

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira – Béjaïa
Faculté des Sciences Exactes
Département d'Informatique



Mémoire de Fin de Cycle

Présenté en vue de l'obtention du diplôme de :

Master en Informatique

Option : Génie Logiciel

Thème :

Prédiction basée sur l'apprentissage profond

Étude de cas : consommation d'électricité
au niveau de Sonelgaz Béjaïa

Réalisé par :

BENSEKHRIA Asma

CHOUF Dihya

Encadré par :

Dr. MIR Foudil

Jury :

Présidente : Dr. GADOUCHE Hania

Examineurs :

- Dr. ZERARGA Lotfi

- Dr. GHANEM Souhila

- Dr. AZIZOU Zahia

Année universitaire : 2024 – 2025

****Remerciements****

Nous tenons tout d'abord à exprimer notre gratitude et nos remerciements à **Dieu Tout-Puissant**, pour nous avoir accordé la santé, la patience et la persévérance nécessaires pour mener à bien ce travail.

Nous souhaitons exprimer notre profonde reconnaissance à notre encadrant, **M. Mir**, pour son encadrement bienveillant, ses conseils précieux et son soutien tout au long de ce projet. Sa disponibilité et ses orientations ont joué un rôle essentiel dans l'aboutissement de ce travail.

Nous tenons également à remercier chaleureusement **M. Dalil Hadjot**, notre maître de stage, pour sa confiance, son accompagnement et ses remarques pertinentes. Sa rigueur et son expertise ont été d'une grande aide dans la concrétisation de ce projet.

Nos remerciements vont également à **nos familles, nos amis** et à toutes les personnes qui nous ont soutenus de près ou de loin durant cette période. Leur appui moral et leurs encouragements constants ont été une source de motivation importante.

Enfin, nous exprimons notre gratitude à l'ensemble des **enseignants**, aux **membres du jury**, ainsi qu'à toutes les personnes ayant contribué, par leurs remarques ou leurs conseils, à la réalisation de ce travail.

Dédicaces

À ceux qui ont su faire naître la lumière dans mes instants de doute. À ceux dont l'amour silencieux a été ma plus grande force. À ceux qui m'ont portée, guidée, protégée, et qui, sans le savoir parfois, ont été l'essence même de ma réussite.

Je dédie ce mémoire à mes parents, les premiers témoins de mes rêves, mes plus fidèles soutiens. Votre amour pur, vos sacrifices sans limite, vos prières murmurées dans le silence... Rien ne serait possible sans vous. Ce travail, chaque mot, chaque étape franchie, est une manière de vous dire merci.

Merci d'avoir été là, toujours, sans condition, avec cette patience infinie et cette foi que seuls les parents savent offrir.

À toi, maman, mon refuge le plus doux. Ta tendresse m'a souvent sauvée, tes paroles m'ont apaisée, ton courage m'a appris à me relever. Tu es mon repère, mon pilier, mon souffle.

À toi, papa, pour ton regard fier, discret mais puissant. Pour tes silences remplis d'amour, ta présence rassurante, et ta force tranquille. Tu m'as appris à ne jamais abandonner.

Je dédie aussi ce mémoire à ma sœur, **Rozina**. Ta présence, ton écoute, ton énergie et ta capacité à me comprendre sans que je parle... Tu as été un rayon de lumière dans mes jours sombres. Tu comptes plus que tu ne l'imagines.

À mes frères, **Arslan** et **Rayane**, vous êtes ma joie, mes rires, mon ancrage. Vos gestes simples, vos attentions spontanées, vos sourires, ont été mon réconfort discret mais profond. Vous êtes une part essentielle de mon équilibre.

Ce mémoire, bien plus qu'un travail académique, est une étape de vie. Et je voulais que vous en fassiez partie, pleinement, sincèrement, éternellement.

Dihya

Dédicaces

Je dédie ce travail principalement à mes parents, ceux qui m'ont toujours soutenue et qui ont été présents à chaque pas, dans chaque moment spécial de ma vie. Vous êtes ma force, mon repère, ce qui me pousse à croire en un avenir meilleur. Les mots ne suffiront jamais à exprimer tout l'amour que je vous porte, ni à décrire celui que j'ai reçu de vous.

À toi, maman, mon refuge... Ton regard bienveillant, ta tendresse et tes prières m'ont portée dans les moments les plus difficiles. Tu es cette présence que rien ne peut remplacer, cette main toujours tendue, même lorsque je suis loin. *I onjoy you, maman.*

À toi, papa. Ta force tranquille, ton respect silencieux pour mes rêves, et ta confiance m'ont poussée à vouloir te rendre fière. Tu as su me guider sans jamais imposer, m'enseigner sans juger, et me soutenir sans jamais faillir.

À mon frère Oussama, Tu es l'épaule sur laquelle je peux m'appuyer, le cœur fort derrière les gestes simples, et la force tranquille qui me rappelle que je ne suis jamais seule. Je te souhaite, de tout mon cœur, un avenir à la hauteur de ta générosité, de ta sagesse et de ta bonté.

Aux personnes proches de moi : Mes cousines Yasmine, Leticia, Nardjes, Chaïma, Inès et Katia — mes sœurs de cœur. Avec vous, j'ai grandi, ri, parfois pleuré, mais toujours aimé profondément. Merci pour chaque instant partagé.

À mes amies Thanina, Dihya, Aïcha et Nora, merci pour votre présence dans les petits et les grands moments.

Asma.

Table des matières

Remerciements	i
Table des matières	iv
Table des figures	viii
Abréviations	x
Liste des tableaux	xi
Introduction Générale	1
1 Les méthodes classiques de prédictions	3
1.1 Introduction	3
1.2 Les séries temporelles	4
1.2.1 Définition des séries temporelles	4
1.2.2 Composantes d'une série temporelles	4
1.2.3 Modélisation des séries temporelles	6
1.2.3.1 Méthodes classiques	6
1.2.3.2 Modèle multiplicatif	8
1.2.3.3 Méthodes de lissage : Moyenne mobile	11
1.3 Processus d'analyse et de prédiction des series temporelles	13
1.3.1 Stationnarité et transformations des séries temporelles	13
1.3.1.1 La stationnarité	13
1.3.1.2 La transformation des séries temporelles	14
1.3.2 Processus stationnaire	15
1.3.2.1 Processus ARMA	15
1.3.3 Processus non stationnaire	20
1.3.3.1 Processus ARIMA	20
1.3.3.2 Processus SARIMA	21
1.3.3.3 Processus SARIMAX	23
1.4 Conclusion	25

2	Les méthodes de deep learning pour la prédiction	26
2.1	Introduction :	26
2.2	L'intelligence artificielle	26
2.3	Le Machine Learning	26
2.3.1	Types d'apprentissage	27
2.4	Définition du Deep Learning	28
2.5	Les Réseaux de Neurones Artificiels (RNA)	28
2.5.1	Structure d'un Réseau de Neurones	28
2.5.2	Types de Réseaux de Neurones	31
2.5.3	Processus d'Apprentissage d'un Réseau de Neurones	33
2.5.4	Les Problèmes Des Réseaux de Neurones Récurents (RNN) :	35
2.5.4.1	Problème de la disparition du gradient (Vanishing Gradient)	36
2.5.4.2	Problème d'explosion du gradient (Exploding Gradient)	36
2.5.4.3	Difficulté à capturer des dépendances à long terme :	36
2.5.5	Long Short-Term Memory (LSTM)	36
2.5.5.1	Propagation Avant (Forward Propagation) :	37
2.5.5.2	Calcul D'Erreur :	39
2.5.5.3	Rétropropagation à Travers le Temps (Backpropagation Through Time) :	39
2.6	Conclusion	39
3	Analyse des Besoins et Conception	41
3.1	Introduction	41
3.2	Présentation de Sonelgaz	41
3.2.1	Historique et rôle stratégique	41
3.2.2	La transformation de SONELGAZ en groupe	42
3.2.3	Les catégories de clients de SONELGAZ	44
3.3	Présentation de l'organisme d'accueil : <i>Sonelgaz Distribution – Direction de Béjaïa</i>	44
3.3.1	Organisation de Sonelgaz Distribution – Direction de Béjaïa	45
3.3.1.1	Division des Relations Commerciales	45
3.4	Problématique	46
3.5	Solutions proposées	47
3.5.1	Fonctionnalités principales de l'application	48
3.6	Les méthodologies de gestion de projet	48
3.6.1	les Méthodes Agile et Traditionnelle	48
3.6.2	Le cadre de travail Scrum :	49
3.6.2.1	Les Rôles principaux de Scrum	49
3.6.2.2	Le processus Scrum et ses concepts clés	50
3.7	Délimitation du domaine d'étude	51
3.7.1	Répartition des rôles selon la méthode Scrum	51

3.7.2	Identification des acteurs	51
3.8	Organisation du projet selon SCRUM	52
3.8.1	Product Backlog	52
3.8.2	Planification des sprints	54
3.9	Diagramme de contexte	55
3.10	Diagramme de cas d'utilisation	57
3.11	Description des cas d'utilisation	57
3.12	Diagramme de séquence	58
3.12.1	Cas d'utilisation s'authentifier	59
3.12.2	Cas d'utilisation "lancer une prédiction"	60
3.12.3	Cas d'utilisation "visualiser les consommations d'un client"	62
3.12.4	Cas d'utilisation "ajouter un client"	64
3.12.5	Cas d'utilisation "insertion de nouvelles données"	65
3.13	Diagramme de classe	67
3.14	Conclusion	68
4	Réalisation	69
4.1	Introduction	69
4.2	Environnement de développement de l'application	69
4.2.1	Langages de programmation	69
4.2.2	Framework de programmation	69
4.2.3	Outils et logiciels utilisé	70
4.2.4	Architecture	70
4.2.5	Librairies Python utilisés	71
4.3	Modélisation des méthodes de prédiction appliquées	72
4.3.1	Modélisation dynamique	72
4.3.1.1	Méthode Box-Jenkins	72
4.4	Processus de prédiction en LSTM	73
4.4.1	Préparation des Données	73
4.4.2	Construction et Entraînement du Modèle LSTM	74
4.5	Aperçu de l'application	75
4.5.1	Développement de l'interface web	75
4.5.2	Les interfaces réalisées	75
4.5.2.1	Interface D'accueil	76
4.5.2.2	Interface profil	76
4.5.2.3	Interface Tableau de bord	76
4.5.2.4	Interface Clients	77
4.5.2.5	Interface historique	78
4.5.2.6	Interface prédiction	79
4.5.2.7	Interface Dashboard Client	80

4.6	Analyse et comparaison des résultats des méthodes	80
4.7	Conclusion	81
4.8	Conclusion et perspectives	82
	Bibliographie	83
	Résumé et Abstract	85

Table des figures

1.1	Représentation graphique de l'évolution mensuelle des cas diagnostiqués de cancer du sein sur la période de 1991 à 1997 [21].	4
1.2	Représentation graphique du dataset utilisée avec ses 3 composantes	5
1.3	Prédiction avec la méthode additif	8
1.4	Prediction avec méthode multiplicatif	11
1.5	ACF et PACF apres la différenciation ordinaire	21
1.6	Prédiction avec la méthode ARIMA.	21
1.7	ACF et PACF apres une différenciation saisonnière	22
1.8	Prédiction avec le modèle SARIMA	23
1.9	Prédiction avec le modèle SARIMAX	25
2.1	Représentation graphique des catégories de données séparées par une frontière de décision [16]	27
2.2	La structure globale des neurones artificiels [22]	29
2.3	Structure détaillée d'un neurone artificiel [9]	29
2.4	la representation graphique des 4 fonctions d'activations	31
2.5	Représentation d'un Réseau de Neurones Récurrent (RNN) déplié [22]	36
2.6	Cellule LSTM [22]	37
3.1	Logo de la société Sonelgaz	42
3.2	Organigramme de la Direction de Distribution de Sonelgaz à Béjaïa [28].	45
3.3	Diagramme de contexte	55
3.4	Diagramme de cas d'utilisation	57
3.5	Diagramme de séquence détaillé du cas d'utilisation "S'Authentifier"	60
3.6	Diagramme de séquence du cas d'utilisation "Lancer une prédiction"	62
3.7	Diagramme de sequance du cas d'utilisation "visualiser des consommation et les predictions d'un client"	63
3.8	Diagramme de séquence cas d'utilisation "ajouter un client "	65
3.9	Diagramme de séquence cas d'utilisation "insertion de nouvelles données "	67
3.10	Diagramme de classe	68
4.1	Interface Accueil	76

4.2	Interface profil de l'administrateur	76
4.3	Interface tableau de bord	77
4.4	Interface clients	77
4.5	Interface Visualiser les données du client	78
4.6	Interface Historique	78
4.7	Interface Prédiction	79
4.8	Interface Dashboard Client	80

Abréviations

Liste des abréviations

- **AIC** : Akaike Information Criterion
- **BIC** : Bayesian Information Criterion
- **AR** : AutoRégressive
- **ARMA** : AutoRegressive Moving Average
- **ARIMA** : AutoRegressive Integrated Moving Average
- **SARIMA** : Seasonal AutoRegressive Integrated Moving Average
- **SARIMAX** : Seasonal AutoRegressive Integrated Moving Average with eXogenous variables
- **ADF** : Dickey-Fuller augmentée
- **ACF** : Fonction Auto-Corrélation
- **PACF** : Fonction Auto-Corrélation Partielle
- **BDD** : Base de données
- **CSS** : Cascading Style Sheets
- **CVS** : Série Corrigée de Variations Saisonnières
- **VSCoDe** : Visual Studio Code
- **DL** : Deep Learning
- **RNN** : Recurrent Neural Networks
- **DNN** : Deep Neural Network
- **CNN** : Convolutional Neural Networks
- **IA** : Intelligence Artificielle
- **LSTM** : Long Short Term Memory
- **ML** : Machine Learning
- **SC** : Série Chronologique
- **UML** : Unified Modeling Language
- **STL** : Seasonal and Trend decomposition using Loess

Liste des tableaux

1.1	Décomposition d'une série temporelle	6
1.2	Signification des symboles utilisés dans le modèle AR	17
3.1	Product Backlog	54
3.2	Planification des sprints	54
3.3	Description tabulaire des messages échangés entre le système et les acteurs	56
3.4	Structure type d'un cas d'utilisation	58
3.5	Description du cas d'utilisation "S'authentifier"	59
3.6	Description du cas d'utilisation "Lancer une prédiction "	61
3.7	Description du cas d'utilisation "visualiser des consommation d'un client "	63
3.8	Description du cas d'utilisation "ajouter un client "	64
3.9	Description du cas d'utilisation "insertion de nouvelles données "	66

Introduction Générale

Dans un contexte mondial marqué par une urbanisation croissante et une accélération du développement économique et démographique, la demande en électricité connaît une augmentation continue. Cette croissance soutenue exerce une pression considérable sur les réseaux de distribution, qui doivent répondre à des besoins énergétiques de plus en plus complexes, tout en garantissant la stabilité et la fiabilité de l'alimentation électrique. En Algérie, cette problématique prend une dimension stratégique pour l'entreprise publique Sonelgaz. Le seul acteur central du secteur énergétique national.

Face à ces enjeux, la prévision de la consommation électrique devient un levier essentiel pour la gestion optimale des ressources. Elle permet non seulement d'ajuster la production à la demande en temps réel, mais aussi d'anticiper les pics de consommation, de minimiser les pertes techniques, et de renforcer l'efficacité globale du réseau. Une prévision précise constitue ainsi un outil d'aide à la décision incontournable, tant à l'échelle régionale que nationale.

Parallèlement, les progrès spectaculaires réalisés dans le domaine des technologies numériques, et plus particulièrement en intelligence artificielle (IA), ont ouvert la voie à des approches innovantes de la modélisation prédictive. Parmi celles-ci, les techniques de deep learning se distinguent par leur capacité à analyser de vastes ensembles de données, à détecter des patterns non linéaires et à apprendre automatiquement les dynamiques complexes du comportement énergétique. Ces modèles, souvent plus performants que les méthodes statistiques classiques, permettent une prédiction plus fine, plus rapide et plus fiable des besoins en électricité.

Dans cette optique, le présent mémoire s'inscrit dans une démarche de recherche et de développement orientée vers l'intelligence artificielle appliquée à l'énergie. Il vise à concevoir et implémenter une application intelligente de prévision de la consommation électrique, destinée à la Direction de Distribution de Sonelgaz à Béjaïa. L'outil développé intégrera un tableau de bord interactif permettant aux responsables de visualiser les données historiques, d'obtenir des prévisions automatisées, et de détecter plus efficacement les tendances futures. L'objectif est ainsi de contribuer à une gestion plus proactive et plus intelligente de la demande électrique, en exploitant les capacités des technologies modernes au service d'un secteur vital.

Ce mémoire est structuré comme suit :

- Le premier chapitre : est consacré à l'étude des séries temporelles et aux méthodes classiques de prédiction. Il présente notamment les modèles ARIMA, SARIMA et leurs variantes, tout en expliquant les notions essentielles liées aux séries chronologiques.
- Le deuxième chapitre : traite des méthodes de deep learning. Il met en avant l'IA , le machine learning et plus particulièrement les réseaux de neurones comme LSTM, qui ont montré leur efficacité dans le traitement des données séquentielles.
- Le troisième chapitre : présente l'étude de cas menée avec Sonelgaz. Il décrit les besoins fonctionnels et techniques, les spécificités de l'environnement de travail, ainsi que les choix méthodologiques adoptés pour répondre aux exigences du projet.
- Enfin, le quatrième chapitre : détaille la réalisation de l'application. Il décrit les outils utilisés, les étapes de développement, l'intégration des modèles de prédiction dans le système, ainsi que les résultats obtenus à travers l'interface développée.

Nous terminerons par une conclusion générale résumant les résultats obtenus et proposant des perspectives d'amélioration.

Chapitre 1

Les méthodes classiques de prédictions

1.1 Introduction

Les séries temporelles constituent une succession de données collectées à des périodes régulières et donnent une vue dynamique de l'évolution d'un phénomène dans le temps. Ces dernières sont présentes dans plusieurs domaines à citer la finance, la météorologie, l'économie, la gestion des ressources, etc., où la compréhension des tendances, saisonnalités et variations aléatoires est vitale pour la prise de décision. C'est dans ce cadre que les modèles de prédiction interviennent pour permettre de prédire les futurs comportements à partir de l'historique.

En fin de compte, ce chapitre s'accorde avec ces considérations en examinant plusieurs facettes de la modélisation des séries temporelles. Les présentes démarches comprennent les modèles additifs et multiplicatifs, ainsi qu'AutoRegressive Moving Average (ARMA), Seasonal AutoRegressive Integrated Moving Average with exogenous variables (SARIMAX), ARIMA, et SARIMA, chacune offrant des perspectives pertinentes sur la modélisation des différentes composantes d'une série temporelle. Les modèles additifs et multiplicatifs décomposent efficacement une série en ses éléments constitutifs, tandis que d'autres modèles, ARMA, ARIMA, SARIMA et SARIMAX permettent de modéliser les dépendances temporelles et les effets saisonniers de manière plus sophistiquée.

Ainsi, l'objectif de ce travail est double : pour l'un, comprendre les forces et les limites de chaque modèle en les appliquant aux données réelles et, pour l'autre, évaluer sa performance prédictive dans différents scénarios. Cela nous aide également à identifier les membres les plus adaptés au modèle pour une application pratique, tout en soulignant simultanément les défis associés à la modélisation des séries chronologiques, tels que la désignation de valeurs pour les membres, la gestion des données manquantes et similaires.

1.2 Les séries temporelles

1.2.1 Définition des séries temporelles

Une série temporelle est un ensemble de données ordonné dans un intervalle de temps réguliers, comme des heures, des jours, des mois ou des années. Elle permet de suivre l'évolution dans le temps, c'est un moyen très utile pour le suivi de l'évolution des données en détectant les tendances ou des fluctuations [18].

On peut représenter ces données par : $X_t = x_t$, où t prend des valeurs numériques entières comme $t = 1, 2, \dots, T$

Les données utilisées dans cette étude représentent l'évolution mensuelle des cas diagnostiqués de cancer du sein sur la période de 1991 à 1997. Elles sont exprimées sous forme de pourcentage et reflètent les variations mensuelles de l'incidence du cancer du sein au fil du temps [21].

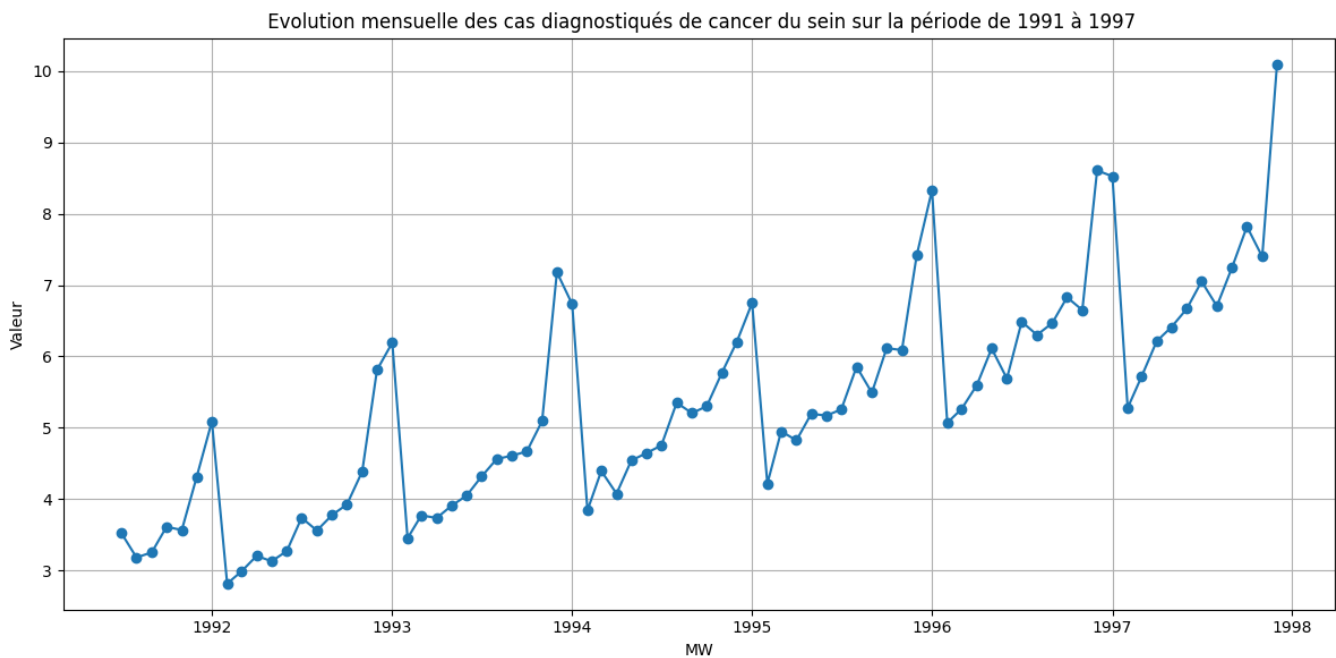


FIGURE 1.1 – Représentation graphique de l'évolution mensuelle des cas diagnostiqués de cancer du sein sur la période de 1991 à 1997 [21].

1.2.2 Composantes d'une série temporelles

Une série temporelles peut être décomposée en trois composantes principales . En séparant une série chronologique en ces trois composantes, on peut mieux analyser, modéliser et prévoir son comportement future. :

La tendance (Trend)

Notée $T(t)$ (voir la formule 1.12) représente le comportement de la série temporelle sur le long terme. En d'autres termes, c'est un mouvement uniforme qui peut être croissant, décroissant ou stable [18].

La composante saisonnière (saisonnalité)

Notée $S(t)$ (voir la formule 1.5) la composante saisonnière représente les changements périodiques de la série temporelle qui se répètent à intervalles fixes. Elle est généralement liée à des effets saisonniers [18].

La composante résiduelle (Bruit ou Résidu)

Notée ϵ (voir la formule 1.9), correspond à des fluctuations irrégulières, en général de faible intensité, mais de nature aléatoire et inattendue. [18].

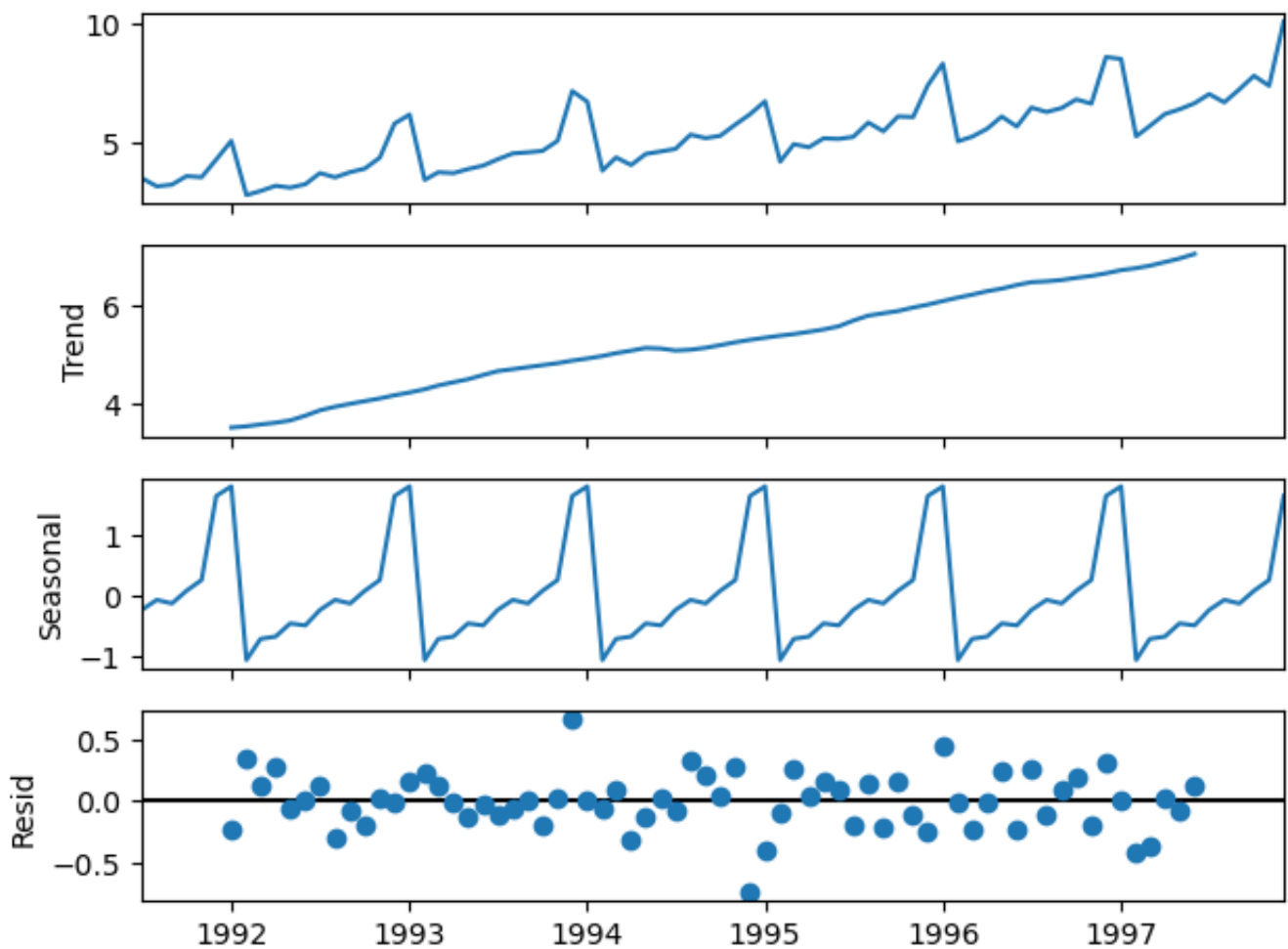


FIGURE 1.2 – Représentation graphique du dataset utilisée avec ses 3 composantes

1.2.3 Modélisation des séries temporelles

1.2.3.1 Méthodes classiques

Modèle additif

Un modèle additif est un type de modèle utilisé en analyse des séries temporelles pour décomposer une série en plusieurs composantes qui s'ajoutent les unes aux autres. Ces composantes représentent généralement par une tendance (T), une saisonnalité (S) et un résidu (R) [14].

