République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

Université A. Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique

MEMOIRE DE MASTER PROFESSIONNEL

En Informatique

Option

Génie logiciel

Approche d'apprentissage profond pour prédire la consommation d'électricité

Cas : Sonelgaz Béjaïa

Présenté par :

Melle CHAFFI Aicha Manel **Melle BELAID Nora**

Devant le jury composé de :

- **Président :** M. SIDER Abderrahmane MCA Université A/Mira de Béjaïa
- **Encadrant :** M. MIR Foudil MCB Université A/Mira de Béjaïa

Examinateurs :

— M. BEDJOU Khaled MCB Université A/Mira de Béjaïa
— Mme. KHOULALENE Nadjet MCB Université A/Mira de Béjaïa
— Mme. GASMI Badrina MCB Université A/Mira de Béjaïa

Promotion 2024–2025

Remerciements

On voudrait avant tout, et dans un premier temps, remercier **ALLAH**,
Tout-Puissant et Miséricordieux, du plus profond de notre cœur,
pour nous avoir donné le courage tout au long de nos études,
ainsi que la santé, la volonté et la patience nécessaires à l'accomplissement de ce travail.

Nous adressons nos sincères remerciements à **M. Mir Foudil**,
pour la confiance qu'il nous a accordée en acceptant d'encadrer ce travail,
ainsi que pour ses explications claires, ses précieux conseils et sa disponibilité tout au long de ce mémoire, malgré ses nombreuses responsabilités.
Ses retours constructifs ont grandement enrichi notre réflexion et affiné notre méthodologie.
Nous lui sommes infiniment reconnaissantes pour l'attention portée à notre travail.

Nos remerciements vont également à **M. HADJOUT Dalil**, responsable du service Statistiques, pour son encadrement pratique. À ce stade, nous exprimons aussi notre gratitude envers l'ensemble du personnel de **Sonelgaz Distribution – Direction de Béjaïa**.

Nous tenons aussi à remercier **les membres du jury**,
pour l'honneur qu'ils nous ont fait en acceptant d'évaluer notre travail.

Enfin, nous souhaitons adresser nos remerciements les plus sincères à nos familles,
qui nous ont soutenues et encouragées tout au long de notre vie et de notre parcours universitaire.
Merci également à nos chers amis, ainsi qu'à **tous nos enseignants**, depuis la maternelle jusqu'à aujourd'hui.

Dédicaces

Je dédie ce mémoire à la petite fille que j'étais, celle qui rêvait en silence. À ses larmes, à ses doutes, mais aussi à sa force et sa persévérance. Aujourd'hui, son rêve devient réalité.

Je dédie cet événement marquant de ma vie à **mes parents bien-aimés**, qu'aucune dédicace ne saurait pleinement exprimer. Vos prières, votre amour inconditionnel et vos sacrifices ont été ma force tout au long de ce parcours. Merci pour votre soutien sans faille, votre patience, votre confiance en moi, et l'éducation précieuse que vous m'avez prodiguée, au prix d'efforts incommensurables.

À toi, **maman**, source infinie d'amour, de tendresse et de courage. Tes mots doux et ta présence rassurante ont illuminé chaque étape de mon chemin. Tu es, et resteras, mon pilier le plus précieux.

À toi, **papa**, pour ta sagesse, ton silence plein de force, et ta confiance discrète mais constante. Ton soutien a été une boussole dans mes choix, et ton exemple, une source d'inspiration. Ta rigueur m'a transmis la motivation de persévérer, même dans les moments les plus difficiles.

À mes petites sœurs **ASSIA**, **LINA** et **IMENE**, pour votre présence à mes côtés. Que Dieu vous protège et vous accorde santé, bonheur et réussite.

À mes tantes, Tata **FATIHA** et sa petite **JANA**, ma petite cousine, à Tata **DJOHRA** qu'Allah te guérisse complètement et rapidement , à mon oncle **DJAMEL**, à mes proches, et à tous ceux qui me donnent de l'amour et de la vivacité.

À mes amies **MANEL**,**LINA**, **ASMA**, **DIHYA** et à toi, ma binôme **AICHA**, avec qui j'ai partagé les défis, les réussites et tant de souvenirs inoubliables. Merci pour votre aide, vos encouragements, et les moments précieux vécus ensemble. Bonne chance à vous, je vous souhaite tout le succès du monde.

NORA

Dédicaces

Je dédie ce mémoire, ainsi que tout le travail accompli et les épreuves traversées à mes parents, que j'ai la chance d'avoir encore à mes côtés aujourd'hui.

À ceux dont un simple regard ou un sourire me donne toute la force nécessaire pour franchir les obstacles qui se dressent sur mon chemin. Je leur adresse une profonde gratitude pour m'avoir transmis des principes, des valeurs, de la force et de la patience. Un simple remerciement ou une dédicace ne suffira jamais à exprimer ce que je ressens, mais en cette fin de chapitre de ma vie, je tiens à vous faire une promesse : celle de ne jamais vous décevoir, d'être une source de bonheur et de fierté, aujourd'hui et pour toujours.

Je tiens également à dédier ce mémoire à ma grand-mère, avec qui j'ai partagé 17 ans de ma vie, et qui, j'en suis sûre, serait fière de sa petite-fille.

AICHA

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

A	ACF	AutoCorrelation Function
	ADF	Augmented Dickey-Fuller test
	AIC	Akaike Information Criterion
	ANN	Artificial Neural Network
	ARIMA	AutoRegressive Integrated Moving Average
	ARMA	AutoRegressive Moving Average
B	BIC	Bayesian Information Criterion
C	CNN	Convolutional Neural Network
D	DP	Deep Learning
E	EMA	Exponential Moving Average
G	GAN	Generative Adversarial Network
	GRU	Gated Recurrent Unit
H	HQIC	Hannan–Quinn Information Criterion
L	LSTM	Long Short Term Memory
M	MA	Moving Average
	MAPE	Mean Absolute Percentage Error
	ML	Machine Learning
	MLP	Multi-Layer Perceptron
P	PACF	Partial AutoCorrelation Function
R	RN	Réseau de Neurones
	RNN	Recurrent Neural Network
S	SARIMA	Seasonal AutoRegressive Integrated Moving Average
	SARIMAX	Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors
	SMA	Simple Moving Average
	STL	Seasonal and Trend decomposition using Loess
W	WMA	Weighted Moving Average

Introduction générale

L'anticipation de la consommation et de la production d'électricité représente aujourd'hui un enjeu majeur, à la fois sur les plans économique, stratégique et environnemental. En raison de l'impossibilité de stocker l'électricité à grande échelle, il est impératif d'ajuster en temps réel la production pour répondre à la demande, en assurant un équilibre optimal entre offre et consommation. Cette régulation permet de minimiser les pertes d'énergie, d'optimiser les coûts et de garantir un approvisionnement fiable aux clients.

Dans ce contexte, les outils de prédiction basés sur l'analyse de séries temporelles offrent une solution puissante pour anticiper les besoins énergétiques. Grâce aux avancées technologiques en intelligence artificielle, et plus particulièrement dans les domaines du Machine Learning et du Deep Learning, il est désormais possible d'atteindre des niveaux de précision élevés dans la prévision de la consommation électrique.

Le présent projet s'inscrit dans cette logique et vise à concevoir une application web intelligente destinée à la Direction de la Distribution de Sonelgaz à Béjaïa. L'objectif est de permettre la prévision automatisée de la consommation des clients ainsi que de la production de l'entreprise, à travers une interface moderne et interactive.

Notre solution repose sur l'intégration et la comparaison de trois approches complémentaires :

SARIMA (Seasonal AutoRegressive Integrated Moving Average), modèle statistique adapté aux séries temporelles saisonnières.

SARIMAX, une version enrichie de SARIMA prenant en compte des variables exogènes.

GRU (Gated Recurrent Unit), un modèle de Deep Learning spécialisé dans la modélisation des dépendances temporelles longues.

En combinant ces approches au sein d'une seule plateforme web, nous visons à automatiser le processus de prévision, tout en laissant à l'utilisateur la possibilité de comparer les performances et les erreurs de chaque méthode. Par ailleurs, ce projet répond également à des problématiques concrètes : déséquilibres entre production et demande, détection de consommations anormales ou frauduleuses, et absence d'outils d'analyse prédictive dans les pratiques actuelles de l'entreprise.

Pour mener à bien ce projet, nous avons adopté une démarche agile, en suivant la méthode Scrum, afin d'assurer une progression itérative, flexible et centrée sur les besoins réels des utilisateurs finaux.

Organisation du mémoire

Notre travail est divisé en quatre chapitres, organisés de la manière suivante :

Chapitre 1 : il traite différentes méthodes de modélisation des séries temporelles, notamment les approches classiques de prévision comme ARIMA, SARIMA et SARIMAX.

Chapitre 2 : nous abordons les concepts fondamentaux du Deep Learning, avec un focus particulier sur les réseaux de neurones récurrents, et plus précisément sur le modèle GRU (Gated Recurrent Unit), utilisé pour la prédiction de séries temporelles.

Chapitre 3 : le troisième chapitre est consacré à la conception de l'application. Il comprend l'analyse des besoins fonctionnels et non fonctionnels, ainsi que la modélisation à l'aide de diagrammes UML.

Chapitre 4 : enfin, le quatrième chapitre présente la réalisation pratique de l'application web, en détaillant les choix technologiques, les interfaces développées, les résultats de prédiction obtenus à l'aide des modèles SARIMA, SARIMAX et GRU, ainsi qu'une comparaison de leurs performances respectives.

Chapitre 1

Méthodes classiques pour la prédiction

1.1 Introduction

Les séries temporelles sont largement utilisées dans divers domaines pour la prévision et l'analyse des tendances. En effet, ces méthodes permettent d'identifier des modèles récurrents, comme les tendances saisonnières ou les cycles économiques, et de prévoir les valeurs futures sur la base de ces observations passées. Dans le contexte de la prédiction des consommations d'énergie, l'analyse des données collectées est essentielle, pour cela diverses méthodes basées sur les séries temporelles sont utilisées, elles permettent d'identifier des tendances saisonnières et des pics de demande pour pouvoir anticiper les besoins futurs en énergie pour but d'optimiser la planification des ressources ou de détecter des anomalies dans la consommation.

1.2 Série temporelle

1.2.1 Définition des séries temporelles

Une série temporelle, aussi appelée série chronologique, est un ensemble d'observations d'une variable statistique économique faites à intervalles réguliers (années, trimestres, mois, jours, etc.) [5]. Elle est généralement notée par l'équation suivante :

$$X_t = x_t, \quad t = 1, 2, \ldots, T, \quad T \in \mathbb{N} \tag{1.1}$$

où $t_1, t_2, t_3, \ldots, t_n$ représentent les instants auxquels les observations ont été effectuées.
Comme indiqué dans l'équation 1.1, la série temporelle est définie à des instants discrets.

1.2.1.1 Analyse d'une Série Temporelle : Cas Pratique

L'exemple de données que nous utiliserons est une série temporelle de consommation d'énergie horaire qui provient du site officiel de PJM[34], une Organisation Régionale de Transport (RTO) aux États-Unis. Il fait partie du réseau d'interconnexion de l'Est exploitant un système de transport d'électricité.

Les données sont exprimées en mégawatts (MW) et correspondent à la consommation horaire sur une période de 4 jours (du 25/12/2004 au 28/12/2004).

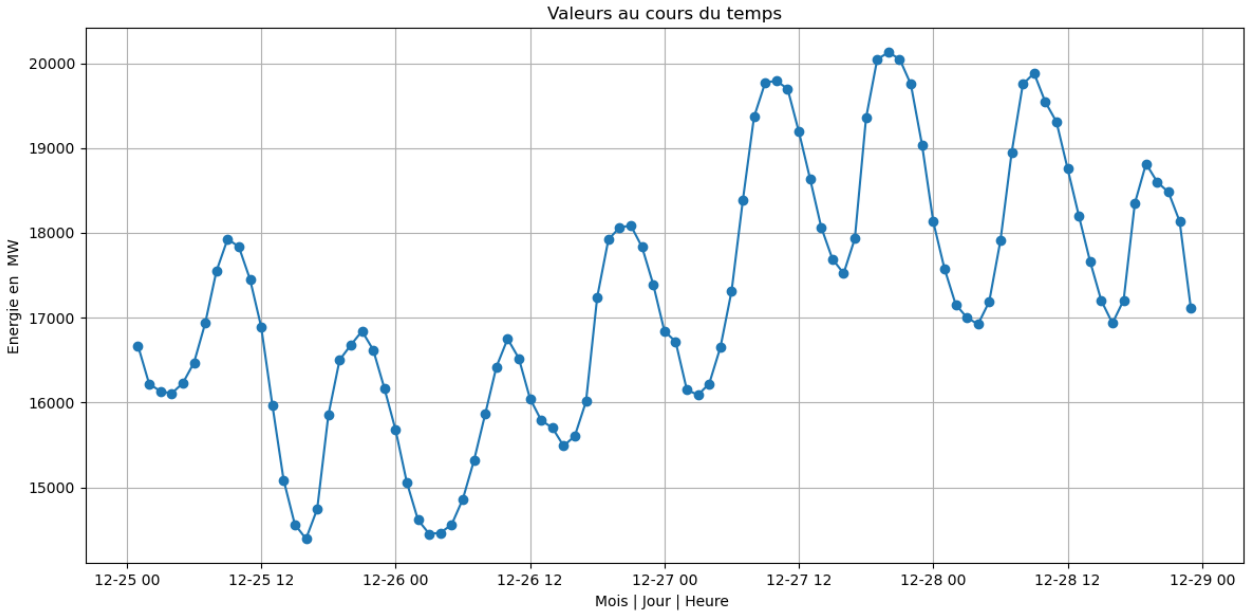


FIGURE 1.1 – Représentation graphique de la série temporelle de l'exemple étudié

1.2.2 Composantes d'une série chronologique

La séparation de la série est effectuée en trois composantes principales :

— **Tendance (le trend) :** notée (T_t) représente l'évolution à long terme d'une série temporelle. Elle traduit le comportement moyen de la série [5].

— **La composante saisonnière (ou saisonnalité) :** notée S_t est une composante cyclique relativement régulière de période intra-annuelle, influencée par des phénomènes de mode, de coutume ou de climat [5].

— **La composante résiduelle (bruit ou résidu) :** notée R_t, regroupe les variations imprévisibles non expliquées par les autres composantes, comme les catastrophes ou grèves. Elle tend à être aléatoire autour de sa moyenne [5].

1.3 Modélisations des séries temporelles

1.3.1 Méthodes classiques de prévision

1.3.1.1 Le modèle additif

Le modèle additif est le modèle classique de décomposition. Il suppose que chaque observation d'une série chronologique est la somme de ses composantes : tendance, saisonnalité et résidu [30], comme le montre l'équation 1.2.

L'équation du modèle additif de la série temporelle est donnée par :

$$X_t = T_t + S_t + R_t \quad \text{pour } t = 1, \ldots, T. \tag{1.2}$$

Symbole	Description
X_t	Variable observée
T_t	Tendance
S_t	Saisonnalité
R_t	Bruit (résidu)

TABLE 1.1 – Définition des composantes du modèle additif

Dans un modèle additif, la saisonnalité est constante et n'est pas influencée par la tendance. Cela signifie que l'amplitude des fluctuations saisonnières reste la même, quelle que soit l'évolution de la tendance. Graphiquement, cela nous permet de tracer des droites parallèles représentant les valeurs maximales et minimales de la série.

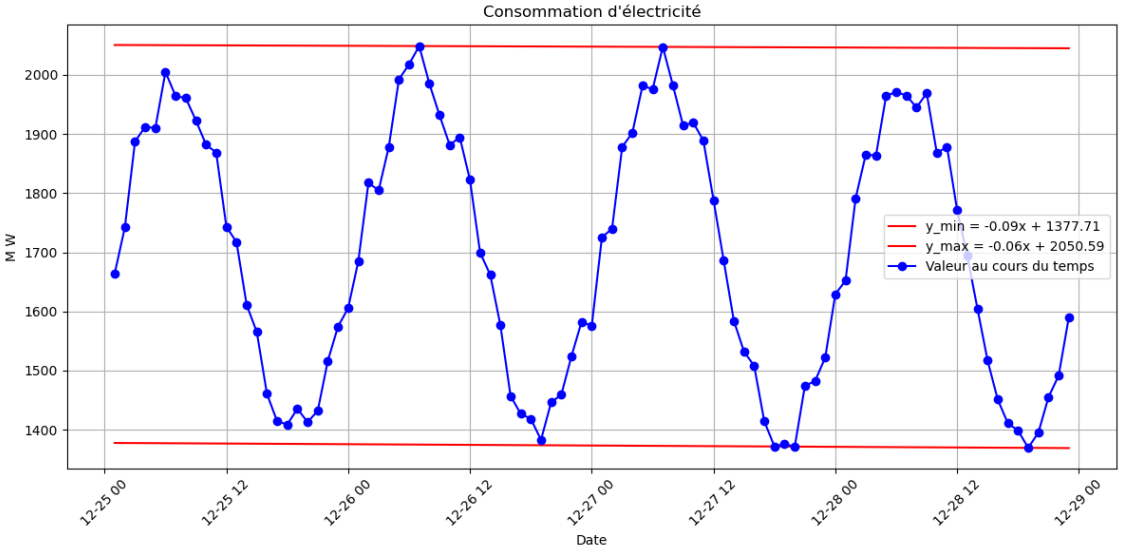


FIGURE 1.2 – Droites des valeurs maximales et minimales d'un exemple d'une série temporelle qui adopte un modèle additif

— **La fonction de prédiction d'un modèle additif**
Nous avons analysé comment modéliser une série temporelle en deux composantes principales : la **tendance** et la **saisonnalité**. Voici la traduction mathématique de ces deux composantes.

— **Tendance :** Nous commençons par calculer la tendance de la série temporelle en utilisant une régression linéaire. La régression linéaire modélise la relation entre les observations de la série temporelle X_t et le temps t.

L'équation de la tendance est la suivante :

$$\text{trend}_t = A \cdot t + B \tag{1.3}$$

où :

Les coefficients A et B sont calculés à partir des sommes suivantes :

$$A = \frac{\sum_{i=1}^{n}(X_t \cdot t) - n \cdot \overline{X} \cdot \overline{t}}{\sum_{i=1}^{n} t^2 - n \cdot \overline{t}^2} \tag{1.4}$$

$$B = \overline{X} - A \cdot \overline{t} \tag{1.5}$$

Symbole	Description
trend_t	Valeur de la tendance à l'instant t
A	Coefficient de pente, représentant le taux de variation de la série temporelle
B	Ordonnée à l'origine, c'est-à-dire l'intersection de la tendance avec l'axe des ordonnées (valeur initiale)
t	Instant dans le temps (index temporel)
$\overline{X}$	La moyenne des valeurs observées X_t
$\overline{t}$	La moyenne des indices temporels t
n	Le nombre d'observations

TABLE 1.2 – Paramètres du modèle de tendance linéaire

— **Saisonnalité :** Une fois la tendance trend_t calculée, la saisonnalité brute est déterminée par la différence entre la valeur observée et la tendance :

$$\text{S}_t = X_t - \text{trend}_t \tag{1.6}$$

La saisonnalité est ensuite ajustée en calculant les coefficients saisonniers moyens pour chaque heure (ou chaque période selon la série temporelle étudiée), suivis du calcul de la moyenne globale de tous ces coefficients saisonniers obtenus :

$$\text{S}_h\_additif = \frac{1}{|H_h|} \sum_{t \in H_h} \text{S}_t \tag{1.7}$$

où H_h représente l'ensemble des instants t correspondant à l'heure h, et $|H_h|$ est le nombre d'éléments dans cet ensemble.
puis :

$$s_{\text{additif}} = \frac{1}{24} \sum_{h=0}^{23} \text{S}_h\_additif \tag{1.8}$$

La somme est effectuée sur 24 heures, car l'axe des abscisses de la série étudiée est en heures. Ainsi, les cycles de 24 heures se répètent, ce qui signifie qu'il y a 24 valeurs de $\text{S}_h\_additif$.

Enfin, les coefficients saisonniers ajustés sont :

$$\text{coefficients\_S\_additif}_h = \text{S}_h\_additif - s_{\text{additif}} \tag{1.9}$$

— **Fonction de prédiction**
Ainsi, le modèle final de la série temporelle est composé de la somme de la tendance et de la saisonnalité ajustée :

$$X_t = (A \cdot t + B) * \text{coefficients\_S\_additif}_h \tag{1.10}$$

1.3.1.2 Le modèle multiplicatif

Le modèle multiplicatif suppose que chaque observation d'une série temporelle est obtenue par le produit de ses composantes : la tendance , la saisonnalité et le résidu. Ce modèle est utilisé lorsque l'amplitude des variations saisonnières dépend de la tendance [5], comme indiqué dans l'équation 1.11.

Symbole	Description
Y_t	Variable observée
T_t	Tendance
S_t	Saisonnalité
R_t	Bruit (résidu)

TABLE 1.3 – Composantes du modèle multiplicatif

Mathématiquement, cela s'exprime par :

$$Y_t = T_t \times S_t \times R_t \quad \text{pour } t = 1, \ldots, T. \tag{1.11}$$

Dans un modèle multiplicatif, les fluctuations saisonnières varient proportionnellement à la tendance. Cela signifie que si la tendance évolue (par exemple, augmente), l'amplitude des fluctuations saisonnières augmente également, et inversement si la tendance diminue. En raison de cette relation proportionnelle, il devient impossible de tracer des droites parallèles, car les fluctuations saisonnières grandissent ou diminuent au fur et à mesure que la tendance évolue comme le montre la série étudiée.

