifie des candidats possibles pour  $p$  et  $q$  et puis on sélectionne

Modèle	ACF	PACF
AR( $p$ )	Décroît exponentiellement ou oscille	Coupure après le lag $p$
MA( $q$ )	Coupure après le lag $q$	Décroît exponentiellement ou oscille
ARMA( $p, q$ )	Décroît exponentiellement ou oscille	Décroît exponentiellement ou oscille

TABLE 1.1 – Identification des ordres  $p$  et  $q$  à l'aide des ACF et PACF

le modèle ayant le *AIC* minimal.

### 1.1.2 LSTM

Les Long short-term memory (LSTM) sont un type de réseau de neurones récurrent (RNN) spécialement conçus pour modéliser des séquences de données avec des dépendances à long terme. Contrairement aux RNN classiques, qui souffrent du problème du gradient qui disparaît, les LSTM sont capables de retenir des informations pendant des périodes plus longues grâce à l'utilisation de portes qui régulent le flux d'informations.

La cellule LSTM comporte plusieurs composants clés :

- L'état de la cellule ( $C_t$ ) : il contient la mémoire de long terme et est mis à jour par les portes d'entrée et d'oubli.
- La **porte d'entrée** ( $i_t$ ) : elle décide quelles informations de l'entrée actuelle  $x_t$  et de l'état caché précédent  $h_{t-1}$  doivent être ajoutées à l'état de la cellule  $C_t$ .
- La **porte d'oubli** ( $f_t$ ) : elle détermine quelles informations de l'état de la cellule précédent  $C_{t-1}$  doivent être oubliées.
- La **porte de sortie** ( $o_t$ ) : elle contrôle la manière dont l'état de la cellule  $C_t$  influence la sortie  $h_t$ .

La mise à jour des différents composants du LSTM à chaque étape  $t$  peut être formulée :

$$\begin{aligned} f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C[h_{t-1}, x_t] + b_C) \\ C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\ o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \\ h_t &= o_t \odot \tanh(C_t) \end{aligned}$$

Où :

- $\sigma$  est la fonction sigmoïde,
- $\tanh$  est la fonction tangente hyperbolique,
- $W_f, W_i, W_C, W_o$  sont les matrices de poids pour chaque porte,
- $b_f, b_i, b_C, b_o$  sont les biais associés,
- $\odot$  représente le produit élément par élément.

Le modèle LSTM est conçu pour éviter les problèmes de vanishing gradient, permettant ainsi un meilleur apprentissage des dépendances à long terme dans les données séquentielles. Chaque composant du modèle LSTM joue un rôle crucial dans la gestion de l'information à travers le temps, ce qui en fait une architecture puissante pour des tâches telles que la prédition de séries temporelles, la traduction automatique, et la modélisation du langage.

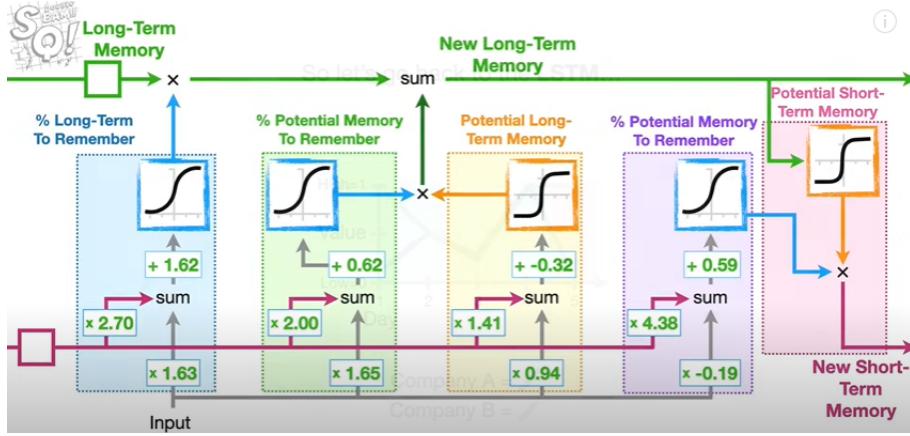


FIGURE 1.1 – Structure d'une cellule LSTM

### 1.1.3 PROPHET

On modélise notre série temporelle  $y$  comme somme de plusieurs composantes.

$$y(t) = g(t) + s(t) + h(t) + \langle \beta, z(t) \rangle + \epsilon_t$$

- $g$  représente la tendance
- $s$  la saisonnalité
- $h$  les effets ponctuels (comme les évènements spéciaux et les jours fériés)
- $z$  vecteur contenant les variables explicatives exogènes
- $\beta$  vecteur des poids
- $\epsilon$  un bruit blanc

#### Modélisation de la tendance

La tendance est souvent modélisée par une fonction affine par morceaux.

$$g(t) = m + k(t - t_0) + \langle \delta, a(t) \rangle$$

où :

- $t_0$  le temps de début de la série temporelle
- $m$  l'offset initial
- $s_1, \dots, s_S$  sont les points de changement de pente
- $\delta = (\delta_s)_{1 \leq s \leq S}$  est un vecteur contenant les changements de pente
- $\forall t, a(t) = (\mathbb{1}_{t \geq t_s}(t - t_s))_{1 \leq s \leq S}$

Les paramètres de  $g$  sont trouvés par une méthode des moindres carrés avec une régularisation  $L_1$  sur  $\delta$ .

Dans le cas d'une série possédant une capacité maximale, le modèle logistique est souvent utilisé :

$$g(t) = \frac{C}{1 + \exp(-(k + \langle \delta, a(t) \rangle))}$$

#### Modélisation de la saisonnalité

La partie saisonnalité représente le comportement périodique de la série temporelle. On peut éventuellement ajouter plusieurs effets saisonniers liés à différentes périodes.

$$s(t) = \sum_{n=1}^N \left( a_n \cos\left(\frac{2\pi n t}{T}\right) + b_n \sin\left(\frac{2\pi n t}{T}\right) \right)$$

$s$  est représentée par une série de Fourier de période  $T$  (choisie au préalable), tronquée à l'ordre  $N$  qui est un hyperparamètre contrôlant la complexité du modèle. Les coefficients de  $s$  sont retrouvés par une régression bayésienne.

## Modélisation des effets ponctuels

On choisit au préalable des événements  $\tau_1, \dots, \tau_H$ .

$$h(t) = \sum_{i=1}^H \alpha_i \mathbb{1}_{t=\tau_i}$$

## 1.2 Pré-traitement des données

### 1.2.1 Elimination des valeurs aberrantes et fusion des datasets

On commence par filtrer les dates sur les 2 datasets pour n'avoir que des données utilisables ; on retire toutes les entrées datant d'avant 2013 et d'après 2026. Puis on modifie les valeurs dans la colonne correspondant au secteur d'activité afin de regrouper toutes les entrées dans un des 4 secteurs d'activité suivants :

- Bâtiment à usage principale de bureau, d'administration ou d'enseignement.
- Autres cas (par exemple : théâtres, salles de sport, restauration, commerces individuels, etc).
- Centre commercial.
- Bâtiment à occupation continue (par exemple : hopitaux, hôtels, internats, maisons de retraite, etc).