Il est particulièrement adapté aux séries temporelles où les variations saisonnières ont une amplitude constante dans le temps. Si l'amplitude des variations saisonnières augmente avec le niveau de la tendance, un modèle multiplicatif est plus approprié.

- Formule générale d'un modèle additif :

$$Y(t) = T(t) + S(t) + R(t) \quad (1.1)$$

où :

Symbole	Signification
$Y(t)$	Valeur observée à l'instant t
$T(t)$	Composante de tendance à l'instant t
$S(t)$	Composante saisonnière à l'instant t
$R(t)$	Résidu à l'instant t

TABLE 1.1 – Décomposition d'une série temporelle

Définir la formule du modèle additif : Pour cela, il est essentiel de suivre des étapes précises afin de déterminer les valeurs des composantes de la formule mathématique :

1. **La tendance :** L'estimation de $T(t)$ (voir la formule 1.12) se calcule avec la régression linéaire

- $T(t)$ peut être modélisée par une droite de la forme :

$$T(t) = a \cdot t + b \quad (1.2)$$

où :

- a : Coefficient de pente.
- b : Ordonnée à l'origine.

Calculer les coefficients a et b :

La méthode des moindres carrés est utilisée afin d'estimer le a et le b .

Formules :

$$a = \frac{\sum_{i=1}^n (Y_t \cdot t) - n \cdot \bar{Y} \cdot \bar{t}}{\sum_{i=1}^n t^2 - n \cdot \bar{t}^2} \quad (1.3)$$

$$b = \bar{Y} - a \cdot \bar{t} \quad (1.4)$$

où :

- \bar{Y} est la moyenne des valeurs observées Y_t ,
- \bar{t} est la moyenne des indices temporels t ,
- n est le nombre de points de données.

2. **Saisonnalité (Composante Saisonnière)** : La première étape consiste à décomposer les données en calculant les coefficients saisonniers $S_{i,j}(t)$, (voir la formule 1.5). Ensuite, on détermine la saisonnalité brute S_j (voir la formule 1.6), puis on calcule sa moyenne \bar{S}_j (voir la formule 1.7) afin d'obtenir la composante saisonnière ajustée $S_{\text{ajustée } j}$, (voir la formule 1.8).

Détendre les données :

$$S_{i,j}(t) = Y(t) - T(t) \quad (1.5)$$

où :

- i : représente une période saisonnière (dans notre exemple ici c'est 24h).
- j : désigne un groupe d'observations dans cette période saisonnière (par exemple, toutes les valeurs pour une période de 1 heure à travers plusieurs jours ou tous les jours de la semaine).

Composante saisonnière brute : Pour estimer S_j pour un groupe d'observations j , on calcule $S_{j,i}$ des différentes périodes de ce groupe.

Elle est donnée par la formule suivante :

$$S_j = \frac{1}{|j|} \sum_{t \in j} S_{i,j} \quad (1.6)$$

où :

- $|j|$: C'est le nombre d'observations dans le groupe j .
- $\sum_{t \in j} S_{i,j}$: C'est la somme des valeurs de saisonnalité $S_{i,j}$ pour toutes les observations dans le groupe j .

Ensuite , la moyennes des composant saisonier est calculer pour assurer qu'elle est proche de zéro dans le modèle additif, afin que les fluctuations saisonnières ne biaisent pas l'analyse.

$$\bar{S}_j = \frac{1}{|i|} \sum_{j=1}^{|i|} S_j \quad (1.7)$$

Enfin , après avoir calculé cette moyenne saisonnière $\overline{S_j}$, on soustrait cette moyenne de chaque valeur pour obtenir une version ajustée de la saisonnalité :

$$S_{\text{ajustée } j} = S_j - \overline{S_j} \quad (1.8)$$

3. **le résidu (bruit)** : Le résidu $R(t)$ (1.5) représente les fluctuations non expliquées par la tendance et la saisonnalité.

La formule pour calculer le résidu est la suivante :

$$R(t) = Y(t) - T(t) - S_{\text{ajustée } j} \quad (1.9)$$

Formule finale de la prédiction :

$$Y_{\text{prédit}}(t) = (a \cdot t + b) + S_{\text{ajustée } j} + (Y_t - T(t) - S_{\text{ajustée } j}) \quad (1.10)$$

- En utilisant ces étapes sur notre dataset on obtient ce résultat , ou les prédictions sont représenté en vert :

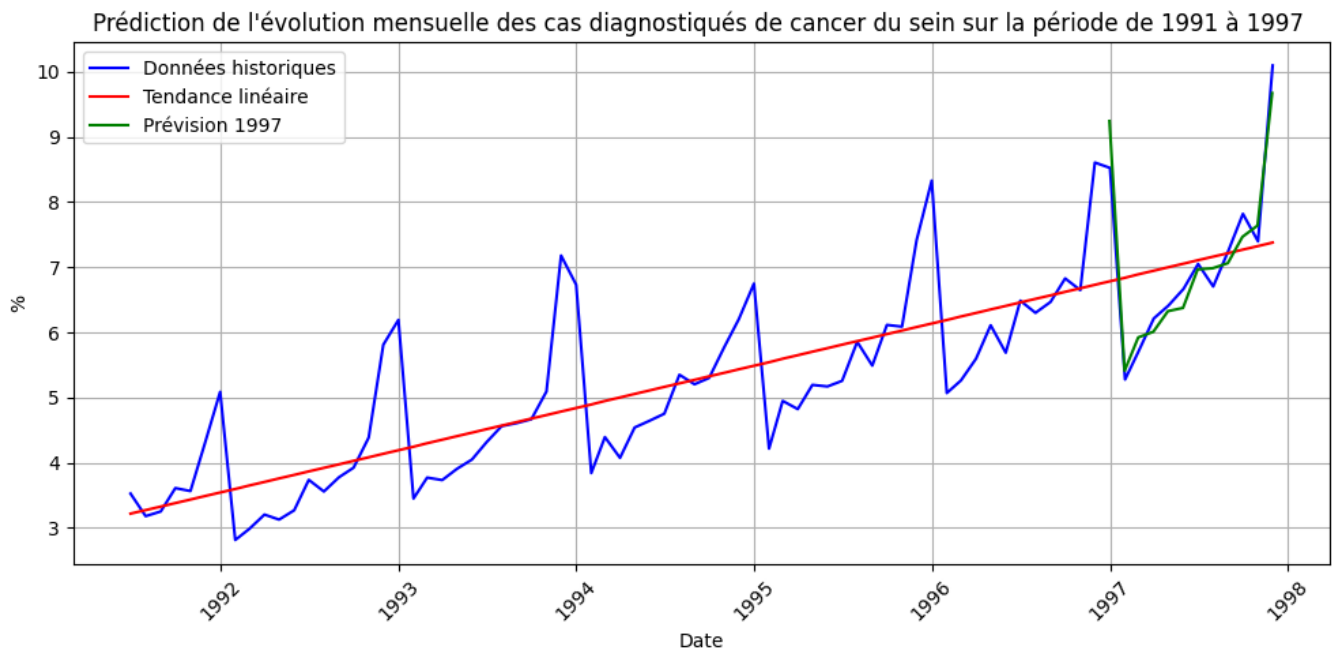


FIGURE 1.3 – Prédiction avec la méthode additif

1.2.3.2 Modèle multiplicatif

Le modèle multiplicatif est une méthode de décomposition d'une série temporelle qui suppose que les différentes composantes de la série (tendance, saisonnalité et résidu) interagissent de manière

multiplicative plutôt qu'additive. Cela signifie que l'effet de chaque composante est proportionnel au niveau de la série, ce qui est particulièrement utile lorsque l'amplitude des variations saisonnières ou des fluctuations aléatoires dépend de la tendance . [20]

Mathématiquement, le modèle multiplicatif s'exprime comme suit :

$$Y(t) = T(t) \times S(t) \times R(t) \tag{1.11}$$

Définir la formule du modèle multiplicatif : cette fois ci également il est donc crucial de suivre une série d'étapes méthodiques pour identifier les valeurs des différentes composantes de la formule mathématique .

1. **La tendance :** sa formule est represente comme suit :

$$T(t) = a \cdot t + b \tag{1.12}$$

les coefficients de la formule sont calculer de la même manière que celle du modele additif .

2. **Saisonnalité (Composante Saisonnière) :** La première étape consiste à détendre les données en calculant les coefficients saisonniers à partir des données observées, puis à calculer la saisonnalité brute et à l'ajuster à la fin.

Détendre les données :

$$S_{i,j}(t) = \frac{Y_t}{T_t} \tag{1.13}$$

où :

- i : représente une période saisonnière (par exemple, 24h).
- j : désigne un groupe d'observations dans cette période saisonnière (par exemple, toutes les valeurs pour une période d'une heure à travers plusieurs jours ou pour tous les jours de la semaine).

Composante saisonnière brute : Pour estimer la composante saisonnière brute S_j d'un groupe d'observations j , on calcule la moyenne des valeurs saisonnières $S_{j,i}$ pour les différentes périodes de ce groupe.

La composante saisonnière brute est obtenue par la formule suivante :

$$S_j = \frac{1}{|j|} \sum_{t \in j} S_{i,j} \tag{1.14}$$

où :

- $|j|$: représente le nombre d'observations dans le groupe j .

- $\sum_{t \in j} S_{i,j}$: la somme des valeurs saisonnières $S_{i,j}$ pour toutes les observations du groupe j .

Pour garantir que les fluctuations saisonnières ne biaisent pas l'analyse dans le modèle multiplicatif, la moyenne des composantes saisonnières est calculée de manière à ce qu'elle soit proche de 1.

$$\overline{S}_j = \frac{1}{|j|} \sum_{j=1}^{|j|} S_j \quad (1.15)$$

Enfin, une fois que cette moyenne saisonnière \overline{S}_j est déterminée, chaque valeur saisonnière est divisée par cette moyenne pour obtenir la version ajustée de la saisonnalité :

$$S_{\text{ajustée } j} = \frac{S_j}{\overline{S}_j} \quad (1.16)$$

- Le résidu (bruit) :** Le résidu $R(t)$ représente les fluctuations qui ne sont pas expliquées par la tendance et la saisonnalité.

La formule pour calculer le résidu devient alors :

$$R(t) = \frac{Y(t)}{T(t) \cdot S_{\text{ajustée } j}} \quad (1.17)$$

La formule finale de la prédiction : La prédiction est obtenue en combinant la tendance, la saisonnalité ajustée et le résidu comme suit :

$$Y_{\text{prédit}}(t) = (a \cdot t + b) \cdot S_{\text{ajustée } j} \cdot \left(\frac{Y(t)}{T(t) \cdot S_{\text{ajustée } j}} \right) \quad (1.18)$$

- En suivant ces étapes en pratique sur le même dataset présenté au départ on obtient ce resultat, avec les prédictions représenté en vert :

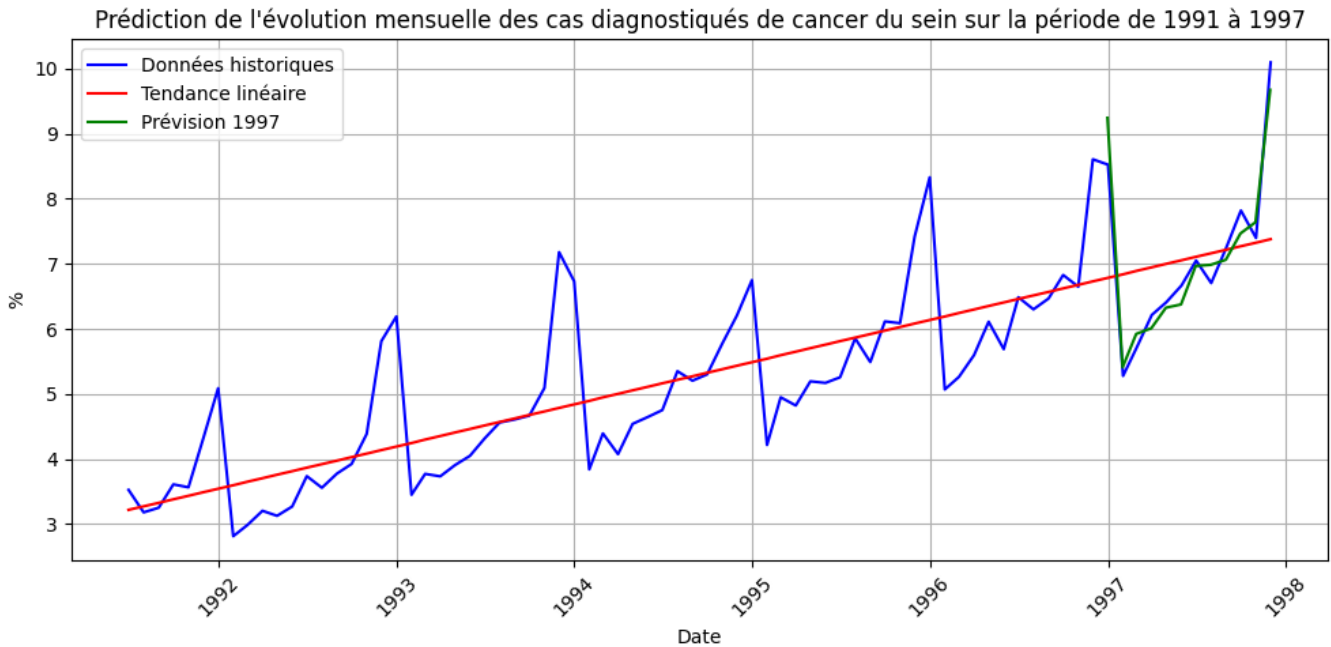


FIGURE 1.4 – Prediction avec méthode multiplicatif

1.2.3.3 Méthodes de lissage : Moyenne mobile

La moyenne mobile est une méthode de lissage largement utilisée en analyse de séries temporelles. Son objectif est de réduire les fluctuations aléatoires afin de faire ressortir les tendances de fond. Développée tout au long du 20 siècle, cette technique se décline en plusieurs variantes selon les besoins d'analyse, notamment la moyenne simple, centrée, pondérée et exponentielle [8] .

1. La Moyenne Mobile Arithmétique (SMA - Simple Moving Average)

La SMA est une méthode simple et largement utilisée pour lisser les séries temporelles. Elle calcule la moyenne des n dernières valeurs, incluant le point actuel. Cette technique permet de dégager les tendances générales en atténuant les fluctuations à court terme. Toutefois, elle reste peu réactive aux variations rapides, ce qui limite son efficacité pour les prévisions à court terme. [8].

Formule :

$$SMA(n) = \frac{P(i) + P(i - 1) + \dots + P(i - n + 1)}{n} \quad (1.19)$$

où :

- $P(i)$: Valeur à l'instant i .
- n : Taille de la fenêtre (nombre de périodes utilisées pour le calcul).
- i : Instant ou index temporel courant (point à prédire).

2. La Moyenne Mobile Centrée (Centered Moving Average)

La moyenne mobile centrée est une extension de la SMA qui utilise un nombre symétrique de valeurs autour d'un point central, afin d'aligner le calcul au cœur de la fenêtre d'observation [8]. Cette approche évite le décalage des tendances, souvent observé avec la SMA classique.

(i) **Fenêtre impaire (n impair) :**

La moyenne est calculée en prenant k valeurs avant et après le point central :

$$SMA_{\text{centrée}}(i) = \frac{1}{n} \sum_{j=i-k}^{i+k} P(j) \quad (1.20)$$

avec $k = \frac{n-1}{2}$.

(ii) **Fenêtre paire (n pair) :**

L'absence d'un point central exact est compensée en transformant la fenêtre en taille impaire, puis en ajoutant une pondération aux extrémités :

$$SMA_{\text{centrée}}(i) = \frac{1}{n} \left[\sum_{k=i-\frac{n}{2}+1}^{i+\frac{n}{2}-1} P(k) + \frac{P(i-\frac{n}{2}) + P(i+\frac{n}{2})}{2} \right] \quad (1.21)$$

Utilisation et limites : Cette méthode est efficace pour détecter les tendances dans des séries stationnaires ou régulières. Toutefois, elle ne peut être appliquée aux extrémités de la série, faute de données suffisantes, et nécessite des ajustements selon la parité de la fenêtre.

- Série Symétrique :

Une série symétrique est une série de données dans laquelle les valeurs sont disposées de manière à présenter une structure miroir autour d'un point central, c'est-à-dire que chaque valeur située avant ce point a une correspondance égale située après celui-ci [19].

Moyenne Mobile Pondérée (WMA - Weighted Moving Average)

La moyenne mobile pondérée (WMA) est une variante de la moyenne mobile simple qui attribue un poids différent à chaque observation de la série temporelle. Contrairement à la moyenne mobile classique où chaque point de la fenêtre a le même poids, ici les valeurs récentes ont plus d'influence que les valeurs plus anciennes [8].

Formule :

$$WMA_t = \frac{\sum_{i=0}^{n-1} w_i X_{t-i}}{\sum_{i=0}^{n-1} w_i} \quad (1.22)$$

où :

- X_{t-i} représente les valeurs de la série temporelle,
- w_i est le poids attribué à chaque valeur,
- $\sum w_i$ est la somme des poids, qui sert à normaliser la moyenne.

Les poids w_i sont souvent choisis de manière à décroître au fur et à mesure que l'on remonte dans le passé, ce qui accorde plus d'importance aux données récentes.

4. Moyenne Mobile Exponentielle (EMA - Exponential Moving Average) La moyenne mobile exponentielle, est une méthode avancée de lissage des séries temporelles. Contrairement à SMA ou WMA, l'EMA attribue un poids exponentiellement décroissant aux données passées. Les valeurs les plus récentes ont un poids plus important, tandis que les données anciennes ont un impact réduit. Cela permet à l'EMA d'être plus réactive aux changements récents tout en tenant compte de l'historique des données.

Elle est adaptée aux données volatiles mais peut être sensible au bruit et dépend du choix du facteur de lissage α . Bien que complexe à initialiser, elle conserve une mémoire des tendances passées tout en réagissant rapidement aux évolutions récentes [8] .

Formule : L'EMA est calculée de manière récursive, c'est-à-dire qu'elle dépend de la valeur précédente de l'EMA :

$$EMA_t = \alpha X_t + (1 - \alpha)EMA_{t-1} \quad (1.23)$$

où :

- X_t est la valeur actuelle de la série,
- EMA_{t-1} est l'EMA du jour précédent,
- α est le facteur de lissage, défini par :

$$\alpha = \frac{2}{n + 1} \quad (1.24)$$

avec n la taille de la période choisie.

Le facteur α contrôle l'influence des anciennes valeurs :

- Si α est proche de 1 (petite période n), l'EMA réagit très vite aux nouvelles valeurs.
- Si α est proche de 0 (grande période n), l'EMA lisse davantage la courbe.
- En utilisant cette méthode et ses types sur notre dataset on obtient ce résultat :

1.3 Processus d'analyse et de prédiction des séries temporelles

1.3.1 Stationnarité et transformations des séries temporelles

1.3.1.1 La stationnarité

La stationnarité d'une série temporelle désigne la propriété selon laquelle les caractéristiques statistiques de la série (comme la moyenne et la variance) ne changent pas au cours du temps. Il est recommandé que la série à prédire soit stationnaire. Pour vérifier cela, plusieurs méthodes existent, dont le test ADF [19].

— **Test ADF (Augmented Dickey-Fuller)**

Le test ADF est l'un des tests les plus couramment utilisés pour évaluer la stationnarité d'une série en testant l'hypothèse de présence d'une racine unitaire, ce qui signifie qu'un processus autorégressif a un coefficient de 1, entraînant une non-stationnarité dans la série temporelle.

Pour déterminer si la série est stationnaire ou non, la valeur critique doit être identifiée. Il s'agit d'une valeur seuil qui détermine, à partir de quelle valeur de la statistique ADF (probability value), l'hypothèse nulle (non stationnarité) peut être rejetée [19]. La valeur critique est identifiée en tenant compte de la taille du dataset et du niveau de significativité, généralement fixé à 5 %, ce qui signifie que l'hypothèse nulle peut être rejetée avec une confiance de 95 %.

Comparaison de la statistique ADF (p-value) avec la valeur critique

- Si la p-value est inférieure à la valeur critique, cela signifie que l'hypothèse nulle (la présence d'une racine unitaire) peut être rejetée, donc la série est stationnaire.
- Si la p-value est supérieure à la valeur critique, cela signifie que l'hypothèse nulle n'est pas rejetée, donc la série est non stationnaire.

1.3.1.2 La transformation des séries temporelles

Dans la pratique, la plupart des séries temporelles ne sont pas stationnaires. Cependant, ces modèles restent applicables tant que la série peut être transformée pour devenir stationnaire. Parmi les transformations couramment utilisées, on trouve le logarithme naturel, la différenciation ou encore la différenciation saisonnière [19].

— **Méthode du passage au logarithme**

Le logarithme naturel est souvent utilisé pour stabiliser la variance d'une série temporelle. Lorsque la variance évolue dans le temps, comme c'est souvent le cas pour les séries dont les valeurs augmentent ou diminuent de manière exponentielle, appliquer le logarithme permet de "réduire" cette variation. Cette transformation est particulièrement efficace pour atténuer les effets des grandes fluctuations relatives tout en conservant la structure des données [19].

— **La différenciation**

Cette technique est utilisée pour éliminer les tendances ou les variations qui ne sont pas constantes dans le temps, ce qui est essentiel pour de nombreux modèles de prévision comme ARIMA et SARIMA, ou encore SARIMAX. Il existe deux types principaux de différenciation dans ce contexte : la différenciation ordinaire et la différenciation saisonnière [19].

— **La différenciation ordinaire**

La différenciation ordinaire permet de rendre une série stationnaire au niveau de la tendance en soustrayant chaque observation de l'observation précédente.

$$Y_t = Y_t - Y_{t-1} \tag{1.25}$$

Où :

- Y_t est la valeur de la série à l'instant t ,
- Y_{t-1} est la valeur de la série à l'instant précédent.

— **La différenciation saisonnière**

Elle est utilisée pour éliminer les effets saisonniers ou périodiques dans une série temporelle. Elle consiste à prendre la différence entre une observation et l'observation correspondante de la saison précédente.

$$Y_t = Y_t - Y_{t-s} \tag{1.26}$$

Où :

- Y_t est la valeur de la série à l'instant t ,
- Y_{t-s} est la valeur de la série à l'instant $t - s$, où s est la période saisonnière.

1.3.2 Processus stationnaire

Le processus cité ci-dessous est adapté aux séries temporelles jugées stationnaires.

1.3.2.1 Processus ARMA

ARMA est un processus qui permet de modéliser les séries temporelles stationnaires.

Elle combine deux techniques pour modéliser les séries temporelles :

- **Auto-régression (AR)** : Les premiers processus autorégressifs ont été introduits par George Udny Yule. Un modèle autorégressif (AR) est un type de modèle statistique pour les séries temporelles. Il essaie de prédire la valeur future d'une variable en utilisant ses valeurs passées, en supposant qu'il y a une relation entre les valeurs passées et futures. Dans un modèle autorégressif, on dit que la variable "se régresse elle-même", c'est-à-dire qu'on prédit une valeur future en fonction de ses propres valeurs passées [8].

L'ordre du modèle AR (p)(voir 1.27) correspond au nombre de jours incorporés dans la formule. On peut le représenter de la manière suivante :

$$\text{Modèle AR}(t) = \sum_{i=1}^p \phi_i x_{t-i} \tag{1.27}$$

où :

- t est l'instant de temps actuel.
- x_t est la valeur observée à l'instant t .
- $\varphi_1, \varphi_2, \dots, \varphi_p$ sont les coefficients du modèle AR correspondant aux jours précédents jusqu'à p jours.
- p est l'ordre du modèle AR.

- **Moyenne Mobile (MA)** : En 1927 Eugen Slutsky a introduit pour la première fois les processus à moyenne mobile dans son article . Ces processus forment une classe flexible de modèles pour de nombreux phénomènes observés. Ils sont construits à partir de l'idée que l'observation au temps t s'explique linéairement par les observations d'un bruit blanc (dépend des erreurs passées) [8].

la MA dans ce contexte est liée a la moyenne mobile pondérée préalablement citée de part l'utilisation des bruits blancs passés avec des poids distincts (valeurs passées pour la SMA) pour lisser ou filtrer la série.