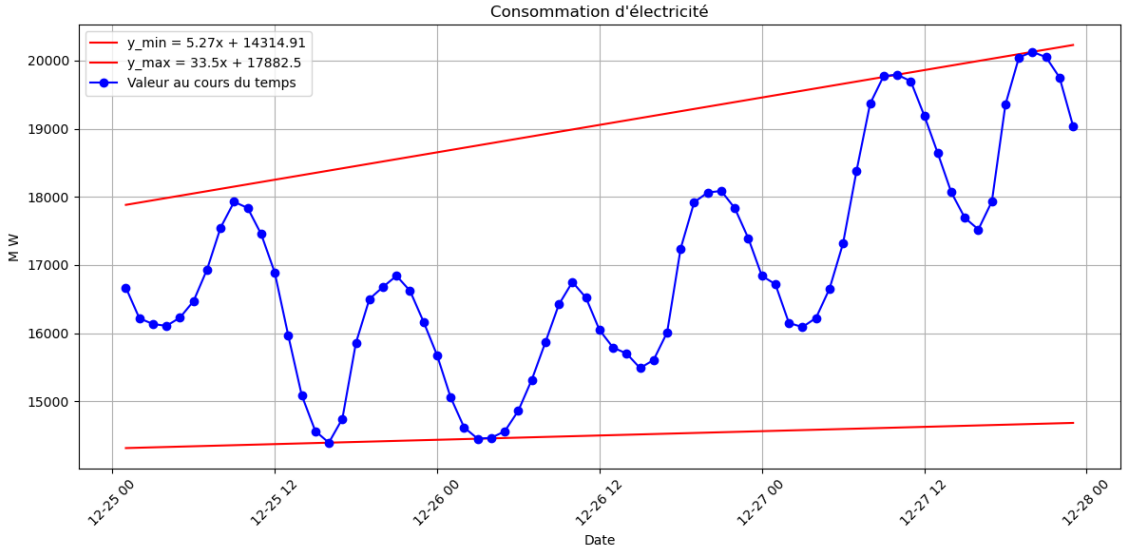


FIGURE 1.3 – Droites des valeurs maximales et minimales

— **La fonction de prédiction d'un modèle multiplicatif**

— **Tendance :** Les coefficients de la tendance sont calculés de la même manière que celle du modèle additif .

$$\text{trend}_t = A \cdot t + B \tag{1.12}$$

— **Saisonnalité :** Une fois trend_t calculée, la saisonnalité brute est obtenue par :

$$\text{S}_t = \frac{Y_t}{\text{trend}_t} \tag{1.13}$$

La saisonnalité est ensuite ajustée, les deux équations suivantes sont calculés de la même manière que celles d'un modèle additif :

$$\text{S}_h\_multiplicatif = \frac{1}{|H_h|} \sum_{t \in H_h} \text{S}_t \tag{1.14}$$

puis :

$$s_{\text{multiplicatif}} = \frac{1}{24} \sum_{h=0}^{23} \text{S}_h\_multiplicatif \tag{1.15}$$

Enfin, les coefficients saisonniers ajustés sont :

$$\text{coefficients\_S\_multiplicatif}_h = \frac{\text{S}_h\_additif}{s_{\text{multiplicatif}}} \tag{1.16}$$

— **Fonction de prédiction**
Le modèle final de la série temporelle est le produit de la tendance et de la saisonnalité multiplicative ajustée :

$$Y_t = (A \cdot t + B) * \text{coefficients\_S\_multiplicatif}_h \tag{1.17}$$

Tel que les résultats obtenus sont :

$$A = 34.25839865621501, \quad B = 15585.01791713326$$

1.3.1.3 Moyenne mobile

La **moyenne mobile**, en anglais **Moving Average (MA)** est une technique de lissage utilisée pour atténuer les fluctuations à court terme d'une série temporelle. Elle consiste à calculer la moyenne des valeurs sur une fenêtre glissante de taille fixe, facilitant ainsi l'identification des tendances [5].

Il existe quatre types de MA :

1. **La Moyenne Mobile Arithmétique (SMA) :** est une méthode de lissage qui consiste à calculer la moyenne des k dernières valeurs observées d'une série temporelle. Elle permet d'atténuer les fluctuations à court terme et de mieux identifier les tendances sous-jacentes [5], comme illustré par l'équation 1.18.

$$SMA_n = \frac{1}{n} \sum_{i=t-n+1}^{t} P(i) \tag{1.18}$$

où :
— $P(i)$ représente la valeur à l'instant i.
— i est l'instant ou l'index temporel courant (point à prédire).
— n est la période de lissage utilisée ou la fenêtre.

2. **La moyenne mobile centrée (SMAcentrée) :** st une variante de la MA simple où chaque valeur lissée est calculée en prenant une fenêtre symétrique autour du point considéré. Cela permet de réduire le décalage temporel observé avec une MA classique.[5].
Cette méthode présente des limites, notamment pour les premières et dernières valeurs de la série où elle ne peut pas être appliquée.

Elle est aussi calculée différemment selon la période utilisée :

— **Cas 1 :** Fenêtre de taille impaire (n impair)

$$SMA_{\text{centrée}}(i) = \frac{1}{n} \sum_{j=i-k}^{i+k} P(j) \tag{1.19}$$

$$k = \frac{n-1}{2} \tag{1.20}$$

Lorsque la période choisie est un nombre impaire, les valeurs prisent avant et après un point de données central sont symétriques avec le point central au milieu.

— **Cas 2 :** Fenêtre de taille paire (n paire) :

Lorsqu'on utilise SMA avec un n pair, on ajuste la fenêtre en la rendant impaire $(n-1)$ et on compense en ajoutant la moitié des valeurs adjacentes avant de diviser par n.

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

SMA est particulièrement utile pour les analyses de données à long terme ou pour détecter des tendances dans des séries temporelles régulières, elle est plus souvent utilisée sur les séries symétriques ou stationnaires.

— **Série symétrique :** une série symétrique est une série où les valeurs sont disposées de manière à ce qu'elles soient les mêmes ou se correspondent des deux côtés d'un point central.

3. **La Moyenne Mobile Pondérée (WMA) :** la WMA attribue des poids décroissants aux observations d'une série temporelle, donnant plus d'importance aux valeurs récentes pour mieux capter les tendances [27], comme illustré par l'équation 1.22.

$$WMA(n) = \frac{\sum_{k=0}^{n-1}(n-k)P(i-k)}{\sum_{k=1}^{n} k} \tag{1.22}$$

La WMA est plus réactive aux changements que la SMA elle réagis donc plus rapidement aux variations ce qui est utile pour ajuster les prévisions à court terme.

4. **La Moyenne Mobile Exponentielle (EMA) :** L'EMA est une méthode de lissage des séries temporelles qui attribue un poids exponentiellement décroissant aux observations passées, mettant davantage l'accent sur les valeurs récentes. Cette approche permet de suivre rapidement les tendances tout en réduisant les fluctuations aléatoires [27], comme le montre l'équation 1.23.

$$EMA(i) = \alpha P(i) + (1-\alpha) EMA(i-1) \tag{1.23}$$

$$\alpha = \frac{2}{n+1} \tag{1.24}$$

La EMA est plus sensible aux changements récents que la WMA . Elle est particulièrement adaptée aux données très volatiles, permettant une réaction rapide aux variations de tendance .
Ce graphe met en application les trois types cités ci-dessus sur le dataset :

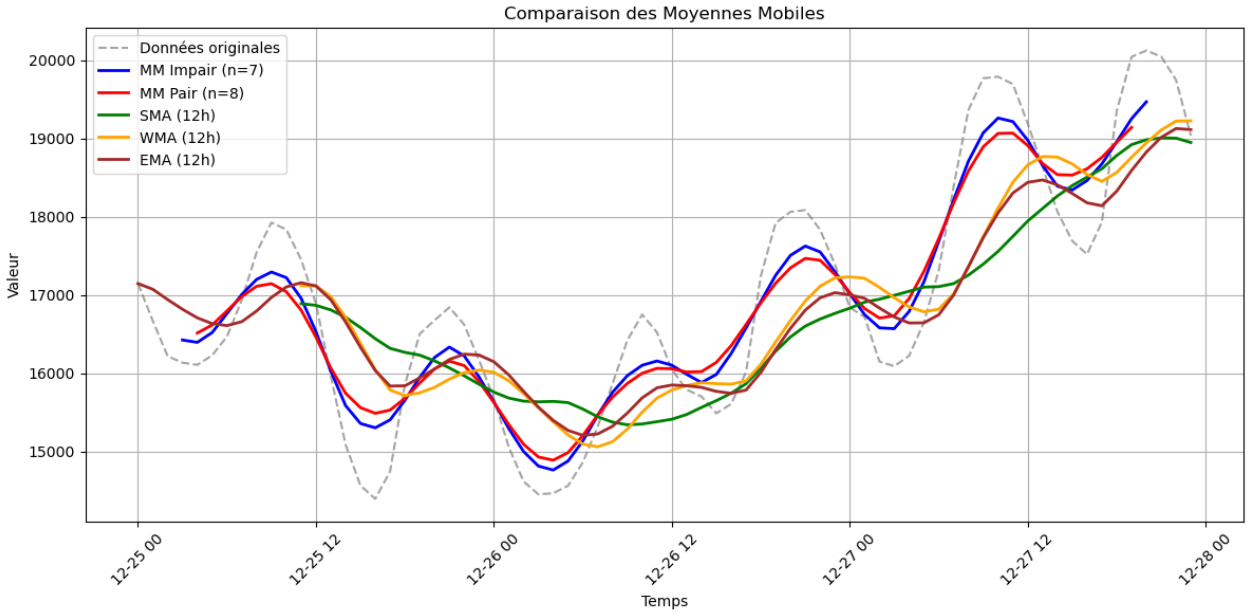


FIGURE 1.4 – Application des méthodes MA sur l'exemple étudié

Le graphique compare les résultats des différentes méthodes de MA appliquées à la série temporelle étudiée. Les courbes lissées (SMA, WMA, EMA) permettent de réduire le bruit des données tout en révélant les tendances. Nous observons que l'EMA réagit plus rapidement aux variations récentes, tandis que la SMA et la WMA offrent un lissage plus stable mais moins réactif.

1.3.2 Modèle stochastique

1.3.2.1 Utilisation de l'approche Box-Jenkins pour l'analyse et la prévision des séries temporelles

— **Définition de l'approche Box-Jenkins**
L'approche Box-Jenkins est une méthodologie systématique pour l'analyse et la modélisation des séries temporelles, développée par **George Box** et **Gwilym Jenkins** .

— **Etapes de l'approche Box-Jenkins**
  — **Identification du modèle**
  Analyse des données : analyser les données de la série temporelle afin d'identifier les tendances, les variations saisonnières et les éventuelles anomalies.
  Différenciation : si la série temporelle présente une tendance ou une saisonnalité, il est conseillé d'effectuer une différenciation afin de la rendre stationnaire.
  Identification des ordres : analyser les graphiques de la fonction d'autocorrélation (ACF) et de la fonction d'autocorrélation partielle (PACF) afin de déterminer les ordres p, d et q du modèle.

— **Estimation des paramètres**
Appliquer des techniques d'estimation pour déterminer les paramètres des modèles statistiques.
— **Vérification du modèle**
Diagnostic du modèle : s'assurer que les résidus du modèle se comportent comme un bruit blanc, c'est-à-dire qu'ils ne présentent ni structure ni corrélation significative.
Réajustement : si le modèle ne valide pas les critères de bruit blanc (p-value significative), modifier les ordres du modèle et recommencer l'analyse.
Validation : valider l'efficacité du modèle en testant ses prédictions sur un jeu de données de validation et en analysant des indicateurs de performance .

1.3.2.2 Stationnarité et transformations des séries temporelles

— **Définition**
Un processus est dit **stationnaire** si ses propriétés statistiques, telles que la moyenne et la covariance, sont invariantes dans le temps. Cela signifie que les caractéristiques du processus ne changent pas au cours du temps [29].
— **Test ADF (Augmented Dickey-Fuller)**
Le test de ADF est un test statistique permettant de vérifier la présence d'une racine unitaire dans une série temporelle. Il a été proposé par **David A. Dickey** et permet de tester la stationnarité en examinant la présence d'une racine unitaire dans un modèle autorégressif [13].
— **Valeur critique**
Pour déterminer si une série est stationnaire ou non, la valeur critique est un seuil statistique à qui le résultat du test ADF est comparé. Elle indique à partir de quel niveau l'hypothèse nulle de non-stationnarité peut être rejetée. La valeur critique est définie en fonction de la taille de l'échantillon et du niveau de signification, généralement fixé à 5%, ce qui signifie que l'hypothèse nulle peut être rejetée avec une confiance de 95% [13].
— **Interprétation de la statistique ADF (p-value) par rapport à la valeur critique :**

1. Lorsque la **p-value** est inférieure à la **valeur critique**, l'hypothèse nulle (présence d'une racine unitaire) est rejetée, indiquant que la série est **stationnaire**.
2. À l'inverse, si la **p-value** est supérieure à la **valeur critique**, l'hypothèse nulle ne peut pas être rejetée, ce qui signifie que la série est **non stationnaire**.

Le test ADF appliqué sur la série étudiée donne une statistique de -1.45 avec une p-value de 0.55, supérieure au seuil de 5%. On ne peut donc pas rejeter l'hypothèse nulle : la série est **non stationnaire** et nécessite une transformation.
— **La transformation des séries temporelles**
Plusieurs techniques sont appliquées aux données afin de stabiliser la variance et améliorer la stationnarité. Parmi les transformations courantes, on trouve la transformation logarithmique, la différenciation.

1. **Passage au logarithme :** la transformation logarithmique est une technique couramment utilisée en analyse des séries temporelles pour stabiliser la variance et atténuer l'effet des valeurs extrêmes. Elle est particulièrement utile lorsque la variance des observations augmente avec le niveau de la série, notamment dans les cas où la série suit une tendance exponentielle.

2. **La différentiation :** la différenciation est une technique essentielle en analyse des séries temporelles pour éliminer les tendances et rendre les séries stationnaires. Elle est particulièrement importante pour les modèles statistique de prévision.

   On distingue principalement deux types de différenciation :

   — **La différenciation ordinaire :** utilisée pour éliminer les tendances globales en calculant la différence entre des observations consécutives, comme illustré par l'équation 1.25.

$$Y_t = y_t - y_{t-1} \tag{1.25}$$

   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 :** appliquée pour supprimer les effets saisonniers en différenciant les observations séparées par une période saisonnière spécifique, comme indiqué dans l'équation 1.26.

$$Y_t = y_t - y_{t-s} \tag{1.26}$$

   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.

1.3.2.3 Processus de prédiction stationnaire

— **Le processus ARMA** (AutoRegressive Moving Average) : est un modèle utilisé en séries temporelles pour capturer les dépendances temporelles dans les données [6], comme exprimé dans l'équation 1.27.

$$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \tag{1.27}$$

où :

— ϕ_i et θ_j sont des coefficients,

— ε_t est un bruit blanc.

Il combine deux composantes :

— **AR(p)** (AutoRegressive) : dépendance linéaire des valeurs passées de la série, comme illustré par l'équation 1.28.

$$\mathrm{AR}(t) = \sum_{i=1}^{p} \phi_i x_{t-i} \tag{1.28}$$

où :

— x_t est la valeur observée à l'instant t.

— p est l'ordre du modèle AR.

— φ_i sont les coefficients du modèle AR qui sont définit à partir de la méthode des moindre carré.

— Méthode des moindres carrés : C'est une technique mathématique développée par **Carl Friedrich Gauss**, utilisée pour estimer les paramètres d'un modèle en minimisant la somme des carrés des écarts entre les valeurs observées et celles prédites par le modèle. Les coefficients sont alors calculés selon l'équation 1.29 :

$$\phi_k = \frac{\sum_{t=i}^{T} \left(Y_{t-k} Y_t - \sum_{j=1}^{k-1} \phi_j Y_{t-k} Y_{t-j} \right)}{\sum_{t=i}^{T} Y_{t-k}^2} \tag{1.29}$$

Symbole	Description
$i = k + 1$	Valeur de départ pour la somme temporelle dans l'équation
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 avec un décalage de k

TABLE 1.4 – Paramètres utilisés dans l'équation des moindres carrés

MA(q) (Moving Average) : dépendance linéaire des erreurs passées. Elle est donnée par l'équation 1.30 :

$$MA(t) = \sum_{i=0}^{q} \theta_i \varepsilon_{t-i}, \quad \text{avec } \theta_0 = 1. \tag{1.30}$$

où :

— les θ_i sont les coefficients qui sont définis à partir de la méthode des moindre carré de la même manière que AR.

— ε_t est un bruit blanc (l'erreur).

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

1. ACF : est un moyen de mesurer la relation linéaire entre une observation à l'instant t et les observations aux instants précédents. Si nous supposons un modèle AR(k), nous pouvons alors souhaiter mesurer uniquement l'association entre x_t et x_{t-k} et filtrer l'influence linéaire des variables aléatoires qui se situent entre les deux [45]. Il est utilisé pour déterminer le paramètre q d'un modèle ARMA en désignant le pic à partir duquel l'ACF ne dépasse plus un certain seuil z 1.31 .

$$z = \left[-\frac{2}{\sqrt{n}}, \frac{2}{\sqrt{n}} \right] \tag{1.31}$$

où :

— n représente le nombre d'observations.

2. PACF : est un outil statistique utilisé pour mesurer la corrélation entre une observation et ses valeurs passées, en tenant compte de l'influence des valeurs intermédiaires. En d'autres termes, elle mesure la relation directe entre deux points d'une série chronologique, en éliminant l'impact des points situés entre eux. Il est utilisé pour déterminer le paramètre p d'un modèle ARMA en désignant le pic à partir duquel le PACF ne dépasse plus le seuil z 1.31 [33].

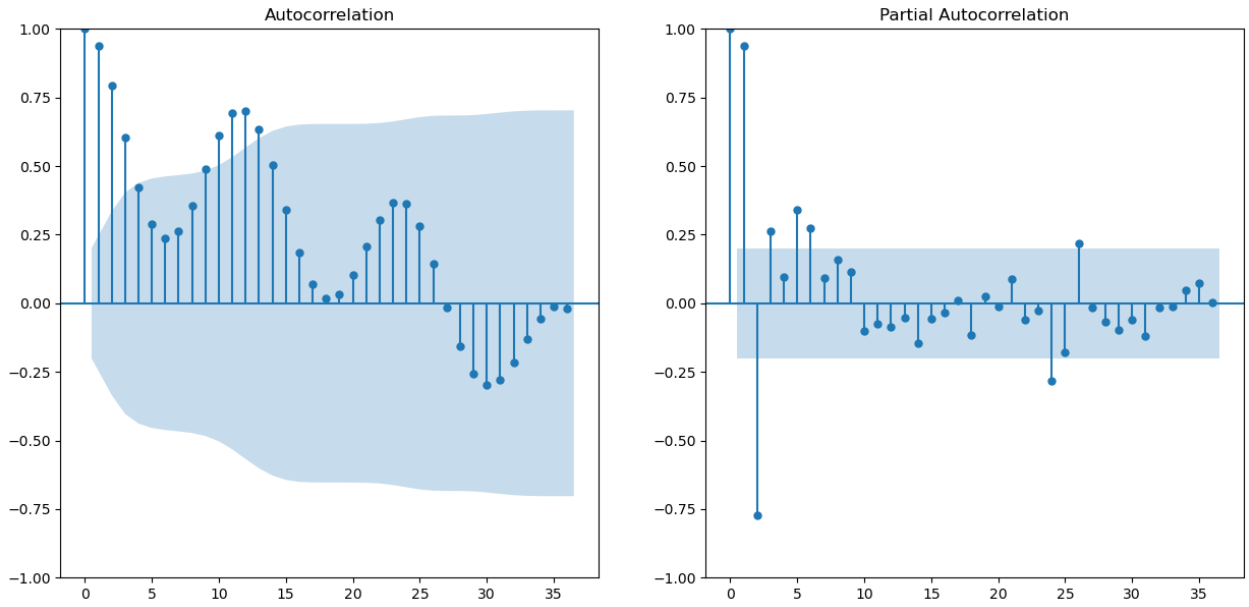


FIGURE 1.5 – ACF et PACF du la série temporelle étudiée

3. Interprétation des résultats graphiques :
   - **Pic positif :** une augmentation de la valeur d'une observation tend à être suivie par une augmentation de la valeur de l'observation à un décalage donné. La corrélation est immédiate pour le PACF (sans influence des valeurs intermédiaires) ou médiatisée pour le ACF.
   - **Pic négatif :** la valeur de deux observations est inversement proportionnelle : quand l'une croît, l'autre décroît (avec ou sans influence des termes intermédiaires selon le graphe étudié)
   - **Pic de faible amplitude :** une corrélation nulle à faible entre deux observation.

— **Les critères d'information**

Les critères d'information servent à comparer et sélectionner le modèle statistique le plus approprié pour un ensemble de données en désignant les paramètres les plus optimaux. Minimiser les critères revient à sélectionner le meilleur modèle.

1. Le AIC (Akaike Information Criterion) : l'AIC est une mesure de la qualité d'un modèle statistique, proposée par **Hirotugu Akaike** en 1973. Il permet de comparer plusieurs modèles en tenant compte à la fois de l'ajustement aux données et de la complexité du modèle.
   L'AIC est défini par l'équation 1.32 :

$$AIC = 2k - 2\ln(L) \tag{1.32}$$

   où k est le nombre de paramètres à estimer du modèle et L est le maximum de la fonction de vraisemblance du modèle. L'AIC vise à estimer la qualité relative des modèles statistiques pour un ensemble de données , en pénalisant la complexité du modèle[8].
2. Le BIC (Bayesian Information Criterion) : également connu sous le nom de critère de Schwarz, le BIC est une mesure utilisée pour la sélection de modèles en statistique. Il pénalise davantage la complexité du modèle que l'AIC. Le BIC est calculé selon l'équation 1.33 :

$$BIC = 2\ln(L) + k\ln(n) \tag{1.33}$$

où n est la taille de l'échantillon.
Le BIC introduit une pénalité plus forte pour les modèles avec un grand nombre de paramètres.

3. Le HQIC (Hannan-Quinn Information Criterion) : le HQIC a été introduit par **Hannan et Quinn** dans leur article de 1979. Il s'agit d'un critère d'information utilisé pour la sélection de modèles statistiques, moins strict que le BIC mais plus conservateur que l'AIC. Il est défini par l'équation 1.34 :

$$HQIC = -2\ln(L) + 2k\ln(\ln(n)) \tag{1.34}$$

Ce critère utilise une pénalité intermédiaire entre l'AIC et le BIC, en tenant compte du nombre de paramètres et de la taille de l'échantillon de manière logarithmique[19].