On veut maintenant gérer la fusion des 2 datasets. Il s'agit donc de mettre bout à bout les colonnes qui se correspondent et auront un intérêt pour la suite, en s'assurant que les entrées sont bien dans la même unité. En particulier, dans le premier dataset, les coordonnées sont données sous forme de latitude et longitude, et non pas sous forme de coordonnées  $x$  et  $y$ . On gère donc la conversion avec une librairie adaptée de Python. On choisit ensuite de faire la fusion selon le tableau suivant :

Dataset 1	Dataset 2
surface_utile	Surface_utile
shon	Surface_(SHON)
date_etablissement_dpe	Date_établissement_DPE
annee_construction	Année_construction
secteur_activite	Secteur_activité
classe_estimation_ges	Etiquette_GES
estimation_ges	Emission_GES_kgCO2/m <sup>2</sup> /an
consommation_energie	Conso_kWhep/m <sup>2</sup> /an
X_L93	Coordonnée_cartographique_X_(BAN)
Y_L93	Coordonnée_cartographique_Y_(BAN)

TABLE 1.2 – Correspondance entre les datasets

Une fois la fusion terminée, on effectue une première étape de filtreage sur les valeurs numériques, qui consiste à retirer toutes les valeurs négatives, qui n'ont aucun sens dans notre cas. On gère ensuite les valeurs manquantes ; sachant que dans le 2ème dataset, un 0 signifie que la valeur est manquante. Pour les colonnes numériques, on remplace les valeurs manquantes par la médiane (une méthode d'imputation plus robuste aux outliers que la moyenne).

En ce qui concerne les cases pour lesquelles le secteur d'activité est absent, on utilise un 3-NN pour imputer le secteur d'activité. Les variables utilisées pour faire l'imputation sont : *shon*, *estimation\_ges* et *consommation\_energie*. On les a centrées et normalisées pour éviter qu'une variable ait un poids plus important.

### 1.2.2 Identification et Suppression des Outliers

Pour identifier les valeurs aberrantes de la colonne *estimation\_ges*, nous avons utilisé la méthode **IQR (Interquartile Range)**. Afin de visualiser ces valeurs aberrantes, nous avons généré :

- Un **boxplot avant suppression des outliers** pour observer la dispersion des valeurs (Figure 1.13).
- La **distribution de estimation\_ges** pour mieux comprendre sa répartition (Figure 1.3).

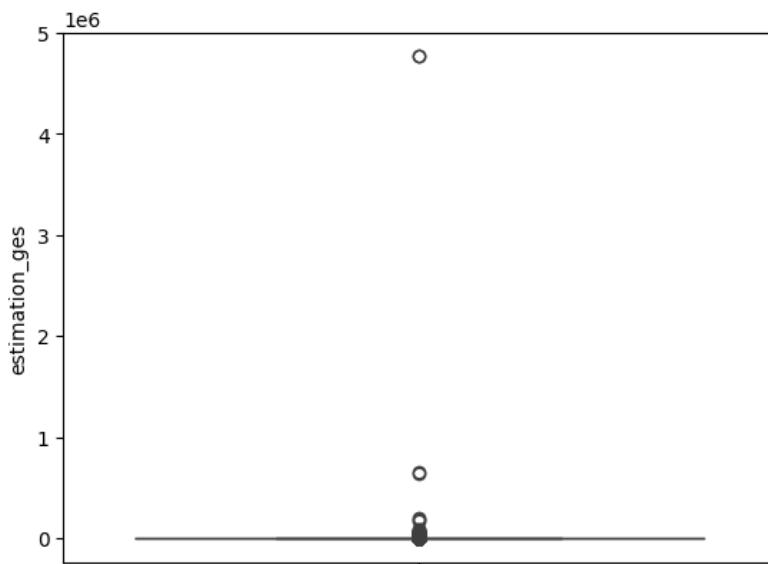


FIGURE 1.2 – Boxplot de l'estimation des GES avant suppression des outliers.

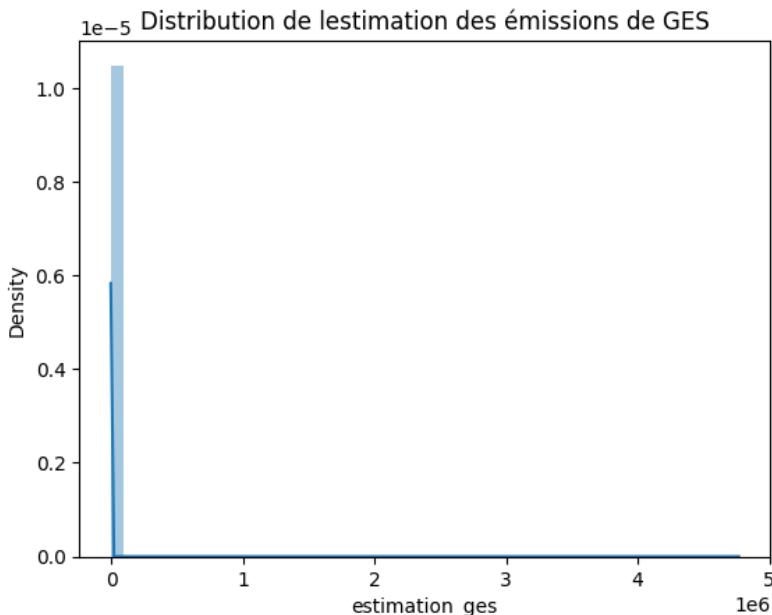


FIGURE 1.3 – Distribution de l'estimation des GES avant suppression des outliers.

Initialement, nous avons appliqué la méthode IQR de manière globale avec un seuil  $Q3 = quantile(0.75)$ , mais cela aurait entraîné la suppression de **16 %** des données. Ce taux étant trop élevé, nous avons ajusté le seuil à  $Q3 = quantile(0.95)$ , réduisant ainsi la suppression à seulement **0.62 %** des données, un compromis plus acceptable.

Cependant, une approche plus logique consiste à **supprimer les outliers par secteur d'activité**, plutôt que globalement. En effet, certains secteurs ont naturellement des niveaux d'émissions de GES plus élevés que d'autres, et appliquer un même seuil à l'ensemble des données risquerait de supprimer des valeurs pertinentes pour ces secteurs.

Nous avons donc procédé à l'identification et la suppression des outliers **spécifiques à chaque catégorie**. Ensuite, nous avons visualisé les résultats à travers plusieurs figures :

- **Boxplots après suppression des outliers** pour chaque secteur d'activité (Figure 1.4).
- **Distribution globale des GES après suppression des outliers** (Figure 1.5).
- **Distributions des GES par catégorie d'activité**, regroupées en une seule figure pour comparaison (Figure 1.6).

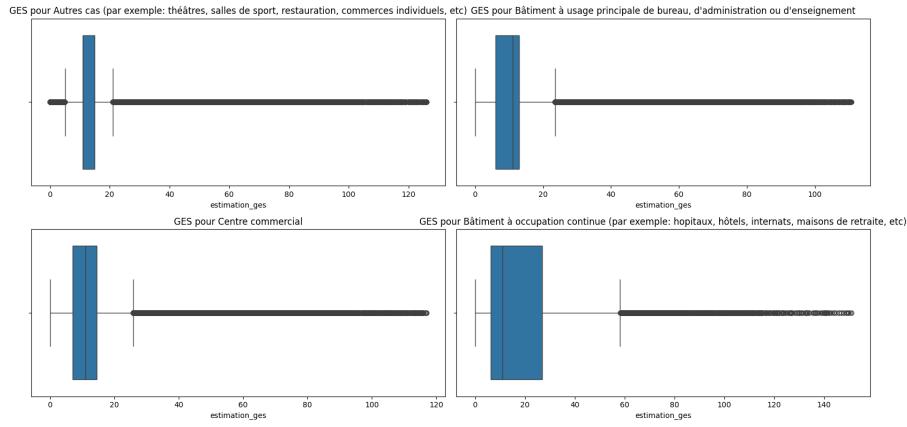


FIGURE 1.4 – Boxplot des émissions de GES après suppression des outliers par catégorie d’activité.