Un processus moyenne mobile d'ordre q , noté MA(q) est donné par :

$$MA(t) = \sum_{i=0}^q \theta_i \varepsilon_{t-i}, \quad \text{avec } \theta_0 = 1. \quad (1.28)$$

où :

- les θ_i sont les coefficients
- ε_t est un bruit blanc de variance σ^2 (l'erreur).

Le modèle ARMA, noté ARMA(p, q), est donc la concaténation des deux techniques précédentes.

Principes utilisés pour le processus ARMA

- **La méthode des moindres carrés**

La méthode des moindres carrés est une technique statistique qui permet de trouver la droite (ou la courbe) qui minimise la somme des carrés des écarts entre les valeurs observées et les valeurs prédites par le modèle , elle permet de trouver les meilleurs coefficients pour l'équation de prédiction pour les modèles de régression linéaire simples.

Calcul des coefficients par la méthode des moindres carrés simple

- **Relation avec formule** : trouver les coefficients (ϕ_k) revient à minimiser la somme des carrés des erreurs entre les valeurs prédites \hat{Y}_t et les valeurs réelles Y_t , ce qui veut dire minimiser l'erreur.

Sachant que les modèles ARMA, ARIMA, SARIMA et SARIMAX sont utilisés principalement pour des fins de prédiction, les valeurs prédites sont le résultat de leur utilisation ce qui fait que cette somme ne peut donc pas être directement calculée, il suffit de trouver les valeurs des coefficients qui minimisent la somme des erreurs.

Cas d'un modèle AR(1) :

$$\phi_1 = \frac{\sum_{t=2}^T Y_{t-1} Y_t}{\sum_{t=2}^T Y_{t-1}^2} \quad (1.29)$$

C'est cette valeur qui minimise l'erreur et donc qui est utilisée comme estimation de ϕ_1 .

Pour généraliser :

$$\phi_k = \frac{\sum_{t=i}^T \left(Y_{t-k} Y_t - \sum_{j=1}^{k-1} \phi_j Y_{t-k} Y_{t-j} \right)}{\sum_{t=i}^T Y_{t-k}^2} \quad (1.30)$$

$$i = k + 1$$

Symbole	Description
ϕ_k	Coefficient d'autorégression d'ordre k à estimer.
T	Nombre total d'observations dans la série.
Y_t	Valeur de la série au temps t .
Y_{t-k}	Valeur passée de la série au lag k .

TABLE 1.2 – Signification des symboles utilisés dans le modèle AR

Cette formule est une simplification directe pour estimer le coefficient ϕ dans un modèle AR sans avoir besoin d'utiliser d'autres méthodes plus complexes comme les calculs matriciels complets. Le même principe est utilisé pour trouver les coefficients du modèle MA.

— **La fonction d'Auto-Corrélation (ACF) et la fonction d'Auto-Corrélation Partielle (PACF)**

1. **ACF** : L'Auto-Corrélation (Autocorrelation Function) mesure la corrélation linéaire entre une observation d'une série temporelle à un instant donné t et les observations à des instants précédents $t - k$. Elle permet de détecter la persistance d'une dépendance dans le temps entre les données. L'ACF est principalement utilisée pour déterminer l'ordre q dans un modèle ARMA.

On calcule la valeur de l'ACF pour chaque décalage temporel k à l'aide de la formule suivante :

$$\rho_k = \frac{\sum_{t=1}^{n-k} (X_t - \bar{X})(X_{t+k} - \bar{X})}{\sum_{t=1}^n (X_t - \bar{X})^2} \quad (1.31)$$

où :

- X_t : valeur de la série au temps t ,
- \bar{X} : moyenne de la série,
- n : nombre total d'observations.

L'interprétation des pics dans le graphique ACF est la suivante :

- **Pic positif important** : forte corrélation entre les valeurs à deux instants séparés par un lag k .

- **Pic négatif** : relation inverse entre les deux valeurs.
- **Pic proche de zéro** : absence de dépendance significative.

Le seuil de significativité utilisé pour interpréter les résultats est défini par :

$$z = \left[-\frac{2}{\sqrt{n}}, \frac{2}{\sqrt{n}} \right] \quad (1.32)$$

2. **PACF** : L'Auto-Corrélation Partielle (Partial Autocorrelation Function) mesure la corrélation directe entre une observation X_t et une observation antérieure X_{t-k} , tout en supprimant l'influence des observations intermédiaires ($X_{t-1}, X_{t-2}, \dots, X_{t-k+1}$).

Le PACF est particulièrement utile pour déterminer l'ordre p d'un modèle ARMA. Il est calculé via une régression multiple où l'on évalue la corrélation entre X_t et X_{t-k} après avoir ajusté tous les lags intermédiaires.

Formellement, la régression est donnée par :

$$x_t = \beta_1 x_{t-1} + \beta_2 x_{t-2} + \dots + \beta_k x_{t-k} + \epsilon_t \quad (1.33)$$

Puis, la corrélation partielle est :

$$\text{PACF}(k) = \frac{\text{Cov}(x_t, x_{t-k} \mid x_{t-1}, \dots, x_{t-k+1})}{\sigma_{x_t} \cdot \sigma_{x_{t-k}}} \quad (1.34)$$

3. Interprétation des résultats graphiques :

- **Pic positif** : relation directe positive entre deux observations à t et $t - k$.
- **Pic négatif** : relation directe inverse entre deux observations.
- **Pic proche de zéro** : absence de relation directe significative.

Détermination des paramètres ARMA(p, q) à partir du PACF et du ACF

- **Le paramètre q** : Rechercher le point à partir duquel l'ACF devient négligeable c'est à dire ne dépasse plus l'intervalle de confiance qui est égale à :

$$\left[-\frac{2}{\sqrt{n}}, \frac{2}{\sqrt{n}} \right]$$

- Où n est la taille du dataset

Le point désigné correspond généralement à la valeur de q ou q_{\max} .

- **Le paramètre p** : Rechercher le point à partir duquel le PACF devient négligeable c'est à dire ne dépasse plus l'intervalle de confiance.

Ce décalage correspond généralement à la valeur de p ou p_{\max} .

Pour plus de précision sur le choix de ces paramètres afin de construire le modèle le plus optimal, des concepts appelés les critères d'information et le MAPE sont utilisés de la même manière que celle du paramètre q .

Critères d'information

Les **critères d'information** sont des outils statistiques permettant de choisir le modèle le plus adapté en équilibrant **la qualité d'ajustement** et **la complexité**. Un critère plus faible indique généralement un meilleur modèle.

— **AIC (Akaike Information Criterion)**

Il pénalise les modèles trop complexes et est souvent utilisé dans le choix des paramètres :

$$\text{AIC} = -2 \log(\mathcal{L}) + 2k$$

Plus l'AIC est faible, meilleur est le modèle.

— **BIC (Bayesian Information Criterion)**

Il pénalise davantage la complexité que l'AIC, surtout avec un grand nombre d'observations :

$$\text{BIC} = -2 \log(\mathcal{L}) + k \log(n)$$

— **HQIC (Hannan-Quinn Information Criterion)**

Il offre une pénalisation intermédiaire entre l'AIC et le BIC :

$$\text{HQIC} = -2 \log(\mathcal{L}) + 2k \log(\log(n))$$

— **MAPE (Mean Absolute Percentage Error)**

Il mesure l'erreur absolue moyenne entre les valeurs prédites et observées :

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^n \left| \frac{Y_t - \hat{Y}_t}{Y_t} \right| \times 100$$

- Un MAPE faible indique une bonne précision des prédictions.
- Un MAPE élevé ($> 20\%$ - 25%) indique une faible fiabilité du modèle.

Combinaison des critères : Les critères AIC, BIC et HQIC aident à évaluer l'ajustement et la complexité du modèle. Le MAPE vient compléter cette évaluation en mesurant la précision sur des données de test.

1.3.3 Processus non stationnaire

Dans le cas où le test ADF affirme l'hypothèse de présence d'une racine unitaire, il existe trois méthodes adaptées à des jeux de données de ce type, chacune d'elle apporte un traitement en plus qui concerne les composantes d'une série temporelle à savoir : la tendance, la saisonnalité et les facteurs externes qui pourraient avoir un impact sur la série. Il est donc nécessaire de déterminer la méthode adaptée à la série temporelle avant d'entamer la recherche des paramètres liée à celle-ci. Pour ce faire, en plus du test ADF, la décomposition STL (Seasonal and Trend decomposition using Loess) est sollicitée [8]

Le choix du modèle approprié

- Si le résultat présente une saisonnalité significative les deux méthodes SARIMA et SARIMAX sont plus adaptées selon la présence d'effets externes (pour SARIMAX) ou pas (pour SARIMA).
- Si le résultat ne présente pas une saisonnalité significative cela suggère que la non stationnarité peut être due à l'effet de la tendance, un modèle ARIMA est donc plus adapté

1.3.3.1 Processus ARIMA

Le modèle ARIMA (AutoRegressive Integrated Moving Average) est une méthode couramment utilisée pour modéliser et prévoir les séries temporelles. Il combine les composantes (AR) et (MA) avec une différenciation (I) pour prendre en compte la composante de la tendance. Le modèle ARIMA (p, d, q) est défini par trois ordres : p, d et q. Les ordres p et q sont identiques à ceux du modèle ARMA déjà définis, et sont déterminés exactement de la même manière.

Le nouveau paramètre d représente [8] :

- **d (Ordre de différenciation)** : L'ordre de différenciation indique combien de fois la série doit être différenciée pour rendre les données stationnaires. Dans la majorité des cas, une seule différenciation ordinaire suffit, ce qui donne $d=1$.

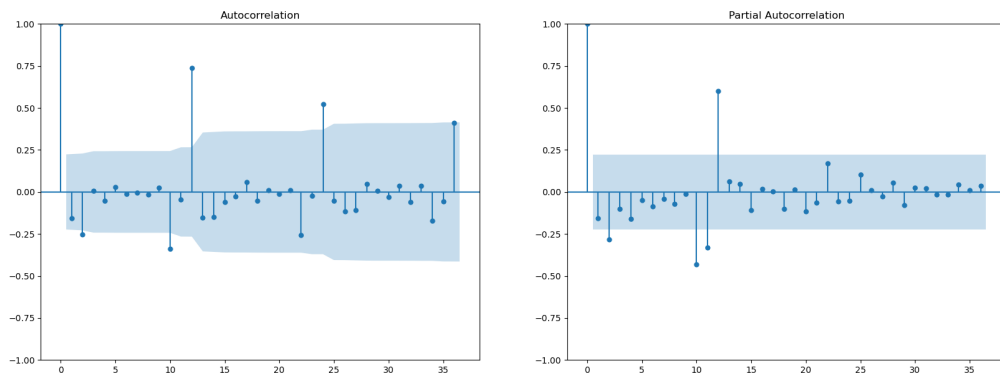


FIGURE 1.5 – ACF et PACF apres la différenciation ordinaire .

Après avoir identifié les composants les plus optimaux du modèle ARIMA, voici le résultat obtenu pour la prédiction :

ARIMA(3,1,2) Avec : $p=3$, $d=1$ et $q=2$

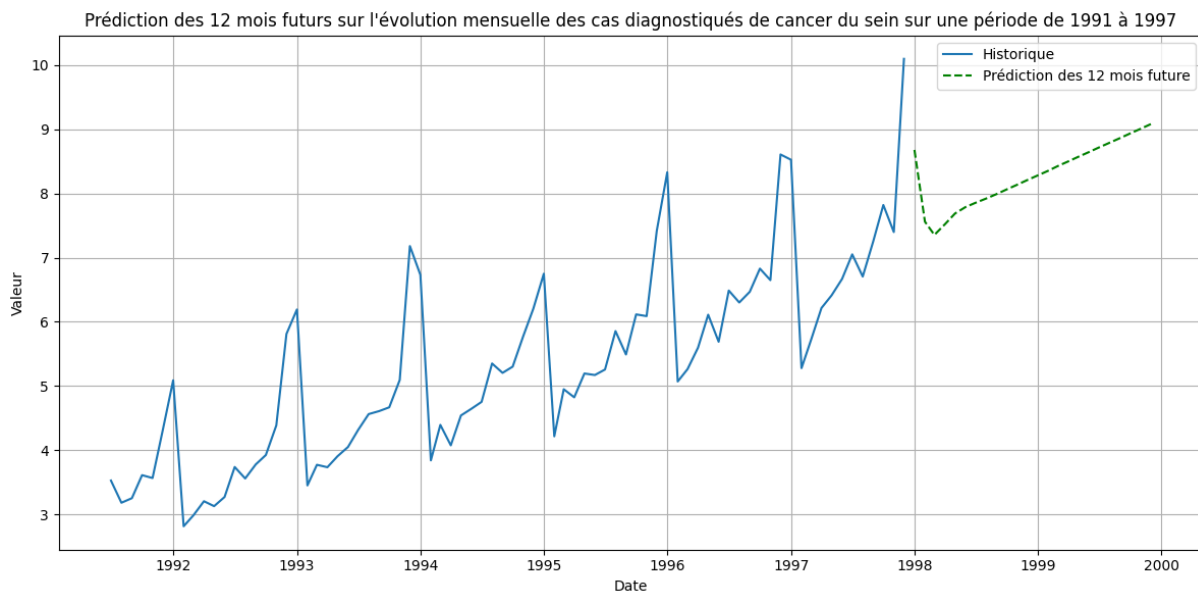


FIGURE 1.6 – Prédiction avec la méthode ARIMA.

1.3.3.2 Processus SARIMA

Le modèle SARIMA est une extension du modèle ARIMA, il permet de prendre en compte la saisonnalité en ajoutant une composante saisonnière. Cela permet de modéliser des cycles réguliers, comme des pics de ventes en fin d'année ou des hausses de consommation énergétique en hiver.

La structure de SARIMA est noté SARIMA(p, d, q)(P, D, Q, s) [8], où :

- Les paramètres p, d, q représentent les mêmes paramètres du modèle ARIMA précédemment présentés.

- Les paramètres P, D, Q concernent la partie saisonnière, avec :
 - P : Le paramètre saisonnier autorégressif .
 - D : Le paramètre de différenciation saisonnière.
 - Q : Le paramètre de moyenne mobile saisonnière.
 - s : La période saisonnière

Détermination des paramètres saisonniers :

- **Le paramètre de différenciation saisonnière (D)** : Indique le nombre de fois que la différenciation saisonnière est appliquée pour diminuer les effets saisonniers afin d'obtenir une série stationnaire [8].
- **P et Q saisonniers** : La détermination des paramètres P et Q saisonniers est similaire à celle des paramètres non saisonniers, le paramètre P est extrait à partir du PACF et le paramètre Q est extrait à partir du ACF, la différence réside dans le fait que le P_{\max} n'est pas désigné comme le lag à partir duquel le PACF devient négligeable mais plutôt le premier lag saisonnier non significatif c'est à dire qu'au lieu de prendre en considération tous les lags, seuls les lags saisonniers (multiples de la période saisonnière) sont pris en compte et le lag à partir du quel les lags saisonniers ne dépassent pas la zone de confiance est désigné comme étant P_{\max} . le même principe est appliqué pour déterminer Q_{\max} [8].

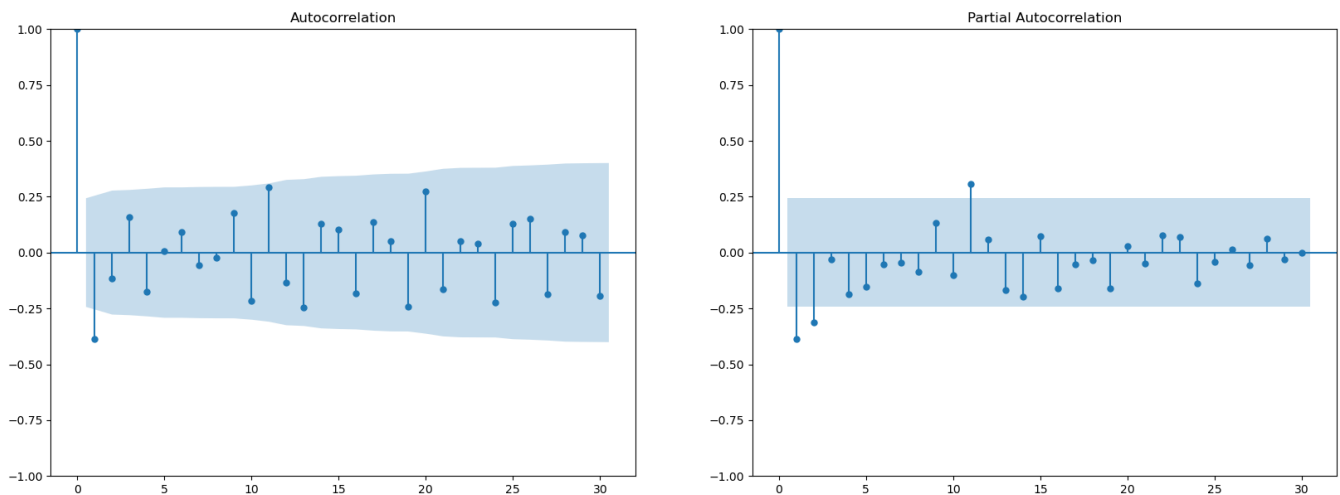


FIGURE 1.7 – ACF et PACF apres une différenciation saisonnière

Voici le résultat de la prédiction obtenue avec la méthode SARIMA après l'identification des paramètres donnant le résultat le plus optimal :

SARIMA (1,1,1)(1,1,0,12) avec : $p=1$, $d=1$, $q=1$, $P=1$, $D=1$, $Q=0$, $s=12$

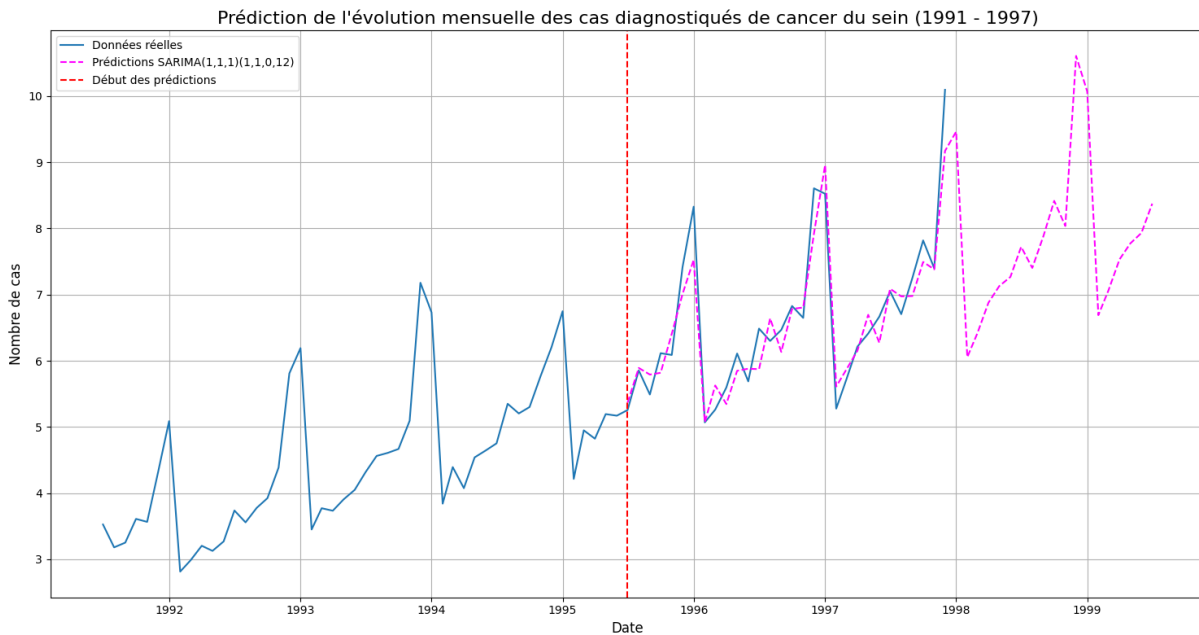


FIGURE 1.8 – Prédiction avec le modèle SARIMA

1.3.3.3 Processus SARIMAX

SARIMAX est une extension du modèle SARIMA qui a été introduit par George Box et Gwilym Jenkins. Il utilise les mêmes paramètres que SARIMA, en incluant un nouveau composant nommé variable exogène (ou régressives externes) [19].

Modèle mathématique : Le modèle mathématique de SARIMAX peut être présenté ainsi :

$$Y_t = \text{SARIMA}(\text{terms}) + \beta_1 x_1(t) + \beta_2 x_2(t) + \dots + \text{résidu}$$

- Y_t : la valeur de la série cible à l’instant t .
- $x_1(t), x_2(t)$: les valeurs des variables exogènes à l’instant t .
- β_1, β_2 : les coefficients estimés pour les variables exogènes, qui permettent de mesurer l’influence de la variable exogène associée sur la variable cible.

Variables exogènes : sont des variables extérieures au modèle qui peuvent influencer le comportement de la série temporelle, améliorant ainsi la prévision de cette série. Elles doivent respecter certaine condition :

- Les dates (instants) des variables exogènes doivent correspondre exactement à celles de la série temporelle principale (Pour prédire les valeurs exogènes, on utilise les modèles de prédiction des séries temporelles préalablement expliqués comme ARIMA, SARIMA).
- les valeurs des variables exogènes doivent être disponibles pour toutes les périodes de prédictions

Pour tester si une variable exogène a un impact statistiquement significatif sur la série temporelle dans un modèle SARIMAX, la p-value qui cherche si une variable exogène a un effet sur les valeurs du dataset est utilisée. elle cherche a savoir si les coefficients β_n associés aux variables exogènes sont significativement différents de zéro.

L'hypothèse nulle H_0 suppose que le coefficient $\beta_n = 0$, ce qui signifie que la variable exogène n'a pas d'effet sur la série cible. L'hypothèse alternative H_1 suppose que

$\beta_n \neq 0$, indiquant une influence significative.

- Si la p-value est inférieure à 0.05, cela signifie que la variable exogène a probablement un effet significatif sur la série temporelle.
- Si la p-value est supérieure ou égale à 0.05, cela suggère que la variable n'a pas d'impact prouvé et pourrait être retirée du modèle.

Dans le contexte de notre dataset, les variables exogènes peuvent inclure des facteurs comme la **pollution atmosphérique** .

Les paramètres utilisés sont les mêmes que ceux du modèle SARIMA, les effets des variable exogènes est ensuite ajouté pour obtenir les résultats de prédiction :

Dans ce cas la p-value trouvée est égale à 0.331, donc on peut déjà approuver que les valeurs exogènes n'ont pas d'effets sur la série

-De plus en visualisant les résultats de la prédiction SARIMAX et en les comparant avec ceux obtenus avec SARIMA il est clair que SARIMA donne des résultats beaucoup plus juste. Dans ce cas il est donc plus favorable d'adopter le modèle SARIMA, mais cela peut différer d'un modèle à un autre

SARIMA(1,1,1) (1,1,0,12)

Le graphe si desous représente la prédictions sur le dataset présenter au départ à l'aide de la méthode sarimax

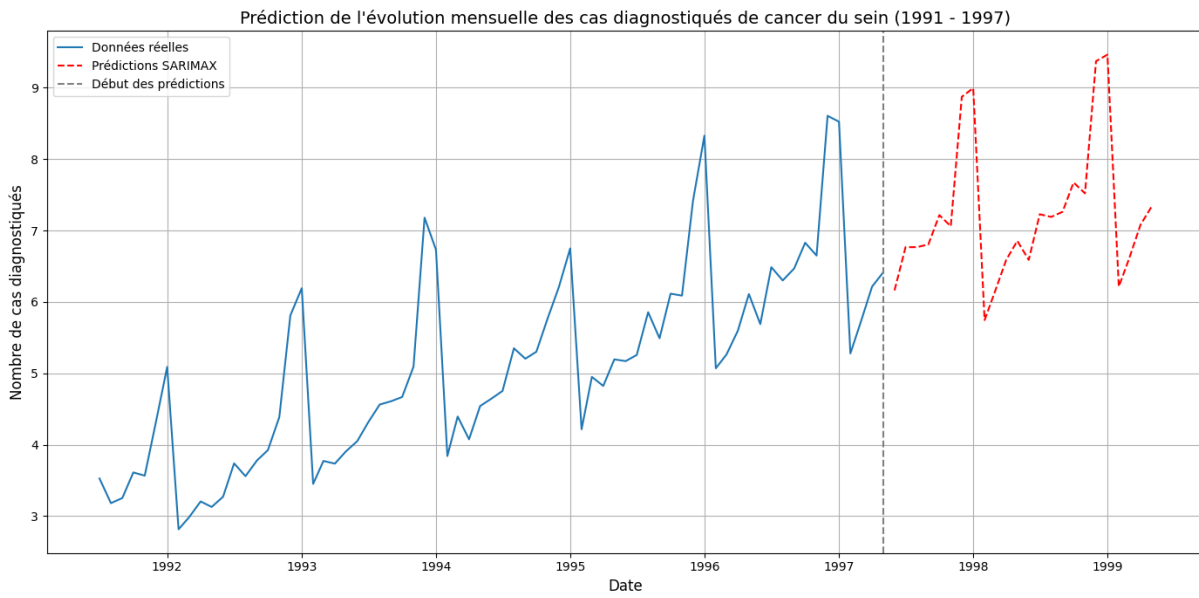


FIGURE 1.9 – Prédiction avec le modèle SARIMAX

1.4 Conclusion

Les méthodes de prédiction forment des outils indispensables pour l'analyse des séries temporelles. Les différentes fonctions et méthodes nous ont permis de visualiser les tendances saisonnières et les pics de données, fournissant ainsi les anticipation des les besoins futurs.

Chapitre 2

Les méthodes de deep learning pour la prédiction

2.1 Introduction :

Le développement technologique accéléré de ces dernières décennies a permis l'émergence de systèmes informatiques capables de simuler certains aspects de l'intelligence humaine. Ces avancées, portées notamment par l'intelligence artificielle (IA), ont profondément transformé de nombreux domaines d'activité, en apportant des solutions innovantes à des problématiques complexes. Dans ce chapitre, nous nous intéressons à l'intelligence artificielle et à l'une de ses branches les plus prometteuses : le Deep Learning, largement utilisé dans les applications de prédiction, comme celles liées à la consommation d'électricité.