— **Le MAPE (Mean Absolute Percentage Error)**
Le MAPE, ou erreur absolue moyenne en pourcentage, est une mesure statistique utilisée pour évaluer la précision des méthodes de prévision. Il exprime l'exactitude des prédictions en pourcentage [44]. La formule du MAPE est donnée par l'équation 1.35 :

$$\text{MAPE} = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \tag{1.35}$$

où :
— n est le nombre d'observations.
— A_t représente la valeur réelle à l'instant t.
— F_t est la valeur prévisionnelle à l'instant t .

Plus le MAPE est proche de 0, plus les prévisions du modèle sont précises.
Un MAPE élevé (supérieur à 20-25%) signale une erreur importante et un manque de fiabilité.

1.3.2.4 Processus de prédiction non stationnaires

Pour l'analyse et la modélisation des séries temporelles non stationnaires, il existe trois processus : **ARIMA** (AutoRegressive Integrated Moving Average), **SARIMA**(easonal AutoRegressive Integrated Moving Average), **SARIMAX**(Seasonal AutoRe-gressive Integrated Moving Average with eXogenous variables). Afin de choisir le processus qui convient mieux au dataset étudié, la décomposition STL est utilisée de manière à identifier une possible composante saisonnière significative.

La décomposition STL (Seasonal and Trend decomposition using Loess) : la décomposition STL est une méthode de décomposition des séries temporelles qui sépare une série en trois composantes :

1. Tendance : elle est obtenue par lissage des données d'origine.
2. Saisonnalité : elle est obtenue en soustrayant d'abord la tendance des données d'origine ensuite en appliquant un lissage.
3. Résidu : il est obtenu en soustrayant la composante de tendance et la composante saisonnière des données d'origine.

Une saisonnalité est considérée comme significative si la dispersion de ses variations (l'écart type) est plus grande que celle des fluctuations résiduelles. La formule de l'écart type est donnée par l'équation 1.36 :

$$\sigma = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(S_t - \bar{S}\right)^2} \tag{1.36}$$

où :

— S_t est la valeur de la série temporelle à l'instant t.
— $\bar{S}$ est la moyenne de la série.
— N est le nombre total d'observations.

Le Choix du modèle approprié :

1. Saisonnalité significative : Les modèles SARIMA et SARIMAX sont les plus appropriés. Le choix entre ces deux modèles dépend de la présence ou de l'absence d'influences extérieures.
2. Saisonnalité non significative : Le modèle ARIMA est plus adéquat.

La décomposition STL appliquée au jeu de données étudié montre que l'écart-type de la composante saisonnière qui est égale à 166 est largement supérieur à celui des résidus qui est égale à 50, ce qui confirme une structure saisonnière marquée. Par conséquent, les modèles SARIMA et SARIMAX sont les plus appropriés pour analyser ces données :

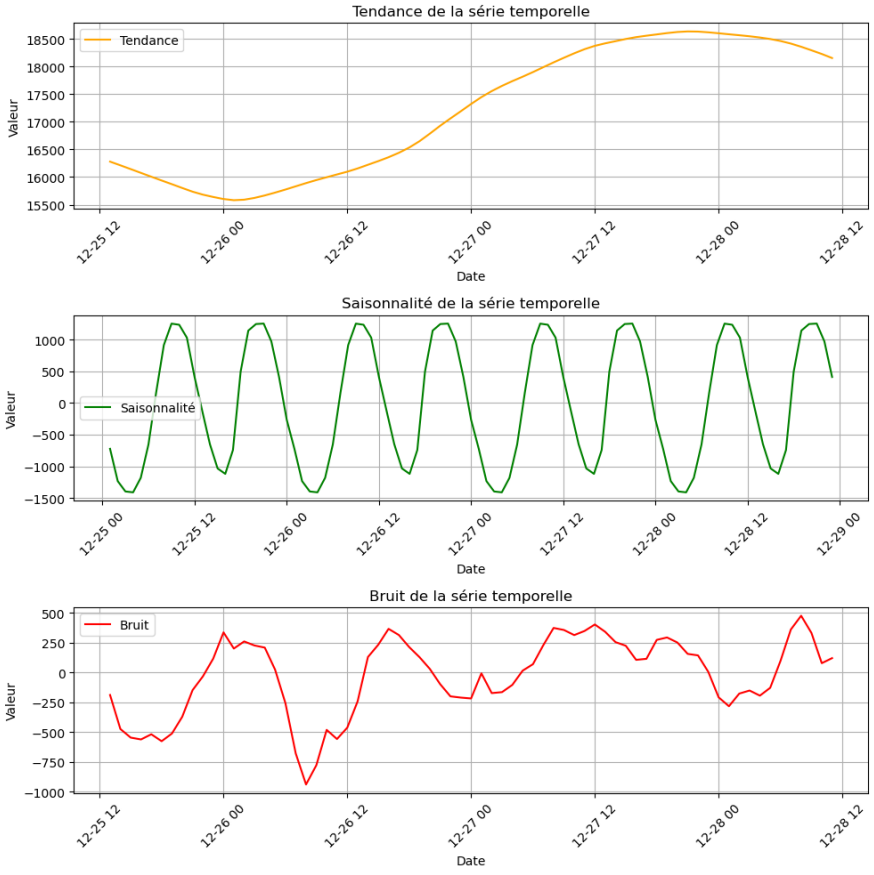


FIGURE 1.6 – Décomposition STL

— **Le processus ARIMA**

Le processus ARIMA est une méthode statistique utilisée pour analyser et prévoir des séries temporelles. Il combine trois composantes :

— Auto-régressive (AR)
— Intégrée (I) : implique la différenciation des données pour les rendre stationnaires.
— Moyenne mobile (MA).

Un modèle ARIMA est noté ARIMA(p, d, q), où :
— p : ordre de la partie auto-régressive.
— d : nombre de différenciations ordinaire nécessaires pour stationnariser la série.
— q : ordre de la partie moyenne mobile .

La formule de prédiction ARIMA est 1.37

$$y'_t = c + \sum_{i=1}^{p} \phi_i y'_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \tag{1.37}$$

où :
— y'_t : valeur prédite de la série à l'instant t, une fois rendue stationnaire par d différenciations,
— c : constante (optionnelle),
— ϕ_i : coefficients auto-régressifs (AR) d'ordre i,
— θ_j : coefficients de la moyenne mobile (MA) d'ordre j,
— ε_t : terme d'erreur ou bruit blanc à l'instant t,

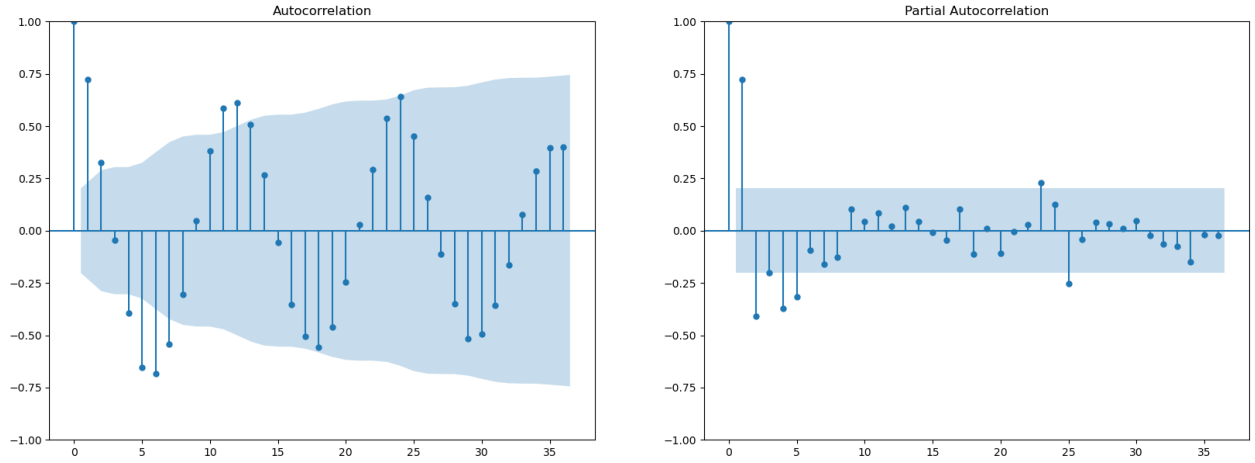


FIGURE 1.7 – Graphiques de l'ACF et du PACF après une différenciation ordinaire sur l'exemple étudié

Après l'application d'une différenciation ordinaire, le graphe ACF montre une réduction de la tendance, mais conserve une structure saisonnière marquée qui s'estime à une période de 12, ce qui suggère la nécessité d'une différenciation saisonnière supplémentaire pour rendre la série pleinement stationnaire.

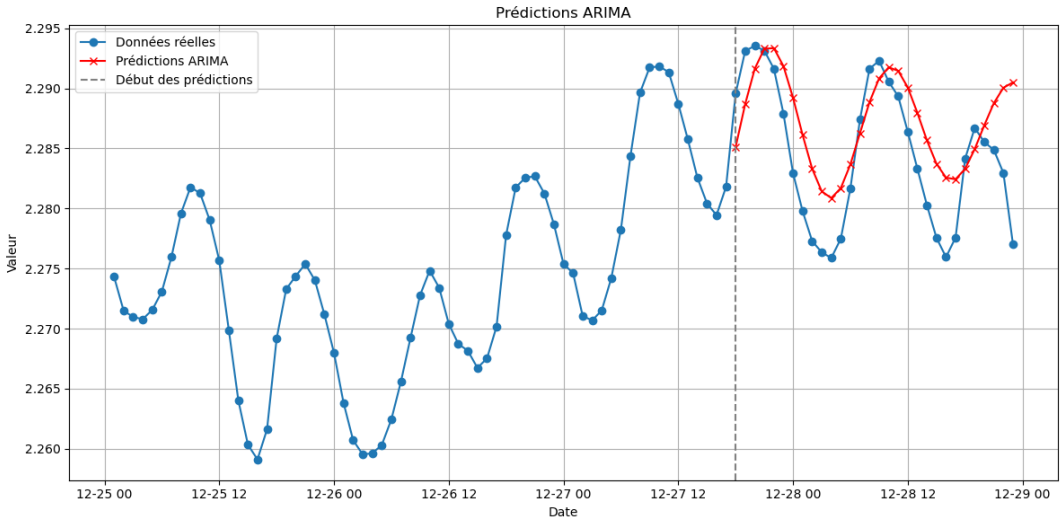


FIGURE 1.8 – Prédiction avec la méthode ARIMA

Les prédictions obtenues avec le modèle ARIMA présentent un écart par rapport aux valeurs réelles, car ce modèle ne prend pas en compte les variations saisonnières présentes dans le graphe ACF.
Le MAPE calculé est de 17,67%.

Le modèle ARIMA retenue est le suivant : ARIMA (3, 1, 2)

— **Le processus SARIMA**

Le processus SARIMA est une extension du modèle ARIMA qui intègre des composantes saisonnières pour analyser et prévoir des séries temporelles présentant des variations périodiques. Il est particulièrement utile lorsque les données montrent des schémas saisonniers réguliers.

Un modèle SARIMA est noté SARIMA(p,d,q)(P,D,Q)[s], où :

Paramètre	**Description**
p, d, q	Paramètres non saisonniers du modèle ARIMA
P	Ordre de la partie auto-régressive saisonnière (correspond au premier pic significatif de l'ACF après une période complète, qui ne dépasse plus le seuil critique z)
D	Nombre de différenciations saisonnières appliquées à la série temporelle
Q	Ordre de la partie moyenne mobile saisonnière (identifié à partir du PACF comme le premier pic saisonnier ne dépassant plus le seuil z)
s	Période de la saisonnalité

TABLE 1.5 – Paramètres du modèle SARIMA

La formulation mathématique d'un modèle SARIMA combine les composantes ARIMA classiques avec des termes saisonniers pour capturer les dépendances à la fois à court terme et saisonnières dans les données [7].

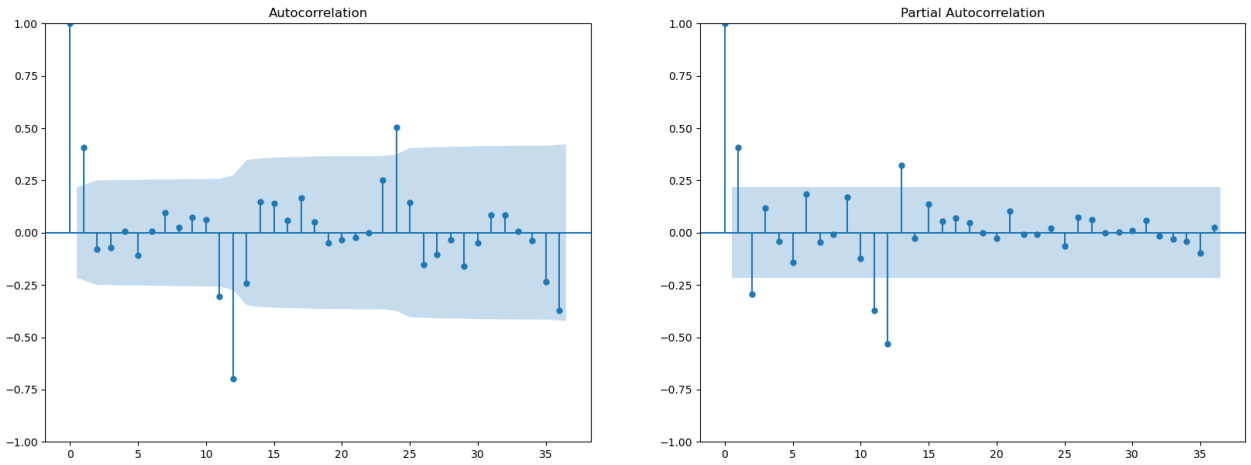


FIGURE 1.9 – Le graphe du ACF et du PACF sur la série étudiée après la différenciation ordinaire et saisonnière

Après l'application d'une différenciation saisonnière, les graphes ACF et PACF montrent une atténuation claire de la composante saisonnière, indiquant que la série est désormais plus proche de la stationnarité et prête à être modélisée avec le modèle SARIMA.

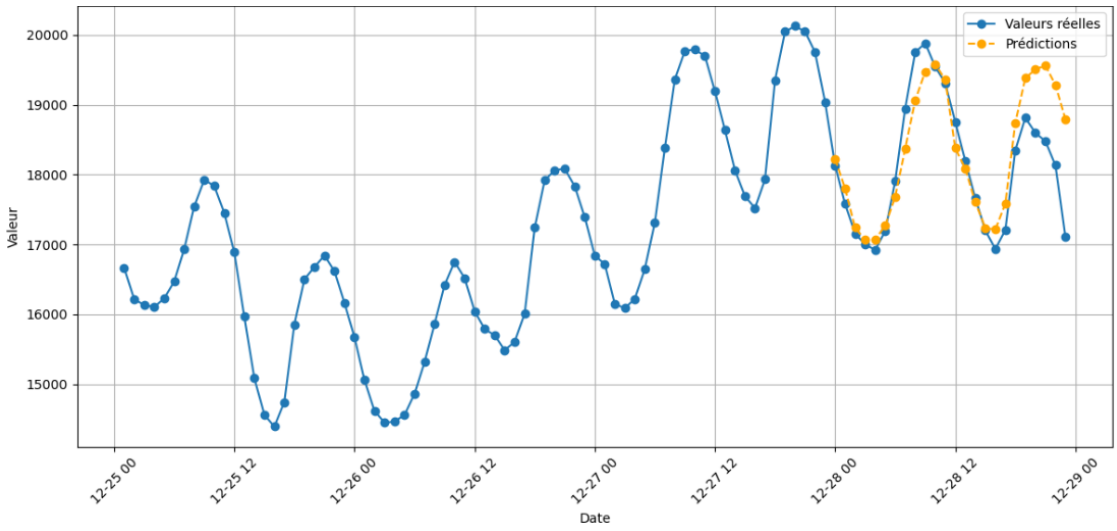


FIGURE 1.10 – Prédiction effectuée sur la série étudiée avec la méthode SARIMA

Ce graphique montre les prévisions avec un modèle SARIMA, la courbe bleue représente les données réelles, et la courbe orange, les valeurs prédites. La ligne rouge marque le début de la prédiction (le 27 décembre). Le modèle suit bien la tendance et la saisonnalité, avec une précision correcte dans les prévisions donnant un taux d'erreur de 10,5% et un AIC de 659.

Le modèle SARIMA retenue est le suivant : SARIMA (0, 1, 1)(1, 1, 0)[12].

— **Le processus SARIMAX**
Le processus SARIMAX est une extension du modèle ARIMA qui intègre à la fois des composantes saisonnières et des variables exogènes pour analyser et prévoir des séries temporelles. Cette approche permet de modéliser des données présentant des schémas

saisonniers tout en tenant compte de facteurs externes susceptibles d'influencer la variable étudiée.

Un modèle **SARIMAX** est généralement noté de la même manière que le modèle **SARIMA**, à la différence qu'il intègre des variables exogènes. Sa notation standard est la suivante :

SARIMAX(p, d, q)(P, D, Q, s)

Les variables exogènes sont des facteurs externes qui peuvent influencer la série temporelle, tels que des indicateurs économiques, des campagnes publicitaires ou des événements spécifiques. En les intégrant dans le modèle, le SARIMAX permet d'améliorer la précision des prévisions en tenant compte de ces influences extérieures[7].

Dans notre cas, les variables exogènes utilisées sont les températures moyennes des villes des États-Unis, enregistrées durant la période allant du 25 décembre 2004 à 01h00 au 28 décembre 2004 à 23h00.

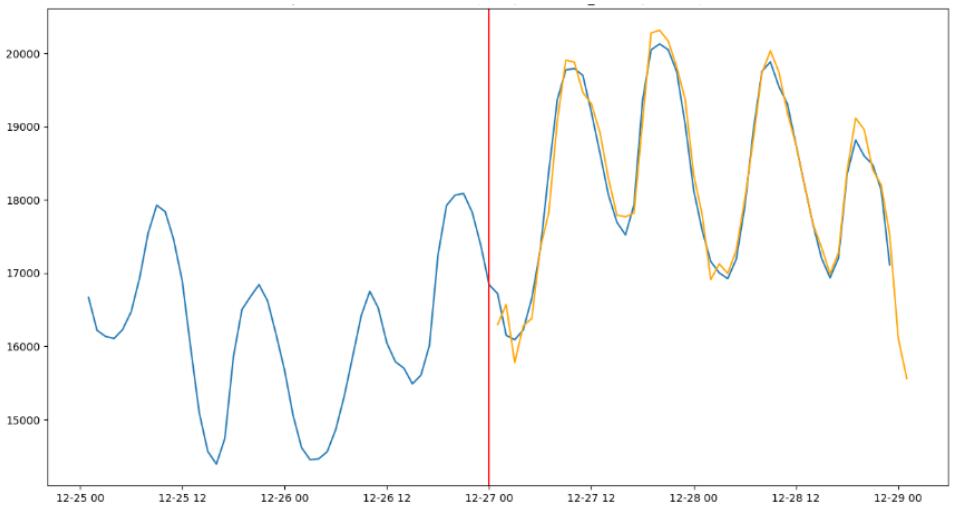


FIGURE 1.11 – Prédiction effectué sur la série étudiée avec la méthode SARIMAX

Les résultats montrent une performance de prédiction supérieur à celle de SARIMA. L'intégration des valeurs exogènes a permis de mieux capturer les différentes composantes saisonnières et les variations des données donnant un taux d'erreur de 5,4% et un AIC de 450.

Le modèle SARIMAX retenue est le suivant : SARIMAX (0, 1, 1)(1, 1, 0)[12]

1.4 Conclusion

Ce chapitre est centré sur l'analyse des série temporelles, il met en évidence leur importance dans la prévision, notamment dans le domaine de la consommation d'énergie. Après avoir défini les séries temporelles et leurs composantes, nous avons présenté différentes méthodes de modélisation, des approches classiques aux modèles statistiques comme ARIMA, SARIMA ou SARIMAX. Ces outils permettent de prédire et d'optimiser la gestion des ressources. Les concepts abordés ici serviront de base à l'introduction du deep learning dans le chapitre suivant.

Chapitre 2

Méthodes de Deep Learning pour la prédiction

La naissance de l'Intelligence Artificielle est attribuée à Alan Turing dans les années 1950, et le terme a été officialisé en 1956 par **John McCarthy**. L'IA est définie comme l'ensemble des théories et techniques mises en œuvre pour créer des machines capables de simuler l'intelligence humaine [22].
Au sain de ce vaste domaine, le Machine Learning (apprentissage automatique) occupe une place centrale dont l'un de ses principaux objectifs est d'améliorer la précision de prédiction et d'automatiser des systemes complexes.

2.1 Le Machine Learning

2.1.1 Définition et Historique

Le Machine Learning (ML) ou L'apprentissage automatique est un domaine d'étude de l'intelligence artificielle qui s'intéresse au développement et à l'étude d'algorithmes statistiques capables d'apprendre à partir de données et de généraliser à des données invisibles, et ainsi d'effectuer des tâches sans instructions explicites .Il prend ses racines dans les années 1950 avec le Perceptron, un modèle de neurone artificiel développé par Frank Rosenblatt en 1957 [26].

Il peut être utilisé dans plusieurs domaines tel que :
— Santé : Diagnostic médical, analyse d'images médicales, prédiction des épidémies.
— Finance : Détection de fraudes, gestion de risques.
— Sécurité : Détection d'intrusions, reconnaissance faciale, surveillance vidéo.

2.1.2 Types d'Apprentissage en ML

L'apprentissage automatique est réparti en trois types principaux : **l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement**.

— **Apprentissage supervisé :** l'apprentissage supervisé est un type d'apprentissage automatique, la branche la plus populaire du ML, dans lequel un algorithme apprend à partir de données étiquetées.