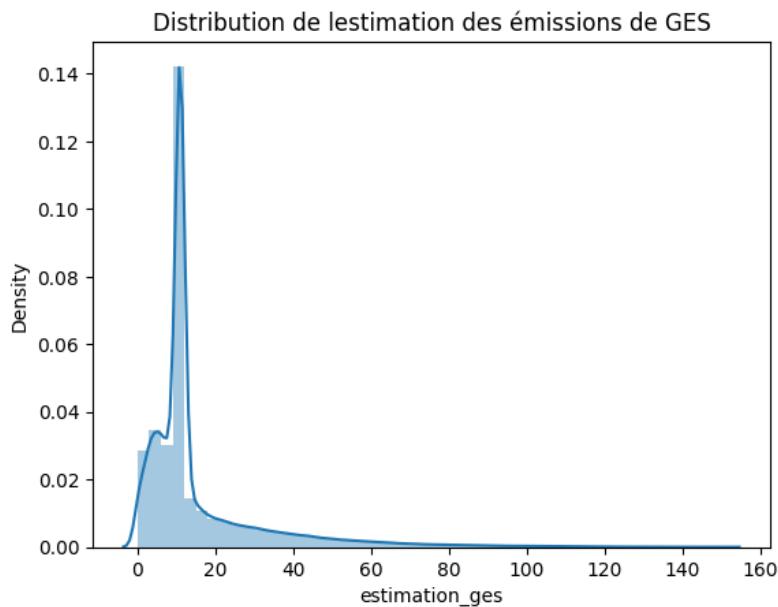


FIGURE 1.5 – Distribution globale des émissions de GES après suppression des outliers.

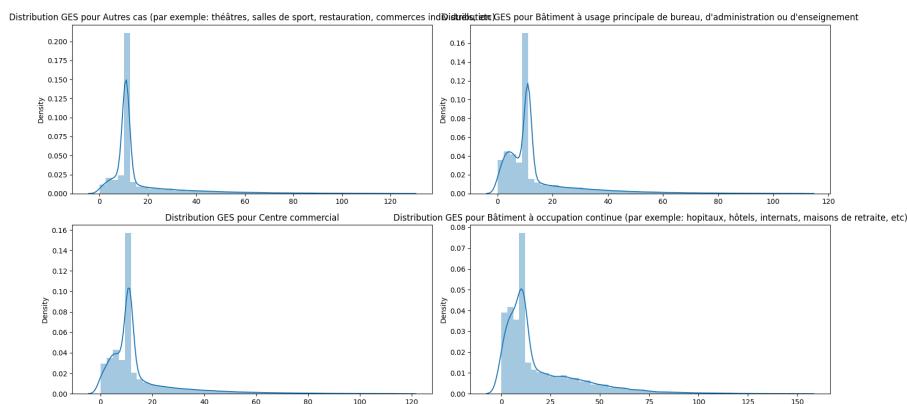


FIGURE 1.6 – Distributions des émissions de GES après suppression des outliers pour chaque catégorie d’activité.

### 1.2.3 Aggrégation des données

Pour appliquer nos modèles de séries temporelles, il est nécessaire d’avoir une seule observation par période (jour, mois ou année selon l’utilisation). Pour ce faire, on regroupe les données par

période et on fait l'aggrégation selon la formule :

$$\text{estimation\_ges moyenne} = \frac{\sum_i \text{estimation\_ges}_i \times \text{shon}_i}{\sum_i \text{shon}_i}$$

Dans un premier temps, on a fait une aggrégation par jour. On a constaté que les données sont très bruitées et volatiles dans ce cas là, notamment parce que le nombre de bâtiments contrôlés par jour est insuffisant pour que la moyenne empirique soit proche de la vraie moyenne. On a choisi après de faire l'aggrégation par mois ce qui a amélioré les performances.

## 1.3 Résultats

### 1.3.1 ARIMA

Lorsqu'on applique le modèle ARIMA aux données agrégées, on obtient les courbes de la figure 1.7. On obtient un MSE entre 8.78 et 19.23 sur la période de test.

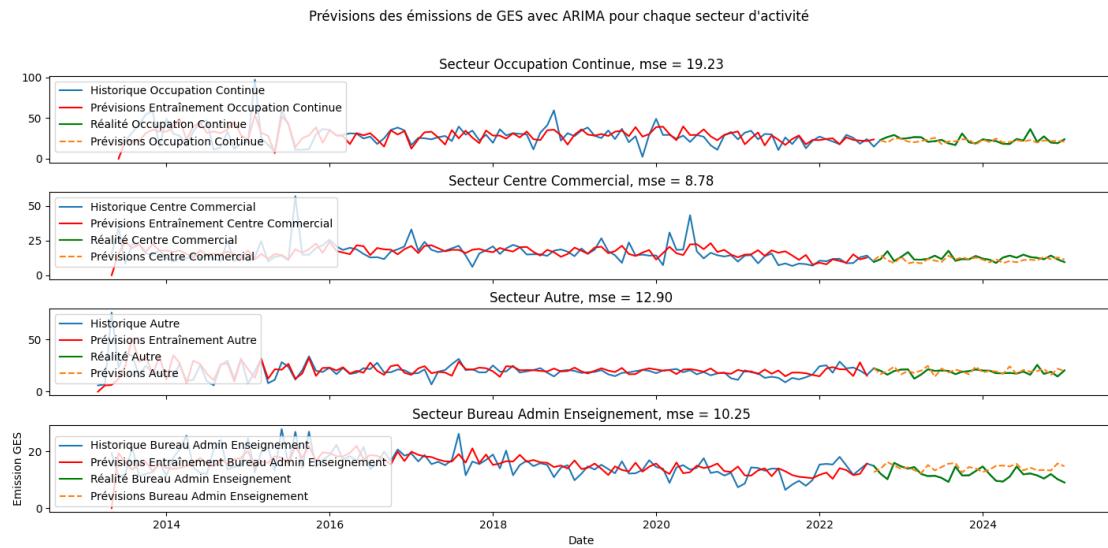


FIGURE 1.7 – Résultat du modèle ARIMA pour les données agrégées

### 1.3.2 LSTM

prediction pour catégorie predictions pour la catégorie Autres par mois avec mse = 11.53

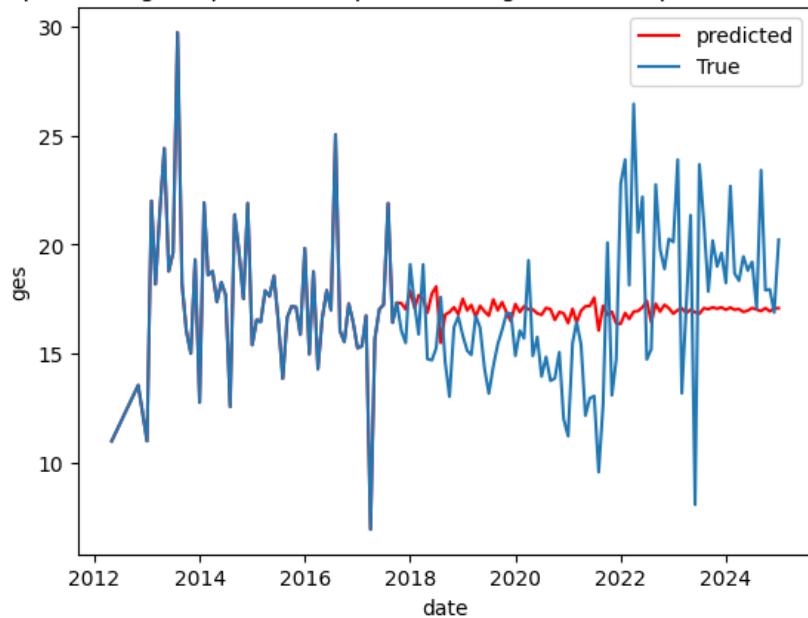


FIGURE 1.8 – Résultat du modèle LSTM pour les données agrégées

prediction pour catégorie predictions pour la catégorie bureau\_admin\_enseignement par mois avec mse = 17.41