2.2 L'intelligence artificielle

L'intelligence artificielle (IA) est un domaine de l'informatique qui vise à concevoir des systèmes capables de reproduire certaines fonctions cognitives humaines, telles que l'apprentissage, le raisonnement, la perception ou encore la prise de décision . Elle repose sur des algorithmes et des modèles mathématiques qui permettent aux machines d'interagir intelligemment avec leur environnement et d'adapter leur comportement à partir de données [27].

2.3 Le Machine Learning

Le ML (ou apprentissage automatique) est une discipline de l'IA qui consiste à développer des algorithmes capables d'apprendre à partir de données pour accomplir des tâches spécifiques sans être explicitement programmés.

Le ML repose sur l'idée que les systèmes peuvent améliorer leurs performances au fil du temps et avec plus de données, en ajustant automatiquement leurs paramètres internes pour

minimiser les erreurs ou maximiser leur efficacité. C'est un outil puissant pour résoudre des problèmes complexes dans des domaines variés, allant de la médecine à la finance, en passant par la reconnaissance vocale ou la recommandation de produits [16].

La Modélisation des Données

La modélisation des données en machine learning consiste à créer des modèles mathématiques pour prédire, classer ou regrouper des données. L'objectif de cette modélisation est de trouver un modèle qui s'ajuste au mieux aux données en minimisant l'erreur (ou la distance) entre les prédictions du modèle et les données réelles. Cela se fait en ajustant les paramètres du modèle pour optimiser ses performances et garantir qu'il généralise bien à de nouvelles données.

- l'axe des X représente les données
- l'axe des Y représente les variable cible

Ce graphe est une illustration de la séparation des deux catégories avec une frontière de décision :

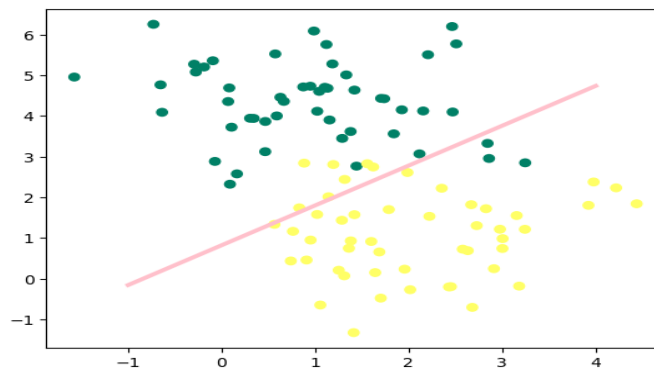


FIGURE 2.1 – Représentation graphique des catégories de données séparées par une frontière de décision [16]

2.3.1 Types d'apprentissage

Les trois grandes catégories d'apprentissage en Machine Learning sont :

Apprentissage Supervisé (Supervised Learning)

L'apprentissage supervisé est une méthode où le modèle apprend à partir de données étiquetées. Cela signifie que chaque exemple dans les données d'entraînement est associé à une "réponse" ou une "étiquette" correcte. Le modèle utilise ces paires (données d'entrée + étiquette) pour apprendre à prédire les étiquettes pour de nouvelles données [16].

Apprentissage Non Supervisé (Unsupervised Learning)

L'apprentissage non supervisé est une méthode où le modèle apprend à partir de données non étiquetées. Il n'y a pas de réponses correctes fournies au modèle. L'objectif est de découvrir des structures, des patterns ou des regroupements naturels dans les données [16].

Apprentissage par Renforcement (Reinforcement Learning)

L'apprentissage par renforcement est une méthode où un agent apprend à interagir avec un environnement en effectuant des actions et en recevant des récompenses ou des pénalités. L'objectif est d'apprendre une stratégie (politique) qui maximise la récompense cumulative sur le long terme [16].

2.4 Définition du Deep Learning

Le DL est une sous-catégorie du ML qui utilise des réseaux de neurones artificiels profonds pour modéliser et résoudre des problèmes complexes. Ces réseaux sont composés de plusieurs couches de neurones interconnectés, ce qui leur permet d'apprendre des représentations hiérarchiques et abstraites des données [16].

2.5 Les Réseaux de Neurones Artificiels (RNA)

Un **réseau de neurones artificiel (RNA)** est un modèle mathématique et computationnel inspiré du fonctionnement des neurones biologiques dans le cerveau humain. Son objectif principal est d'apprendre à partir de données pour effectuer des tâches telles que des prédictions, des classifications ou des reconnaissances de motifs [16].

Un RNA est un ensemble de neurones artificiels organisés en couches et reliés entre eux par des poids qui sont ajustés durant l'apprentissage.

2.5.1 Structure d'un Réseau de Neurones

Un réseau de neurones est composé de trois types de couches principales, chacune jouant un rôle spécifique dans le traitement des données :

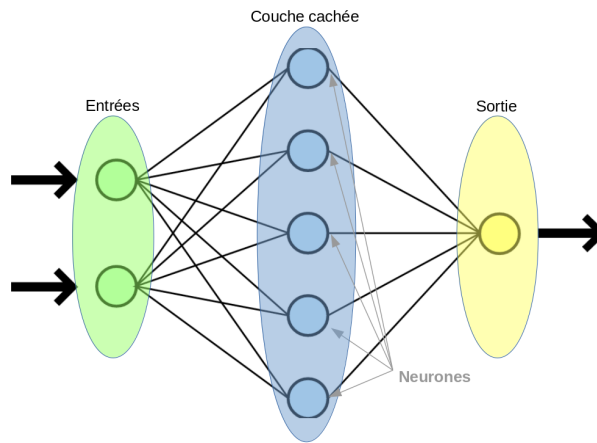


FIGURE 2.2 – La structure globale des neurones artificiels [22]

- **La couche d'entrée (Input layer)** : Contient les neurones d'entrée (X), qui représentent les caractéristiques des données. Son rôle est de recevoir les données brutes.
- **Les couches cachées (Hidden layers)** : Ces couches constituent le cœur du réseau, elles combinent les entrées avec des poids (W) et des biais (b) pour apprendre des représentations de plus en plus abstraites. Plus il y a de couches cachées, plus le réseau est profond (d'où le terme *deep learning*).
- **La couche de sortie (Output layer)** : Donne le résultat final (Y). Le nombre de neurones dans cette couche dépend du type de tâche que le réseau doit effectuer (classification, régression, etc.).

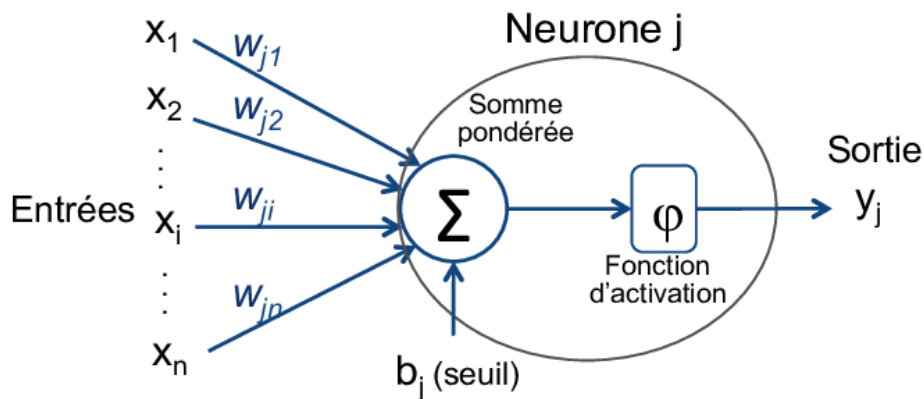


FIGURE 2.3 – Structure détaillée d'un neurone artificiel [9]

Fonction d'Activation : La fonction d'activation est une transformation mathématique appliquée à la sortie d'un neurone afin d'introduire de la non-linéarité. Elle permet aux réseaux de neurones d'apprendre des relations complexes entre les données d'entrée et d'améliorer leur capacité de généralisation [16].

- Sans fonction d'activation, un réseau de neurones ne serait qu'une simple combinaison linéaire de ses entrées. Cela limiterait fortement sa capacité à résoudre des problèmes complexes, notamment ceux impliquant des motifs non linéaires.

Principales fonctions d'activation et leurs formules : Il existe plusieurs fonctions d'activation, parmi lesquelles on peut mentionner

1. Fonction seuil (Step Function) : Elle retourne 1 si l'entrée dépasse un certain seuil, sinon elle retourne 0. Elle est principalement utilisée dans les perceptrons simples.

$$f(z) = \begin{cases} 1, & \text{si } z \geq 0 \\ 0, & \text{sinon} \end{cases}$$

2. Fonction sigmoïde (Sigmoid Function) : Elle transforme l'entrée en une valeur comprise entre 0 et 1, ce qui est utile pour les problèmes de classification binaire.

$$f(z) = \frac{1}{1 + e^{-z}}$$

3. Fonction tangente hyperbolique (Tanh Function) : Similaire à la sigmoïde, mais elle produit des valeurs entre -1 et 1, ce qui facilite l'apprentissage dans certaines architectures.

$$f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

4. ReLU (Rectified Linear Unit) : Très utilisée dans les réseaux profonds, cette fonction renvoie 0 pour les valeurs négatives et la valeur elle-même pour les valeurs positives, accélérant ainsi l'apprentissage.

$$f(z) = \max(0, z)$$

Voici le résultat graphique de chaque fonctions d'activations mentionnées :

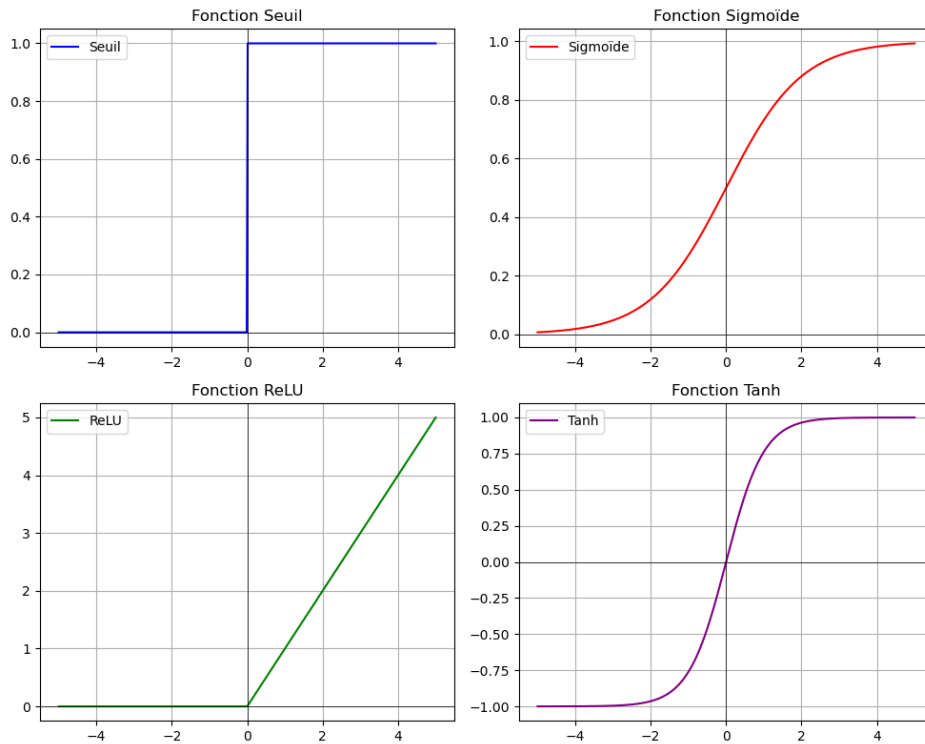


FIGURE 2.4 – la representation graphique des 4 fonctions d'activations

-Le choix de la fonction d'activation dépend de la nature du problème (régression ou classification binaire/multi-classes), des données, et de la vitesse d'entraînement.

2.5.2 Types de Réseaux de Neurones

Il existe plusieurs types de réseaux de neurones, chacun adapté à un type de problème spécifique :

Le Perceptron simple

Le perceptron est le modèle de réseau de neurones le plus élémentaire, introduit par Frank Rosenblatt en 1958. Il s'agit d'un algorithme de classification binaire qui prend plusieurs entrées (appelées attributs), applique des poids à chacune, puis les additionne avec un biais. La somme est ensuite transformée par une fonction d'activation, généralement un seuil, pour produire une sortie : 0 ou 1. Par exemple, un perceptron peut être utilisé pour prédire si une plante est toxique ou non en fonction de la longueur et de la largeur de ses feuilles.

Toutefois, ce modèle est limité aux problèmes linéairement séparables, ce qui signifie qu'il ne peut pas résoudre des cas complexes où les données ne peuvent être séparées par une simple droite ou un plan. Pour pallier cette limitation, des réseaux plus avancés ont été développés, comme le perceptron multicouche [16].

Le perceptron multicouche (MLP)

Le perceptron multicouche, ou MLP (Multilayer Perceptron), est une extension du perceptron simple. Il est constitué de plusieurs couches de neurones : une couche d'entrée, une ou plusieurs couches cachées, et une couche de sortie. Chaque neurone réalise une combinaison linéaire de ses entrées, suivie d'une fonction d'activation non linéaire, comme ReLU ou tanh. Ce type de réseau permet de modéliser des relations non linéaires et d'apprendre des représentations plus abstraites des données. Les MLP sont utilisés dans de nombreuses applications, comme la reconnaissance de chiffres manuscrits, la prédiction de séries temporelles simples, ou encore la classification d'images. Leur structure relativement simple mais puissante en fait une base importante dans le domaine du deep learning [16].

Les réseaux de neurones convolutifs (CNN)

Les réseaux de neurones convolutifs (CNN – Convolutional Neural Networks) sont conçus spécifiquement pour le traitement de données structurées sous forme de grille, comme les images. Contrairement aux MLP, les CNN exploitent la structure spatiale des données à l'aide de couches de convolution qui appliquent des filtres (ou noyaux) glissants sur l'image afin de détecter des motifs visuels tels que des bords, des textures ou des formes. Ces caractéristiques sont ensuite combinées par des couches de pooling et des couches pleinement connectées pour produire une sortie. Les CNN sont extrêmement performants en vision par ordinateur et sont utilisés dans des tâches telles que la reconnaissance faciale, la détection d'objets, l'analyse médicale par imagerie ou la conduite autonome [16].

Les réseaux adversariaux génératifs (GAN)

Les réseaux adversariaux génératifs (GAN – Generative Adversarial Networks) ont été introduits en 2014 par Ian Goodfellow. Ils sont constitués de deux réseaux de neurones opposés : un générateur, chargé de produire de fausses données (par exemple des images), et un discriminateur, qui essaie de distinguer les données réelles des données synthétiques. Les deux réseaux s'entraînent simultanément : le générateur apprend à créer des données de plus en plus réalistes pour tromper le discriminateur, tandis que ce dernier améliore sa capacité à détecter les fausses données. Ce mécanisme de compétition permet d'obtenir des résultats remarquablement réalistes. Les GAN sont utilisés dans la génération d'images, la création artistique, les deepfakes, la super-résolution et la simulation de données rares ou coûteuses à obtenir [16].

Les réseaux de neurones récurrents (RNN) :

Les réseaux de neurones récurrents (RNN – Recurrent Neural Networks) sont adaptés aux données séquentielles, où l'ordre des éléments est important, comme les séries temporelles,

le texte ou l'audio. Contrairement aux réseaux classiques, les RNN possèdent une mémoire interne qui leur permet de prendre en compte le contexte des étapes précédentes lors du traitement d'une séquence. Chaque sortie dépend ainsi non seulement de l'entrée actuelle, mais aussi des états précédents. Cette capacité rend les RNN particulièrement efficaces dans des applications comme la traduction automatique, la génération de texte, la reconnaissance vocale ou la prévision de consommation énergétique. Des variantes plus puissantes comme les LSTM et GRU ont été développées pour surmonter certaines limites des RNN de base, notamment le problème de gradient qui disparaît sur les longues séquences [16].

2.5.3 Processus d'Apprentissage d'un Réseau de Neurones

L'apprentissage d'un réseau de neurones suit un processus structuré en plusieurs étapes. Ces étapes sont essentielles pour permettre au modèle de s'améliorer progressivement et de minimiser l'erreur entre les prédictions et les valeurs réelles [16].

Voici les étapes générales de l'apprentissage d'un réseau de neurones :

1. Initialisation des poids

Les poids (W) et les biais (b) du réseau sont initialisés aléatoirement .

2. Propagation Avant (Forward Propagation)

La première phase consiste à transmettre les données d'entrée à travers les différentes couches du réseau afin d'obtenir une sortie. Chaque neurone effectue un calcul basé sur les poids associés aux connexions et une fonction d'activation.

La formule générale appliquée dans chaque neurone est la suivante :

$$\sigma = f(W \cdot X + b) \tag{2.1}$$

où :

- X représente les données d'entrée,
- W correspond aux poids attribués à chaque connexion,
- b est le biais introduit pour ajuster la sortie,
- f est la fonction d'activation appliquée au résultat pour introduire de la non-linéarité.

Cette étape permet d'obtenir une sortie basée sur les poids actuels du modèle.

Le résultat final est donc une prédiction (sortie du réseau).

3. Calcul de l'Erreur

Une fois la sortie obtenue, il est nécessaire d'évaluer la performance du modèle en comparant la sortie prédite avec la valeur réelle. Cette comparaison est réalisée à l'aide d'une fonction de perte, qui mesure l'écart entre ces deux valeurs.

La fonction de coût (ou fonction de perte) mesure l'erreur entre les prédictions du modèle et les vraies valeurs. Pour le perceptron, la fonction de coût utilisée est souvent la fonction logistique, qui est dérivée de la vraisemblance. Elle quantifie à quel point les prédictions du modèle s'écartent des données réelles. L'objectif est de minimiser cette fonction pour améliorer les performances du modèle.

Pour un modèle de classification binaire, la fonction de coût utilisée est souvent la fonction **Log Loss**, qui est donnée par :

$$loss = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(h(x^{(i)})) + (1 - y^{(i)}) \log(1 - h(x^{(i)}))] \quad (2.2)$$

où :

- m est le nombre total d'exemples d'entraînement.
- $y^{(i)}$ est la vraie étiquette (0 ou 1) pour l'exemple i .
- $h(x^{(i)})$ est la sortie du modèle (probabilité) pour l'exemple i , généralement calculée à l'aide de la fonction sigmoïde.

Pour les problèmes de régression, on utilise généralement l'**Erreur Quadratique Moyenne** (Mean Squared Error - MSE) :

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

où :

- y_i est la valeur réelle,
- \hat{y}_i est la prédiction du modèle,
- N est le nombre d'échantillons.

4. Rétropropagation (Backpropagation)

Après le calcul de l'erreur, une mise à jour des poids est nécessaire pour améliorer la précision du modèle. Cette mise à jour est réalisée grâce à l'algorithme de rétropropagation.

Propagation arrière (Backward Pass) : L'erreur est propagée en sens inverse à travers le réseau, de la couche de sortie jusqu'à la couche d'entrée. Le gradient de l'erreur est calculé par rapport à chaque poids du réseau en appliquant la règle de la chaîne :

$$\frac{\partial loss}{\partial W} = \frac{\partial loss}{\partial y} \times \frac{\partial y}{\partial W} \quad (2.3)$$

Chaque poids est mis à jour en fonction de l'influence qu'il a eu sur l'erreur. L'objectif de cette étape est d'identifier comment chaque poids doit être ajusté pour minimiser l'erreur globale du modèle.

Formule de mise à jour des poids (Descente de Gradient) : Une fois les gradients calculés, la descente de gradient est utilisée pour mettre à jour les poids du réseau et ainsi réduire progressivement l'erreur.

$$W = W - \alpha \frac{\partial loss}{\partial W} \quad (2.4)$$

où :

- α est le taux d'apprentissage, qui contrôle la vitesse d'adaptation du modèle,
- $\frac{\partial loss}{\partial W}$ est le gradient de l'erreur par rapport au poids W .

5. Répétition (Époques)

Les étapes 2 à 4 sont répétées pour plusieurs époques (passages complets sur le jeu de données) jusqu'à ce que le réseau apprenne bien les données.

6. Validation et test

Le réseau est évalué sur un ensemble de validation pour éviter le surapprentissage (overfitting).

Enfin, il est testé sur un ensemble de test pour mesurer ses performances sur des données invisibles.

2.5.4 Les Problèmes Des Réseaux de Neurones Récurents (RNN) :

Le principal problème des RNN (réseaux de neurones récurrents) est leur difficulté à capturer des dépendances à long terme dans les séquences. Ce problème est souvent appelé le problème du gradient qui disparaît (vanishing gradient) ou, dans certains cas, le problème du gradient qui explose (exploding gradient).

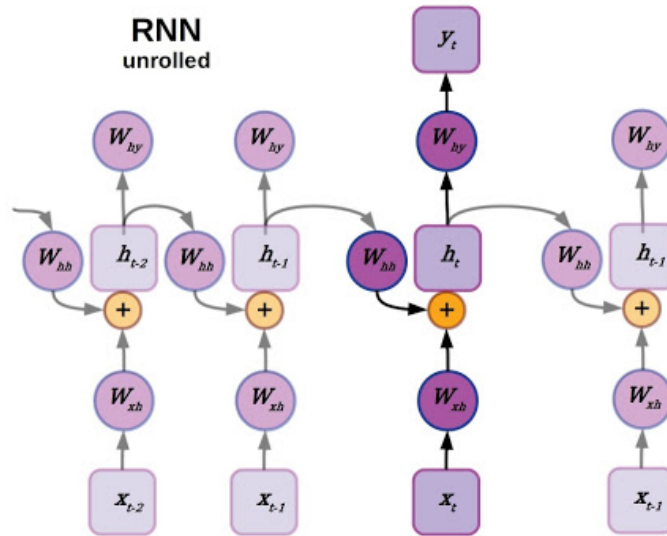


FIGURE 2.5 – Représentation d'un Réseau de Neurones Récurent (RNN) déplié [22]

2.5.4.1 Problème de la disparition du gradient (Vanishing Gradient)

Les RNN utilisent des séquences de données et propagent l'information d'un pas de temps à l'autre via leur état caché. Pendant l'entraînement, la backpropagation through time (BPTT) calcule les gradients de la perte par rapport aux paramètres du réseau. Ces gradients sont multipliés à chaque pas de temps (en raison de la règle de la chaîne). Si les gradients sont inférieurs à 1, ils deviennent de plus en plus petits à mesure qu'ils se propagent à travers les pas de temps. Au bout d'un certain temps, ils deviennent si petits qu'ils n'ont plus d'impact sur la mise à jour des poids [16].

2.5.4.2 Problème d'explosion du gradient (Exploding Gradient)

À l'inverse du problème du gradient qui disparaît, si les gradients sont supérieurs à 1, ils peuvent devenir exponentiellement grands à mesure qu'ils se propagent à travers les pas de temps. Cela se produit lorsque les poids du réseau sont mal initialisés ou trop grands [16].

2.5.4.3 Difficulté à capturer des dépendances à long terme :

les RNN ont une mémoire limitée. Ils ont tendance à "oublier" les informations des premiers pas de temps lorsqu'ils traitent des séquences longues.

2.5.5 Long Short-Term Memory (LSTM)

Les LSTM sont une variante des RNN conçue pour capturer des dépendances à long terme dans les données séquentielles. Ils ont été proposés par Hochreiter et Schmidhuber en 1997

dans leur article séminal [17]

Son architecture a été conçue spécifiquement pour résoudre les problèmes des RNN (réseaux de neurones récurrents), en particulier le vanishing gradient (gradient qui disparaît) et la difficulté à capturer des dépendances à long terme.

Contrairement aux RNN standard, les LSTM introduisent une structure de mémoire appelée cellule (ou *memory cell*) qui permet de conserver l'information sur de longues périodes. Dans cette cellule on y trouve 3 entrées (l'entrée du dataset x_t , L'état caché précédent h_{t-1} hidden state en anglais et L'état de la cellule précédente C_{t-1} aussi appelé cell state en anglais) .

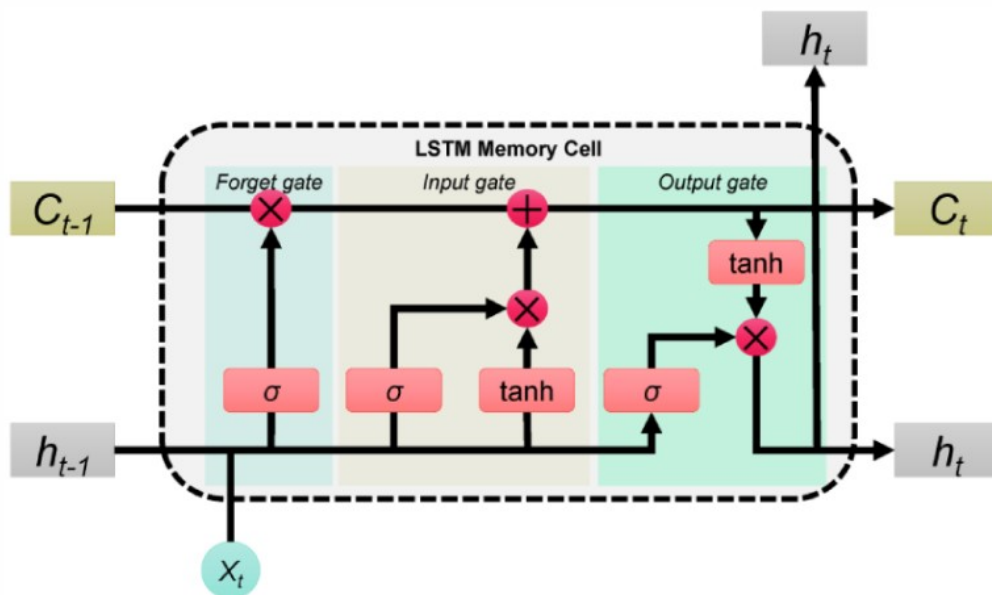


FIGURE 2.6 – Cellule LSTM [22]

2.5.5.1 Propagation Avant (Forward Propagation) :

Ces 3 entrées sont contrôlées par trois portes :

— 1. Porte d'oubli (Forget Gate)

La porte d'oubli détermine quelle partie de l'information précédente doit être oubliée. Elle prend en entrée (h_{t-1} et x_t)

Sa Formule est représentée comme suite :

$$f_t = \sigma(W_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \quad (2.5)$$

Où :

- W_{f2} : Poids de la porte d'oubli,
- b_f : Biais de la porte d'oubli,
- σ : Fonction sigmoïde (produit des valeurs entre 0 et 1).