Ces données possèdent des caractéristiques précises et sont associées à une sortie connue, ce qui permet au modèle d'établir des relations entre les entrées et les sorties

pour effectuer des prédictions. Dans ce cadre, on distingue deux grands types de problèmes :

1. Classification : la classification est une tâche de l'apprentissage supervisé où l'objectif est de prédire une étiquette ou une catégorie à partir des caractéristiques d'une observation.
   Plus précisément, étant donné un ensemble de données d'apprentissage contenant des exemples étiquetés, un modèle de classification est formé pour apprendre la relation entre les entrées (les caractéristiques) et les sorties (les étiquettes). Ce modèle est ensuite utilisé pour prédire l'étiquette d'observations inconnues, souvent dans un cadre où les catégories sont discrètes [17].

   Des algorithmes comme les Support Vector Machines (SVM) et les Random Forests sont souvent utilisés pour résoudre ce type de problème.
2. La régression : la régression est une tâche d'apprentissage supervisé qui vise à prédire une valeur continue à partir de variables explicatives. Contrairement à la classification, où la sortie est une catégorie discrète, la régression permet d'estimer une quantité numérique en modélisant la relation entre les variables d'entrée et la variable cible [14].

   Contrairement à la classification qui attribue une catégorie ("catégorie A" ou "catégorie B"), la régression vise à estimer des quantités continues, comme le prix d'une maison, la température ou le revenu. Cela en fait un outil essentiel pour des applications telles que la prévision de ventes, l'estimation de prix ou la prédiction de tendances économiques.

— **Apprentissage non supervisé**
L'apprentissage non supervisé est une approche du ML où un modèle est entraîné sur des données non étiquetées, c'est-à-dire sans réponse attendue. L'objectif est d'identifier des structures, des motifs ou des regroupements dans les données sans supervision humaine [9].
Contrairement à l'apprentissage supervisé, aucune supervision n'est nécessaire, l'algorithme apprend de lui-même à organiser ou comprendre les données.
On distingue deux principaux types de tâches :

1. Le Clustering (Regroupement) : le clustering consiste à regrouper les données en fonction de leurs similitudes. Le modèle apprend tout seul à classer les données en différentes catégories sans qu'on lui dise quelles sont ces catégories à l'avance.K-Means et DBSCAN sont des algorithmes utilisés pour le clustering.
2. La réduction de dimension : elle vise à simplifier les données en réduisant le nombre de caractéristiques tout en conservant les informations importantes.Lorsqu'il y a beaucoup de variables (caractéristiques), certaines peuvent être inutiles ou redondantes. La réduction de dimension permet de les simplifier.

— **Apprentissage par renforcement**
L'apprentissage par renforcement (Reinforcement Learning - RL) est une approche du ML où un système apprend à prendre des décisions en interagissant avec un environnement afin de maximiser une récompense. L'agent explore différentes actions et reçoit des retours sous forme de récompenses positives ou négatives, ce qui lui permet d'améliorer progressivement sa stratégie [43].

En d'autres termes c'est un processus de ML qui apprend à accomplir une tâche en fonction de ses succès et échecs.
L'algorithme prend des actions dans un environnement et ajuste ses stratégies en fonction des récompenses obtenues pour ses bonnes actions et des pénalités pour les mauvaises, afin d'améliorer ses performances à long terme.

2.2 Le Deep Learning (DL)

Le DL est une branche du ML qui utilise des réseaux de neurones artificiels profonds pour modéliser et apprendre des représentations complexes à partir de grandes quantités de données. Il repose sur le même principe que le ML, qui consiste à entraîner des modèles à partir de données, mais se distingue par l'utilisation exclusive des réseaux de neurones artificiels pour l'apprentissage automatique [23].
Le DL construit un modèle en s'entraînant sur des données existantes afin d'ajuster ses paramètres et obtenir un modèle optimal pour la tâche cible.

2.2.1 Définition des Réseaux de Neurones

Les RN artificiels sont des modèles d'apprentissage inspirés de la structure du cerveau humain. Ils sont constitués de couches de neurones interconnectés qui ajustent dynamiquement leurs poids pour reconnaître des motifs complexes et faire des prédictions [39].

2.2.2 Evolution et Historique des RN

Les RN artificiels naissent en 1943 avec **McCulloch** et **Pitts**. En 1957, le psychologue Frank Rosenblatt développe le perceptron qui est un modèle très simple de l'IA, mais ses limites mises en évidence en 1969 par Minsky et Papert conduisent à l'hiver de L'IA (1974-1980). En 1986, **Hinton** relance le domaine avec le perceptron multicouche et la rétropropagation. Dans les années 1990, en 1997, d'autres méthodes sont développées comme le LSTM qui améliorent le traitement des séquences. Enfin, en 2012, la compétition ImageNet qui est une énorme base de données d'images marque l'essor du DL, révolutionnant l'intelligence artificielle et la force du DL.

2.2.3 Structure d'un Neurone Artificiel

Un neurone artificiel est un composant de base dans un réseau de neurones. Il traite une ou plusieurs entrées, applique une transformation et produit une sortie.
— **Les entrées :** les entrées d'un neurone sont représentées par un **vecteur de valeurs numériques**, comme indiqué dans l'équation 2.1 :

$$\mathbf{x} = (x_1, x_2, ..., x_n) \tag{2.1}$$

— **Transformation :**

1. Chaque entrée est pondérée par un **poids** w_i, qui ajuste son importance. Un **biais** b est également ajouté pour améliorer la flexibilité du modèle. La combinaison linéaire pondérée est alors exprimée par l'équation 2.2 :

$$z = \sum_{i=1}^{n} w_i x_i + b \tag{2.2}$$

2. La fonction z est ensuite transformée par une fonction d'activation pour produire la sortie du neurone. Les fonctions d'activation les plus utilisées en DL sont présentées ci-dessous :

   ReLU (Rectified Linear Unit) : utilisée principalement dans les couches cachées des grands réseaux de neurones. Sa fonction est définie par l'équation 2.3 :

$$f(z) = \max(0, z) \tag{2.3}$$

   Sigmoïde : principalement utilisée pour les problèmes de classification binaire. Définie par l'équation 2.4 :

$$f(z) = \frac{1}{1 + e^{-z}} \tag{2.4}$$

   Tanh (Tangente hyperbolique) : surtout utilisée dans les couches cachées des petits réseaux de neurones. Définie par l'équation 2.5 :

$$f(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \tag{2.5}$$

   Softmax : utilisée pour les tâches de classification multiple (choisir la classe la plus probable parmi plusieurs options). Définie par l'équation 2.6 :

$$f(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{2.6}$$

3. Sortie : la sortie y du neurone dépend de la fonction d'activation choisie. Elle peut être :
   - Une valeur continue (ex : prédiction de prix en régression).
   - Une probabilité (ex : classification binaire avec sigmoïde).
   - Un vecteur de probabilités (ex : classification multi-classes avec softmax).

2.2.4 Les types de RN

2.2.4.1 Les réseaux de Neurones Artificiels (ANN)

Les ANN sont des modèles mathématiques organisés en couches (entrée, cachées et sortie). Ils prennent une entrée et génèrent une sortie sans mémoire des états précédents, ce qui signifie qu'ils traitent chaque entrée indépendamment des autres. Cependant, cette limitation représente un défi pour les données séquentielles. Pour pallier cette limite, les RNN ont été développés.

2.2.4.2 Perceptron Multicouche (MLP)

Un MLP est un ANN feedforward (l'information circule uniquement dans une seule direction, de la couche d'entrée vers la couche de sortie), qui se compose de plusieurs couches de neurones, où chaque couche est entièrement connectée à la suivante et applique une fonction d'activation non linéaire [23].

2.2.4.3 Réseaux de Neurones Convolutifs (CNN)

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].

2.2.4.4 Réseaux de Neurones Récurrents(RNN)

est un réseau de neurones spécialisé dans le traitement d'une séquence de valeurs[11].

Les RNN sont des réseaux de neurones conçus pour mémoriser les informations passées grâce à leurs connexions récurrentes. Ils sont particulièrement utiles dans des domaines où le contexte temporel est crucial, comme le traitement du langage naturel ou l'analyse de séries temporelles.

2.2.4.5 LSTM (Long Short-Term Memory)

Les LSTM (Long Short-Term Memory) sont une variante des réseaux de neurones récurrents (RNN) conçue pour capturer les dépendances à long terme dans les données séquentielles. Ils ont été introduits par Hochreiter et Schmidhuber en 1997 dans leur article fondateur[18].

Le LSTM est une version améliorée du RNN, capable de mieux mémoriser les longues séquences et d'éviter le problème de la disparition du gradient, grâce à une cellule mémoire qui peut stocker et gérer l'information à l'aide de portes de régulation. Ces portes permettent de contrôler le flux d'informations, assurant ainsi la conservation des informations importantes sur une longue durée.

- La porte de l'oubli (Forget Gate) : Elle détermine quelles informations de l'état de la cellule précédente doivent être conservées et quelles informations doivent être oubliées.
- La porte d'entrées (Input Gate) : Elle détermine quelles nouvelles informations de l'entrée doivent être stockées et ajoutées dans l'état de la cellule.
- La porte de sortie (Output Gate) : Elle détermine quelles informations de l'état de la cellule doivent être utilisées pour produire la sortie de la cellule.

2.2.4.6 Le GRU (Gated Recurrent Unit)

Le GRU (Gated Recurrent Unit), introduit par Cho et al. (2014), est une version simplifiée du LSTM. Il utilise deux portes pour contrôler le flux d'information et permet de capturer les dépendances à long terme tout en étant plus léger et plus rapide à entraîner que le LSTM[12]. Le GRU utilise deux portes :

- La porte de mise à jour : qui est la combinaison entre la porte d'oubli et la porte d'entrée des LSTM, elle reprend les fonctions de ces deux portes.
- Porte de réinitialisation : elle contrôle combien d'informations passées doivent être oubliées.

2.2.4.7 Autoencodeurs (AE - Autoencoders)

Un autoencodeur est un type de réseau de neurones artificiels utilisé pour l'apprentissage non supervisé, visant à apprendre une représentation efficace des données en les encodant dans une dimension inférieure, puis en les décodant pour reconstruire les données originales.

Cette technique est couramment employée pour la détection d'anomalies et la génération de données[2].
Il prend des données en entrée, il les transforme en une version plus compacte, et il essaie ensuite de recréer les données d'origine à partir de cette version compacte.

2.2.4.8 GAN (Generative Adversarial Networks)

Un GAN est un type de ANN utilisé pour l'apprentissage non supervisé, introduit par **Ian Goodfellow** en 2014. Il repose sur l'interaction de deux réseaux de neurones opposés :

— **Le générateur** : produit des données synthétiques à partir d'un bruit aléatoire.
— **Le discriminateur** : tente de distinguer les données réelles des données générées.

Ces deux réseaux s'entraînent ensemble, ce qui permet au générateur d'améliorer progressivement ses données synthétiques jusqu'à ce qu'elles ressemblent aux données réelles [23].

2.2.4.9 Transformers

les Transformers sont une architecture de RN utilisée en traitement du langage naturel (NLP). Grâce à leur mécanisme d'auto-attention, ils traitent toute une séquence en parallèle, ce qui les rend plus efficaces que les modèles récurrents pour capturer les relations à longue distance [32].

2.2.5 Apprentissage des RN

L'entraînement d'un RN est un processus itératif qui vise à minimiser l'erreur entre les prédictions du réseau et les valeurs réelles. La première étape de l'apprentissage est **la forward propagation** (propagation avant) qui consiste à faire passer les données d'entrée pour obtenir une sortie. La deuxième est **la fonction de coût** mesure l'erreur du modèle, ensuite vient **la back propagation** (rétropropagation) qui calcule les gradients de cette erreur par rapport aux poids du réseau qui sont utilisés par **la descente de gradient** pour améliorer le modèle. Ce processus permet au réseau d'apprendre et d'améliorer ses prédictions au fil du temps.

1. **Forward Propagation** : c'est la première étape de l'apprentissage des RN, elle vise a faire propager les données d'entrées d'une couche à une autre en appliquant la pondération z 2.2 et la fonction d'activation qui applique une transformation non linéaire aux sorties des neurones.
   La sortie d'une couche devient l'entrée de la couche suivante et la sortie finale produit une valeur prédite.

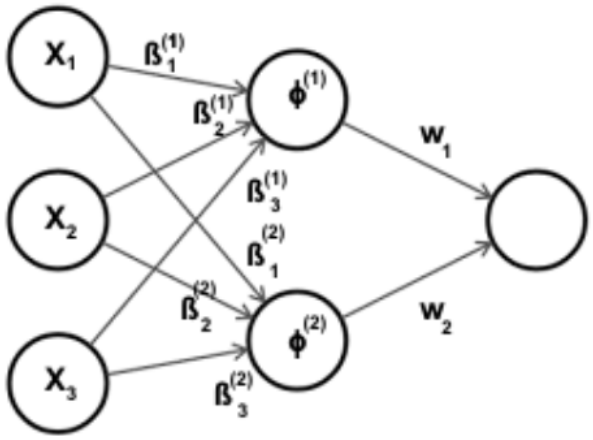


FIGURE 2.1 – Architecture d'un réseau de neurones avec 3 variables d'entrée, une couche cachée contenant 2 neurones et une couche de sortie composée d'un seul neurone.

Modélisation mathématique d'un neurone artificiel

— Couche cachée : Les combinaisons linéaires pour chaque neurone de la couche cachée sont données par les équations 2.7 et 2.8 :

$$z_1 = \beta_1^{(1)} X_1 + \beta_2^{(1)} X_2 + \beta_3^{(1)} X_3 + b_1 \tag{2.7}$$

$$z_2 = \beta_1^{(2)} X_1 + \beta_2^{(2)} X_2 + \beta_3^{(2)} X_3 + b_2 \tag{2.8}$$

Symbole	**Signification**
z_1, z_2	Valeurs nettes d'activation (somme pondérée + biais) pour les neurones 1 et 2 de la couche cachée.
X_1, X_2, X_3	Variables d'entrée (features).
$\beta_j^{(i)}$	Poids synaptique associé à la connexion entre l'entrée X_j et le neurone z_i.
b_1, b_2	Biais (valeurs ajoutées à la somme pondérée) pour les neurones z_1 et z_2.

TABLE 2.1 – Signification des composantes des équations d'entrée des neurones

Ces valeurs sont ensuite transformées par une fonction d'activation, comme indiqué dans les équations 2.9 et 2.10 :

$$a_1 = \phi(z_1) \tag{2.9}$$

$$a_2 = \phi(z_2) \tag{2.10}$$

— Couche de sortie : le calcul de la sortie du modèle est obtenue à partir des activations des couches cachées, selon les équations 2.11 et 2.12 :

$$z_{\text{out}} = w_1 a_1 + w_2 a_2 + b \tag{2.11}$$

$$a_{\text{out}} = \sigma(z_{\text{out}}) \tag{2.12}$$

2. **Fonction de Coût** : Aussi appelée fonction de perte ou fonction d'erreur, la fonction de coût permet d'évaluer la performance du modèle elle mesure l'erreur entre la sortie a_{out} du réseau et la vraie valeur y.

   Il existe plusieurs fonctions de coût, la fonction adéquate au modèle est choisie selon le type de problème (classification, régression, séries temporelles, etc.), la nature des sorties (discrètes, continues, probabilités, etc.), les propriétés de convergence.
   Dans ce qui suit les fonctions de coût de la classification et de la régression (prédiction des séries temporelles) seront présentées.

   — **Fonction de coût pour la classification binaire :** la fonction de coût (entropie croisée) pour un problème de classification binaire est donnée par l'équation 2.13 :

$$J = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)\right] \tag{2.13}$$

   Où :
   — N : nombre total d'exemples dans l'ensemble d'entraînement.
   — $y_i \in \{0,1\}$: vraie classe de l'exemple i (0 ou 1).
   — $\hat{y}_i$: probabilité prédite pour la classe 1 (sortie de la fonction sigmoïde).

   — **Fonction de coût pour la classification multiclasse :** pour un problème de classification à K classes, la fonction de coût généralisée (entropie croisée multiclasse) est donnée par l'équation 2.14 :

$$J = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{i,k} \log \hat{y}_{i,k} \tag{2.14}$$

   Où :
   — K : nombre total de classes.
   — $y_{i,k} \in \{0,1\}$: indicateur binaire indiquant si l'exemple i appartient à la classe k.
   — $\hat{y}_{i,k}$: probabilité prédite pour la classe k (sortie de la fonction softmax).

   — **Fonction de coût pour la régression (MSE) :** pour les problèmes de régression et séries temporelles, on utilise généralement l'erreur quadratique moyenne, définie par l'équation 2.15 :

$$J = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.15}$$

   Où :
   — y_i : valeur réelle pour l'exemple i.
   — $\hat{y}_i$: valeur prédite pour l'exemple i
   .

3. **Rétropropagation (Backpropagation) :** la rétropropagation est une étape essentielle dans l'apprentissage des réseaux de neurones. Elle permet au modèle de corriger ses erreurs en ajustant les poids à l'aide des gradients calculés par la dérivée de la fonction de coût. Ce processus détermine comment la fonction de coût évolue de la couche de sortie vers la couche d'entrée, à travers le calcul des dérivées partielles (gradients).

La forme générale du gradient de la fonction de coût J par rapport à un poids w_i est donnée par l'équation 2.16 :

$$\frac{\partial J}{\partial w_i} = \frac{\partial J}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial w_i} \tag{2.16}$$

Et pour le biais b_i, l'équation 2.17 :

$$\frac{\partial J}{\partial b_i} = \frac{\partial J}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial b_i} \tag{2.17}$$

TABLE 2.2 – Signification des notations utilisées dans le calcul des gradients

Symbole	**Description**
J	Fonction de coût, qui mesure l'erreur du modèle
$\hat{y}$	Sortie prédite du modèle (valeur produite par la fonction d'activation)
y	Valeur réelle (label de l'observation)
z	Sortie linéaire du neurone, avant application de la fonction d'activation
w_i	Poids associé à l'entrée X_i

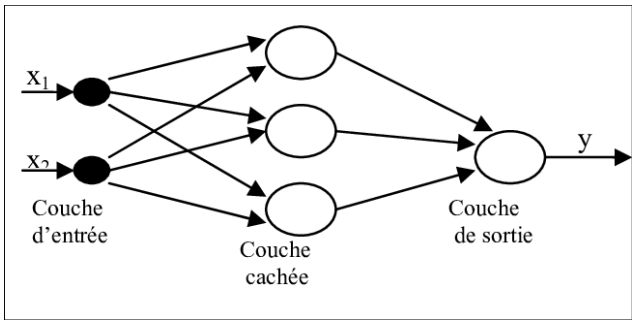


FIGURE 2.2 – Réseau de neurone à deux couches

Les gradients entre la couche de sortie et la couche cachée de cet exemple sont donnés par les équations 2.18 et 2.19 :

$$\frac{\partial J}{\partial W_2} = \frac{\partial J}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial Z_2} \cdot \frac{\partial Z_2}{\partial W_2} \tag{2.18}$$

$$\frac{\partial J}{\partial b_2} = \frac{\partial J}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial Z_2} \cdot \frac{\partial Z_2}{\partial b_2} \tag{2.19}$$

Les gradients entre la couche cachée et la couche d'entrée, sont donnés par les équations 2.20 et 2.21 :

$$\frac{\partial J}{\partial W_1} = \frac{\partial J}{\partial A_1} \cdot \frac{\partial A_1}{\partial Z_2} \cdot \frac{\partial Z_2}{\partial A_1} \cdot \frac{\partial A_1}{\partial Z_1} \cdot \frac{\partial Z_1}{\partial W_1} \tag{2.20}$$

$$\frac{\partial J}{\partial b_1} = \frac{\partial J}{\partial A_1} \cdot \frac{\partial A_1}{\partial Z_2} \cdot \frac{\partial Z_2}{\partial A_1} \cdot \frac{\partial A_1}{\partial Z_1} \cdot \frac{\partial Z_1}{\partial b_1} \tag{2.21}$$

TABLE 2.3 – Signification des notations utilisées dans la rétropropagation

Symbole	Description
W_1, W_2	Matrices des poids : W_1 entre l'entrée et la couche cachée, W_2 entre la couche cachée et la couche de sortie
b_i	Vecteur des biais pour la couche i
Z_i	Somme pondérée des entrées de la couche i : $Z_i = W_i X + b_i$
A_i	Sortie activée de la couche i, après application de la fonction d'activation : $A_i = f(Z_i)$

L'utilisation des matrices rend les calculs plus rapides, plus efficaces et plus généralisables pour entraîner des réseaux de neurones de grande taille.

Ces gradients sont ensuite utilisés dans la **Descente de Gradient** pour mettre à jour les poids et améliorer l'apprentissage du réseau.

4. **Descente de Gradient (Gradient Descent)** : le gradient indique dans quelle direction modifier les poids pour réduire l'erreur. La descente de gradient est un algorithme d'optimisation utilisé pour ajuster les poids d'un réseau de neurones dans le but de minimiser la fonction de coût.

   Les équations 2.22 et 2.23 décrivent la mise à jour des poids W_i et des biais b_i à chaque itération :

$$W_i \leftarrow W_i - \eta \frac{\partial J}{\partial W_i} \tag{2.22}$$

$$b_i \leftarrow b_i - \eta \frac{\partial J}{\partial b_i} \tag{2.23}$$

   où :

   — η est le learning rate (taux d'apprentissage) qui est un hyperparamètre qui contrôle la vitesse à laquelle un modèle ajuste ses poids lors de l'entraînement des mises à jour, ses valeurs les plus courantes sont :

   — Petit réseau simple : 0.01 à 0.1 .
   — Réseaux profonds : 0.0001 à 0.01 .
   le η est ajusté en fonction des résultats .

   (a) Si $\frac{\partial J}{\partial W}$ est positif, cela signifie que J augmente si W augmente, donc nous **diminuons** W (même principe pour $\frac{\partial J}{\partial b}$).

   (b) Si $\frac{\partial J}{\partial W}$ est négatif, cela signifie que J diminue si W augmente, donc nous **augmentons** W (même principe pour $\frac{\partial J}{\partial b}$).

2.3 Le RNN (Recurrent Neural Networks)

2.3.1 Définition des RNN

Uu RNN est un type de ANN doté d'une mémoire lui permettant de se rappeler des informations passées. Cette mémoire est représentée par des connexions récurrentes entre

les neurones ou les unités de neurones, formant ainsi un cycle récurrent, d'où le terme récurrent dans RNN.