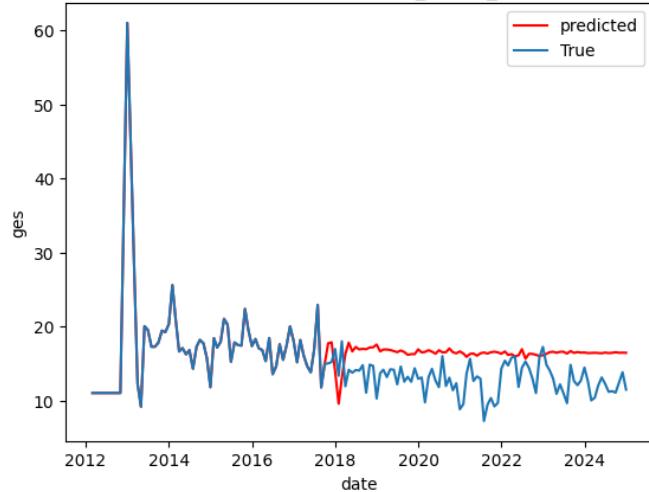


FIGURE 1.9 – Résultat du modèle LSTM pour les données agrégées

prediction pour catégorie predictions pour la catégorie centre\_commercial par mois avec mse = 51.80

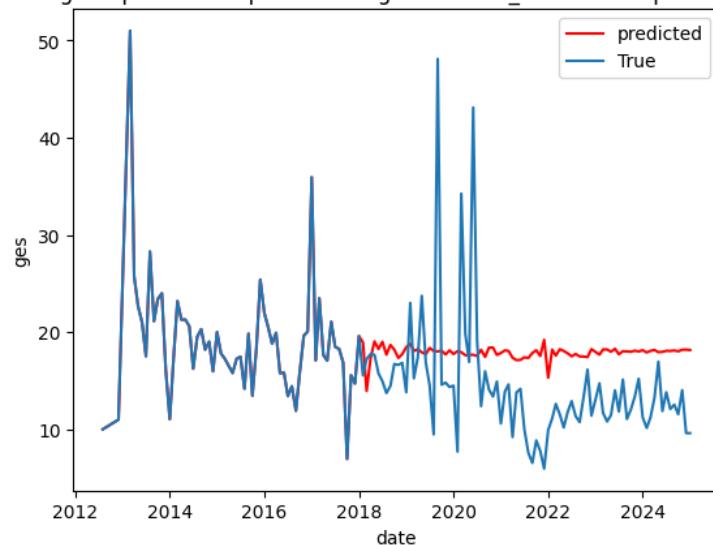


FIGURE 1.10 – Résultat du modèle LSTM pour les données agrégées

prediction pour catégorie predictions pour la catégorie occupation\_continue par mois avec mse = 84.90

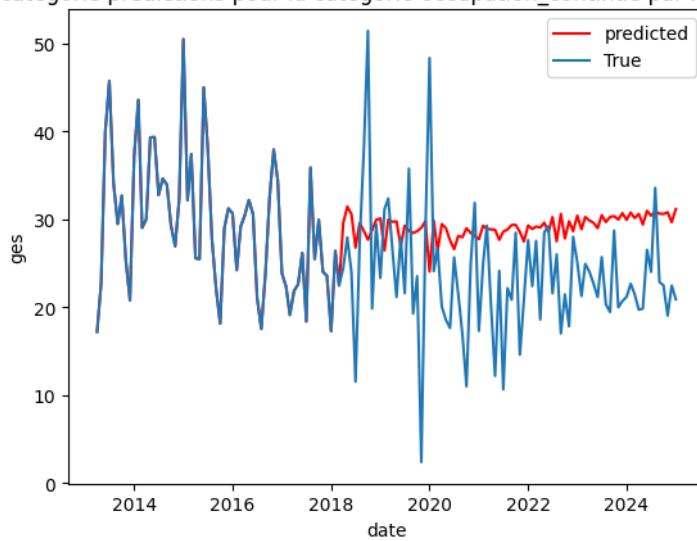


FIGURE 1.11 – Résultat du modèle LSTM pour les données agrégées

### 1.3.3 PROPHET

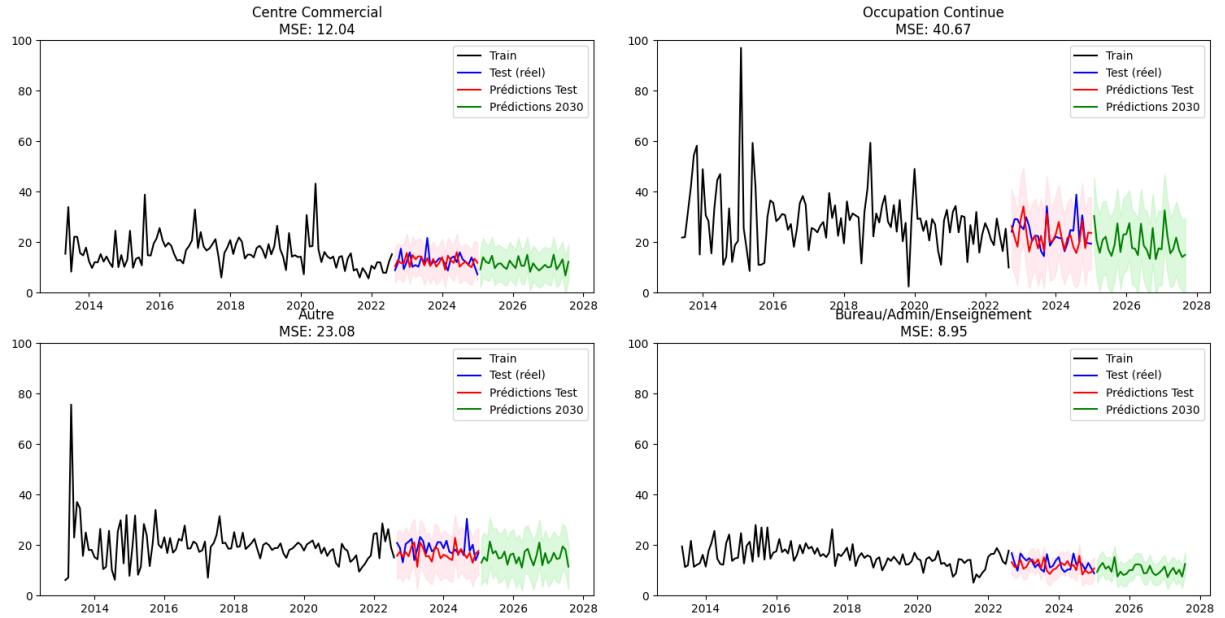


FIGURE 1.12 – Résultats obtenus par le modèle prophet.

Le modèle obtient des résultats assez bons, du même ordre que ceux obtenus par ARIMA, hormis sur le secteur d’activité occupation continue.

### 1.3.4 Modèle naïf

On remarque que malgré les tentatives de fine tuning, les performances des modèles précédents sont discutables. En effet, le modèle suivant (la moyenne) obtient des performances en terme d’erreur qui sont du même ordre, voire meilleures qu’ARIMA et PROPHET.