* Elle produit une valeur entre 0 et 1 pour déterminer quelle proportion de la valeur C_{t-1} doit être conservée :

- 0 : Oublie complètement l'information.
- 1 : Garde complètement l'information.

et donc la sortie de cette étape est :

$$\text{partie oubliée} = f_t \cdot C_{t-1} \quad (2.6)$$

— **2. Porte d'entrée (input Gate) :**

La Porte d'oubli décide quelles nouvelles informations doivent être ajoutées à l'état de la cellule. Cette étape peut être divisée en 2 parties :

1. calculer le potentiel du LTM sous cette formule :

$$\tilde{C}_t = \tanh(W_C \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C) \quad (2.7)$$

Où :

- W_{f2} : Poids de la porte d'entrée,
- b_f : Biais de la porte d'entrée,
- \tanh : Fonction d'activation (produit des valeurs entre -1 et 1).

2. Mise à jour de l'état de la cellule :

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

où i_t est une sortie sigmoïde qui contrôle combien d'informations nouvelles doivent être ajoutées :

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.8)$$

- **3. Porte de sortie (Output Gate) :** La porte de sortie contrôle quelle partie de l'état de la cellule doit être envoyée comme sortie et utilisée pour le prochain état caché.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.9)$$

L'état caché final est alors obtenu via une activation tanh appliquée à l'état de la cellule mis à jour :

$$h_t = o_t \cdot \tanh(C_t) \quad (2.10)$$

la sortie finale est :

$$y_t = W_y \cdot h_t + b_y \quad (2.11)$$

2.5.5.2 Calcule D'Erreur :

Après la forward propagation, on compare les prédictions du réseau aux vraies valeurs (cibles) à chaque pas de temps.

On calcule une fonction de perte (par exemple, l'erreur quadratique moyenne) qui mesure l'écart entre les prédictions et les cibles.

$$L = \sum_{t=1}^T \text{loss}(y_t, \text{target}_t) \quad (2.12)$$

2.5.5.3 Rétropropagation à Travers le Temps (Backpropagation Through Time) :

La **BPTT** consiste à propager l'erreur à travers le réseau, mais en remontant le temps (d'où le nom "through time").

Voici comment ça se passe :

1. **Initialisation** : On commence par le dernier pas de temps $t = T$ et on calcule le gradient de la perte par rapport à l'état caché à ce pas de temps.
2. **Rétropropagation** : On remonte le temps, pas à pas, de $t = T$ à $t = 1$. À chaque pas de temps, on met à jour les gradients de la perte par rapport :
 - Aux états cachés précédents.
 - Aux poids et biais du réseau.
3. **Dépendance temporelle** : Comme l'état caché à un pas de temps dépend des états cachés précédents, les gradients doivent être propagés à travers tous les pas de temps précédents.

2.6 Conclusion

Cette partie a permis d'explorer les fondements théoriques et techniques des méthodes de Deep Learning appliquées à la prédiction. Nous avons d'abord introduit les concepts clés de l'IA et du ML

Ensuite, nous nous sommes concentrés sur le Deep Learning, en détaillant la structure et le fonctionnement des réseaux de neurones artificiels (RNA) et le (RNN)

Pour pallier ces limitations, nous avons présenté les LSTM , une architecture avancée capable de gérer efficacement les séquences temporelles grâce à leur mécanisme de portes et leur rétropropagation à travers le temps.

Chapitre 3

Analyse des Besoins et Conception

3.1 Introduction

Avant de commencer tout développement informatique, il est essentiel de bien comprendre ce que l'on attend du futur système. L'analyse des besoins permet justement de poser les bases du projet, en identifiant les attentes des utilisateurs, les contraintes du terrain, et les objectifs à atteindre. Cette étape vise à éviter toute confusion ou incompréhension qui pourrait nuire à la qualité ou à la pertinence de la solution finale.

Une fois les besoins clairement définis, on passe à la phase de conception. Celle-ci consiste à réfléchir à la manière dont le système va être organisé : quelles seront ses différentes parties, comment elles vont interagir entre elles, et comment les données vont circuler. C'est une étape clé pour assurer que le projet sera réalisable, cohérent et capable d'évoluer dans le temps.

L'analyse et la conception sont donc deux phases complémentaires et indispensables pour garantir le bon déroulement d'un projet informatique. Elles permettent d'anticiper les difficultés, de structurer le travail, et d'avoir une vision claire avant de passer à la réalisation concrète.

3.2 Présentation de Sonelgaz

3.2.1 Historique et rôle stratégique

Sonelgaz constitue l'opérateur historique de la fourniture d'électricité et de gaz en Algérie. Depuis sa création en 1969, elle s'engage à répondre aux besoins énergétiques des citoyens et des entreprises, participant ainsi à l'amélioration de leur quotidien .

Initialement structurée comme une entreprise intégrée verticalement, Sonelgaz a évolué, avec l'adoption de la loi sur l'électricité et la distribution du gaz par canalisations, vers un modèle de holding. Aujourd'hui, elle pilote un groupe industriel diversifié, composé de multiples sociétés aux métiers complémentaires.

Son engagement en faveur de la politique énergétique nationale se traduit par des réalisations majeures, notamment en matière d'électrification rurale et de distribution publique du gaz. Ces efforts ont permis d'atteindre un taux de couverture en électricité de 99 %, desservant plus de 12 millions de clients, et un taux de pénétration du gaz de 67 %, avec plus de 8 millions d'abonnés.

Structuré autour de 10 filiales directement gérées par la holding et de 04 sociétés en partenariat avec des tiers, le groupe Sonelgaz continue de jouer un rôle central dans la dynamique énergétique algérienne, tout en préservant son savoir-faire historique et son engagement envers le progrès national. [28] .



FIGURE 3.1 – Logo de la société Sonelgaz

3.2.2 La transformation de SONELGAZ en groupe

À partir de 2004, dans le cadre de la mise en œuvre de la loi n° 02-01 du 5 février 2002, SONELGAZ a entamé une profonde restructuration. Cette réforme a permis :

- L'ouverture à la concurrence dans la production d'électricité ;
- La séparation des activités (production, transport, distribution) en filiales distinctes, organisées sous forme de sociétés par actions (SPA). [28]

Ce processus a conduit à la création progressive de plusieurs entités spécialisées :

2004

- **GRTG** (Gestionnaire du Réseau de Transport du Gaz) : Société dédiée au transport du gaz.
- **SPE** (Sonelgaz Production de l'Électricité) : Regroupement des centrales existantes pour la production électrique.
- **GRTE** (Gestionnaire du Réseau de Transport de l'Électricité) : Société assurant le transport de l'électricité et la gestion transitoire du système production-transport.
- Première réforme du **FOSC** (Fonds des Œuvres Sociales et Culturelles).

2005

- **CREG** (Commission de Régulation de l'Électricité et du Gaz) : Autorité indépendante chargée de :
 - Garantir le service public ;
 - Conseiller les pouvoirs publics sur l'organisation des marchés de l'électricité et du gaz ;
 - Surveiller le respect des réglementations sectorielles.
- **SMT** (Société Civile de Médecine du Travail) : Pour la santé au travail.
- **CREDEG** (Centre de Recherche et Développement en Électricité et Gaz) : Dédié à l'innovation.
- Fusion des sociétés de maintenance :
 - Regroupement des 4 sociétés de maintenance véhicules (MPV) en une seule ;
 - Fusion des 3 sociétés de maintenance des transformateurs en une seule (SMT).
- Préparation de la filialisation de la distribution : Création de 4 directions régionales.

2006

- Création des sociétés de distribution régionales :
 - **SDA** (Sonelgaz Distribution Alger) ;
 - **SDC** (Sonelgaz Distribution Centre) ;
 - **SDE** (Sonelgaz Distribution Est) ;
 - **SDO** (Sonelgaz Distribution Ouest).
- Mise en place de l'**OS/SPT** (Opérateur Système Production-Transport).
- Nouvelle réforme du FOSC.
- Réintégration des 5 sociétés de travaux : **KAHRAKIB**, **KANAGAZ**, **KAHRIF**, **ETTERKIB** et **INERGA**.

2007

- **IFEG** (Industrie de Formation en Électricité et Gaz) : Pour la formation professionnelle.
- Réorganisation de la **SPE** en 4 pôles nationaux de production (dont un pôle Sud).

2009

- **SOPIG** (Société de Gestion du Patrimoine Immobilier) : Pour la gestion des actifs immobiliers.
- **CEEG** (Compagnie d'Engineering en Électricité et Gaz) : Pour les études et l'ingénierie.
- **ELIT** (Société des Systèmes) : Spécialisée dans les solutions technologiques.

Cette transformation a structuré SONELGAZ en un groupe intégré, avec des entités autonomes couvrant l'ensemble de la chaîne de valeur énergétique. [28]

3.2.3 Les catégories de clients de SONELGAZ

Les clients de SONELGAZ sont classés en fonction du niveau de tension électrique utilisé : basse tension (BT) ou haute tension (HT). Cette segmentation permet une adaptation optimale des infrastructures de distribution aux besoins des différents consommateurs.

1. Clients HTB (Haute Tension B) – 63 kV ou 90 kV

Profil : Grands consommateurs industriels et infrastructures énergivores.

Rôle du réseau :

- Transport de l'électricité à l'échelle régionale et locale ;
- Alimentation des industries lourdes (sidérurgie, cimenteries, etc.) ;
- Desserte des réseaux ferroviaires et des grands sites industriels.

2. Clients HTA (Haute Tension A) – 15 kV à 30 kV

Profil : Petites et moyennes entreprises (PME), commerces et industries légères.

Rôle du réseau :

- Distribution locale vers les zones d'activité économique ;
- Alimentation des transformateurs qui fournissent la basse tension aux particuliers.

3. Clients BT (Basse Tension) – 230 V ou 400 V

Profil : Ménages, artisans et petits commerces.

Rôle du réseau :

- Distribution finale vers les foyers et les petits professionnels ;
- Alimentation des appareils électroménagers et des installations domestiques.

Cette classification permet à SONELGAZ d'optimiser la gestion des flux énergétiques et d'adapter ses services aux besoins spécifiques de chaque catégorie de consommateurs.

3.3 Présentation de l'organisme d'accueil : *Sonelgaz Distribution – Direction de Béjaïa*

La Direction de Distribution de Béjaïa constitue une antenne régionale de la **Société Algérienne de Distribution de l'Électricité et du Gaz de l'Est (SDE)**, filiale du groupe Sonelgaz, dont le siège social est situé à Constantine. Elle assure la gestion opérationnelle de la distribution de l'énergie électrique et du gaz naturel dans la wilaya de Béjaïa. [28] .

Sur le plan organisationnel, cette direction se compose de plusieurs structures fonctionnelles, parmi lesquelles :

- le secrétariat général,
- les assistants du Directeur de Distribution,

— les responsables des affaires juridiques, de la communication et de la sécurité.

Elle s'appuie également sur une organisation territoriale comprenant :

— neuf divisions techniques,

— dix agences commerciales situées à : Béjaïa Cité Tobal, Béjaïa 4 Chemins, El Kseur, Amizour, Sidi Aïch, Seddouk, Akbou, Tazmalt, Aokas et Kherrata,

— cinq districts répartis entre : Béjaïa, Akbou, Sidi Aïch, Amizour et Kherrata.

[28] .

3.3.1 Organisation de Sonelgaz Distribution – Direction de Béjaïa

La structure organisationnelle des différentes divisions de la Direction de Distribution de Sonelgaz à Béjaïa est illustrée dans la figure 3.2 ci-dessous.

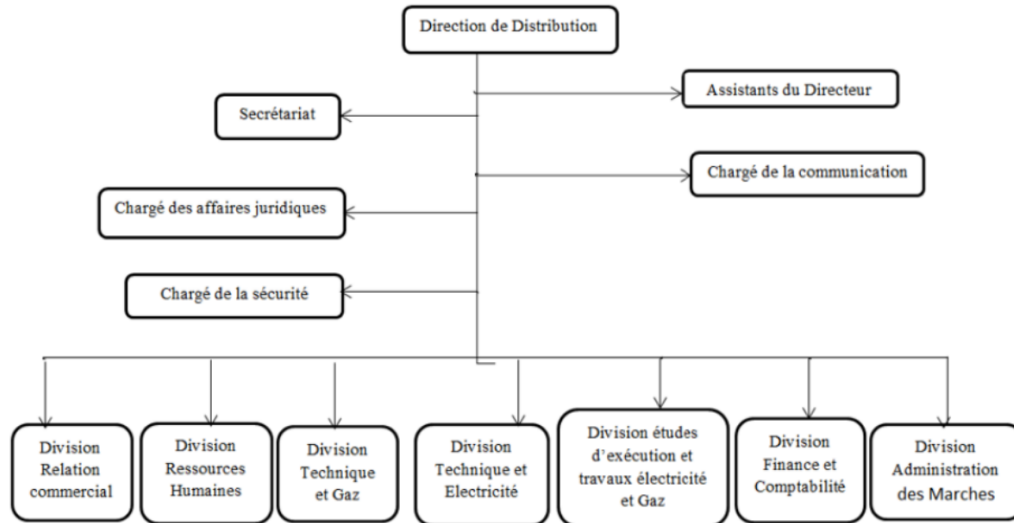


FIGURE 3.2 – Organigramme de la Direction de Distribution de Sonelgaz à Béjaïa [28].

3.3.1.1 Division des Relations Commerciales

La Division des Relations Commerciales occupe une place stratégique au sein de la Direction de Distribution. Elle est chargée de plusieurs missions essentielles [28], notamment :

— la commercialisation de l'électricité et du gaz,

— la gestion des relations clients,

— la facturation sur mémoire (FSM),

— l'analyse statistique (notamment le suivi des pertes techniques et non techniques),

— l'amélioration continue de la qualité de service et de la satisfaction client.

Cette division est organisée en trois services distincts :

1. Service Développement des Ventes (RCN)

Ce service a pour principales responsabilités :

- la prise en charge des demandes de raccordement des clients au réseau électrique,
- l'orientation et le conseil des clients Moyenne Tension / Moyenne Pression (MT/MP) concernant les modalités de raccordement et le niveau de tension ou de pression requis,
- la facturation des devis liés aux raccordements (RCN) ainsi que des prestations diverses,
- l'établissement des ordres d'exécution des travaux après règlement des devis par les clients.

2. Service Recouvrement

Ce service assure :

- la gestion des comptes clients HT/IIP, MT/MP, BT/BP,
- le respect des délais de facturation pour les différents groupes,
- l'émission d'avertissements en cas de retard de paiement (ordres de coupure, mises en demeure, etc.),
- le contrôle et l'archivage de l'ensemble des pièces comptables,
- le suivi personnalisé des grands comptes,
- le suivi du processus de facturation et de recouvrement dans le cadre du système FSM.

3. Service Grands Comptes

Les missions de ce service incluent :

- la gestion, la télérelève et la facturation mensuelle des clients MT/MP et HT/HP,
- l'établissement de contrats pluriannuels (d'une durée de 5 ans) à signer avant la mise en service des postes clients MT/MP,
- le suivi et l'évaluation de la qualité de service dédiée aux grands comptes.

3.4 Problématique

Le secteur de la distribution de l'électricité est confronté à des défis de plus en plus complexes, notamment en raison de la croissance de la demande, des exigences de performance, et de la nécessité d'optimiser les ressources. Dans ce contexte, la capacité à produire de l'électricité de manière équilibrée, à détecter efficacement les fraudes, et à prendre des décisions fondées sur des données fiables représente un enjeu stratégique pour les entreprises du domaine.

La société Sonelgaz, à travers sa Direction de Distribution de Béjaïa, illustre bien ces problématiques. Au cours de notre stage au sein de cette entité, plusieurs défis opérationnels majeurs ont été identifiés :

- **Une gestion déséquilibrée de la production électrique** : Sonelgaz éprouve des difficultés à ajuster la production à la demande réelle. Une production insuffisante entraîne des pénuries, tandis qu'une surproduction provoque des pertes, l'électricité n'étant pas stockable. Cette instabilité rend la gestion énergétique complexe.
- **Une détection de la fraude encore insuffisante** : La présence de systèmes de comptage défectueux chez certains clients complique l'évaluation précise de la consommation réelle, ce qui engendre des erreurs de facturation et des pertes financières.
- **Des méthodes d'analyse traditionnelles** : L'entreprise repose encore sur des outils classiques pour l'analyse des ventes et l'aide à la décision, ce qui limite sa capacité à réagir efficacement aux évolutions du marché et aux attentes des clients.

L'intégration de l'IA apparaît alors comme une solution pertinente pour répondre à ces enjeux. En effet, l'analyse prédictive, basée sur des données historiques et en temps réel, permettrait d'anticiper la demande et d'optimiser la production. De plus, des systèmes intelligents de surveillance pourraient améliorer la détection des fraudes, tandis que des outils d'analyse avancés offriraient une meilleure compréhension des comportements clients.

Dans un contexte où l'électricité constitue un levier fondamental du développement économique, l'adoption de technologies intelligentes représente une opportunité pour Sonelgaz d'améliorer son efficacité opérationnelle, de renforcer sa performance et de garantir un service de qualité.

3.5 Solutions proposées

Dans un contexte où la prise de décision en matière de ventes devient de plus en plus complexe, il est essentiel pour les entreprises de s'appuyer sur des méthodes avancées d'analyse et de prévision. L'essor de l'IA, en particulier les techniques de ML et de DL, a ouvert de nouvelles perspectives, incitant de nombreuses entreprises à adopter ces technologies pour améliorer leur gestion opérationnelle.

L'intelligence artificielle constitue aujourd'hui un levier incontournable pour développer des outils d'analyse prédictive fiables, capables de répondre aux enjeux liés à la gestion des ventes. Dans le cadre de ce projet, il s'agit de concevoir une application informatique intelligente dédiée à l'analyse et à la prédiction des consommations électriques des clients, répondant aux besoins de la Direction de Distribution de Béjaïa – Sonelgaz.

Ce système permettra une meilleure compréhension des habitudes de consommation des clients, une anticipation plus précise de la demande énergétique. L'objectif est d'apporter à Sonelgaz un outil d'aide à la décision performant, capable d'optimiser la gestion des ventes d'électricité et de réduire les pertes énergétiques.

3.5.1 Fonctionnalités principales de l'application

L'application intelligente que nous proposons intégrera les fonctionnalités suivantes :

1. Authentification et inscription des utilisateurs.
2. Visualisation des ventes électriques des différents clients.
3. Visualisation de la consommation globales des clients .
4. Prédiction des consommations à l'aide de modèles LSTM (*Long Short-Term Memory*).
5. Prédiction des consommations via les méthodes ARIMA et SARIMA.
6. Visualisation de l'historique des prédictions de la consommation globale.
7. Exportation des données au format PDF , PNG ou CSV.

En résumé, l'application développée offrira à Sonelgaz une solution technologique innovante pour améliorer la prise de décision stratégique en matière de ventes. Grâce à l'intégration des techniques d'intelligence artificielle les plus récentes, elle contribuera à une meilleure gestion des ressources énergétiques, tout en apportant une vision fine et actualisée des comportements de consommation de la clientèle.

3.6 Les méthodologies de gestion de projet

La gestion de projet repose sur des méthodologies permettant d'organiser, de planifier et de contrôler efficacement les différentes phases d'un projet. Deux grandes approches sont couramment utilisées dans le domaine : la méthode traditionnelle et la méthode agile. Chacune présente des avantages et des inconvénients selon la nature du projet, les objectifs fixés, et le contexte organisationnel.

3.6.1 les Méthodes Agile et Traditionnelle

1. La méthode traditionnelle : La méthode traditionnelle suit une approche séquentielle, où chaque étape du projet doit être terminée avant de passer à la suivante. Elle est souvent utilisée pour les projets à exigences bien définies dès le départ [4].

Caractéristiques principales :

- Définition claire des besoins en début de projet.
- Phases bien distinctes : analyse, conception, développement, tests, déploiement.
- Documentation complète à chaque étape.
- Faible flexibilité en cas de changements en cours de projet.

2. La méthode Agile La méthode Agile repose sur une approche itérative et incrémentale. Elle est conçue pour être flexible et réactive face aux changements de besoins, avec une forte implication des parties prenantes tout au long du projet [4].

Caractéristiques principales :

- Division du projet en itérations courtes (sprints).
- Livraison fréquente de versions fonctionnelles.
- Collaboration étroite avec le client.
- Adaptation rapide aux changements.

Comparaison globale

Critère	Méthode Traditionnelle	Méthode Agile
Structure	Séquentielle	Itérative
Flexibilité	Faible	Élevée
Implication du client	Faible	Forte
Livraison	Une seule fois à la fin	Progressive, à chaque sprint
Adaptation au changement	Difficile	Facile

3.6.2 Le cadre de travail Scrum :

Scrum n’est pas une méthode stricte, mais un cadre de travail (framework) destiné à gérer des projets complexes et évolutifs. Il vise à livrer des produits de haute valeur de manière productive, créative et collaborative.

Ce framework repose sur des itérations courtes et incrémentales, permettant à une équipe de progresser par amélioration continue afin de mieux répondre aux besoins des clients [5].

Scrum s’appuie sur trois piliers fondamentaux :

- **Transparence** : Tout le monde dans l’équipe doit avoir accès aux informations importantes. Cela permet de mieux travailler ensemble et de comprendre où en est le projet. Par exemple, chacun sait ce qui a été fait, ce qu’il reste à faire, et s’il y a des problèmes [5].
- **Inspection** : L’équipe prend régulièrement le temps de vérifier le travail réalisé. Cela se fait souvent à la fin de chaque sprint. Le but est de voir si ce qui a été produit répond bien aux besoins, et s’il y a des erreurs ou des choses à améliorer [5].
- **Adaptation** : Une fois les problèmes ou les améliorations identifiés, l’équipe s’adapte. Elle change sa manière de faire ou modifie le produit pour mieux répondre aux attentes. L’idée est de toujours chercher à s’améliorer et à rester flexible [5].

3.6.2.1 Les Rôles principaux de Scrum

Scrum repose sur **trois rôles principaux** pour aider l’équipe à bien travailler ensemble :

- **Le Product Owner** (ou propriétaire du produit) : il représente les besoins des clients. Il gère la liste des tâches à faire (le backlog) et décide ce qui est prioritaire pour apporter le plus de valeur possible [3].
- **Le Scrum Master** : c'est comme un coach. Il aide l'équipe à bien appliquer les règles de Scrum, à résoudre les blocages et à s'améliorer. Il veille à la bonne communication et au bon déroulement des réunions [3].
- **L'Équipe de développement** : ce sont les personnes qui font le travail. Elles s'organisent elles-mêmes pour réaliser les tâches du sprint, construire le produit et livrer des résultats de qualité [3].

Chacun a son rôle bien défini, mais tous travaillent ensemble pour atteindre un même objectif, créer un produit utile, de façon efficace et continue.

3.6.2.2 Le processus Scrum et ses concepts clés

Scrum se repose sur des cycles courts appelés *sprints*, au cours desquels une équipe pluridisciplinaire travaille de façon collaborative à la réalisation d'un ensemble d'objectifs clairement définis.

Le processus Scrum commence par la définition des exigences, où l'équipe identifie la vision du produit, les fonctionnalités souhaitées, la valeur apportée aux utilisateurs, ainsi que le public cible. Ces éléments alimentent le Product Backlog, une liste ordonnée et priorisée des fonctionnalités, améliorations et corrections à apporter au produit. Cette liste est maintenue par le Product Owner, qui veille à ce que les besoins métier soient correctement reflétés.

La planification du sprint (ou Sprint Planning) est la réunion qui ouvre chaque sprint, période fixe généralement comprise entre 2 et 4 semaines. Lors de cette réunion, l'équipe sélectionne un ensemble d'éléments du Product Backlog qu'elle s'engage à réaliser durant le sprint. Cet engagement permet de fixer un objectif clair et réalisable.

Pendant le sprint, l'équipe travaille sur les tâches planifiées. Chaque jour, une réunion rapide de 15 minutes, appelée Daily Scrum ou mêlée quotidienne, permet aux membres de synchroniser leurs efforts, de faire le point sur l'avancement, d'identifier les blocages éventuels et d'adapter leur organisation si nécessaire.

À la fin du sprint, une revue du sprint (Sprint Review) est organisée. L'équipe présente le travail accompli au Product Owner et aux parties prenantes. Cette étape favorise les retours constructifs, la validation des fonctionnalités, et la révision éventuelle du backlog.

Enfin, une Rétrospective du sprint (Sprint Retrospective) est tenue en interne, afin d'analyser ce qui a bien fonctionné, les difficultés rencontrées, et de définir des actions d'amélioration pour les prochains sprints. [34]

3.7 Délimitation du domaine d'étude

3.7.1 Répartition des rôles selon la méthode Scrum

Dans le cadre de ce projet, la méthodologie Scrum a été adoptée pour assurer une gestion agile, itérative et collaborative du développement. La répartition des rôles s'est faite comme suit :

- **Scrum Master : M. Foudil MIR**

Il joue le rôle de facilitateur entre le Product Owner et l'équipe de développement. Il veille à la bonne application du cadre Scrum, supprime les obstacles rencontrés par l'équipe, favorise la communication interne, et encourage l'amélioration continue de la productivité ainsi que du savoir-faire collectif. Il a aussi la responsabilité de garantir que les réunions Scrum (daily meeting, sprint planning, sprint review, sprint retrospective) se déroulent correctement.

- **Product Owner : M. Dalil HADJOUT**

Il est responsable de la vision du produit. Il définit les exigences fonctionnelles du projet en tenant compte des besoins du client. Il établit et priorise les éléments du Product Backlog, veille à ce que l'équipe de développement comprenne bien les tâches à réaliser, et valide les fonctionnalités livrées à la fin de chaque sprint.

- **Équipe de développement :**

Elle est chargée de transformer les exigences fonctionnelles en un produit logiciel conforme aux attentes. Elle participe activement à toutes les phases du sprint : analyse, développement, tests et intégration continue. L'équipe se charge également de la gestion du code source, de sa maintenance et de l'amélioration de ses performances.