Contrairement aux RN traditionnels, les RNN peuvent traiter des données séquentielles en reconnaissant des motifs dans des séquences comme le texte, les séries temporelles, les signaux audio et la vidéo. Leur mémoire interne leur permet de retenir des informations sur de longues séquences.

2.3.2 Historique des RNN

Les Réseaux de Neurones Récurrents (RNN) ont été développés dans les années 1980 par **John Hopfield** et **David Rumelhart** afin de capturer les dépendances séquentielles dans les données. Dans les années 1990, **Sepp Hochreiter** et **Jürgen Schmidhuber** ont mis en évidence le problème de disparition du gradient, limitant leur efficacité sur de longues séquences.

2.3.3 Architecture RNN

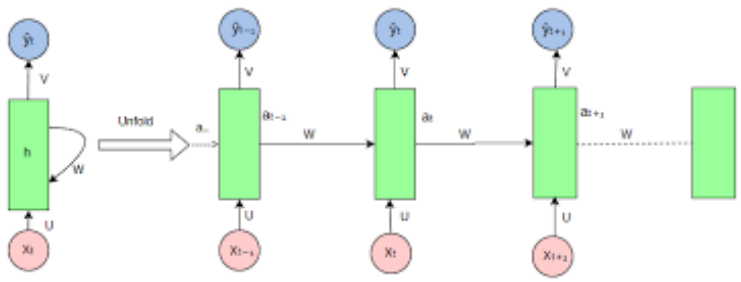


FIGURE 2.3 – Architecture générale d'un Réseau de Neurones Récurrent (RNN

En plus de l'entrée x^t, l'unité neuronale reçoit comme entrée supplémentaire la sortie de l'unité précédente. Cette entrée correspond à l'**état caché** à l'instant $t-1$. Par exemple, pour le second bloc, nous utilisons deux entrées : x^2 et a^1 (état caché à $t = 1$).
Cela illustre la connexion récurrente, propre aux RNN, où chaque bloc est connecté à lui-même au fil du temps.

L'état initial est souvent défini comme :

$$a^0 = 0 \tag{2.24}$$

L'unité récurrente de base est alors décrite par l'équation 2.25 :

$$a^t = \sigma(W_x \cdot x^t + W_a \cdot a^{t-1} + b_a) \tag{2.25}$$

La sortie à l'instant t, donnée par l'équation 2.26, est :

$$y^t = \sigma(W_y \cdot a^t + b_y) \tag{2.26}$$

Où :
— W_y : poids de la couche de sortie
— b_y : biais de la sortie

TABLE 2.4 – Signification des paramètres de l'unité récurrente

Symbole	Description
a^t	État caché au temps t
x^t	Entrée à l'instant t
W_x	Poids liés aux entrées
W_a	Poids des connexions récurrentes
b_a	Biais appliqué à l'état caché
σ	Fonction d'activation (ex. ReLU, tanh)

2.3.4 Les Différents Schémas d'Architecture des RNN

— **One to Many :** un modèle One-to-Many prend une seule entrée et génère une séquence de sorties. Ce type de modèle est utile lorsqu'on souhaite produire une série d'éléments à partir d'un seul point de départ. Par exemple, il peut être utilisé pour générer une description textuelle à partir d'une image : une seule image en entrée produit une séquence de mots qui forment une légende descriptive.

Le fonctionnement général d'un modèle One-to-Many est illustré par l'équation 2.27 :

$$X \to Y_1, Y_2, Y_3, \ldots, Y_n \tag{2.27}$$

— **Many to One :** un modèle Many-to-One prend une séquence d'éléments en entrée et produit une seule sortie. Cela signifie qu'il analyse plusieurs données successives pour aboutir à un résultat unique. Par exemple, dans l'analyse des sentiments, une phrase complète (une suite de mots) est traitée pour déterminer un sentiment global, comme positif, négatif ou neutre.

La structure générale de ce modèle est représentée par l'équation 2.28 :

$$X_1, X_2, X_3, \ldots, X_n \to Y \tag{2.28}$$

— **Many to Many :**un modèle Many-to-Many prend une séquence d'entrées et produit une séquence de sorties. Ce type de modèle est utilisé lorsque l'entrée et la sortie sont toutes deux des séquences, que celles-ci aient des longueurs variables ou égales. Pour gérer cela, une architecture courante est celle de l'encodeur-décodeur :

1. Encodeur : reçoit une séquence en entrée et la transforme en un vecteur de contexte, une représentation compacte qui capture l'essentiel de l'information de la séquence.
2. Décodeur : utilise le vecteur de contexte pour générer pas à pas la séquence de sortie, en produisant un élément de la séquence à la fois.

L'un des avantages de cette architecture encodeur-décodeur :

— Gérer des séquences d'entrée et de sortie de différentes longueurs .

— L'encodeur traite l'ensemble de la séquence d'entrée : Cela permet de capturer une compréhension globale du contexte, ce qui améliore la qualité de la séquence de sortie générée par le décodeur.

2.4 GRU (Gated Reccurent Unit)

Malgré l'introduction des RNN pour le traitement des séries séquentielles, un problème persiste, celui de la disparition ou de l'explosion du gradient en conséquence de

la multiplication répétée des dérivées rendant difficile la capture des dépendances à long terme.

2.4.1 Explosion de gradient

L'explosion de gradient se déroule lors de la rétropropagation, les gradients deviennent extrêmement grands ce qui rend les mises à jour des poids instables, les RNN peuvent devenir instables pendant l'entraînement rendant difficile la convergence vers une solution optimale, ce problème peut se dérouler lorsque l'initialisation des poids est faite avec des valeurs élevées, ce qui peut aussi rendre les gradients élevés.
Comme solution, une technique appelée **régularisation** qui limite la taille des gradients à une valeur maximale prédéfinie pour éviter l'explosion est utilisée.

2.4.2 Disparition du gradient

La disparition du gradient est un problème fréquent lors de la rétropropagation dans les réseaux de neurones récurrents (RNN). Elle survient lorsque les gradients deviennent extrêmement petits au fur et à mesure qu'ils sont propagés en arrière dans le temps. Ce phénomène est particulièrement accentué lors de l'utilisation de fonctions d'activation comme la tangente hyperbolique.

Il en résulte une difficulté à apprendre les dépendances à long terme, car les mises à jour des poids deviennent négligeables. Le modèle stagne alors dans l'apprentissage et n'arrive pas à réduire efficacement la fonction de coût.

La mise à jour de l'état caché, qui illustre ce phénomène, est décrite par l'équation 2.29 :

$$a^t = \sigma(W_x \cdot x^t + W_a \cdot a^{t-1}) \tag{2.29}$$

Symbole	Signification
a^t	État caché (ou activation) à l'instant t.
x^t	Entrée du réseau à l'instant t.
a^{t-1}	État caché (ou activation) à l'instant précédent $t-1$.
W_x	Matrice de poids associée aux entrées.
W_a	Matrice de poids associée à l'état caché précédent.
σ	Fonction d'activation non linéaire (ex. tanh ou sigmoid).

TABLE 2.5 – Signification des composantes de l'équation d'activation récurrente

Lors de la rétropropagation, le gradient de la perte J par rapport à un état caché précédent a_t est multiplié par W_a à chaque étape :

$$\frac{\partial J}{\partial a_3} = 1 \quad \text{(on suppose un gradient initial de 1)}$$

$$\frac{\partial J}{\partial a_2} = W_a \times 1$$

$$\frac{\partial J}{\partial a_1} = W_a \times W_a \times 1 = W^2$$

Exemple numérique : Si $W_a = 0.5$, alors :

$$\frac{\partial J}{\partial a_2} = 0.5$$

$$\frac{\partial J}{\partial a_1} = 0.5 \times 0.5 = 0.25$$

Le gradient diminue rapidement, ce qui illustre le phénomène de **disparition du gradient**.

Pour résoudre le problème, le **LSTM** a été proposé en 1997, une nouvelle structure de réseaux a été introduite avec des cellules de mémoire et des mécanismes de portes qui permettent de contrôler le flux d'informations de manière adaptative, ce qui fait augmenter la complexité du modèle et le nombre de paramètres, c'est pour cela qu'apparaît le **GRU** afin de simplifier l'architecture LSTM tout en gardant son efficacité.

2.4.3 Définition des réseaux GRU

Les GRU sont une variante des RNN qui introduisent des portes de **mise à jour** et de **réinitialisation** pour contrôler le flux d'information au sein de la cellule récurrente. Elles permettent de résoudre le problème du gradient évanescent et de capturer des dépendances à long terme tout en réduisant la complexité des calculs par rapport aux LSTM et en offrant des performances comparables [11].

2.4.4 Historique des réseaux GRU

Kyunghyun Cho et son équipe ont introduit les GRU en 2014, dans un article dont l'idée principale est de proposer un nouveau modèle pour la traduction se basant sur un RNN de type encodeur décodeur, les auteurs ont présenté une nouvelle cellule qui utilise deux portes de base pour simplifier le LSTM. Le nom GRU a été attribué quelque mois plus tard dans leur nouvel article.

2.4.5 Structure d'une cellule GRU

Une cellule GRU possède comme :
- Entrées : la donnée x^t et l'état caché a^{t-1} à l'instant $t-1$
- Sorties : la prédiction y^t et l'état caché a^t

Son architecture possède deux portes :
- **La porte de réinitialisation (reset gate)** : elle permet de réinitialiser partiellement ou totalement l'état précédent en contrôlant la quantité d'information à oublier avant de calculer le nouvel état caché (c'est-à-dire : quoi ignorer).
  Cela facilite l'intégration de nouvelles informations, tout en atténuant l'influence des données passées non pertinentes.

  L'équation de la porte de réinitialisation est donnée par 2.30 :

$$r^t = \sigma(W_r x^t + U_r a^{t-1} + b_r) \tag{2.30}$$

  La fonction sigmoïde utilisée est :

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

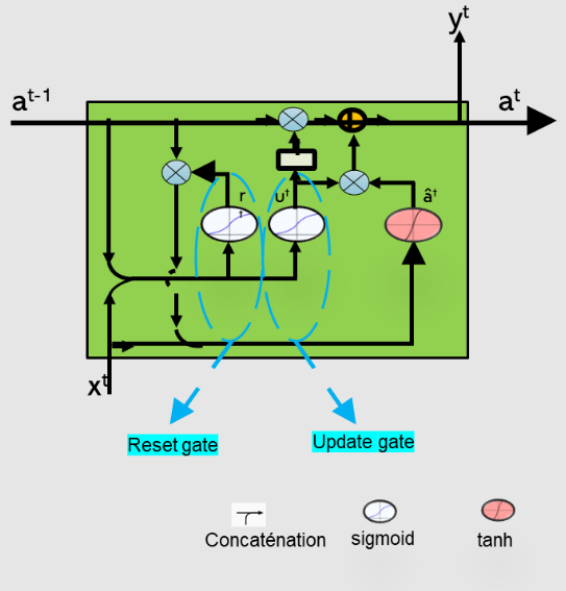


FIGURE 2.4 – Architecture d'une cellule GRU

TABLE 2.6 – Signification des paramètres de la porte de réinitialisation

Symbole	Description
r^t	Vecteur de la porte de réinitialisation (valeurs entre 0 et 1)
σ	Fonction sigmoïde
W_r	Matrice des poids appliqués à l'entrée x^t
U_r	Matrice des poids appliqués à l'état précédent a^{t-1}
b_r	Biais associé à la porte de réinitialisation

Le résultat est un vecteur de valeurs entre 0 et 1 : une valeur proche de 1 signifie que l'information passée est largement conservée, tandis qu'une valeur proche de 0 signifie que les nouvelles informations dominent.

Ensuite, le résultat de la porte de réinitialisation est utilisé pour calculer **l'état candidat** $\tilde{a}^t$, qui sera combiné à la porte de mise à jour pour produire l'état final à l'instant t. L'équation 2.31 représente ce calcul :

$$\tilde{a}^t = \tanh(W_a x^t + U_a(r^t \odot a^{t-1}) + b_a) \tag{2.31}$$

TABLE 2.7 – Signification des paramètres de l'état candidat dans GRU

Symbole	Description
$\tilde{a}^t$	Nouvel état candidat au temps t
W_a	Matrice des poids appliqués à l'entrée x^t
U_a	Matrice des poids appliqués à a^{t-1}, l'état précédent
r^t	Vecteur de la porte de réinitialisation
$\odot$	Produit élément par élément (Hadamard)
b_a	Biais associé à l'état candidat
tanh	Fonction d'activation hyperbolique contraignant la sortie entre -1 et 1

— **La porte de mise à jour (update gate)** : elle permet de déterminer combien de l'état précédent doit être conservé et combien d'informations nouvelles doivent être intégrées pour produire le nouvel état (autrement dit : garder quoi et ajouter quoi).

Le résultat est un vecteur de valeurs entre 0 et 1, indiquant la part de l'état précédent qui sera conservée. L'équation 2.32 définit cette porte :

$$u^t = \sigma(W_u x^t + U_u a^{t-1} + b_u) \tag{2.32}$$

Ce vecteur est ensuite utilisé pour calculer **l'état actuel** a^t à l'instant t, selon l'équation 2.33 :

$$a^t = (1 - u^t) \odot \tilde{a}^t + u^t \odot a^{t-1} \tag{2.33}$$

Donc :

— Si $u^t \simeq 0$, alors $1 - u^t \simeq 1$: l'état candidat $\tilde{a}^t$ a une contribution majeure dans l'état actuel $a^t = \tilde{a}^t$. Le réseau accorde donc plus d'importance aux nouvelles informations.

— Si $u^t \simeq 1$, alors $1 - u^t \simeq 0$: l'état précédent a^{t-1} est majoritairement conservé $a^t = a^{t-1}$. Le réseau privilégie la rétention des informations passées par la cellule précédente.

Pour finir, le y^t et souvent égale à l'état caché final a^t, il peut être parfois transformé en raison du contexte.

2.4.6 Le rôle du GRU dans la gestion de la disparition de gradient

— La porte de réinitialisation r_t permet au modèle d'oublier une partie de l'ancien état et d'éviter des accumulations inutiles qui réduiraient les gradients.

— La porte de mise à jour u_t permet de conserver le gradient lorsqu'elle est proche de 1.

Si $u_t = 1$ alors :

$a_t = a_{t-1}$

Le gradient se propage directement sans diminution :

$$\frac{\partial L}{\partial a_t} = 1, \quad \frac{\partial L}{\partial a_{t-1}} = 1, \quad \frac{\partial L}{\partial a_{t-2}} = 1, \quad \frac{\partial L}{\partial a_{t-3}} = 1$$

— Les connexions directes permettent une rétropropagation plus stable et un meilleur apprentissage des longues séquences en transférant directement les informations dont le $u_t = 1$, ce qui conserve mieux le contexte et prédit correctement .

2.4.7 Construction du modèle GRU

Les différentes étapes du processus de construction du modèle GRU sont présentées ci-dessous :

2.4.7.1 Préparation des données

Cette étape consiste à importer les données selon leur format initial qui seront convertis en une représentation de données tabulaires, pour s'adapter aux exigences du modèle.

2.4.7.2 Normalisation des données

Cette étape consiste à ramener les données sur une échelle définie, généralement entre 0 et 1 ou entre -1 et 1, en appliquant la technique du **Min-Max Scaling**. Cela permet d'assurer une distribution cohérente des caractéristiques et d'éviter qu'une variable à forte amplitude ne domine les autres.
Cette normalisation :

— Facilite l'apprentissage du modèle.
— Prévient l'explosion du gradient.
— Améliore les performances de convergence.

La formule de la mise à l'échelle min-max, définie pour un intervalle $[a, b]$, est donnée par l'équation 2.34 :

$$X_{\text{scaled}} = a + \frac{(X - X_{\min})(b - a)}{X_{\max} - X_{\min}} \tag{2.34}$$

TABLE 2.8 – Description des termes de la normalisation Min-Max

Symbole	Description
X	Valeur d'origine à normaliser
X_{scaled}	Valeur après normalisation
$X_{\min}$	Valeur minimale du jeu de données
$X_{\max}$	Valeur maximale du jeu de données
a	Borne inférieure de l'échelle cible (souvent 0 ou -1)
b	Borne supérieure de l'échelle cible (souvent 1)

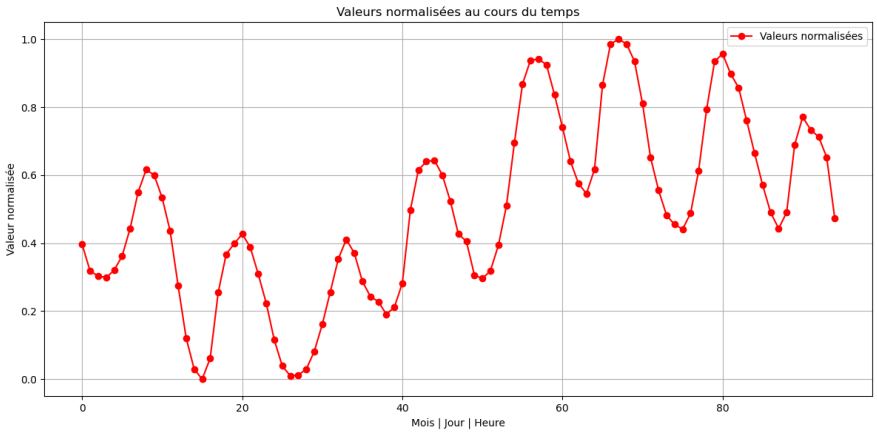


FIGURE 2.5 – Le graphique de la série étudiée après la normalisation sur un intervalle [0, 1]

Après la normalisation, les valeurs du graphe ont été redimensionnées pour varier entre 0 et 1, cela permet une meilleure comparaison et un traitement plus efficace par les algorithmes.

2.4.7.3 Création des séquences temporelles

Les réseaux GRU traitent les séries temporelles sous forme de séquences de valeurs et non de données individuelles, ce qui fait que les données sont structurées en séquences

temporelles, où X représente une fenêtre d'observations passées, chacune contenant le même nombre de valeurs passées pour prédire la suivante.
Cette structuration permet au modèle GRU de capturer les dépendances temporelles et les relations entre les éléments d'une séquence.

2.4.7.4 Division en ensembles d'entraînement et de test

Une fois les données préparées et structurées en séquences temporelles, elles sont réparties en deux ensembles distincts :

- **Ensemble d'entraînement** (70-80% des données) : utilisé pour ajuster les paramètres du modèle GRU pendant l'apprentissage.
- **Ensemble de test** (20-30% des données) : utilisé pour évaluer objectivement les performances et l'efficacité du modèle après son entraînement.

2.4.7.5 Définition du modèle GRU

La définition du modèle GRU est une étape de base pour l'entraînement de données, et c'est sa structure qui permet d'apprendre à comprendre et à traiter les données séquentielles.
Elle est définie ainsi :

- Une couche d'entrée qui reçoit les séquences de données
- Une ou plusieurs couches GRU, chacune contenant un nombre spécifique de cellules. Les calculs propres aux GRU y sont effectués. À la fin de chaque couche, à l'exception de la dernière, la totalité des états cachés générés par la séquence est transmise à la couche suivante. Pour la dernière couche, seul le dernier état caché est transmis.
- Une ou plusieurs couches de désactivation (Dropout) qui désactive un certain pourcentage de neurones pendant l'entraînement pour éviter le sur-apprentissage qui est un problème qui peut être rencontré lorsqu'un modèle apprend trop bien les détails des données d'entraînement, au point de perdre sa capacité de généralisation sur de nouvelles données. Pour y remédier le Dropout désactive à chaque itération, un sous-ensemble aléatoire de neurones, et les poids des autres sont ajustés pour compenser cette suppression.
- Une couche de sortie qui a pour fonction de transformer la dernière sortie du réseau et la réduire en une unique valeur, qui constitue la prédiction finale.

2.4.7.6 Compilation du modèle

Cette étape prépare le modèle à apprendre en définissant **la fonction de coût** pour évaluer la performance du modèle ainsi que **l'optimiseur** qui est la fonction permettant de mettre à jour les poids du modèle pour minimiser la perte en utilisant la descente de gradients.

2.4.7.7 Entraînement du modèle

C'est l'étape où le modèle apprend à faire des prédictions correctes en ajustant ses poids et biais à partir des données d'entraînement et de leurs résultats. Dans cette partie, le nombre de fois où le modèle réapprend les données est définit pour une meilleure mémorisation et une meilleure généralisation.
Pour une meilleure performance le modèle GRU reçoit en entrée une série de séquences qui seront traitées en parallèle dont le nombre est aussi déterminé lors de cette étape.

2.4.7.8 Évaluation et prédiction

Voici le résultat de la prédiction sur les dernières 24h.

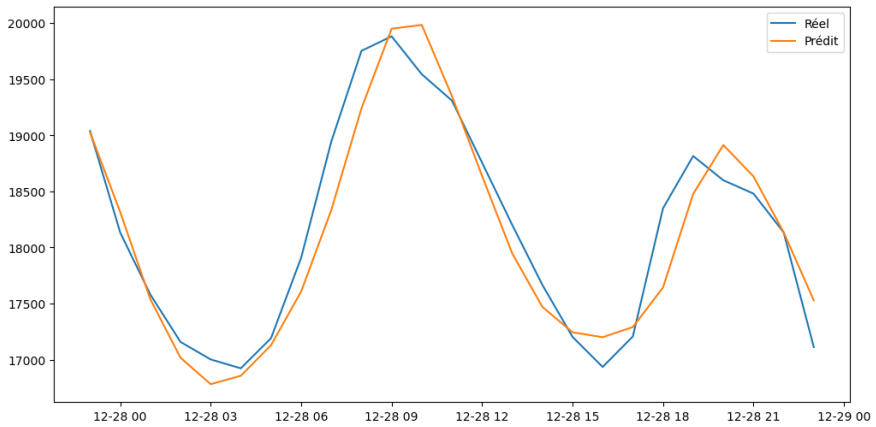


FIGURE 2.6 – Prédiction sur les dernières 24h de la série étudiée avec GRU

Nous observons une bonne correspondance générale entre les valeurs réelles et la prédiction, avec quelques écarts visibles aux pics et aux creux. Cela montre que le modèle de prédiction capte bien la tendance globale.