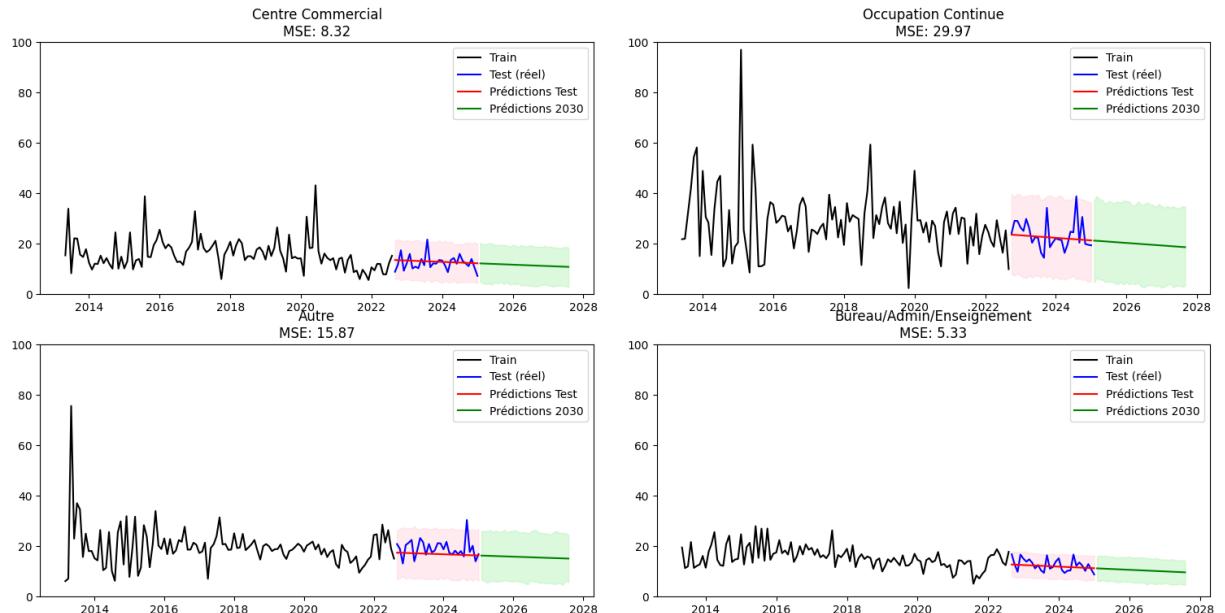


FIGURE 1.13 – Résultats obtenus par le modèle naïf.

## Chapitre 2

# Optimisation de l'intensité du portefeuille sous contraintes

Il était à l'origine voulu que nous utilisions un modèle de type Markowitz pour cette partie du projet. Cependant nous nous sommes rendu compte que le cadre ne s'y prêtait pas. Nous avons donc fait le choix de d'utiliser des méthodes d'optimisation sous contraintes "classiques" pour cette partie du projet.

### 2.1 Première approche : programmation linéaire

#### 2.1.1 Le modèle

Nous avons d'abord utilisé de la programmation linéaire pour résoudre le problème suivant : minimiser les émissions moyennes de notre portefeuille (pondéré par l'investissement) tout en garantissant un rendement minimum et un investissement maximum de 100 millions par actif. Formellement :

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \sum_{i=1}^n e_i x_i \\ \text{sous contraintes :} \quad & \sum_{i=1}^n x_i = 1 \\ & \sum_{i=1}^n r_i x_i \geq R_{\min} \\ & 0 \leq x_i \leq \frac{10^8}{I_{\text{total}}}, \quad \forall i = 1, \dots, n, \end{aligned}$$

où :

- $x_i$  : investissement dans l'actif  $i$ ,
- $e_i$  : émissions de  $CO_2$  associées à l'actif  $i$ ,
- $r_i$  : rendement attendu de l'actif  $i$ ,
- $R_{\min}$  : rendement minimum requis,
- $I_{\text{total}}$  : investissement total alloué,
- $n$  : nombre d'actifs dans le portefeuille.

On fait le choix de fixer un investissement maximum par actif pour deux raisons principales : sinon le modèle choisirait un unique actif et investirait tout le portefeuille dedans (l'actif tel que le rendement est supérieur au rendement minimum et avec la plus petite émission de GES), mais également car cela ne ferait pas de sens d'investir des sommes arbitrairement grandes dans des projets.

#### 2.1.2 Les résultats

À l'aide du modèle, obtient la frontière efficiente suivante :

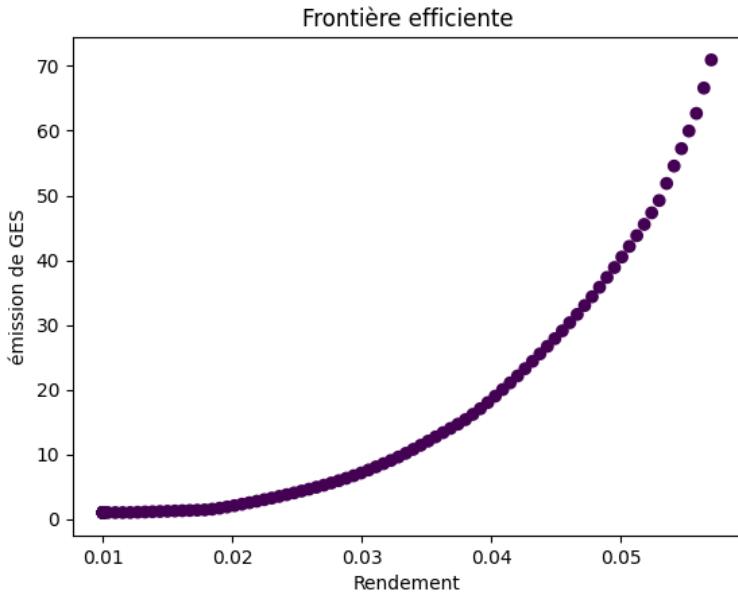


FIGURE 2.1 – Résultat du modèle de programmation linéaire

### 2.1.3 Remarques sur le modèle

On aurait également pu avoir la démarche inverse : maximiser le rendement en mettant une limite sur les émissions du portefeuille, mais le résultat aurait été le même. En effet, comme on peut l'observer sur la figure 2.2, les émissions et les rendements des deux programmes se superposent.

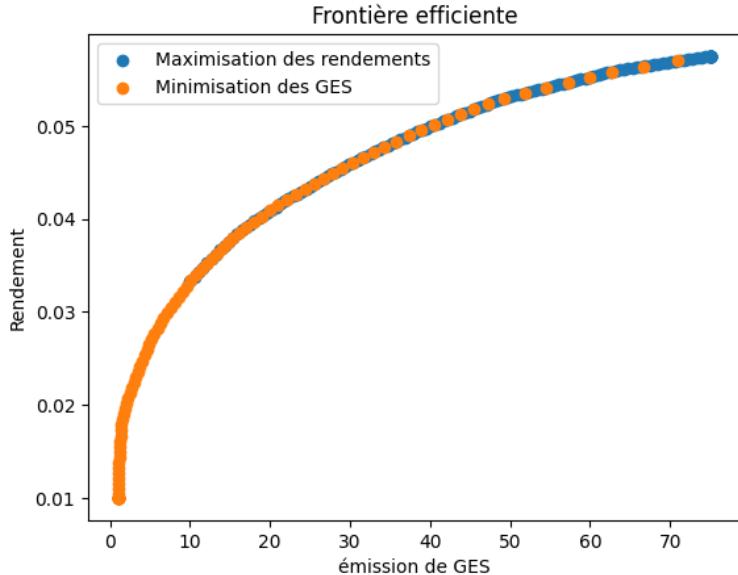


FIGURE 2.2 – Résultats pour la maximisation et la minimisation.