L'équipe est composée de :

- Mlle **BENSEKHRIA Asma.**
- Mlle **CHOUF Dihya.**

3.7.2 Identification des acteurs

Dans cette section, nous identifions les différents acteurs susceptibles d'interagir avec le système.

Un acteur peut être défini comme une entité (humaine ou logicielle) qui joue un rôle particulier dans l'interaction avec le système à modéliser.

Les principaux acteurs de notre système sont les suivants :

- **Administrateur :** Il s'agit d'une personne disposant des droits d'accès à l'espace administrateur après son authentification. il a la capacité de gérer les différentes ressources du système.

- **Utilisateur** : Il s'agit d'un individu déjà enregistré dans la base de données, qui se connecte à la plateforme afin d'accéder aux fonctionnalités qui lui sont réservées.

3.8 Organisation du projet selon SCRUM

3.8.1 Product Backlog

Le product backlog est une liste priorisée de toutes les fonctionnalités, améliorations, corrections ou tâches nécessaires pour développer le produit. Il est géré par le Product Owner et évolue en permanence selon les retours, les priorités ou les besoins du client [34].

ID du User Story	Nom du UserStory	Description du User Story	Priorité	Statut
US01	Authentification	<ul style="list-style-type: none"> — En tant qu'administrateur, je dois pouvoir me connecter à l'application en saisissant mon identifiant et mon mot de passe pour accéder aux fonctionnalités de gestion tout en sécurité. — En tant qu'utilisateur, je souhaite me connecter et quitter mon compte en toute sécurité pour gérer mes accès. 	Haute	Fait
US02	Visualisation des consommations globale des clients	<ul style="list-style-type: none"> — En tant qu'administrateur, je souhaite visualiser les données de consommation électrique des clients du secteur économique, sur une période donnée, sous forme de graphiques et de tableaux . Je pourrais aussi télécharger ces données sous différent format (CSV , PDF où PNG). 	Haut	Fait
→ voir suite...				

ID du User Story	Nom du UserStory	Description du User Story	Priorité	Statut
US03	Prédiction des consommations globales des clients	<ul style="list-style-type: none"> — En tant qu'administrateur, je veux pouvoir prédire les consommations globales des clients , avec différentes méthodes et je veux que les détails de cette prédiction s'affiche comme la période prédite , les paramètres utilisés ... ect . à la fin je veux pouvoir télécharger le résultat afficher sous différents formats (CSV , PDF ou PNG) . 	Haute	Fait
US04	Visualisation de la consommation personnelle d'un client avec leurs prédictions	<ul style="list-style-type: none"> — En tant qu'administrateur, je souhaite visualiser les données de consommation électrique de chaque client, ainsi que leurs consommations futures, tout en ayant la possibilité d'importer de nouvelles données . Et aussi télécharger le résultat affiché sous différent format (CSV , PDF ou PNG) . — En tant que client, je souhaite visualiser mes données réelles ainsi que celles du prochain trimestre .Et aussi télécharger le résultat affiché sous différent format (CSV , PDF ou PNG) . 	Haut	Fait
→ voir suite...				

ID du User Story	Nom du UserStory	Description du User Story	Priorité	Statut
US05	Sauvgarde des anciennes prédictions	— En tant qu’administrateur, je préfère que les anciennes prédictions soient sauvegardées dans une page différente.	Faible	Fait

TABLE 3.1 – Product Backlog

3.8.2 Planification des sprints

Un sprint est une période de travail courte et définie (souvent 1 à 4 semaines) dans la méthode Agile, durant laquelle une équipe développe une partie fonctionnelle et livrable du produit. L’objectif est de livrer une version utilisable ou testable à la fin de chaque sprint [34].

Sprint	Nom du sprint	Période
Sprint 1	Page d’accueil et Authentification	1 semaine
Sprint 2	Gestion des clients et visualisation de leurs consommations avec leur prédiction	2 semaine
Sprint 3	Visualisation de la consommation et de leur prédiction dans chaque compte d’un client	1 semaine
Sprint 4	Visualisation des consommations globales	1 semaine
Sprint 5	Prédiction des consommation globale avec des modèles arima et sarima, , avec possibilité d’impression des données CSV, PNG ou PDF	2 semaines
Sprint 6	Prédiction des consommation globale avec un modèle du deep learning (lstm) , avec possibilité d’impression des données CSV, PNG ou PDF	1 semaine
Sprint 7	Visualisation des prédictions historique	1 semaine

TABLE 3.2 – Planification des sprints

3.9 Diagramme de contexte

Un diagramme de contexte est une vue très haute du système qui met en évidence les interactions entre le système principal et ses acteurs externes, sans détailler son fonctionnement interne [33]. Les intervenants qui interagissent avec le système au moyen de différents messages dans notre application sont les suivants :

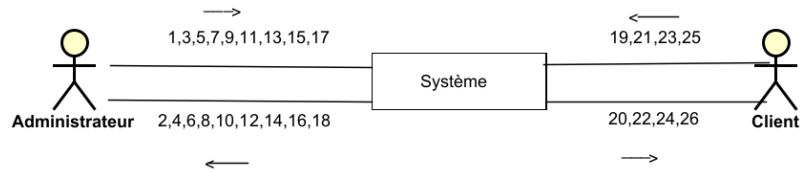


FIGURE 3.3 – Diagramme de contexte

Acteur	Numéro	Message	Numéro	Message
Administrateur	1	Demande d'authentification	2	Afficher le profil
	3	Demande d'analyse statistique	4	Afficher le dashboard
	5	Demande d'afficher la liste des clients	6	Affichage de l'interface des clients inscrit
	7	effectuer une recherche d'un client	8	afficher le client choisi
	9	Demander la gestion des clients	10	Accès aux fonctionnalités de gestion
	11	Demande de visualisation des consommations et prédictions des clients	12	Affichage de l'interface des données et prédiction
	13	Demande de prédiction	14	Afficher l'interface de prédiction.
	15	Effectuer un choix de prédiction	16	Affichage de résultat de prédiction
	13	Demande d'accès à l'historique de prédiction	14	Affichage de l'interface historique
	15	Demande de télécharger les données pdf, png ou csv	16	Document téléchargé
	17	Demande de déconnexion	18	Affichage de l'interface d'accueil
Client	19	Demande d'authentification	20	Affichage du profil client
	21	Demande d'affichage des consommations et des prédictions personnelles	22	Affichage de l'interface dashboard
	23	Demande de téléchargement des données en pdf, png ou csv	24	Document téléchargé
	25	Demande de déconnexion	26	Affichage de l'interface d'accueil

TABLE 3.3 – Description tabulaire des messages échangés entre le système et les acteurs

3.10 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation (en anglais *Use Case Diagram*) est un type de diagramme UML (Unified Modeling Language) utilisé pour représenter les interactions entre un système et ses utilisateurs (appelés acteurs). Il permet de modéliser les fonctionnalités d'un système du point de vue des utilisateurs.

La figure 3.4 représente le diagramme de cas d'utilisation de notre application.

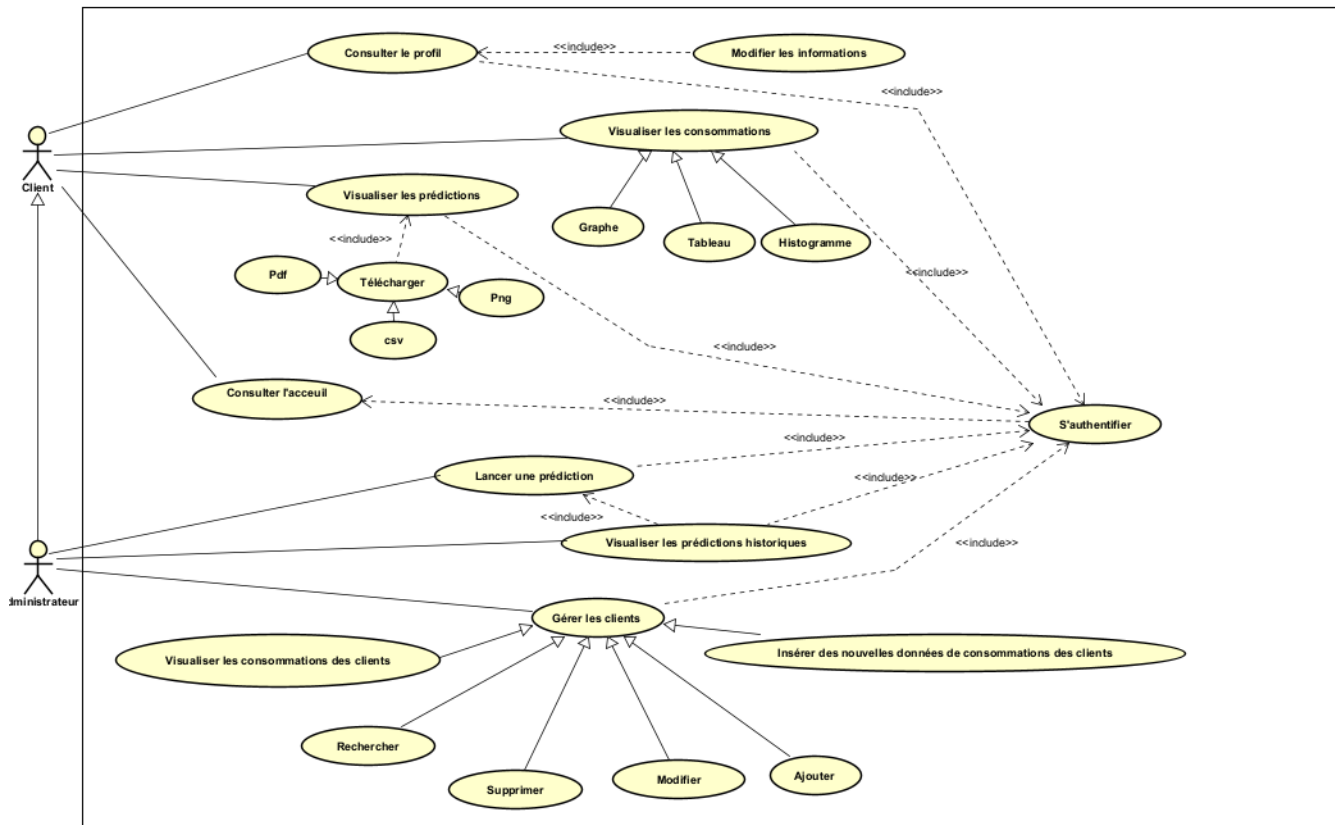


FIGURE 3.4 – Diagramme de cas d'utilisation

3.11 Description des cas d'utilisation

La description textuelle d'un cas d'utilisation constitue une formalisation précise d'un scénario fonctionnel, permettant de modéliser les interactions entre un ou plusieurs acteurs et un système donné. Présentée sous forme de tableau, elle explicite de manière structurée les différentes composantes du cas d'utilisation, notamment les objectifs, les conditions préalables, le déclencheur, les étapes du scénario principal, les cas alternatifs ainsi que les postconditions.

Cette description vise à clarifier le comportement attendu du système, tout en facilitant

l'analyse des besoins fonctionnels et la communication entre les parties prenantes du projet logiciel.

Nom du cas d'utilisation	
Acteurs	Les acteurs qui vont réaliser le cas d'utilisation
Description	Une description résumée du cas d'utilisation
Auteurs	Les intervenants de l'élaboration de la fiche
Date	La date de rédaction ou mise à jour de la fiche
Pré-conditions	Les conditions nécessaires pour déclencher le cas d'utilisation
Description des scénarios	
Scénario nominal	La description des interactions entre acteur et système, dans le meilleur des cas
Scénario alternatif	Description d'une situation hypothétique où les événements diffèrent de la trajectoire principale
Post-conditions	État du système après réalisation du cas d'utilisation

TABLE 3.4 – Structure type d'un cas d'utilisation

3.12 Diagramme de séquence

Un **diagramme de séquence** est un diagramme d'interaction UML qui décrit comment les objets interagissent entre eux au fil du temps. Il illustre l'enchaînement chronologique des messages échangés entre les différents acteurs et composants d'un système pour réaliser un scénario spécifique d'un cas d'utilisation [6]. Dans le cadre de cette étude, nous avons choisi cinq cas d'utilisation représentatifs à modéliser à l'aide de diagrammes de séquence : l'authentification, lancer une prédiction, visualisation des consommations et prédictions d'un client, l'ajout d'un client et l'insertion de nouvelles données.

3.12.1 Cas d'utilisation s'authentifier

Le tableau fournit une description textuelle détaillée du cas d'utilisation intitulé «S'authentifier».

S'authentifier	
Acteurs	Administrateur
Description	L'administrateur s'authentifie en utilisant ses identifiants (nom d'utilisateur et mot de passe) pour accéder à l'application
Auteurs	CHOUF Dihya, BENSEKHRIA Asma
Date	1 Avril 2025
Pré-conditions	L'administrateur doit avoir un compte valide (un nom d'utilisateur et un mot de passe valide)
Description des scénarios	
Scénario nominal	<ol style="list-style-type: none">1. L'administrateur ouvre l'application et est redirigé vers l'écran d'authentification.2. Il saisit son nom d'utilisateur et son mot de passe.3. Le système vérifie la validité des identifiants saisis.4. Si les informations sont correctes, l'administrateur est redirigé vers l'interface principale de l'application.
Scénario alternatif	<ol style="list-style-type: none">1. L'administrateur saisit un nom d'utilisateur ou un mot de passe invalide.2. Le système affiche un message d'erreur indiquant que les informations sont incorrectes et invite l'administrateur à les ressaisir.3. L'administrateur corrige les identifiants et soumet à nouveau.4. Le système vérifie à nouveau les informations saisies et, si elles sont valides, redirige l'administrateur vers l'interface principale de l'application.
Post-conditions	L'administrateur est authentifié avec succès et peut accéder aux fonctionnalités de l'application.

TABLE 3.5 – Description du cas d'utilisation "S'authentifier"

La figure 3.5 représente le diagramme de séquence détaillé du cas d'utilisation "S'Authentifier"

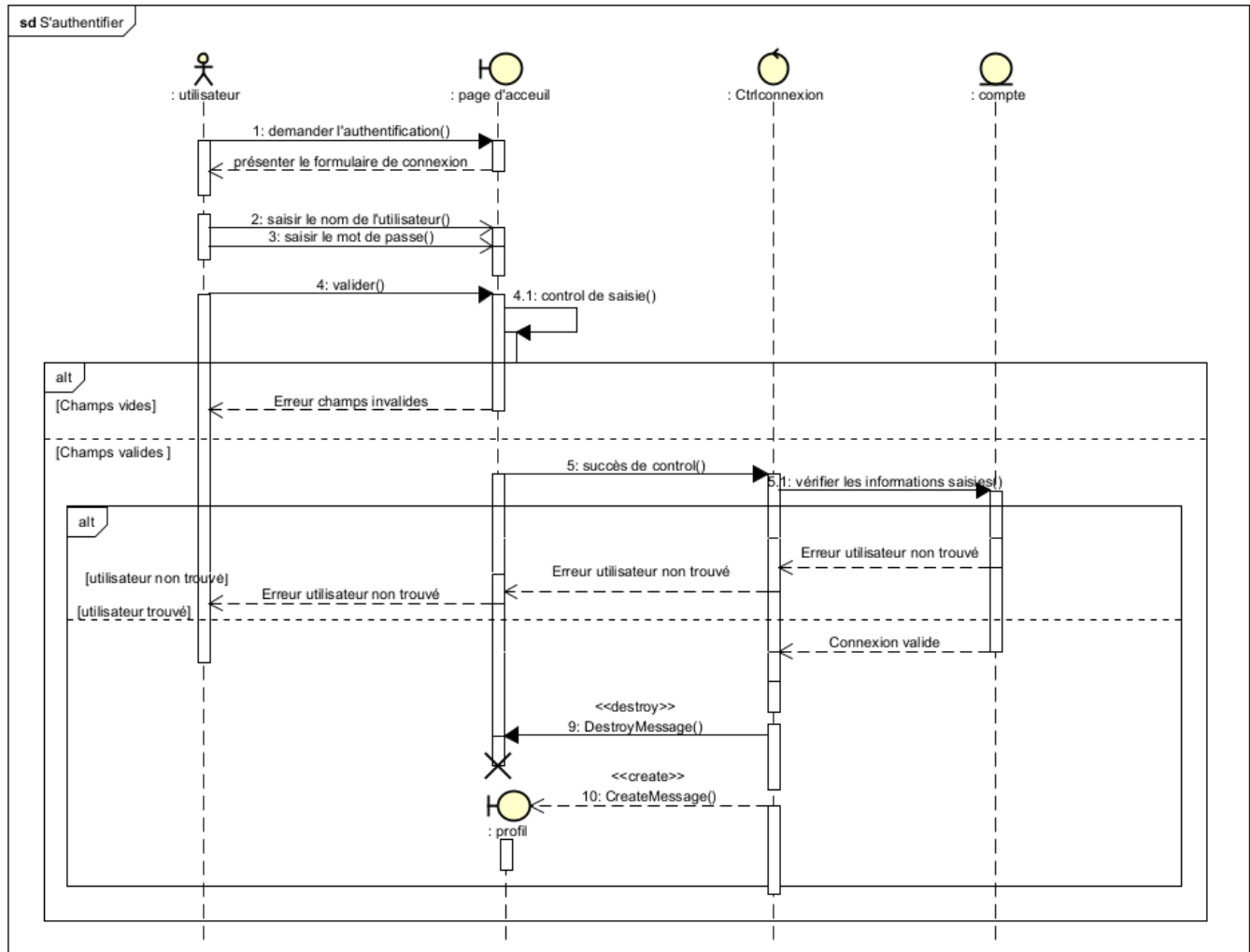


FIGURE 3.5 – Diagramme de séquence détaillé du cas d'utilisation "S'Authentifier"

3.12.2 Cas d'utilisation "lancer une prédiction"

Le tableau fournit une description textuelle détaillée du cas d'utilisation intitulé «lancer une prédiction».

Lancer une prédiction	
Acteurs	Administrateur
Description	Cette fonctionnalité permet à l'administrateur de visualiser les prédictions des consommations liées à la consommation globale des clients. Les prédictions sont présentées sous forme de graphiques et de tableaux.
Auteurs	CHOUF Dihya, BENSEKHRIA Asma
Date	8 Mai 2025
Pré-conditions	L'administrateur doit être authentifié et avoir accès aux données des clients.
Description des scénarios	
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur clique sur créer une prédiction et choisit la méthode de prédiction. 2. Le système affiche les périodes disponible. 3. L'administrateur choisit la période souhaitée et lance la prédiction. 4. Le système génère et affiche les résultats de la prédiction. 5. L'administrateur peut exporter les résultats aux formats PDF, PNG ou CSV.
Scénario alternatif	<ol style="list-style-type: none"> 1. L'administrateur lance une prédiction. 2. Le système vérifie la disponibilité des données de consommation et affiche une erreur des données indisponible . 3. L'administrateur insère les données. 4. Le système vérifie à nouveau les données saisies et, si elles sont valides, il affiche la prédiction.
Post-conditions	Les prédictions sont générées et affichées à l'administrateur.

TABLE 3.6 – Description du cas d'utilisation "Lancer une prédiction "

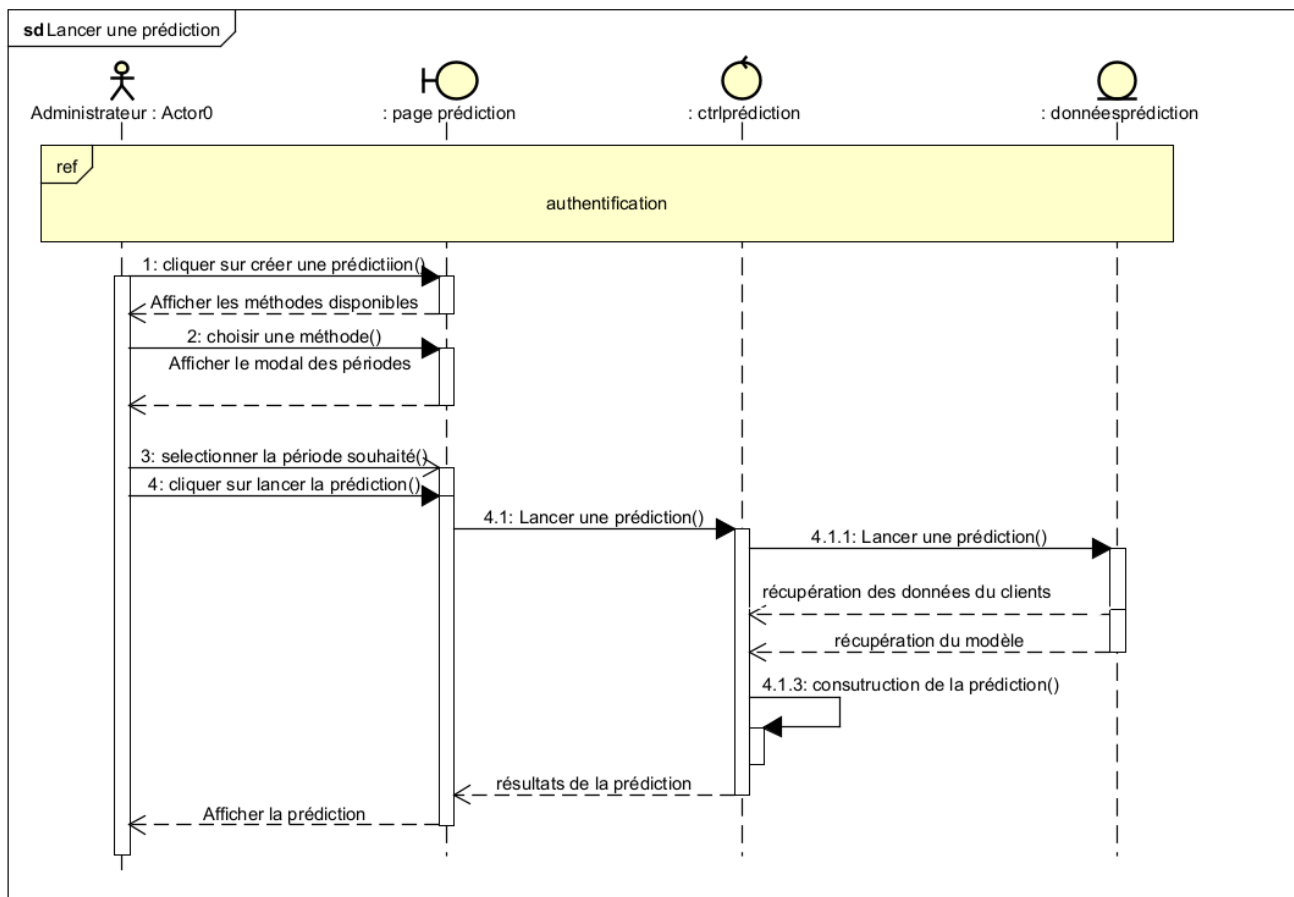


FIGURE 3.6 – Diagramme de séquence du cas d'utilisation "Lancer une prédiction"

3.12.3 Cas d'utilisation "visualiser les consommations d'un client"

visualiser les consommation d'un client	
Acteurs	Administrateur
Description	offre la possibilité d'ajouter un nouveau client tout en inserant ces données de consommations
Auteurs	BENSEKHRIA Asma , CHOUF Dihya
Date	11 Mai 2025
Pré-conditions	l'authentification de l'utilisateur
Description des scénarios	
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur effectue une recherche d'après le code du client qu'il desire chercher 2. le résultat de la recherche s'affiche sur la même interface 3. L'utilisateur demande une visualisation des données réel et future du client du'il désirer visualiser 4. Le système affiche les Résultats.
Post-conditions	l'affichage du résultat

TABLE 3.7 – Discription du cas d'utilisation "visualiser des consommation d'un client "

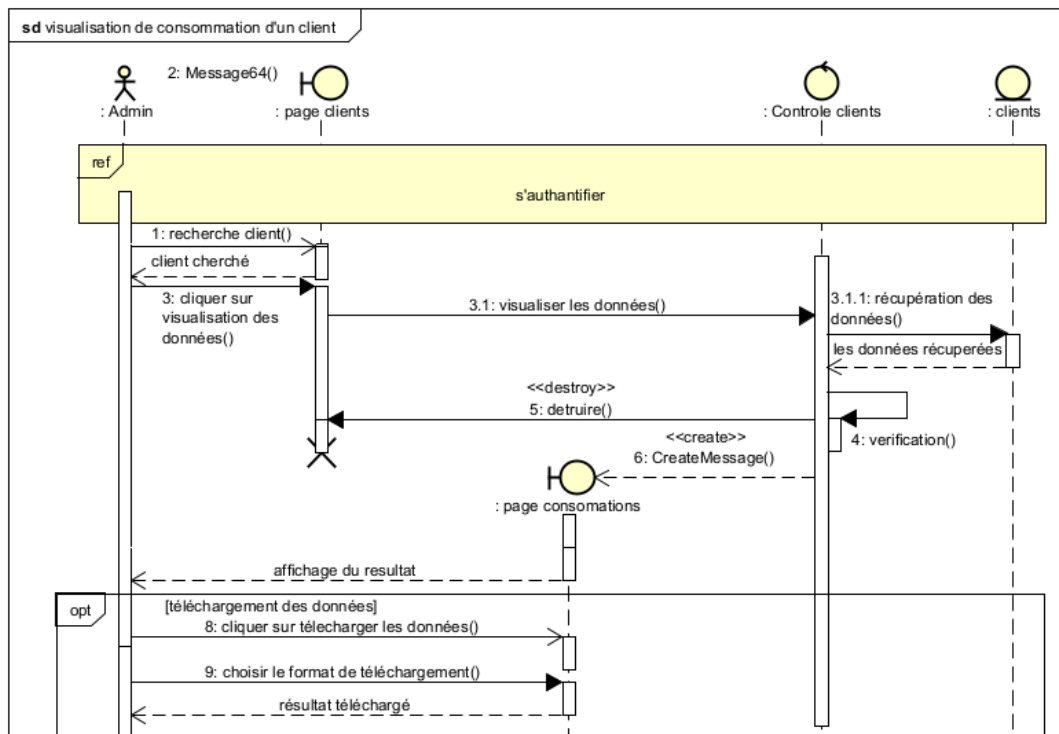


FIGURE 3.7 – Diagramme de sequence du cas d'utilisation "visualiser des consommation et les predictions d'un client"