2.4.7.9 Hyperparamètres du GRU

Les méthodes d'apprentissage automatique notamment le GRU se basent sur plusieurs hyperparamètres que l'on peut ajuster afin d'optimiser la performance du modèle. Ils impactent fortement la capacité du modèle à apprendre et sa généralisation sur de nouvelles données ou sur un plus grand nombre de données. Nous distinguons deux types :

— **Les hyperparamètres du modèle**
Ce sont ceux qui contrôlent la structure du modèle GRU ou son comportement interne.

1. **units** : un modèle peut avoir plusieurs couches et units représente le nombre de cellules GRU dans la couche, Une série plus complexe nécessite la sollicitation d'un plus grand nombre d'units, cependant un trop grand nombre d'units peut entraîner un surapprentissage.
2. **activation** et **reccurent_activation** : qui sont respectivement la fonction d'activation des sorties du GRU et la fonction d'activation utilisée dans les portes du GRU. Par défaut, la fonction tangente hyperbolique est attribuée au paramètre activation et la fonction sigmoïde pour reccurent_activation
3. **dropout** : permet de désactiver aléatoirement un certain pourcentage de neurones.

— **Les hyperparamètres de l'entraînement**
Ce sont ceux qui controlent le processus d'apprentissage, ils influencent la vitesse et la stabilité de l'entrainement.

1. **loss** : c'est la fonction de perte, le choix de cet hyperparamètre dépend du problème à résoudre, si il s'agit d'une régression, mean_absolute_error (MAE) et mean_squared_error (MSE) sont recommandées sinon si il s'agit d'une classification d'autres fonctions sont proposées.

2. **optimizer :** l'optimiseur est l'algorithme qui décide comment mettre à jour les poids du réseau pour réduire l'erreur(la loss), c'est une version intelligente de la descente de gradient, dans le cas de GRU l'optimiseur le plus utilisé est adam (Adaptive Moment Estimation)
3. **batch_size** : c'est le nombre d'échantillons de données d'entraînement que le modèle traite avant de mettre à jour ses poids.
   Un batch_size petit donne un apprentissage long mais précis contrairement a une valeur trop grande qui donne un résultat moins précis mais plus rapide.
4. **epochs** : sa valeur détermine combien de fois le modèle passe sur toutes les données d'entraînement. Le modèle risque un surapprentissage si la valeur de epochs est trop grande.

C'est principalement en modifiant les valeurs des hyperparamètres que le modèle GRU s'adapte afin de donner de meilleurs résultats

Pour les résultats de prédiction ci dessus voici les hyperparamètres utilisés :

TABLE 2.9 – Paramètres du modèle LSTM

Paramètre	**Valeur**
units	64
activation	tanh
recurrent_activation	sigmoid
dropout	0.4014
loss	MSE
optimizer	adam
batch_size	16
epochs	36

2.5 Conclusion

Ce chapitre a permis de comprendre les bases de L'IA et du DP. Nous avons présenté les principaux types de réseaux de neurones, en mettant l'accent sur les RNN et les GRU, qui sont bien adaptés à l'analyse de données chronologiques. Ces connaissances serviront à appliquer concrètement les modèles de DP à la prédiction de la consommation d'énergie dans les prochains chapitres.

Chapitre 3

Analyse des besoins et conception

3.1 Introduction

L'analyse des besoins est le processus de compréhension des besoins des utilisateurs et des parties prenantes, tandis que la conception est le processus de création d'une solution pour répondre à un ou des besoins identifiés, ces deux étapes sont complémentaires et liées : L'analyse des besoins alimente la conception en informations et la conception peut révéler de nouveaux besoins ou des contraintes qui nécessitent une révision de l'analyse, c'est pour cela que ce sont des étapes fondamentales dans le processus de création d'une solution pour une problématique identifiée.

3.2 Présentation de l'entreprise d'accueil

FIGURE 3.1 – Logo de la Sonelgaz Direction Distribution de BÉJAIA

SONELGAZ est l'opérateur historique de la fourniture d'énergie électrique et gazière en Algérie. Fondée en 1969, Sonelgaz a servi le citoyen algérien pendant plus d'un demi-siècle, fournissant une source d'énergie essentielle à la vie quotidienne. Avec la promulgation de la loi sur l'électricité et la distribution de gaz par canalisations, Sonelgaz est passée d'une entreprise verticalement intégrée à une holding pilotant un groupe industriel composé de multiples sociétés et métiers.
Sonelgaz a toujours joué un rôle crucial dans le développement économique et social du pays.
Sa contribution à la politique énergétique nationale est illustrée par les importants programmes réalisés en électrification rurale et distribution publique de gaz, permettant

d'atteindre un taux de couverture en électricité de 98 taux de pénétration du gaz de 65 gaz se compose de 14 sociétés filiales gérées directement par la holding et de 12 sociétés en participation avec des tiers[42].

3.2.1 Transformation de SONELGAZ en groupe

À partir de 2004, le processus de transformation de Sonelgaz, lié à la loi n° 02-01 du 5 février 2002, a permis non seulement l'ouverture de la production d'électricité à la concurrence, mais aussi la séparation des fonctions de production, de transport et de distribution de l'électricité et du gaz en filiales distinctes sous forme de sociétés par actions (SPA). Ce processus a conduit à la création de nouvelles filiales à partir de 2004 [42].

En 2004, les nouvelles entités créées incluent :

- Une société de transport du gaz : le gestionnaire du réseau de transport du gaz (GRTG).
- Une société de production regroupant les centrales existantes : Sonelgaz Production de l'Électricité (SPE).
- Une société de transport de l'électricité, gestionnaire transitoire du système de production-transport : le gestionnaire du réseau de transport de l'électricité (GRTE) [42].
- La première modification des statuts du Fonds des œuvres sociales et culturelles (FOSC).

En 2005, plusieurs entités et réformes importantes ont vu le jour :

- Création de la Commission de Régulation de l'Électricité et du Gaz (CREG), un organisme indépendant et autonome doté de la personnalité juridique, avec trois missions principales : assurer et contrôler le service public, conseiller les pouvoirs publics sur l'organisation et le fonctionnement des marchés de l'électricité et du gaz, et surveiller le respect des lois et règlements relatifs à ces marchés [42].
- Création de la Société Civile de Médecine du Travail (SMT).
- Création d'une société de recherche et développement de l'électricité et du gaz (CREDEG).
- Fusion des quatre sociétés de Maintenance et de Prestations de Véhicules (MPV) en une seule entité.
- Fusion des trois sociétés de Maintenance des Transformateurs en une seule Société (SMT).
- Préparation à la filialisation de la distribution avec la création de quatre directions générales régionales [42].

En 2006, quatre nouvelles sociétés de distribution ont été créées :

1. Sonelgaz Distribution Alger (SDA)
2. Sonelgaz Distribution Centre (SDC)
3. Sonelgaz Distribution Est (SDE)
4. Sonelgaz Distribution Ouest (SDO)

En 2006 :

- Mise en place de la société opérateur système production-transport : OS, SPT.
- Nouvelle modification des statuts du Fonds des œuvres sociales et culturelles.
- Réintégration des cinq sociétés de travaux : KAHRAKIB, KANAGAZ, KAHRIF, ETTERKIB et INERGA [42].

En 2007 :
— Création de l'Institut de Formation en Électricité et Gaz (IFEG).
— Réorganisation de la SPE en quatre pôles nationaux de production d'électricité, dont un pôle Sud.
En 2009 :
— Création de la Société de Gestion du Patrimoine Immobilier (SOPIG).
— Création de la Compagnie d'Engineering de l'Électricité et du Gaz (CEEG).
— Création de la Société des Systèmes (ELIT).

3.2.2 Les différents types de clients de Sonelgaz

Organisation de la Sonelgaz Distribution Direction de Bejaia Les types de clients sont définis en fonction du niveau de tension (basse tension ou haute tension) qui leur est attribué [42].Nous nous distinguons parmi eux :

— Clients HTB (Haute Tension B) : Ces clients utilisent des lignes HTB qui constituent le réseau de répartition ou d'administration régionale, permettant le transport de l'électricité à l'échelle régionale ou locale. Ces lignes, avec une tension de 63 kV ou 90 kV, desservent les industries lourdes et les grands consommateurs électriques tels que les transports ferroviaires, et se connectent avec le réseau secondaire.
— Clients HTA (Haute Tension A) : Ces clients bénéficient des lignes HTA qui transportent l'électricité à l'échelle locale vers les petites industries, PME et commerces. Ces lignes ont une tension comprise entre 15 kV et 30 kV.
— Clients BT (Basse Tension) : Ces clients utilisent les lignes BT, les plus petites du réseau, avec une tension de 230 V ou 400 V. Ces lignes alimentent les ménages et les artisans, fournissant l'énergie nécessaire pour les appareils ménagers et autres usages quotidiens.

3.2.3 Présentation de l'organisme d'accueil (Sonelgaz Distribution – Direction de Béjaïa)

La direction de distribution de Béjaïa est rattachée à la Société Algérienne de Distribution de l'Électricité et du Gaz de l'Est (SDE), dont le siège est situé à Constantine. Cette direction est composée des unités suivantes :

— Le secrétariat
— Les assistants du Directeur de Distribution
— Le chargé des affaires juridiques
— Le chargé de la communication
— Le chargé de la sécurité
— 9 divisions et 10 agences commerciales : Béjaïa Cité Tobal, Béjaïa 4 Chemins, El Kseur, Amizour, Kherrata, Sidi Aïch, Seddouk, Akbou, Tazmalt, Aokas
— 5 districts : Béjaïa, Akbou, Sidi Aïch, Amizour, Kherrata

3.2.4 Organisation de la Sonelgaz Distribution – Direction de Béjaïa

La structure organisationnelle des différentes divisions de la Direction de Sonelgaz à Béjaïa est présentée dans la figure 3.2.

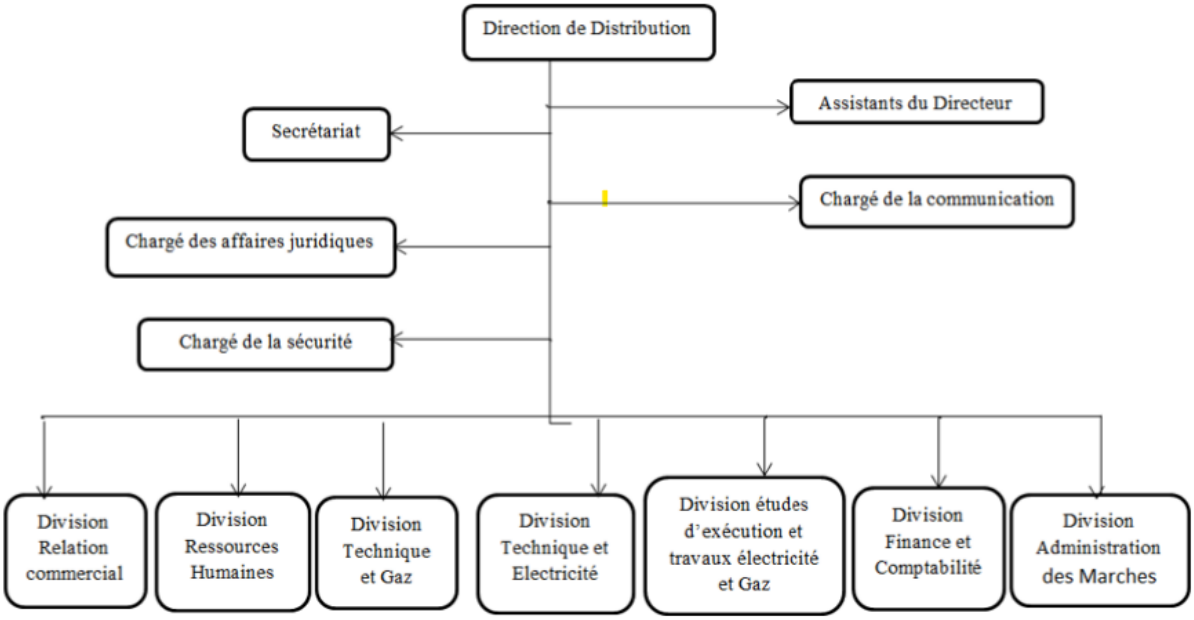


FIGURE 3.2 – Organigramme de la Sonelgaz Distribution – Direction de Béjaïa

3.2.4.1 Division Relations Commerciales

Cette division joue un rôle crucial au sein de la Direction de Distribution [42], avec pour principales missions :

— La vente d'électricité et de gaz.
— La gestion des relations clients.
— La facturation sur mémoire (FSM).
— L'analyse statistique (taux de pertes).
— L'assurance de la qualité de service et la satisfaction client.

Elle est subdivisée en trois services :

3.2.4.2 Service Développement des Ventes (RCN) :

— Prise en charge de toutes les demandes de raccordements des clients en électricité.
— Orientation et conseil des clients MT/MP concernant les modes de raccordement et la détermination du niveau de tension/pression.
— Facturation des devis RCN et de toutes les prestations.
— Établissement des ordres d'exécution des travaux après paiement des devis par les clients.

3.2.4.3 Service Recouvrement :

— Gestion des comptes clients IIT/IIP, MT/MP, BT/BP.

— Respect des délais de facturation pour les groupes en HT/IIP, MT/MP, BT/BP.
— Avertissement des clients en cas de retard de paiement des factures (ordres de coupure, mise en demeure, etc.).
— Contrôle et archivage de toutes les pièces comptables.
— Suivi des clients grands comptes.
— Suivi de la facturation et du recouvrement FSM (facturation sur mémoire).

3.2.4.4 Service Grands Comptes :

— Gestion, télérelève et facturation mensuelle des clients MT/MP, HT/HP.
— Établissement de contrats de 5 ans à signer avant la mise en service d'un poste client (livraison) MT/MP.
— Veille à la qualité de service.

Notre stage s'est déroulé au sein du **Service Grands Comptes**, dont la mission principale est la gestion et le suivi des clients à haute ou moyenne tension. Ce service traite des données de consommation mensuelle, ce qui a permis de développer une application web de prédiction à l'aide de modèles de séries temporelles comme SARIMA, SARIMAX et GRU.

3.2.5 Problématique

L'électricité est un pilier du développement économique mondial. Elle devient de plus en plus essentielle avec l'industrialisation, les progrès technologiques et la croissance continue de la demande énergétique. Face à ces évolutions, les entreprises du secteur, comme Sonelgaz, sont confrontées à des défis majeurs liés à la production, la gestion et la distribution de l'électricité.

Lors de notre stage à la Direction de Distribution de Sonelgaz à Béjaïa, nous avons identifié plusieurs problématiques clés entravant l'efficacité de l'entreprise :

— **Sous-production d'électricité :** la production reste insuffisante pour satisfaire les besoins croissants des clients, causant un déséquilibre entre l'offre et la demande.
— **Pertes d'énergie :** l'électricité étant non stockable, une part importante est perdue durant le transport et la distribution, impactant le rendement global du réseau.
— **Fraude et consommation non mesurée :** certains clients manipulent les instruments de comptage, rendant difficile une facturation équitable et fiable.
— **Gestion des clients et manque d'outils d'aide à la décision :** la gestion manque de numérisation et d'outils avancés, ce qui complique l'optimisation des ressources.

Dans ce contexte, quelques travaux antérieurs ont été menés au sein de Sonelgaz pour aborder cette problématique en exploitant diverses techniques d'intelligence artificielle. Le premier travail de [24] a proposé le développement d'une application web dédiée à la prévision des ventes d'électricité, en s'appuyant sur une approche combinée utilisant les modèles LSTM et SARIMA, contribuant ainsi au renforcement des capacités de planification stratégique de l'entreprise. Par ailleurs, [25] a exploré une architecture hybride combinant LSTM et GRU, démontrant une amélioration significative de la précision de prévision par rapport aux méthodes classiques. Plus récemment, [15] a développé une approche intégrant les modèles Transformers, LSTM et la régression linéaire pour optimiser la prédiction de la consommation électrique. L'ensemble de ces études convergent vers la confirmation de l'apport considérable de l'intelligence artificielle dans la gestion énergétique et mettent en évidence la nécessité d'outils performants pour améliorer l'efficacité opérationnelle du secteur électrique.

3.2.6 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 éléctriques 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.

3.3 Analyse des besoins

3.3.1 Objectif de l'application

L'objectif principal de l'application est de fournir un outil de prédiction permettant aux entreprises d'anticiper leurs résultats sur différentes périodes en fonction de données historiques et cela à l'aide d'algorithmes avancés tel que les algorithmes de séries temporelles basés sur des modèles statistiques (SARIMA , SARIMAX) et les algorithmes d'apprentissage profond basé sur le GRU, la plateforme sélectionnera le modèle le plus adapté pour offrir les meilleurs prévisions afin d'aider les entreprises à mieux gérer leurs ressources, leur budget et leur production.

3.3.2 Identification des utilisateurs et de leurs besoins

— **Les utilisateurs**

1. L'administrateur de l'entreprise Sonelgaz : responsable de la gestion des clients, des données de consommation, des statistiques, etc.
2. Les clients : les consommateurs d'énergie ayant accès à leur propre profil, leur historique de consommation, et éventuellement des prévisions.

— **Les besoins fonctionnels**

— Coté client :

1. Authentification des clients (consommateurs) à l'aide de leur adresse e-mail et de leur mot de passe.
2. Affichage des consommations historiques des utilisateurs et de la prédiction du trimestre en cours ou du prochain sous forme de graphiques.
3. Accès au profil personnel avec possibilité de modifier certaines informations.

— Coté administrateur :

1. Authentification de l'administrateur de Sonelgaz à l'aide de l'adresse e-mail et du mot de passe.

2. Visualisation des prédictions de la production d'électricité de l'entreprise sous forme de graphiques et de tableaux, accompagnées des caractéristiques des modèles utilisés et de leurs performances.
3. Gestion complète des clients : ajout, suppression, modification des informations, visualisation de la consommation, et génération de prédictions futures.
4. Consultation de toutes les anciennes prédictions accompagnées d'une comparaison avec les données réelles.

— **Les besoins non fonctionnels**

1. Sécurité de l'authentification.
2. Temps de réponse rapide pour l'affichage des résultats des méthodes statistiques.
3. Résponsivité de l'application web.
4. Une interface utilisateur claire et facile à utiliser.

— **Fonctionnalités principales**

1. Gestion des utilisateurs.
2. Prédiction et analyse des données.

— **Contraintes**

1. L'authentification doit être sécurisée.
2. Le site doit pouvoir gérer de grands volume de donnees.
3. Trouver une solution pour limiter le temps de l'entrainement des modèles de prédiction.
4. La protection des données de l'entreprise.

3.4 Conception

3.4.1 La méthodologie agile

Une méthode agile est une approche flexible de gestion de projet informatique, basée sur des cycles courts et itératifs, permettant une livraison continue de fonctionnalités et une adaptation rapide aux besoins du client.

Les méthodes agiles reposent sur des principes fondamentaux tels que la simplicité, la communication et l'adaptabilité, favorisant un développement itératif et incrémental pour mieux répondre aux évolutions des besoins[28].
Parmi les méthodes agiles existantes **SCRUM** est utilisée pour superviser et contrôler l'avancement du travail.

3.4.2 Scrum

Scrum est une méthode agile de gestion de projets complexes. Elle repose sur des cycles de développement itératifs et incrémentaux, appelés **sprints**, favorisant la collaboration et l'adaptabilité. Inspiré des travaux de Takeuchi et Nonaka (1986), il a été formalisé dans les années 1990 par Schwaber et Sutherland, devenant un standard des approches agiles modernes[40].

3.4.2.1 Répartition des rôles en Scrum

Chaque projet utilisant **Scrum** s'organise autour d'une équipe auto-organisée et multifonctionnelle. Trois rôles principaux structurent cette organisation :

- **Product Owner :** est un acteur clé de la méthodologie Scrum, dont le but est de maximiser la valeur du produit. Il élabore et hiérarchise les fonctionnalités en fonction des besoins des utilisateurs et des parties prenantes. En étroite collaboration avec l'équipe de développement, il s'assure que les exigences sont clairement comprises et mises en œuvre de manière efficace[40].
- **Scrum Master :** est le garant de l'application de Scrum et un facilitateur pour l'équipe. Il élimine les obstacles, organise les événements Scrum et favorise l'auto-organisation. En tant que leader-serviteur, il soutient l'équipe et le Product Owner, encourageant la collaboration et l'amélioration continue afin d'assurer une livraison efficace et régulière de valeur[40].
- **Développeurs :** forment une équipe autonome et collaborative, responsable de la création et de la livraison des fonctionnalités du produit à chaque sprint. Ils transforment le Backlog Produit en valeur, s'adaptent aux évolutions et garantissent la qualité et le succès du sprint[40].

3.4.2.2 Les Étapes et Événements de Scrum

- **Planification du Sprint :** le processus commence par la planification du Sprint, où le Product Owner, en collaboration avec l'équipe de développement, définit l'objectif du Sprint et sélectionne les éléments prioritaires du Product Backlog à développer. Cette réunion, d'une durée pouvant aller jusqu'à 8 heures pour un Sprint d'un mois, permet de clarifier les attentes, d'aligner les priorités et d'estimer la charge de travail nécessaire[40].
- **Exécution et Suivi Quotidien :** pendant le Sprint, l'équipe de développement progresse sur les tâches définies et organise quotidiennement une mêlée quotidienne (Daily Scrum), une réunion de 15 minutes maximum. Cet échange permet de synchroniser les activités, d'identifier les obstacles et d'ajuster le plan d'action afin d'assurer l'avancement du Sprint[40].
- **Revue du Sprint :** à la fin du Sprint, une revue de Sprint est organisée avec les parties prenantes afin d'inspecter l'incrément développé et de recueillir des retours. Cette réunion, pouvant durer jusqu'à 4 heures pour un Sprint d'un mois, permet d'évaluer le travail réalisé et d'ajuster le Product Backlog en fonction des évolutions, des nouvelles priorités et des besoins identifiés[40].
- **Rétrospective du Sprint :**Après la revue du Sprint, l'équipe Scrum tient une rétrospective de Sprint, un moment clé pour analyser son fonctionnement, identifier des axes d'amélioration et renforcer l'efficacité collective. Cette réunion, d'une durée maximale de 3 heures pour un Sprint d'un mois, aboutit à un plan d'actions visant à optimiser les prochains Sprints et à favoriser une meilleure collaboration[40] .