## 2.2 Optimisation du Portefeuille d’Investissement

Nous considérons un problème d’optimisation visant à allouer des proportions d’investissement entre plusieurs actifs en tenant compte du rendement et de l’impact environnemental (émissions de gaz à effet de serre, GES) en même temps. Nous cherchons à résoudre le problème d’optimisation suivant :

$$\begin{aligned}
\max_x \quad & \sum_{i=1}^n (R_{\text{new},i}^2 - \lambda GES_{\text{new},i}) x_i - \alpha \sum_{i=1}^n x_i^2 \\
\text{sous contraintes} \quad & \sum_{i=1}^n x_i = 1, \\
& x_i \geq 0, \quad \forall i \in \{1, \dots, n\}.
\end{aligned}$$

où :

- $R_{\text{new},i}$  est le taux de crédit normalisé pour l'actif  $i$ ,
- $GES_{\text{new},i}$  est l'émission de GES normalisée pour l'actif  $i$ ,
- $x_i$  est la proportion d'investissement à allouer à l'élément  $i$ ,
- $\lambda$  est un paramètre de pondération pour les émissions de GES contrôlant l'importance du critère environnemental,
- $\alpha$  est un coefficient de régularisation favorisant une allocation plus uniforme.

L'objectif est de maximiser une fonction combinant le rendement au carré et les émissions de GES pondérées, tout en respectant une contrainte d'allocation totale de 1 et une non-négativité des proportions.

La régularisation empêche l'algorithme d'investir uniquement dans l'actif qui maximise  $R^2 - \lambda GES$ , favorisant ainsi une meilleure diversification.

### 2.2.1 Choix des Paramètres $\alpha$ et $\lambda$

Le réglage de  $\alpha$  et  $\lambda$  influence fortement la stratégie d'investissement :

- Si  $\lambda$  est trop élevé, l'optimisation priviliege les actifs avec un faible GES, au détriment du rendement.
- Si  $\lambda$  est trop faible, l'allocation sera uniquement guidée par la maximisation de  $R^2$ .
- Un  $\alpha$  trop grand force une répartition quasi-égale entre tous les actifs, limitant ainsi la performance potentielle.
- Un  $\alpha$  trop faible peut entraîner une concentration excessive du portefeuille sur un unique actif.

### 2.2.2 Approche pour la Sélection des Paramètres

Afin de déterminer les valeurs optimales de  $\alpha$  et  $\lambda$ , plusieurs stratégies peuvent être envisagées comme Recherche en grille (Grid Search) : tester plusieurs combinaisons de paramètres et observer l'impact sur la répartition des proportions.

#### Optimisation de $\alpha$ à $\lambda$ fixé

Dans cette partie, on fixe  $\lambda = 10^{-3}$ , on fait varier le paramètre de régularisation  $\alpha$  et on observe le comportement du rendement et des émissions GES.

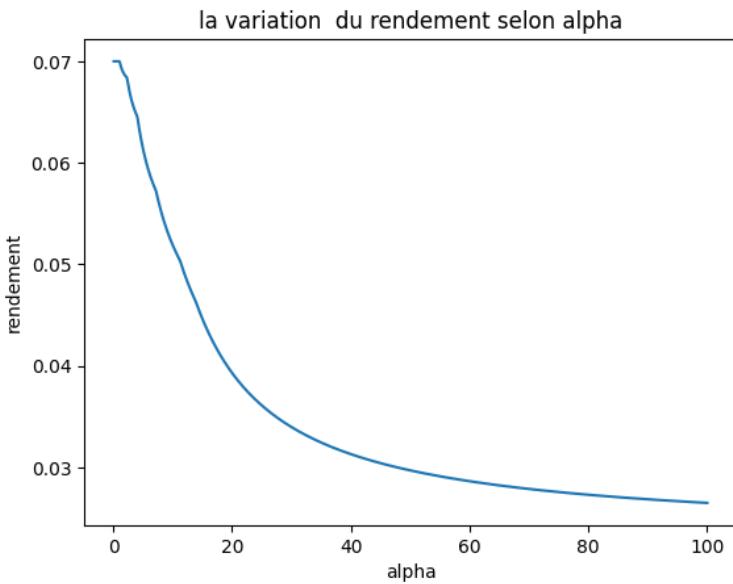


FIGURE 2.3 – Variation du rendement moyen du portefeuille  $\langle x, R \rangle$  en fonction de  $\alpha$

On remarque que le rendement moyen du portefeuille diminue d'autant plus que  $\alpha$  est grand, ce qui prévisible. En effet, dans le cas où  $\alpha = 0$  (pas de pénalisation des portefeuilles non diversifiés), l'algorithme investit dans l'actif au rendement maximal. Par contre, pour des valeurs plus grandes de  $\alpha$ , l'algorithme est obligé de diversifier le portefeuille ce qui fait diminuer le rendement.

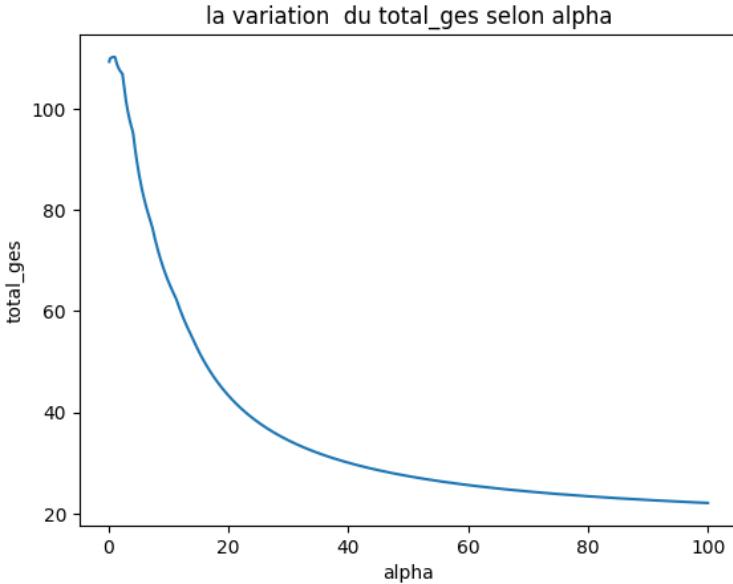


FIGURE 2.4 – Variation de l'estimation GES moyenne du portefeuille  $\langle x, G \rangle$  en fonction de  $\alpha$

On remarque que l'émission GES moyenne du portefeuille diminue lorsque  $\alpha$  augmente.

### Frontière efficiente

Dans cette partie, on va tracer la frontière efficiente pour notre approche d'optimisation. On prépare une grille de couples  $(\alpha, \lambda)$  et on calcule le portefeuille pour chaque couple de la grille. On calcule l'estimation GES maximale et le rendement minimal des portefeuilles de la grille. Après, on fixe à chaque fois un seuil d'émissions GES (resp. une borne inférieure de rendement) qu'il ne faut pas dépasser (resp. qu'il faut atteindre) inférieur à l'émission maximale (resp. supérieure au rendement minimal) sur la grille et on trouve le portefeuille de notre grille répondant à cette condition qui maximise le rendement (resp. minimise les émissions GES). On trace les courbes suivantes :