3.12.4 Cas d'utilisation "ajouter un client"

ajouter un client	
Acteurs	Administrateur
Description	offre la possibilité d'insérer de nouvelles données d'après le client choisit précédement et de rounouvler la prediction dapres ces nouvelles données
Auteurs	BENSEKHRIA Asma , CHOUF Dihya
Date	8 Avril 2025
Pré-conditions	l'authentification de l'utilisateur
Description des scénarios	
Scénario nominal	<ol style="list-style-type: none"> 1. l'utilisateur clique sur ajouter un nouveau client et affichage du formulaire 2. remplir le formulaire 3. sauvgarder les données envoyée
Scénario alternatif	l'enchainement démmare du 2ème point du scénario nominal <ol style="list-style-type: none"> 1. le système verifie les données inserer 2. le système detecteune erreur de saisie ou l'existence de l'utilisateur 3. affichage d'un message d'erreur
Post-conditions	l'affichage du client dans la liste des clients

TABLE 3.8 – Discription du cas d'utilisation "ajouter un client "

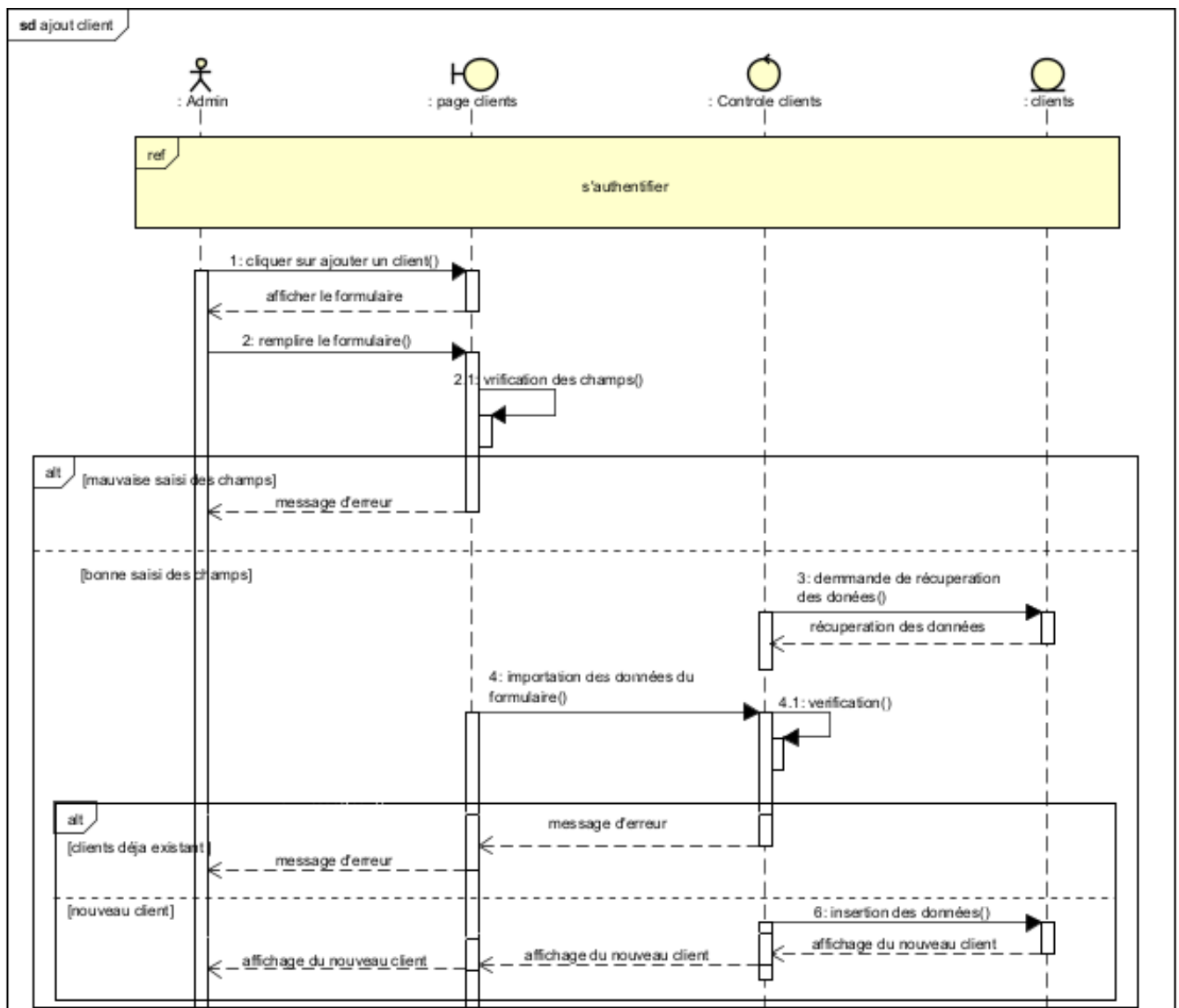


FIGURE 3.8 – Diagramme de séquence cas d'utilisation "ajouter un client "

3.12.5 Cas d'utilisation "insertion de nouvelles données"

insertion de nouvelles données	
Acteurs	Administrateur
Description	offre la possibilité d'insérer de nouvelles données d'après le client choisit précédement et de rounouveler la prediction dapres ces nouvelles données
Auteurs	BENSEKHRIA Asma , CHOUF Dihya
Date	5 Juin 2025
Pré-conditions	l'authentification de l'utilisateur
Description des scénarios	
Scénario nominal	<ol style="list-style-type: none"> 1. l'utilisateur choisi de d'importet de nouvelles données du client 2. Ces données sont s'auvgarder 3. le modèle se renouvlé d'après les nouvelles données inserée 4. la visualisation des données réel et predites se change d'après la nouvelle insertion
Scénario alternatif	<p>l'enchainement démmare du 5ème point du scénario nominal</p> <ol style="list-style-type: none"> 1. le système verifie les données inserer 2. le système detecte un grand ecard entres les dates des données ancienne et des données inserer 3. affichage d'un message d'erreur
Post-conditions	l'affichage du résultat

TABLE 3.9 – Discription du cas d'utilisation "insertion de nouvelles données "

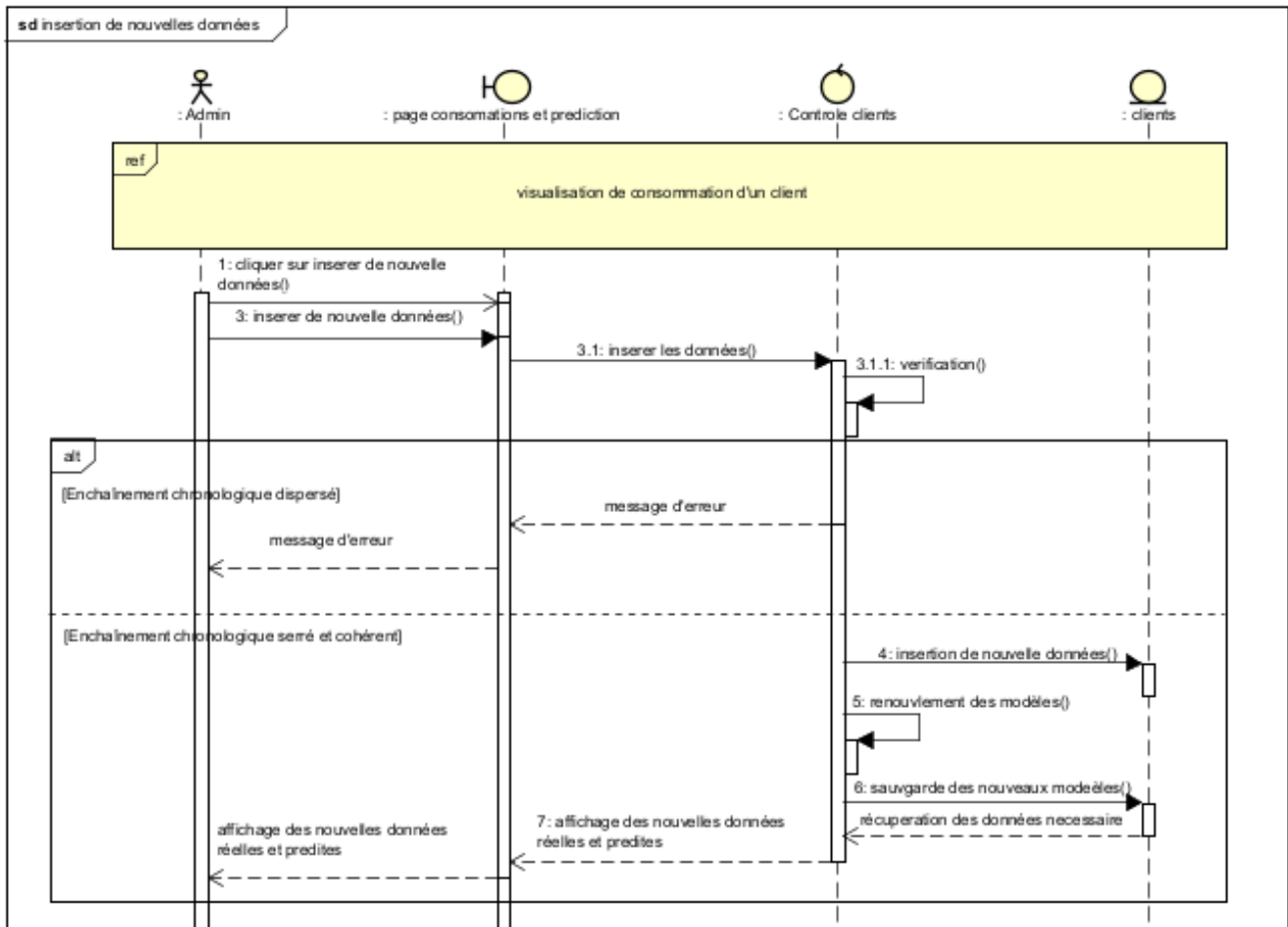


FIGURE 3.9 – Diagramme de séquence cas d'utilisation "insertion de nouvelles données "

3.13 Diagramme de classe

Un diagramme de classes est un diagramme structurel de l'UML (Unified Modeling Language) qui décrit la structure statique d'un système logiciel [6]. Il montre les classes du système, leurs attributs, leurs méthodes (opérations), ainsi que les relations statiques (associations, agrégations, compositions, généralisations) entre elles. Ce diagramme permet de modéliser les entités principales d'un système et leurs interactions structurelles.

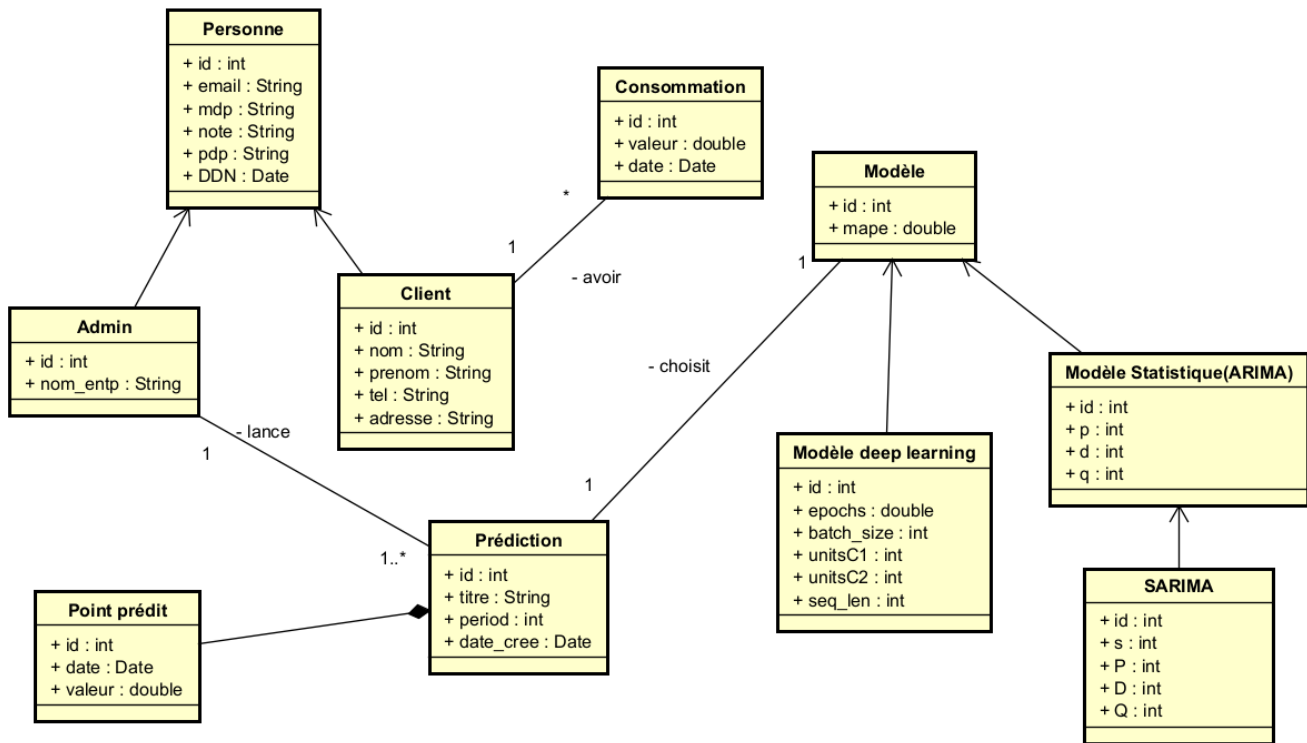


FIGURE 3.10 – Diagramme de classe

3.14 Conclusion

Ce chapitre nous a permis de détailler les fonctionnalités nécessaires que l’application devait contenir. Nous avons commencé par la présentation de l’entreprise SONELGAZ ainsi que de ses différentes structures d’accueil, pour ensuite définir la problématique rencontrée par l’entreprise et proposer des solutions concrètes, adaptées aux utilisateurs.

Par ailleurs, le choix de la méthodologie agile (SCRUM) nous a permis de réaliser le projet de manière flexible, tout en facilitant son adaptation aux éventuels changements au cours de son évolution.

Enfin, ce chapitre se conclut par une phase de conception qui a permis de poser les bases techniques et fonctionnelles du projet.

Chapitre 4

Réalisation

4.1 Introduction

Le développement d'une application consiste à utiliser de différents outils et de concepts de technologie interconnectée. La présentation des outils utilisés dans le mémoire est fondamentale pour éclaircir l'environnement dont l'application a été développée. De plus, une présentation des interfaces de l'application ainsi que les résultats obtenus sont cruciaux pour démontrer le résultat attendu de ce projet.

4.2 Environnement de développement de l'application

4.2.1 Langages de programmation

- **Python** : python est un langage de programmation interprété, orienté objet, et de haut niveau, connu par sa syntaxe claire et concise. Il est largement utilisé dans le développement web, l'analyse de données, l'intelligence artificielle, et bien d'autres domaines. Python met l'accent sur la lisibilité du code et la productivité des développeurs [13].
- **JavaScript** : est un langage de programmation interprété, principalement utilisé pour rendre les pages web interactives. Il fonctionne côté client (dans le navigateur) mais peut également être utilisé côté serveur via des environnements comme Node.js [12].

4.2.2 Framework de programmation





Un Framework de programmation est une infrastructure logicielle réutilisable qui offre des bibliothèques et des outils permettant de structurer son code et de faciliter le développement de l'application en imposant souvent un cadre de travail à suivre.

- **REACT JS** : est une bibliothèque javascript open source développée par Meta (anciennement Facebook), est conçue pour le développement des interfaces utilisateur (UI)

, ça particularité est que le code est découper en commposants différent ce qui offre la possibilité de réutiliser cette partie du code , ces compossants gère leurs propre état (state) ce qui mettent automatiquement à jour lorsqu'il change, ce qui rend l'interface réactive et fluide .(Bien que React soit officiellement une bibliothèque JavaScript dédiée à la construction d'interfaces utilisateur, elle est souvent utilisée comme un framework grâce à son écosystème riche. Dans ce travail, nous l'avons adoptée comme fondation du front-end.) . [24]

- **Taiwind CSS** : est un Framework utilitaire (ensemble de classes CSS très simples) , contrairement a des Framework comme Bootstrap qui offre des composant prés a utiliser . [29]
- **Fast API** : est un framework web moderne et rapide pour Python, il est basé sur les la création des API d'une manière rapide , facile et propre en créant des API REST . [26]

4.2.3 Outils et logiciels utilisé

-  **Visual Studio Code (VS Code)** : VS Code est un éditeur de code source léger, gratuit et open-source développé par Microsoft. Il prend en charge de nombreux langages de programmation, notamment Python, JavaScript , HTML, C++, etc. Il offre des fonctionnalités avancées comme l'autocompilation, le débogage, l'installation d'extensions [25].
-  **Anaconda** : Anaconda est une distribution libre et open-source de python conçue pour le calcul scientifique, l'analyse de données, l'intelligence artificielle et l'apprentissage automatique. Elle inclut des outils populaires comme Jupyter Notebook, Spyder, et plus de 1500 packages scientifiques [1].
-  **Astar UML** : Astar UML est un logiciel de modélisation qui permet de créer des diagrammes UML , tels que les diagrammes de classes , de séquence de cas d'utilisation,etc. Il est utilisé principalement en ingénierie logicielle pour visualiser et concevoir des système orientés objets [2].
-  **GitHub** : Github est une paletforme de développement collaboratif basée sur le système de gestion de versions Git. Elle permet aux développeurs de stocker, gérer et partager leurs code source, collaborer sur des projets, suivre les modifications , et gérer les versions [15].

4.2.4 Architecture

L'application à été développé en utilisant le framework **React**, dont l'architecture repose sur une approche modulaire et orienté composants. cette architecture permet une organisation

calire du code, favorise la réutilisabilité des éléments, et facilite la maintenance ainsi que l'évolution du projet.

l'architecture mise en place s'articule autour des éléments suivants :

- **Composants** : chaque fonctionnalité de l'interface utilisateur est encapsulée dans un composant autonome (ex : boutons, cartes d'information). Ces composants sont réutilisables dans les différentes parties de l'application.
- **Pages** : les pages correspondent aux vues principales de l'application, telles que la page d'accueil, la page profil. Elles sont construites en assemblant plusieurs composants.
- **Gestion de l'état** : l'état local des composants est géré à l'aide des hooks React (useState, useEffect). Pour les données partagées entre plusieurs composants, un système d'état global est mis en place via le Context API.
- **Services** : les interactions avec le backend (appel d'API, envoi de données, récupération d'information) sont centralisées dans les modules de services, afin de séparer la logique métier de la couche présentation.
- **Ressources** : les fichiers statistiques tels que les images ou les feuilles de style sont regroupés dans un répertoire dédié (assets/), ce qui permet une gestion centralisée des ressources.

Cette architecture modulaire permet d'organiser efficacement le projet, en assurant une séparation des responsabilités, une meilleure lisibilité du code, et une facilité d'évolution du système.

4.2.5 Bibliothèques Python utilisés

- **Pandas** : est une bibliothèque open source conçue pour la manipulation et l'analyse des données comme les DataFrames, facilitant la lecture, l'écriture, le filtrage et le traitement efficace des données tabulaires [31].
- **NumPy** : est une bibliothèque fondamentale pour les calculs scientifique et mathématiques en Python. Elle offre de nombreuses fonctionnalités et outils utiles pour les projets de data science [11].
- **Matplotlib** : est une bibliothèque de visualisation de données en 2D. Elle permet de créer des graphiques statique, interactifs et personnalisables , comme des courbes, histogrammes, etc [10].
- **TensorFlow** : est une bibliothèque open source développée par Google, destinée au calcul numérique et à l'apprentissage automatique. Elle est principalement utilisée pour l'entraînement de modèles de machine learning et deep learning à grande échelle [32].
- **Keras** : est une bibliothèque open source pour la construction et l'entraînement des réseaux de neurones. Elle fournit une interface simple pour TensorFlow, et est utilisée pour des tâches d'apprentissage profond [30].

- **Scikit-Learn** : est une bibliothèque simple et efficace pour l'apprentissage automatique en Python. Elle fournit une large gamme d'outils pour le traitement de données. Elle est construite sur des bibliothèques fondamentales comme NumPy, Matplotlib, ce qui la rend puissante tout en restant facile à utiliser. Elle est largement utilisée dans les projets de data science pour la création des modèles prédictifs [23].

4.3 Modélisation des méthodes de prédiction appliquées

4.3.1 Modélisation dynamique

Contrairement à la modélisation simple la modélisation dynamique se concentre sur l'évolution d'un système dans le temps. Elle prend en compte les changements des variables et les interactions entre elles au fil du temps . Parmi les nombreuses approches de modélisation dynamique on peut citer la méthode Box-Jenkins .

4.3.1.1 Méthode Box-Jenkins

La méthode Box-Jenkins est une approche systématique utilisée pour analyser et modéliser des séries temporelles, afin de prévoir ou de comprendre leur comportement future [7]. Elle se base sur quatre étapes principales :

1. Identification du modèle :

- **Objectif** : Déterminer la structure du modèle qui décrit le mieux le dataset.
- **Étapes clés** :
 - **décomposition de la série** : La décomposition consiste à séparer la série temporelle en plusieurs composantes (tendance, saisonnalité , résidu) . Cela permet de mieux comprendre la structure de la série et d'identifier les composantes dominantes.
 - **Stationnarité** : Vérifier si la série temporelle est stationnaire (c'est-à-dire si sa moyenne et sa variance sont constantes dans le temps). Si ce n'est pas le cas, appliquer des transformations (comme la différenciation) pour la rendre stationnaire.
 - **Analyse des autocorrélations** : Sur la base des graphiques ACF et PACF, déterminer les ordres des termes autorégressifs (AR) et des termes de moyenne mobile (MA) qui seront inclus dans le modèle.

2. Estimation des paramètres :

- **Objectif** : Estimer les coefficients du modèle sélectionné à l'étape d'identification.
- **Étape clés** : Utiliser des méthodes statistiques (comme la méthode du maximum de vraisemblance ou les moindres carrés) pour estimer les paramètres du modèle.

3. Validation du modèle :

- **Objectif** : Vérifier que le modèle est bien adapté aux données et qu'il capture correctement la structure de la série temporelle.
- **Étapes clés** :
 - **Analyse des résidus** : Les résidus (erreurs) du modèle doivent ressembler à un bruit blanc (moyenne nulle , variance constante et absence d'autocorrélation) .
 - **Critères d'information** : Comparer différents modèles en utilisant des critères comme l'AIC (Akaike Information Criterion) ou le BIC (Bayesian Information Criterion) pour sélectionner le modèle le plus approprié.

Si le modèle n'est pas valide, retourner à l'étape d'identification pour ajuster les ordres ou explorer d'autres structures de modèle.

4. Prévision :

- **Objectif** : Utiliser le modèle validé pour prédire les valeurs futures de la série temporelle.
- **Étape clés** : Générer des prévisions en utilisant les valeurs passées de la série et les erreurs passées .

4.4 Processus de prédiction en LSTM

4.4.1 Préparation des Données

1- Normalisation des données

La normalisation des données est la première étape à réaliser dans ce processus. Elle consiste à mettre les données à une échelle comprise entre 0 et 1 . Cette étape est essentielle pour l'architecture de la cellule LSTM, qui utilise des fonctions d'activation telles que tanh ou sigmoid, sensibles aux valeurs d'entrée trop grandes ou trop petites.

Par ailleurs, la normalisation permet à toutes les variables d'être sur un pied d'égalité, en évitant qu'une variable n'écrase l'influence des autres.

Pour cela, une méthode appelée MinMaxScaler est utilisée.

2-Séparation Des Données

Cette étape est nécessaire pour garantir une évaluation fiable sur les prédictions réaliser , l'idée est de diviser les données en deux parties :

- **Train (entraînement)** se sont les données du qui sont utiliser pour construire et entrainer notre modèle .elle represente 80% des données réelles .

- **Test (évaluation)** c'est avec ces données qu'on teste la prédiction et de l'évaluer en faisant une comparaison .elle représente 20% des données du dataset .

3- Creation des séquences

Pour entraîner un modèle LSTM, les données temporelles doivent être transformées en séquences ,cela est essentielle pour crée des liens logiques ou statistiques entre les valeurs d'une série au fil du temps. Voici en comment ces données sous forme de sequanaces sont organiser :

- **Entrée (X)** : Une matrice de séquences de taille $N \times T \times F$, où :
 - N = nombre d'échantillons.
 - T = longueur de la séquence .
 - F = nombre de caractéristiques .
- **Sortie (y)** : Un vecteur de valeurs à prédire correspondant à l'élément $T + 1$.

4.4.2 Construction et Entraînement du Modèle LSTM

1- Construction

La conception du modèle LSTM implique la configuration de son architecture interne, qui doit être adaptée aux spécificités des données temporelles traitées. Cette phase inclut la définition de la structure du réseau, avec le choix du nombre de couches, de leur dimension, des fonctions d'activation et des mécanismes de régularisation, afin d'optimiser le modèle pour l'entraînement.

2- Entraînement et évaluation

Cette étape constitue le cœur du processus de modélisation. Elle permet d'ajuster le modèle aux données d'apprentissage et de vérifier sa capacité à généraliser sur de nouvelles données.

a- Entraînement : L'entraînement vise à apprendre les paramètres internes du modèle (poids et biais) afin de minimiser l'erreur entre les prédictions et les valeurs réelles.

- **Objectif des données d'entraînement** : `X_train` contient les séquences temporelles utilisées comme entrées, et `y_train` les valeurs à prédire. Le but est de faire correspondre la sortie du modèle aux valeurs cibles.
- **Objectif des paramètres d'entraînement** :
 - `epochs` définit le nombre de fois que l'ensemble des données est parcouru, ce qui permet un apprentissage progressif.
 - `batch_size` permet de diviser l'apprentissage en petits sous-ensembles, améliorant la stabilité et la vitesse de l'optimisation.