Scrum suit un cycle itératif et incrémental, fondé sur les principes de transparence, d'inspection et d'adaptation, afin d'assurer une amélioration continue du processus et des livrables.

3.4.2.3 Les Artefacts de SCRUM

- **Le Product Backlog :** ou Carnet de Produit, est une liste hiérarchisée et évolutive de toutes les tâches possibles dans un projet Scrum. Géré par le Product

Owner, il s'adapte en permanence aux retours de l'équipe de développement, aux changements du marché et aux besoins des clients. Cette liste constitue la base de la planification des sprints, avec les éléments les plus prioritaires qui sont choisis pour être traités en premier.

— **Le Sprint Backlog :** ou Carnet du sprint, est un plan détaillé des tâches que l'équipe de développement s'engage à accomplir pendant un sprint pour arriver à l'objectif fixé. Il est composé des éléments choisis du Product Backlog et est entièrement géré par l'équipe, qui l'adapte jour après jour pour suivre et ajuster le progrès des travaux.
— **L'Incrément :** est l'ensemble des éléments du Product Backlog terminés pendant ce sprint, et les précédents terminés pendant les sprints antérieurs.

3.4.2.4 Répartition des rôles

— **Product Owner : M. Dalil HADJOUT** qui est responsable de maximiser la valeur du produit en définissant les spécifications fonctionnelles, en priorisant les besoins, en validant les fonctionnalités développées et en assurant une bonne compréhension des exigences par l'équipe de développement.
— **Scrum Master : M. Foudil MIR** qui est le facilitateur entre le Product Owner et l'équipe de développement. Il veille à une communication efficace, supervise les activités de l'équipe, encourage la collaboration et s'assure du respect de la méthode Agile tout au long des sprints. Il accompagne également l'amélioration continue de la performance de l'équipe.
— **Développeurs :** l'équipe de développement est composée de Melle Nora BELAID et Melle Aicha Manel CHAFFI, responsables de la création et de la livraison des produits ou services à chaque sprint. Elles assurent également la réalisation des User Stories afin de répondre efficacement aux besoins des clients.

3.4.2.5 Le Product backlog

ID	Nom du User Story	Description du User Story	Priorité	Statut
US1	Authentification	En tant qu'administrateur, je souhaite pouvoir m'authentifier afin d'accéder aux différentes fonctionnalités du système. – En tant que client, je souhaite pouvoir m'authentifier afin d'accéder aux différentes fonctionnalités du système.	Haute	Fait
US2	Lancer une prédicition	En tant qu'administrateur, je souhaite prédire la quantité d'énergie future à produire. –En tant qu'administrateur, je souhaite pouvoir sauvegarder le modèle de prédiction avec ses paramètres, afin de le réutiliser ultérieurement. –En tant qu'administrateur, je souhaite consulter les anciennes prédictions effectuées ainsi que ses paramètres afin de les comparer avec les donnees réelles.	Haute	Fait
US3	Visualisation des consumations/prédicitions	En tant que client, je souhaite pouvoir visualiser mes consommations sous forme de graphique. –En tant que client, je souhaite pouvoir visualiser les prédictions de ma consommation future, ainsi qu'estimer le montant que je vais payer au prochain trimestre. –En tant que client, je souhaite pouvoir exporter les résultats de mes prédictions.	Haute	Fait

ID	Nom du User Story	Description du User Story	Priorité	Statut
US4	Gestion des comptes clients	En tant qu'administrateur, je souhaite pouvoir ajouter de nouveaux clients et en supprimer, afin de gérer efficacement la base de données des utilisateurs. –En tant qu'administrateur, je souhaite pouvoir modifier les informations d'un client, afin de maintenir les données à jour. –En tant qu'administrateur, je souhaite pouvoir visualiser les consommations et les prédictions de chaque client, afin de suivre leur activité énergétique..	Moyenne	Fait
US5	Gestion de profil	En tant que client, je souhaite pouvoir modifier mes informations personnelles (comme mon nom, mon adresse ou mon mot de passe). –En tant qu'administrateur, je souhaite pouvoir modifier les informations de l'entreprise (comme son nom, son adresse ou le mot de passe).	Faible	Fait

Table 3.1 – Product Backlog

3.4.2.6 Planification des sprints

Les User Stories du Product Backlog sont classées par priorité pour implémenter en premier les fonctionnalités à forte valeur ajoutée. Le développement est organisé en sprints, des itérations courtes (2 à 4 semaines) permettant de livrer des versions utilisables du produit. Chaque sprint a un objectif précis et débute dès la fin du précédent. La planification des sprints est initialement établie, mais elle peut être ajustée selon la vélocité de l'équipe.

Table 3.2 – Planification des sprints

Sprint	Nom du sprint	Période
Sprint 1	Authentification et gestion de profil	2 semaines
Sprint 2	Visualisation des consommations	1 semaine
Sprint 3	Lancer une prédiction	3 semaines
Sprint 4	Visualisation des prédictions	1 semaine
Sprint 5	Gestion des comptes clients	2 semaines

3.4.3 Diagrammes de conception

Les diagrammes de conception sont des représentations graphiques utilisées pour représenter la structure et le comportement d'un système logiciel avant sa réalisation. Dans le cadre de notre application, deux acteurs essentiels y sont représentés : l'administrateur et le client, chacun ayant des interactions spécifiques avec le système selon le type du diagramme.

3.4.3.1 Diagramme de contexte

Le diagramme de contexte est une représentation graphique de haut niveau d'un système, dans laquelle ce dernier est modélisé comme une seule entité interagissant avec des acteurs ou systèmes externes. Il permet de visualiser les flux d'informations entrants et sortants, et de définir les limites fonctionnelles du système sans en détailler la structure interne [41].

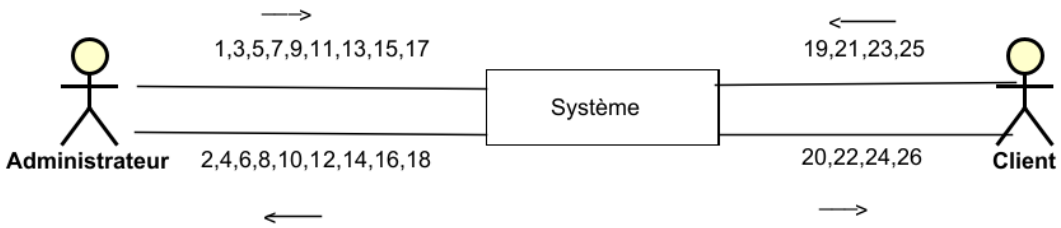


FIGURE 3.3 – Diagramme de contexte

Acteur	Numéro	Message	Numéro	Message
Administrateur	1	Demande d'authentification	2	Afficher le profil de l'administrateur
	3	Demande d'analyse statistique	4	Afficher le dashboard
	5	Demande d'accès qu coté client	6	Affichage de l'interface des clients inscrits
	7	Rechercher d'un client	8	afficher les informations du client recherché
	9	Demander l'ajout d'un client	10	Affichage du formulaire d'ajout
	11	Demande de visualisation des consommations et prédictions d'un client	12	Affichage du graphique contenant les consommations et prédictions du client
	13	Demande de prédiction	14	Afficher l'interface de prédiction.
	15	lancer une prédiction	16	Affichage de résultat de prédiction en graphe
	13	Demande d'accès à l'historique de prédiction	14	Affichage de l'interface historique
	15	Demande de téléchargement des resultats de prédiction en format pdf	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 des graphiques en pdf	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

3.4.4 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation (Use Case Diagram) est un diagramme UML qui permet de modéliser les interactions entre les acteurs externes (utilisateurs ou systèmes) et le système étudié, à travers des scénarios appelés cas d'utilisation. Il décrit les fonctionnalités attendues du système sous forme de comportements observables par les utilisateurs [4].

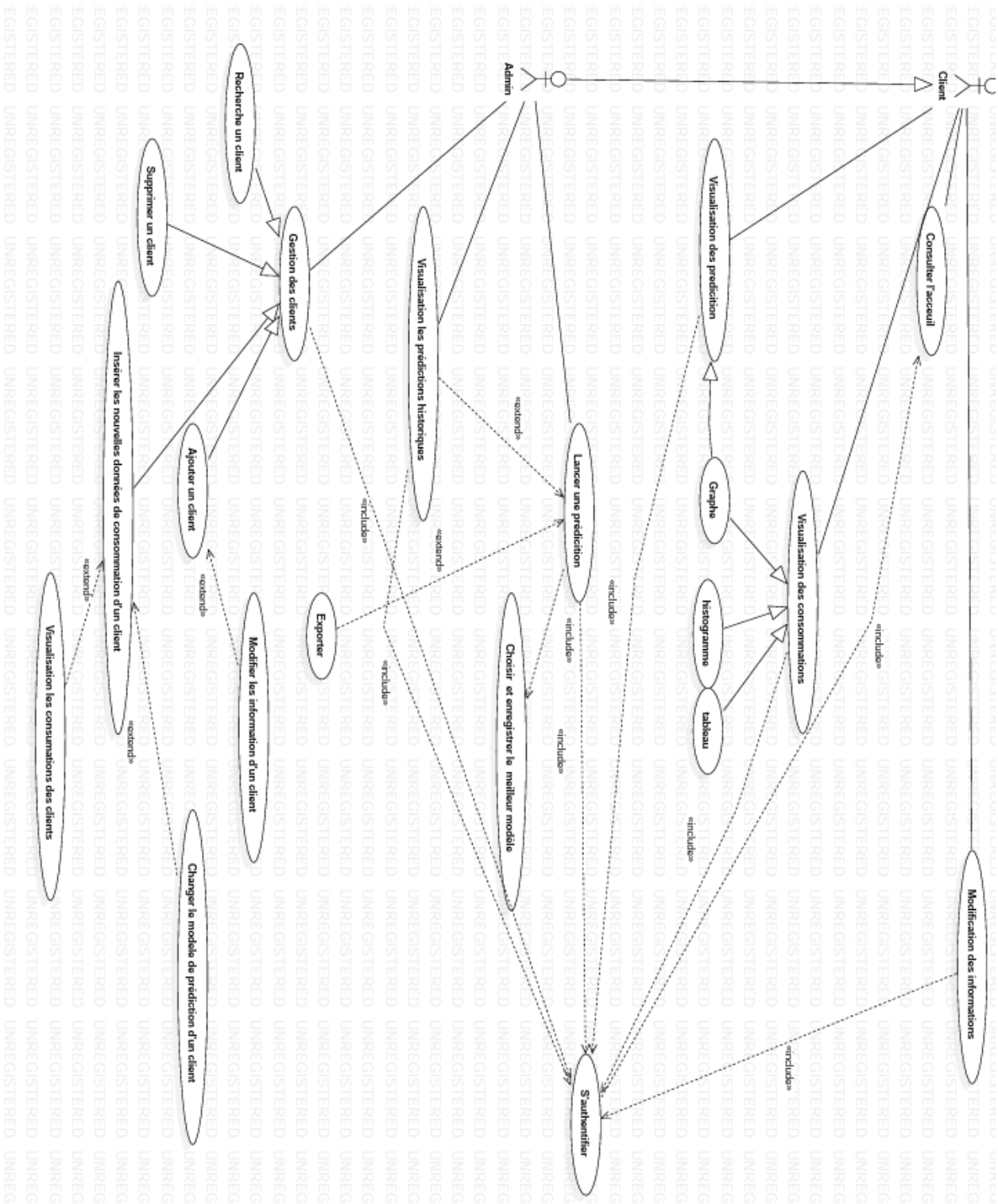


FIGURE 3.4 – Diagramme de cas d'utilisation

3.4.5 Diagrammes de séquences

Le diagramme de séquence est une description des interactions entre un acteur et les objets du système, mettant l'accent sur la chronologie des échanges de messages.

3.4.6 Diagramme de séquence détaillé

Les diagrammes de séquence détaillés offrent une représentation claire de l'ordre d'exécution des actions, des conditions et des boucles, ainsi que des appels de méthodes, des retours, et des interactions entre les différents objets du système[38].
Il démontre les interactions dynamiques entre les différents composants de l'application dans un scénario donné. Pour structurer ces composants de manière claire, nous avons mis en œuvre le modèle architectural **MVC** (Modèle – Vue – Contrôleur) ou "Boundary – Control – Entity" (BCE) en anglais, qui permet de dissocier la logique métier, la présentation et le contrôle des actions utilisateur.

- Contrôleur : il reçoit les actions de l'utilisateur via l'interface, traite les requêtes, appelle les modèles, et renvoie la réponse.
- Interface (Vue) : c'est l'élément avec lequel l'utilisateur interagit directement tel que les pages web ou les formulaires.
- Entité (Modèle) : Elle contient les données et la logique métier.

3.4.6.1 Diagramme de séquence détaillé du cas d'utilisation "S'authentifier"

Le diagramme de séquence détaillé du cas d'utilisation « Authentification » commence lorsque l'utilisateur remplit le formulaire de connexion avec son adresse e-mail et son mot de passe. Ensuite, une vérification est effectuée pour confirmer l'existence du compte.

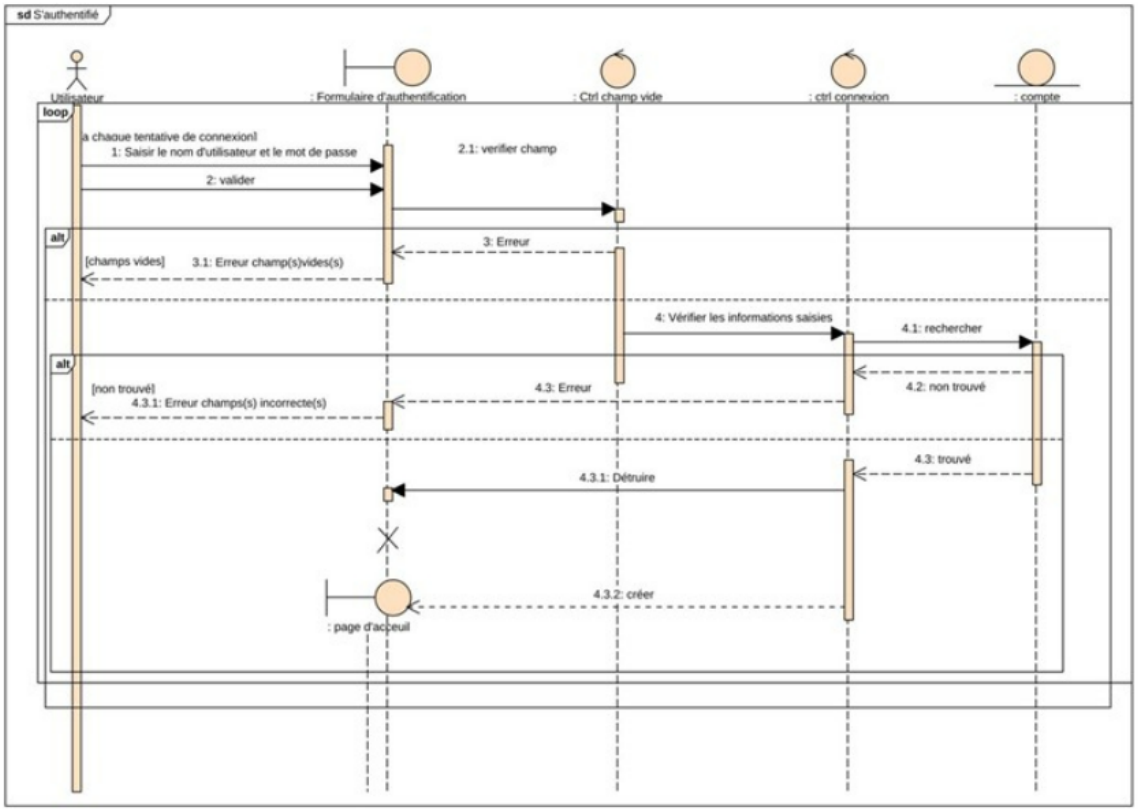


FIGURE 3.5 – Diagramme de séquence détaillé du cas d'utilisation "S'authentifier"

3.4.6.2 Diagramme de séquence détaillé du cas d'utilisation "Lancer une prédiction"

Le diagramme de séquence détaillé du cas d'utilisation « Lancer une prédiction » commence après l'authentification du client. Il décrit une séquence d'actions et d'échanges de messages entre les acteurs et les composants du système, incluant notamment la saisie de la période de prédiction, le déclenchement de l'entraînement, ainsi que toutes les étapes jusqu'à l'obtention du résultat final. Le scénario complet est représenté dans la figure suivante.

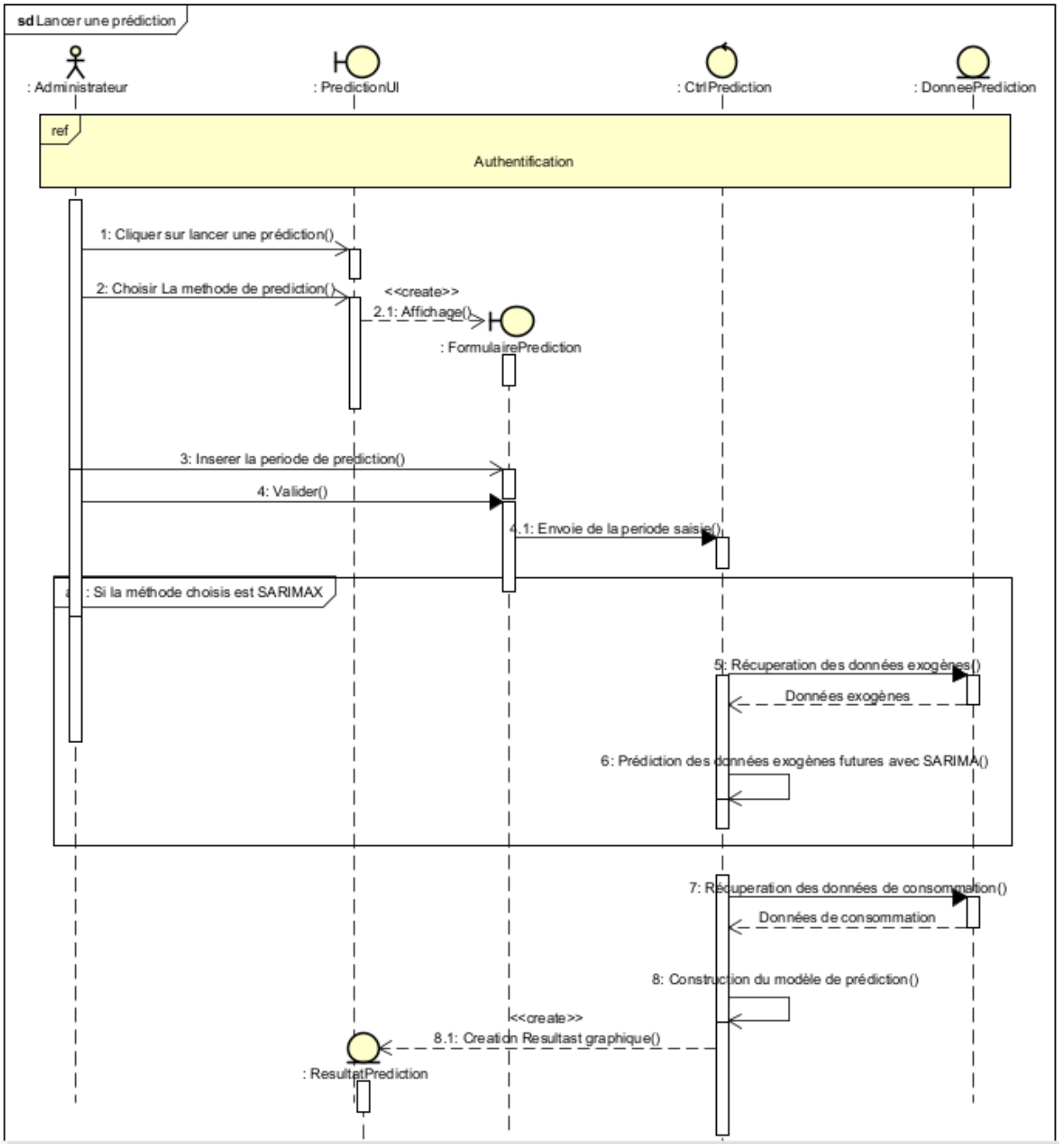


FIGURE 3.6 – Diagramme de séquence détaillé du cas d'utilisation "Lancer une prédiction"

3.4.6.3 Diagramme de séquence détaillé du cas d'utilisation "Visualiser une prédiction historique"

Le diagramme de séquence détaillé du cas d'utilisation «Visualiser une prédiction historique» illustre la séquence d'actions et les échanges de messages entre les acteurs et les composants du système lorsqu'un utilisateur souhaite consulter une prédiction antérieure. Il met en évidence les étapes clés, telles que la sélection de la date de prédiction, la récupération des données correspondantes, jusqu'à l'affichage détaillé des résultats.