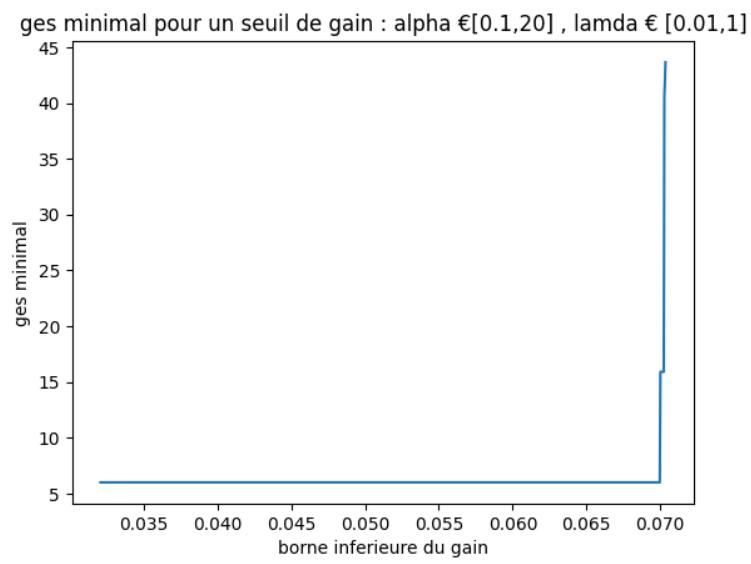


FIGURE 2.5 – GES minimal pour un seuil de gain ; grille  $100 \times 100$  avec  $\alpha \in [0.1, 20]$ ,  $\lambda \in [0.01, 1]$

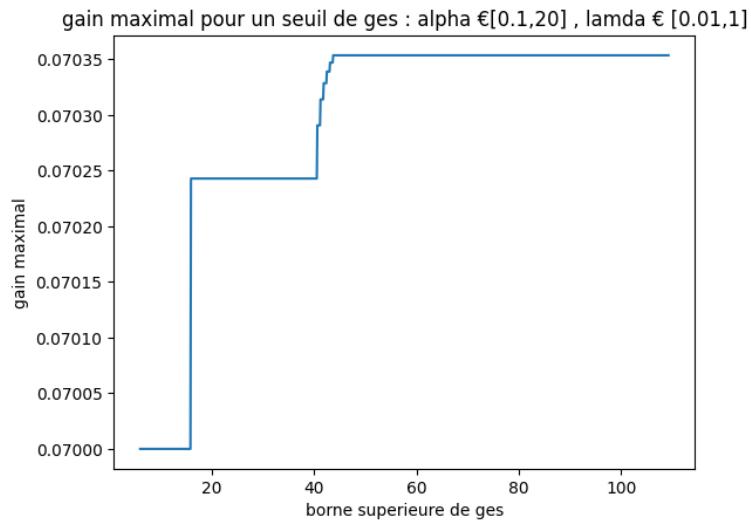


FIGURE 2.6 – Gain maximal pour un seuil de GES ; grille  $100 \times 100$  avec  $\alpha \in [0.1, 20]$ ,  $\lambda \in [0.01, 1]$

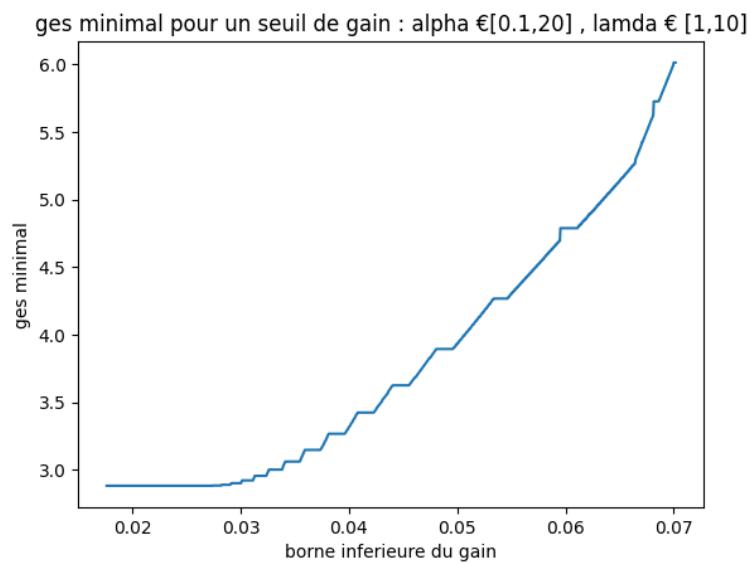


FIGURE 2.7 – GES minimal pour un seuil de gain ; grille  $100 \times 100$  avec  $\alpha \in [0.1, 20]$ ,  $\lambda \in [1, 10]$

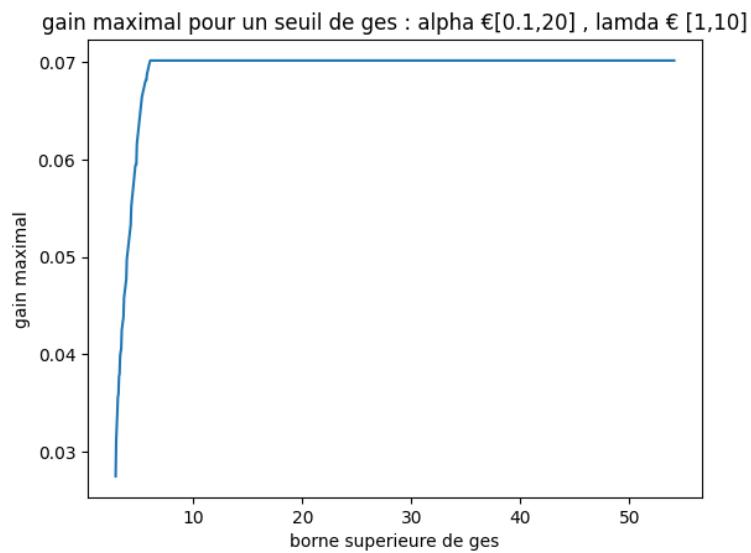


FIGURE 2.8 – Gain maximal pour un seuil de GES ; grille  $100 \times 100$  avec  $\alpha \in [0.1, 20]$ ,  $\lambda \in [1, 10]$

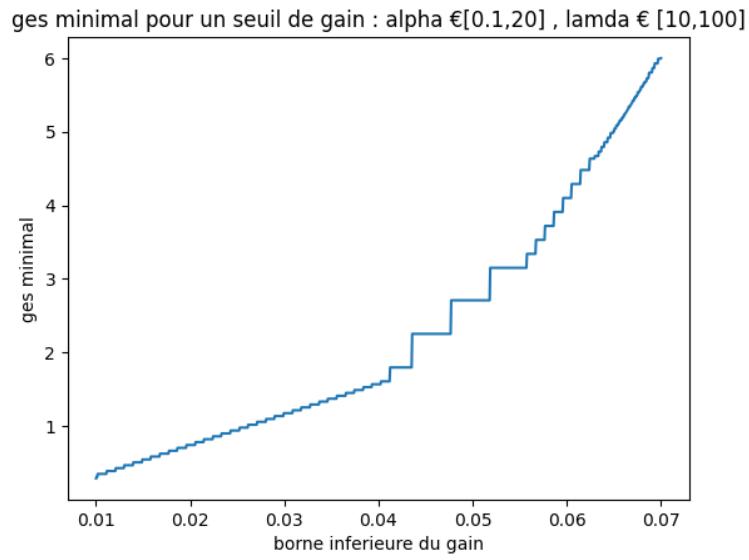


FIGURE 2.9 – GES minimal pour un seuil de gain ; grille  $100 \times 100$  avec  $\alpha \in [0.1, 20]$ ,  $\lambda \in [10, 100]$