- `validation_split` permet de réserver une portion des données pour évaluer les performances du modèle en dehors de l'apprentissage, détectant les risques de sur-apprentissage.
 - **Rôle de l'arrêt anticipé (EarlyStopping)** : Il permet de stopper automatiquement l'entraînement dès que les performances sur les données de validation cessent de s'améliorer, évitant ainsi que le modèle ne s'adapte trop fortement aux données d'apprentissage.
- b- Évaluation** : L'évaluation permet de vérifier la **précision du modèle sur des données qu'il n'a jamais vues**, ce qui reflète sa capacité de généralisation.
- **Utilisation du RMSE (Root Mean Squared Error)** : Cette métrique mesure l'écart moyen entre les valeurs prédites et les valeurs réelles. Un RMSE faible indique que le modèle fait des prédictions proches de la réalité.
 - **Visualisation des résultats** : La comparaison graphique entre les prédictions du modèle et les données réelles permet une évaluation visuelle et intuitive des performances. Cela aide à identifier d'éventuels écarts systématiques ou des erreurs majeures.

4.5 Aperçu de l'application

4.5.1 Développement de l'interface web

L'interface utilisateur de notre application web a été développée à l'aide du framework React.js. Ce dernier permet de construire des interfaces dynamiques et réactives en structurant l'application sous forme de composants réutilisables.

Chaque composant correspond à une partie spécifique de l'interface, comme les tableaux de données, les filtres de recherche ou encore les formulaires de prédiction. Grâce à React, l'affichage s'adapte automatiquement aux changements d'état, offrant une expérience fluide et interactive.

La communication avec le backend se fait à travers des requêtes HTTP (`fetch`), ce qui permet de récupérer ou d'envoyer des données (par exemple : lancer une prédiction, consulter l'historique, supprimer une entrée). Les données reçues sont ensuite utilisées pour mettre à jour l'interface en temps réel, sans recharger la page.

React apporte ainsi une grande souplesse dans le développement de l'interface, tout en garantissant une bonne maintenabilité et évolutivité du projet.

4.5.2 Les interfaces réalisées

Nous allons maintenant présenter différentes interfaces de notre application et expliquer leurs utilités respective :

4.5.2.1 Interface D'accueil

L'interface accueil présenté dans la figure 4.1, est la première page affichée à l'utilisateur (client et administrateur), elle lui offre la possibilité de se connecter en lui offrant un formulaire de connexion lui permettant d'accéder aux fonctionnalités du système.



FIGURE 4.1 – Interface Accueil

4.5.2.2 Interface profil

L'interface profil présente dans la figure 4.2 est la première interface qui s'affiche à l'administrateur après sa connexion. Elle affiche ses informations personnelles et lui permet de les modifier.

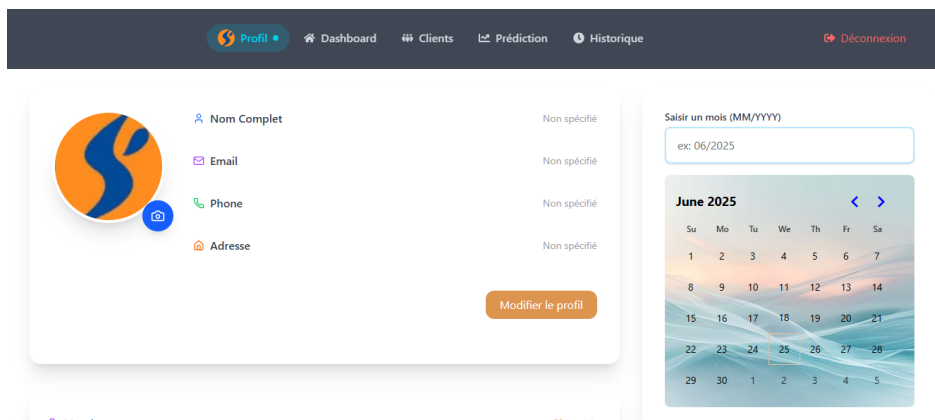


FIGURE 4.2 – Interface profil de l'administrateur

4.5.2.3 Interface Tableau de bord

L'interface du Dashboard (tableau de bord), présentée dans la figure 4.3, permet à l'administrateur de visualiser les données globales de l'entreprise ainsi que les informations relatives

aux caractéristiques clés. Elle offre également la possibilité de télécharger ces données aux formats PDF, CSV et PNG.

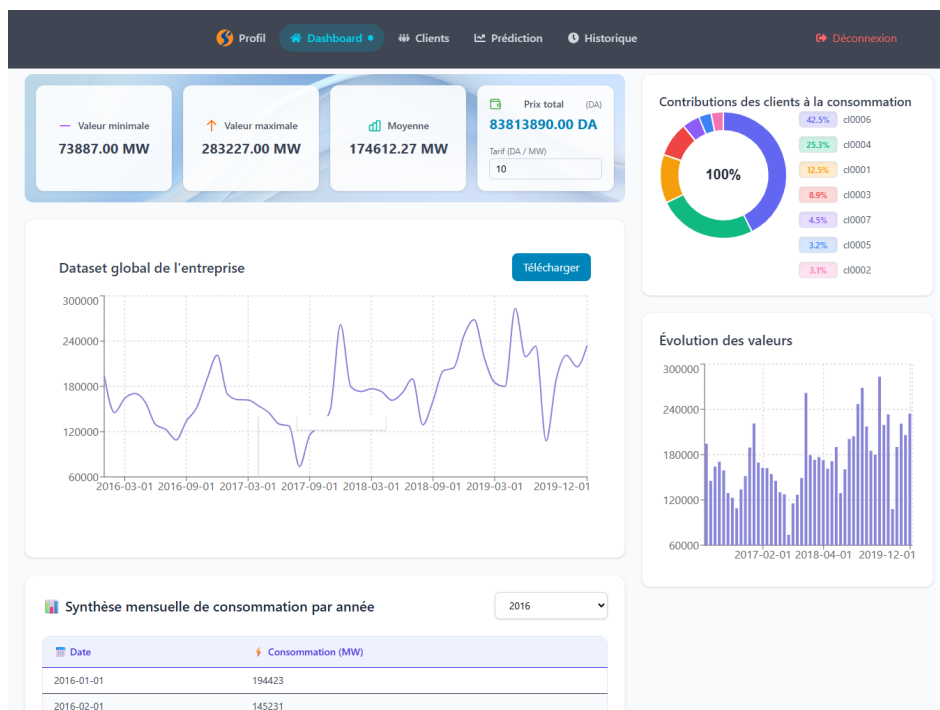


FIGURE 4.3 – Interface tableau de bord

4.5.2.4 Interface Clients

L'interface Clients, illustrée dans la figure 4.4, permet de visualiser la liste des clients de l'entreprise Sonelgaz ainsi que leurs informations personnelles. L'administrateur a la possibilité d'ajouter un client, de modifier ou supprimer ses données, ainsi que d'y insérer de nouvelles données de consommation.

ID	Code Client	Nom	Prénom	Date de Naissance	Adresse	Téléphone	Email	Actions
19	c0001	client1	clNom	1999-09-09	aoukass	0559730653	c1@gmail.com	Visualiser les données
20	c0002	client2	clNom	1999-09-09	Melbou	0669730653	c2@gmail.com	Visualiser les données
21	c0003	client3	clNom	1999-09-09	Amizour	0779730653	c3@gmail.com	Visualiser les données
22	c0004	client4	clNom	1999-09-09	Béjaja	0669738653	c4@gmail.com	Visualiser les données
23	c0005	client5	clNom	1999-09-09	tichy	0559730888	c5@gmail.com	Visualiser les données

FIGURE 4.4 – Interface clients

Une fois connecté à l'interface Clients, l'administrateur peut accéder à l'interface Visualiser les données, illustrée dans la figure 4.5. Celle-ci lui permet de consulter les données de consommation des clients, d'afficher les prédictions sous différents formats, ainsi que de visualiser le taux d'erreur associé à la prédiction.

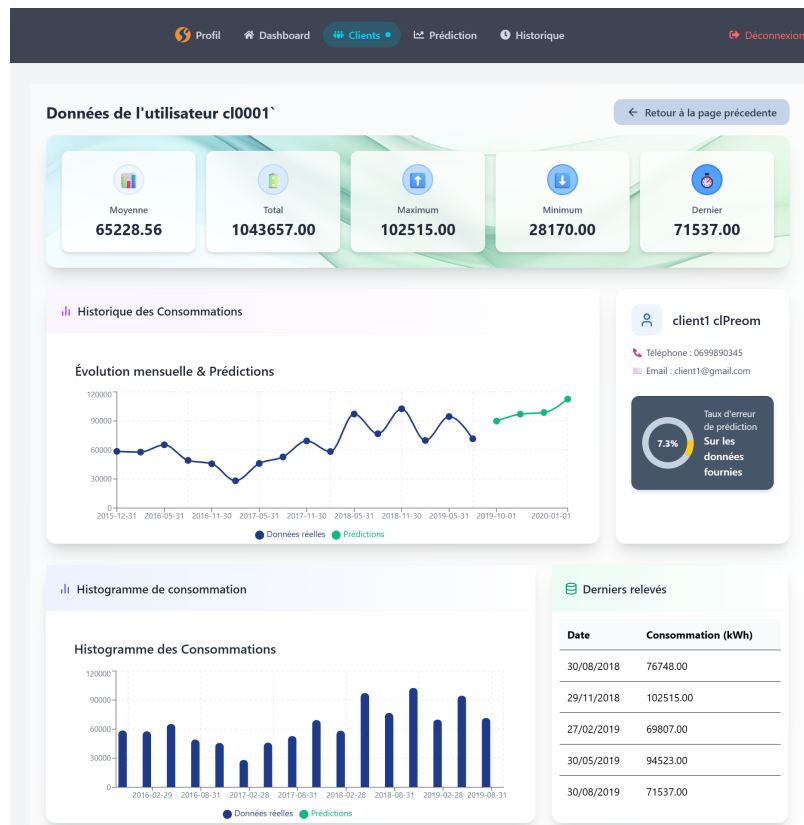


FIGURE 4.5 – Interface Visualiser les données du client

4.5.2.5 Interface historique

L'interface illustrée dans la figure 4.6 répertorie les anciennes prédictions, sauvegardées automatiquement lors de chaque lancement. Elle permet d'afficher les détails de chaque prédiction, avec la possibilité d'une visualisation approfondie sous forme de graphiques et d'informations détaillées.

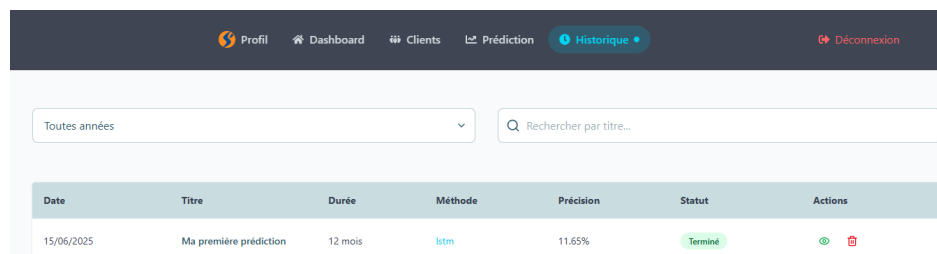


FIGURE 4.6 – Interface Historique

4.5.2.6 Interface prédiction

L'interface illustrée dans la figure 4.7 permet de lancer et de visualiser les prédictions effectuées sur le dataset global de consommation des clients. Elle propose trois méthodes de prédiction : ARIMA, SARIMA et LSTM. L'administrateur peut choisir la méthode souhaitée ainsi que la période à prédire (mois, trimestre ou année). Une fois ces choix effectués, les résultats de la prédiction sont affichés sous forme de graphique ou de tableau, accompagnés des informations clés associées.

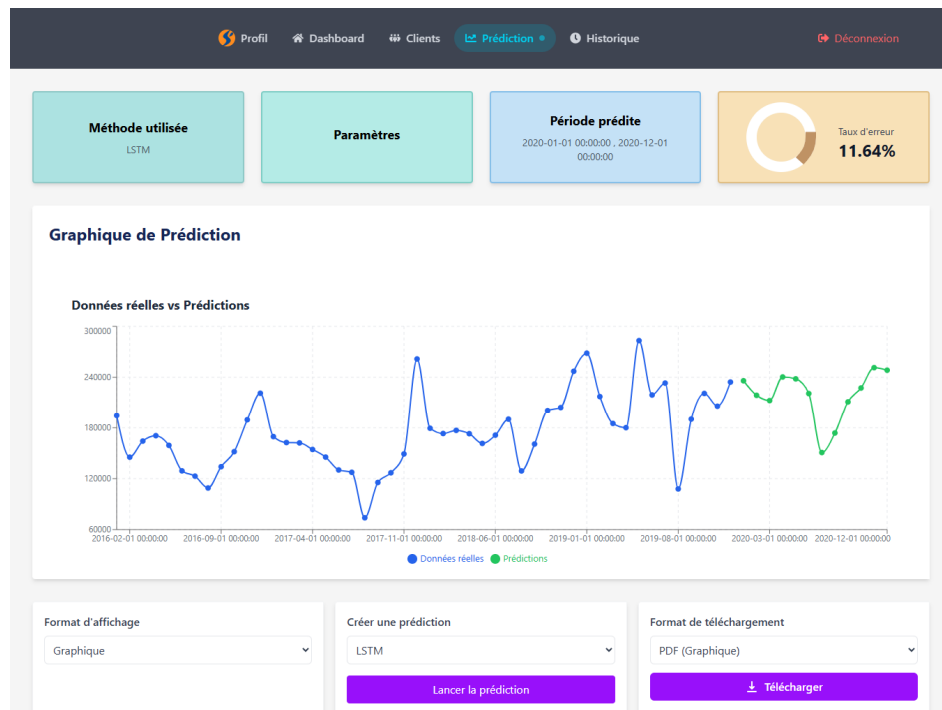


FIGURE 4.7 – Interface Prédiction

4.5.2.7 Interface Dashboard Client

L'interface illustrée dans la figure 4.8 représente l'espace auquel le client accède afin de visualiser ses données personnelles, accompagnées des prédictions pour les quatre prochains trimestres, incluant une estimation des paiements prévus pour chacun d'eux.

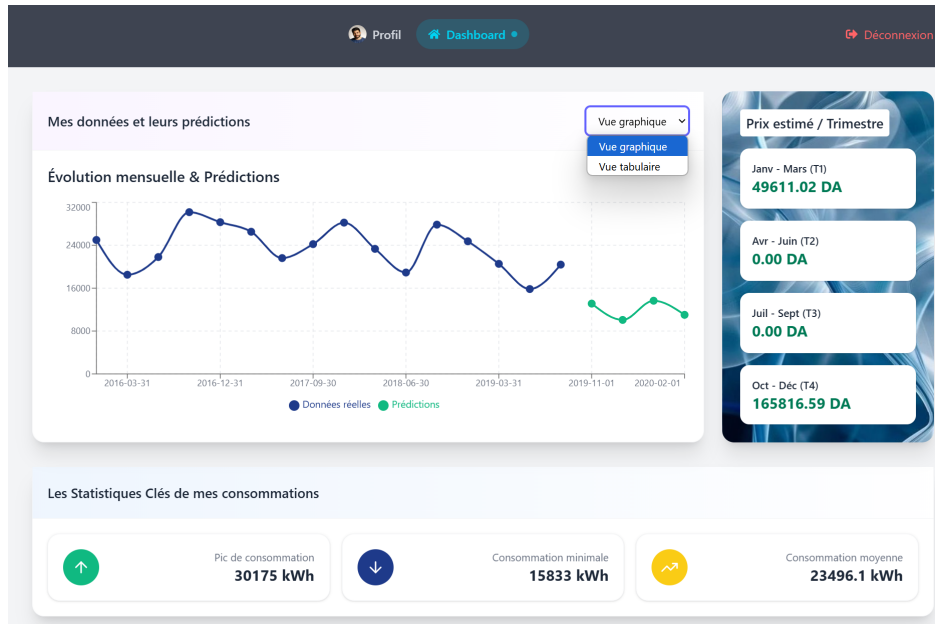


FIGURE 4.8 – Interface Dashboard Client

4.6 Analyse et comparaison des résultats des méthodes

Dans le cadre de notre étude, nous avons appliqué trois approches différentes pour prédire la consommation d'électricité des sept clients d'une durée de 4 années dont l'entreprise nous a fournis : les modèles **ARIMA**, **SARIMA** et **LSTM**. On été évalué sur la base du *Mean Absolute Percentage Error* (MAPE) afin de mesurer la précision des prédictions générées.

Le modèle **ARIMA** a fourni des résultats corrects, avec un **taux d'erreur avoisinant les 23%**. Bien qu'efficace pour des séries sans saisonnalité prononcée, il s'est montré moins performant dans notre cas, où les données comportent des variations cycliques. Son principal atout réside dans sa simplicité et sa rapidité de mise en œuvre.

Le modèle **SARIMA**, qui prend en compte les composantes saisonnières de la série, a permis d'améliorer les performances, avec un **MAPE réduit à environ 15%**. Il a su mieux représenter les fluctuations périodiques caractéristiques de la consommation électrique, ce qui a contribué à des prédictions plus cohérentes sur le moyen terme.

Le modèle **LSTM**, quant à lui, s'est distingué par ses **résultats les plus satisfaisants**, avec un **taux d'erreur inférieur à 10%**. Grâce à sa capacité à apprendre les dépendances temporelles complexes sur de longues séquences, il a généré des prédictions plus proches des

valeurs réelles. Ce modèle d'apprentissage profond s'est avéré particulièrement efficace pour saisir les dynamiques non linéaires et les irrégularités du comportement énergétique.

Ces résultats soulignent l'importance d'adapter le choix du modèle en fonction de la nature des données. L'intégration de méthodes classiques et modernes permet d'obtenir une vue plus complète et plus précise, et ce travail confirme l'intérêt de recourir à des modèles avancés tels que les **LSTM** dans les problématiques industrielles de prévision.

4.7 Conclusion

Ce chapitre a permis de présenter l'ensemble de travail réalisé, depuis le choix des technologies et outils de développement jusqu'à la modélisation des prévisions à l'aide de techniques d'apprentissage automatique et profond. Enfin, des captures d'écran fournies en fin de chapitre illustrent de manière visuelle le fonctionnement de l'application, mettant en lumière les fonctionnalités développées et les interfaces utilisateur.

4.8 Conclusion et perspectives

Dans un contexte énergétique marqué par une demande croissante en électricité, la capacité à anticiper la consommation est devenue un enjeu stratégique pour les entreprises du secteur, notamment Sonelgaz. À travers ce mémoire, nous avons exploré différentes approches de prévision dans le but de développer une application intelligente capable d'analyser et de prédire la consommation électrique à Béjaïa.

Nous avons d'abord étudié les fondements des séries temporelles et mis en œuvre des modèles statistiques classiques tels que ARIMA et SARIMA, reconnus pour leur simplicité et leur efficacité sur des données linéaires et saisonnières. Ensuite, nous nous sommes tournés vers les méthodes issues du deep learning, notamment les réseaux de neurones récurrents (RNN) et Long Short-Term Memory (LSTM), qui se sont révélés particulièrement performants pour capturer des dynamiques complexes, non linéaires et à long terme dans les données.

Cette double approche a permis de comparer les performances des différentes méthodes de prédiction, et de tirer parti de leurs avantages respectifs. L'intégration de ces modèles au sein d'un tableau de bord interactif constitue une véritable valeur ajoutée pour les décideurs de Sonelgaz, leur offrant une visualisation claire des données et des prévisions fiables pour une meilleure planification énergétique.

Ce projet a permis de démontrer l'apport concret de l'intelligence artificielle dans l'optimisation de la gestion énergétique. Il a également mis en lumière les défis liés à la qualité des données, à la complexité du choix des modèles et à leur intégration dans un outil fonctionnel.

Parmi les perspectives futures, l'amélioration des performances par l'utilisation d'architectures plus récentes comme les Transformers, l'intégration de données externes (météo, événements spéciaux), ou encore le déploiement de l'application à plus grande échelle sont autant de pistes prometteuses. En somme, ce travail constitue une première étape vers la modernisation des outils de prévision énergétique, à l'appui des technologies intelligentes.

Bibliographie

- [1] Anaconda Inc. Anaconda Distribution. <https://www.anaconda.com>, 2025. Consulté le 17 mai 2025.
- [2] Astah. Astah UML Tool. <https://astah.net>, 2025. Consulté le 16 mai 2025.
- [3] Atlassian. Rôles dans scrum. <https://www.atlassian.com/fr/agile/scrum/roles>, 2025. Consulté le 20 avril 2025.
- [4] Axiocode. Méthode agile vs classique : Quelle méthode utiliser ? <https://www.axiocode.com/methode-agile-vs-classique-quelle-methode-utiliser/>, 2024. Consulté le 17 avril 2025.
- [5] AxioCode. Méthode agile vs classique : Quelle méthode utiliser ? <https://www.axiocode.com/methode-agile-vs-classique-quelle-methode-utiliser/>, 2024. Consulté le 20 avril 2025.
- [6] Grady Booch. *The unified modeling language user guide*. Pearson Education India, 2005.
- [7] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis : Forecasting and Control*. Wiley, 4th ed. edition, 2008.
- [8] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis : forecasting and control*. John Wiley & Sons, 2015.
- [9] Mon Coach Data. Comprendre les réseaux de neurones, 2023. Consulté le 5 mars 2025.
- [10] Matplotlib Developers. Matplotlib documentation, 2023. Consulté le 3 mai 2025.
- [11] NumPy Developers. Numpy documentation, 2023. Consulté le 6 mai 2025.
- [12] Mozilla Foundation. Javascript documentation - mdn, 2024. Consulté le 18 mai 2025.
- [13] Python Software Foundation. Python official website, 2024. Consulté le 19 mai 2025.
- [14] Gwilym Jenkins et Gregory Reinsel George Box. *Time Series Analysis : Forecasting and Control*. Wiley, Hoboken, NJ, 5e édition edition, 2015.
- [15] GitHub Inc. GitHub. <https://github.com>, 2025.
- [16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, 2016. Consulté le 15 mars 2025.
- [17] Alex Graves and Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012.
- [18] James D. Hamilton. *Time Series Analysis*. Princeton University Press, 1994.

- [19] James D. Hamilton. *Time Series Analysis*. Princeton University Press, Princeton, NJ, 1994.
- [20] R. J. Hyndman and G. Athanasopoulos. *Forecasting : Principles and Practice*. OTexts, 3rd ed. edition, 2021. Consulté le 10 avril 2025.
- [21] Kaggle. Kaggle datasets, 2025. Consulté le 21 juin 2025.
- [22] Pieter-Jan Kindermans, Kristof T Schütt, Maximilian Alber, Klaus-Robert Müller, Dumitru Erhan, Been Kim, and Sven Dähne. Learning how to explain neural networks : Patternnet and patternattribution. 2017.
- [23] Scikit learn Developers. Scikit-learn documentation, 2023. Consulté le 5 juin 2025.
- [24] Meta Platforms, Inc. React documentation. <https://react.dev>, 2025. Consulté le 5 juin 2025.
- [25] Microsoft. Visual Studio Code. <https://code.visualstudio.com>, 2025. Consulté le 18 mai 2025.
- [26] Sebastián Ramírez. Fastapi documentation. <https://fastapi.tiangolo.com/>, 2018. Consulté le 1 juin 2025.
- [27] Stuart Russell and Peter Norvig. *Artificial Intelligence : A Modern Approach*. Pearson, Upper Saddle River, NJ, 4 edition, 2021.
- [28] Sonelgaz. Site officiel de sonelgaz. <https://www.sonelgaz.dz/fr/>, 2024. Consulté le 17 avril 2025.
- [29] Tailwind Labs. Tailwind css documentation. <https://tailwindcss.com>, 2025. Consulté le 15 juin 2025.
- [30] Keras Team. Keras documentation, 2023. Consulté le 9 mars 2025.
- [31] Pandas Development Team. Pandas documentation, 2023. Consulté le 6 mai 2025.
- [32] TensorFlow Team. Tensorflow documentation, 2023. Consulté le 7 juin 2025.
- [33] Venngage. Diagramme de contexte : Définition. <https://venngage.com/blog/context-diagram/>, 2024. Consulté le 1 mai 2025.
- [34] Yookkan. La méthode scrum : définition, rôles et étapes. <https://yookkan.com/logiciel-de-gestion-de-projet-gratuit-en-francais/la-methode-scrum/>, 2024. Consulté le 20 avril 2025.

Résumé

Avec l'essor de l'intelligence artificielle, les avancées récentes en apprentissage automatique (ML) et en apprentissage profond (DL) offrent aux entreprises des outils puissants pour améliorer leurs performances. Face aux défis liés à la gestion de l'énergie, comme l'augmentation constante de la demande, ces technologies apparaissent comme une solution innovante.

Ce mémoire s'inscrit dans cette dynamique et propose le développement d'une application web permettant d'analyser et de prédire la consommation électrique au sein de la direction commerciale de Sonelgaz à Béjaïa. L'application repose sur trois approches de prévision des séries temporelles : ARIMA, SARIMA et les réseaux de neurones LSTM. Elle offre des fonctionnalités de visualisation, de prédiction et de génération de rapports.

Pour structurer le développement, la méthodologie agile **Scrum** a été adoptée, assurant un pilotage itératif et collaboratif du projet. L'évaluation de l'application a été réalisée à partir des données réelles de consommation mensuelle de 7 clients, collectées sur une période de 4 ans dans la région de Béjaïa.

Mots-clés : Séries temporelles, Intelligence Artificielle, Deep Learning, Méthode Agile, Prédiction de consommation électrique.

Abstract

With the rise of artificial intelligence, recent advances in machine learning (ML) and deep learning (DL) have provided companies with powerful tools to enhance their performance. Faced with challenges such as the growing demand for energy, these technologies offer innovative solutions.

This thesis presents the development of a web application for analyzing and forecasting electricity consumption within the commercial division of Sonelgaz in Béjaïa. The application is based on three time series forecasting methods : ARIMA, SARIMA, and LSTM neural networks. It includes features for data visualization, prediction, and report generation.

The project was developed using the agile **Scrum** methodology, ensuring iterative and collaborative management. The application was tested on real monthly consumption data from 7 clients in the Béjaïa region, covering a 4-year period.

Keywords : Time Series, Artificial Intelligence, Deep Learning, Agile Method, Electricity Consumption Forecasting.