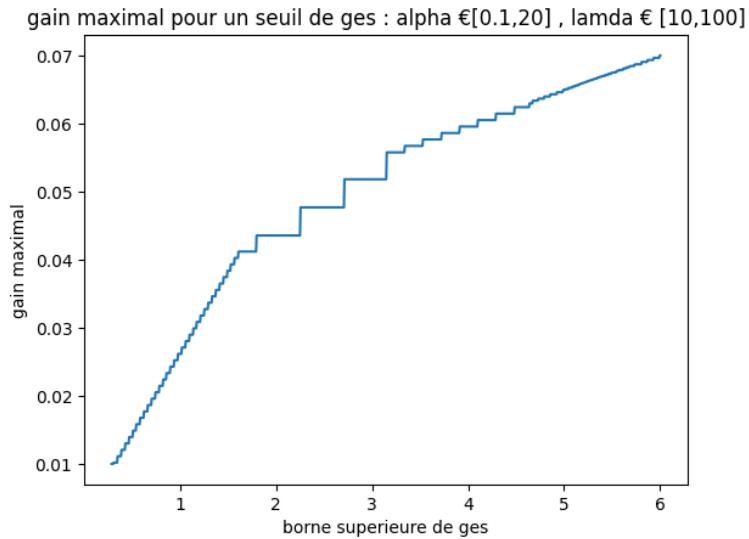


FIGURE 2.10 – Gain maximal pour un seuil de GES ; grille  $100 \times 100$  avec  $\alpha \in [0.1, 20]$ ,  $\lambda \in [10, 100]$

On remarque que lorsque l'ordre de grandeur de  $\lambda$  augmente, l'émission GES maximale atteinte par les portefeuilles de la grille diminue, ce qui est prévisible : On donne plus de poids à l'émission de GES.

Dans la suite, on travaille avec la dernière grille  $100 \times 100$  avec  $\alpha \in [0.1, 20]$ ,  $\lambda \in [10, 100]$

la variation du total\_ges selon alpha avec lamda=lamda\_best

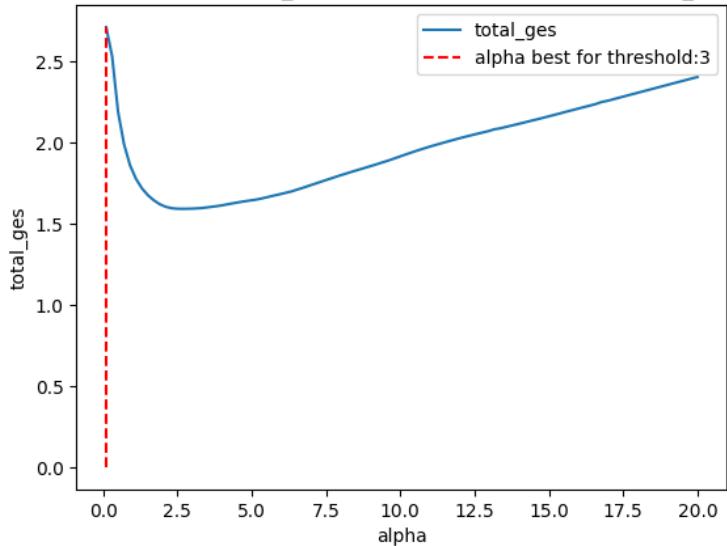


FIGURE 2.11 –

la variation du gain selon alpha avec lamda=lamda\_best

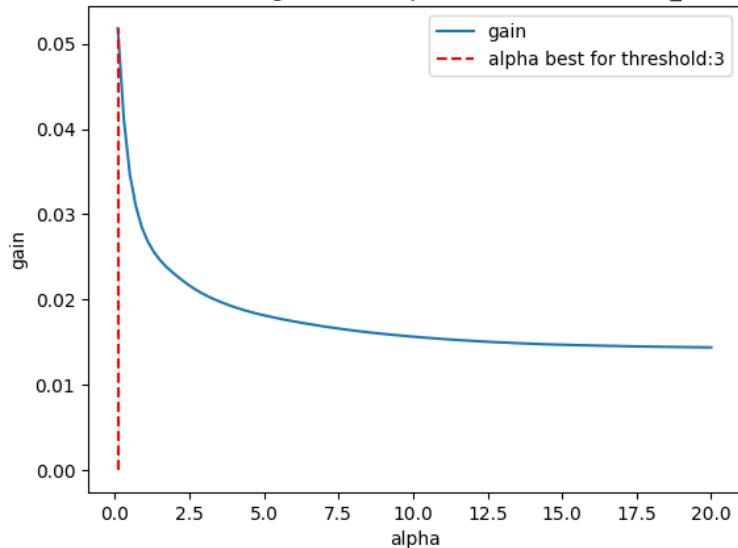


FIGURE 2.12 –

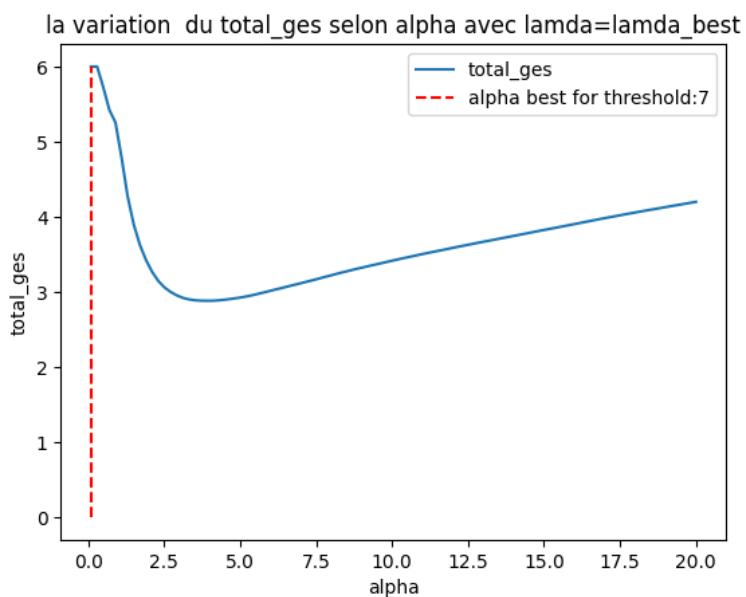


FIGURE 2.13 –

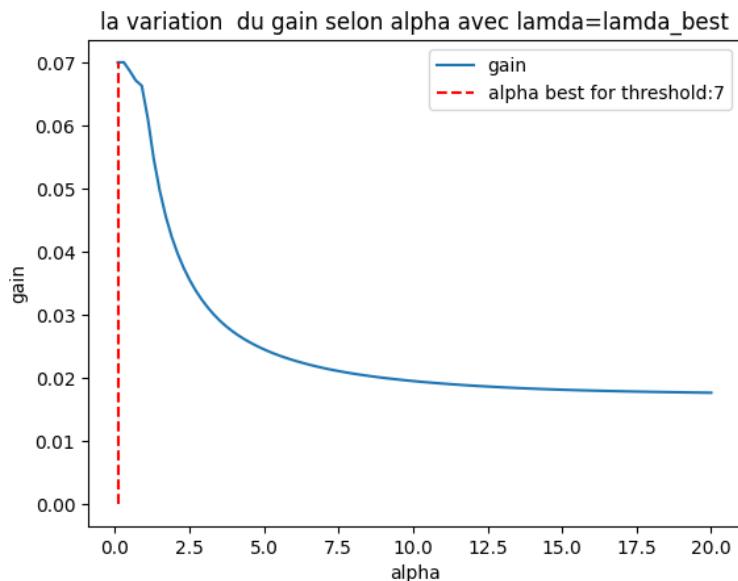


FIGURE 2.14 –

En conclusion, on utilise un seuil d'émissions GES de l'ordre de 7,  $\alpha\_best = 0.1$  et  $\lambda\_best = 10$ . Dans ce cas le total d'émissions est 6.39 et un rendement de 7%