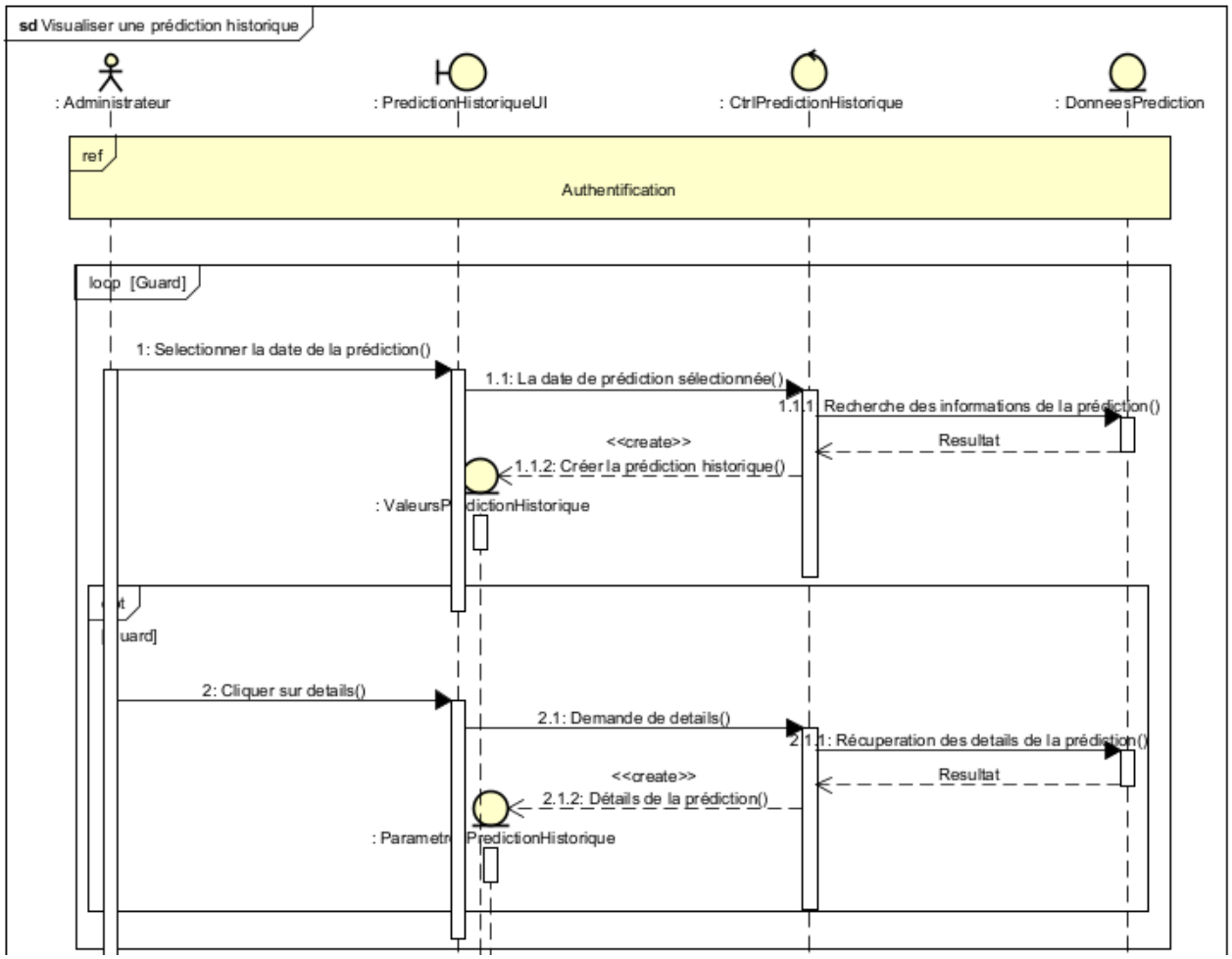


FIGURE 3.7 – Diagramme de séquence détaillé du cas d'utilisation "Visualiser une prédiction historique"

3.4.6.4 Diagramme de séquence détaillé du cas d'utilisation "Rechercher un client "

Le diagramme de séquence détaillé du cas d'utilisation «Rechercher un client» illustre la séquence d'actions et les échanges de messages entre les acteurs et les composants du système lorsqu'un utilisateur souhaite rechercher un client. Après avoir saisi le code du client dans la barre de recherche, le système récupère et affiche les informations complètes du client correspondant.

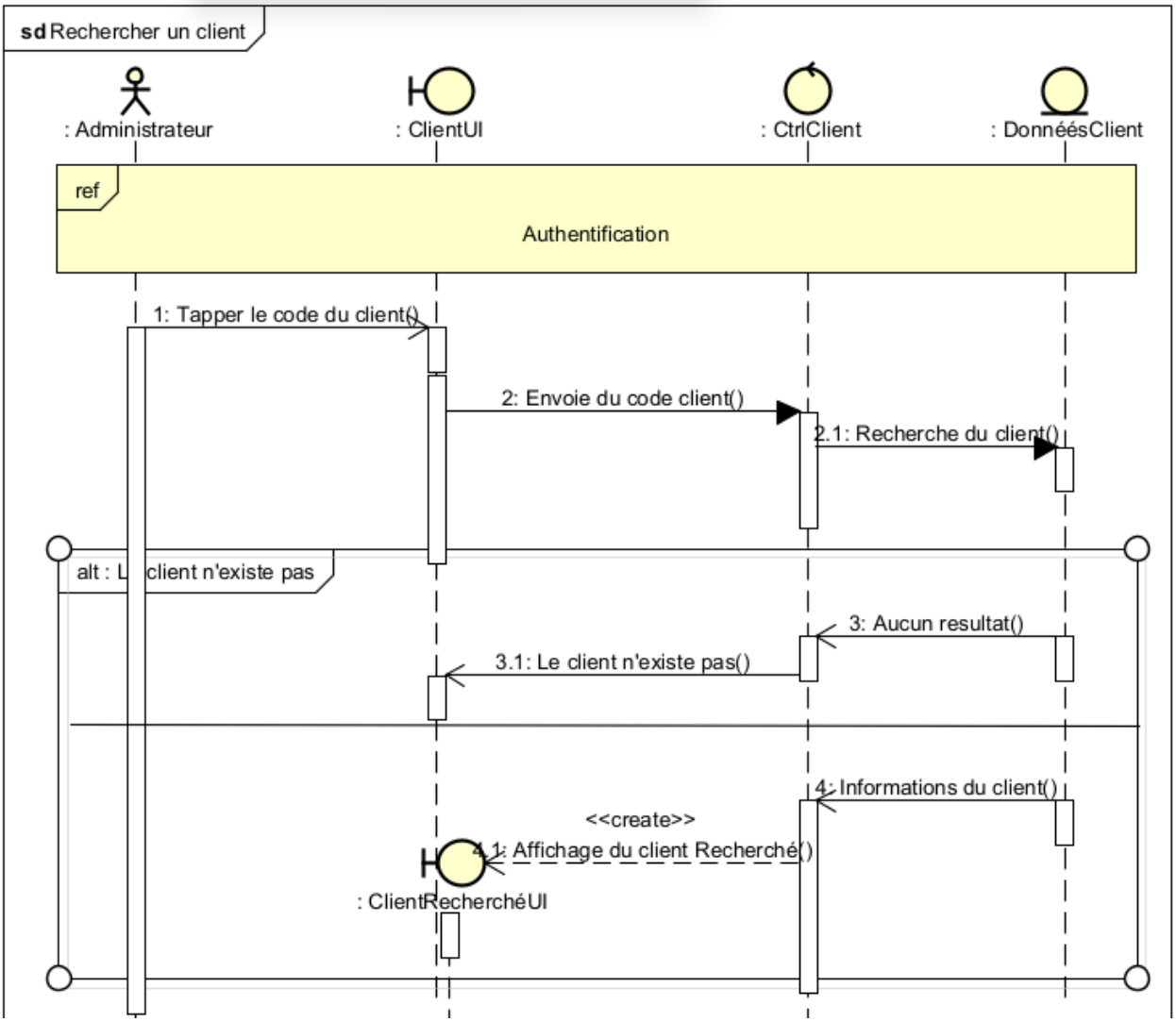


FIGURE 3.8 – Diagramme de séquence détaillé du cas d'utilisation "Rechercher un client"

3.4.6.5 Diagramme de séquence détaillé du cas d'utilisation "Insérer des nouvelle données de consommation d'un client"

Le diagramme de séquence détaillé du cas d'utilisation «Insérer de nouvelles données de consommation d'un client» illustre la séquence d'actions et les échanges de messages entre les acteurs et les composants du système lorsqu'un utilisateur souhaite ajouter de nouvelles données de consommation pour un client spécifique. Après avoir recherché ce client à l'aide de son code, les nouvelles données sont saisies via le formulaire de modification et enregistrées dans le système.

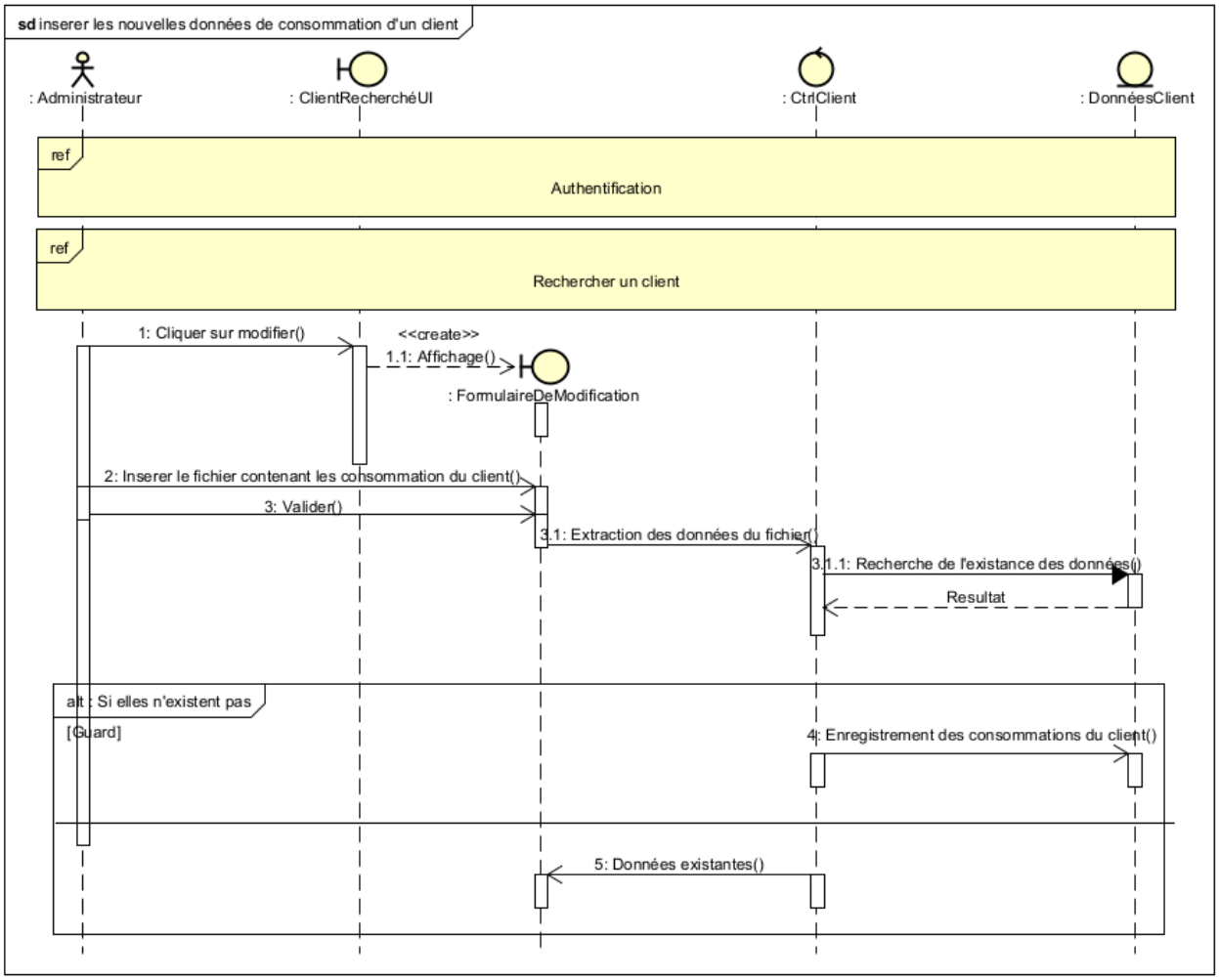


FIGURE 3.9 – Diagramme de séquence détaillé du cas d'utilisation "Insérer de nouvelles données de consommation"

3.4.7 Diagrammes de Classes

Un diagramme de classes est un diagramme structurel qui représente la structure statique d'un système. Il décrit les classes, leurs attributs, leurs méthodes ainsi que les relations entre elles (associations, généralisations, dépendances, etc.). Ce type de diagramme est essentiel pour la modélisation orientée objet, car il offre une vision claire de l'organisation et des interactions entre les composants du système[38].

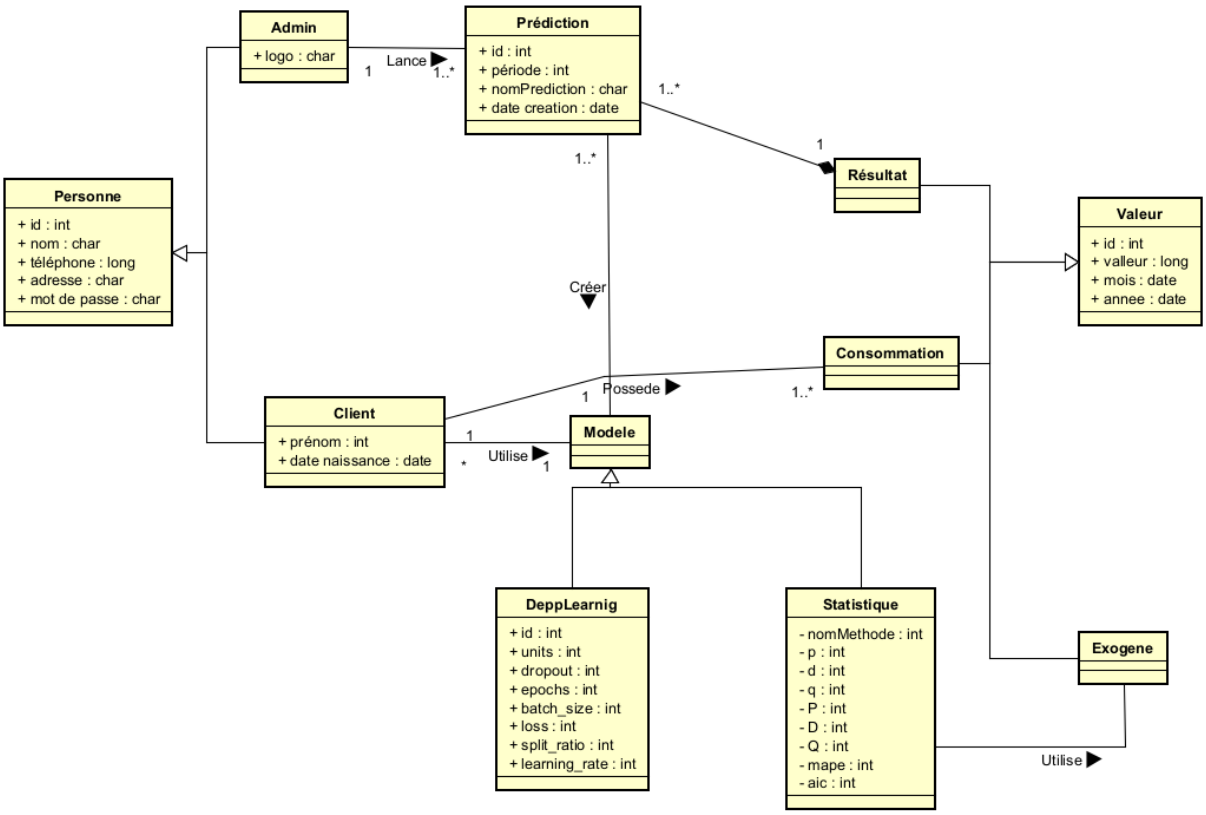


FIGURE 3.10 – Diagramme de classe

3.4.8 Conclusion

Dans ce chapitre, nous avons abordé plusieurs points essentiels pour l'identification et la structuration des principales fonctionnalités de notre système. Nous avons commencé par l'application de la méthode Scrum à notre projet, à travers la définition du backlog et des sprints. Ensuite, nous avons présenté les différentes étapes de développement à l'aide de plusieurs diagrammes, notamment des diagrammes de séquence détaillés. Ainsi, ce chapitre constitue une base solide qui facilitera la gestion et la mise en œuvre de la phase de réalisation abordée dans le chapitre suivant.

Chapitre 4

Réalisation

4.1 Introduction

Ce dernier chapitre est consacré à la partie pratique du projet. Cette phase de réalisation constitue l'étape concrète du développement de l'application, où les choix techniques et les spécifications fonctionnelles définis en amont sont traduits en code. Il s'agit de la dernière étape avant d'obtenir un produit final opérationnel. Elle marque le passage de la théorie à la mise en œuvre effective.

Il contiendra l'environnement de développement choisi, les bibliothèques et les outils utilisés ainsi que le résultat de l'application des différentes méthodes sur les données transmises par l'entreprise .Une présentation détaillée de l'application avec des captures d'écran suivra accompagnée d'une analyse comparative des résultats et d'une conclusion finale.

4.2 Architecture de l'application

L'architecture de l'application repose sur une séparation claire entre le client (interface utilisateur) et le serveur (logique métier et traitement de données), selon une architecture **client-serveur moderne**.

- Front-end (client) : développé avec React, il s'agit d'une interface web interactive et dynamique . Il expose à l'utilisateur la possibilité de se charger ses données, d'afficher les graphiques, les résultats de prédictions , et d'interagir avec l'application de manière fluide.
- Back-end (serveur) : mis en développement avec FastAPI, un framework Python rapide et efficace, il se charge d'accueillir l'appel du client, de traiter les séries temporelles (SARIMA, SARIMAX, GRU), de gérer les modèles de prédiction, et pour finir d'envoyer les résultats réponse JSON.
- Communication : front-end et back-end communiquent via API REST, permettant ainsi l'échange structuré de données (par exemple : envoi des fichiers, récupération des prédictions).
- Traitement des données : les modèles sont traités côté serveur, on renvoie ensuite les résultats au client afin de les visualiser.

4.3 Environnement de développement de l'application

4.3.1 Environnement Logiciel

- **Anaconda** : anaconda est une distribution open-source de Python spécialement conçue pour simplifier le développement en science des données et en apprentissage automatique. Cette plateforme intègre un ensemble complet de bibliothèques (NumPy, Pandas, Matplotlib, Scikit-Learn) et d'outils interactifs (Jupyter Notebook, Spyder), optimisant ainsi les workflows analytiques. Elle inclut également Conda, un gestionnaire de paquets puissant permettant d'installer des dépendances, de gérer des environnements virtuels et de garantir la reproductibilité des projets[46].
- **Jupyter Nootbook** : jupyter Notebook est un environnement de développement interactif permettant de créer et exécuter du code directement depuis un navigateur web. Il permet d'intégrer de manière cohérente du code (Python, R, etc.), des visualisations dynamiques (graphiques, tableaux) et des explications textuelles au sein d'un même document. Cette polyvalence en fait un outil incontournable pour la recherche en science des données, le développement de modèles de machine learning, et la pédagogie interactive[10].
- **Visual Studio Code (VSCode)** : visual Studio Code est un éditeur de code multiplateforme léger développé par Microsoft, réputé pour sa richesse fonctionnelle et son extensibilité. Il inclut le débogage, la coloration syntaxique, ainsi qu'un contrôle de version Git intégré. Grâce à un large éventail d'extensions, il permet de travailler avec presque tous les langages et frameworks, tout en restant léger .
- **HaSTA-UML** : HastaUML est un logiciel de modélisation UML destiné à l'apprentissage de la conception orientée objet. Il permet de créer des diagrammes UML, tels que les diagrammes de cas d'utilisation ou de classes, à travers une interface simple et adaptée au contexte pédagogique. Il est largement utilisé dans les établissements francophones pour initier les étudiants à la modélisation logicielle [37].
- **Invite de commande (cmd)** : C'est un outil qui permet d'interagir directement avec le système d'exploitation à travers des instructions textuelles. Dans le cadre de ce projet, il a été utilisé pour :
  - naviguer dans les dossiers du projet.
  - exécuter le serveur.
  - gérer les environnements virtuels.
  - installer les bibliothèques.
- **Git et GitHub** : Git a permit de suivre l'évolution du code source et d'assurer un bon historique des modifications. La plateforme GitHub a servi à héberger le code et à le partager entre les membres du binome, facilitant ainsi le travail collaboratif à distance.

4.3.2 Langages de programmation

- **Python** : python est un langage interprété, de haut niveau et multi-paradigme, optimisé pour une syntaxe claire et une productivité accrue. Flexible, il intègre les paradigmes procédural, orienté objet et fonctionnel. Son typage dynamique et sa gestion automatique de la mémoire en font un outil phare en développement web,

analyse de données, IA et automatisation. Son succès s'appuie sur une bibliothèque standard complète et un écosystème de frameworks adaptés aux besoins actuels [35].
Il est utilisé pour le traitement des données, l'implémentation des algorithmes de machine learning et de deep learning.

— **JavaScripts** : JavaScript est le langage du web : c'est le seul langage exécuté dans tous les navigateurs web, et le seul capable d'interagir directement avec la page web elle-même, c'est un langage de haut niveau, dynamique, non typé, et interprété, distinct de Java[21].
Il est utilisé pour le développement de l'interface utilisateur web.

4.3.3 Bibliothèques et Framworks utilisés

4.3.3.1 Front-End

— **React** : react est une bibliothèque JavaScript qui est souvent utilisée pour construire des interfaces interactives et dynamiques grace au principe des composants qu'elle adopte pour créer des applications web rapides, modulaires et faciles à maintenir.
Dans le cadre de ce projet, React a été utilisé pour développer la partie front-end de l'application.

— **Tailwind CSS :** Tailwind CSS est un framework CSS utilitaire qui permet de concevoir des interfaces utilisateur en appliquant des classes directement dans le code HTML. Contrairement aux frameworks traditionnels basés sur des composants préconçus, Tailwind propose une approche "utility-first", offrant plus de flexibilité et de personnalisation dans la conception des interfaces [47].

— **Recharts** : c'est une bibliothèque React spécialisée dans la création de graphiques interactifs. Elle est utilisée pour afficher les résultats de prédiction et les données de consommation sous forme de graphiques.

— **html2canvas** : c'est une bibliothèque JavaScript qui permet de capturer une portion d'une page web et de la convertir en image. Elle est utilisée pour transformer un graphique en image avant impression et exportation.

— **jsPDF** :C'est une bibliothèque pour générer des PDF en JavaScript. Elle est utilisée pour exporter les graphiques en PDF.

— **lucide-react** : Bibliothèque d'icônes modernes et personnalisables pour React.

4.3.3.2 Back-End/API

— **FastAPI** : c'est un framework Web moderne, rapide (haute performance) pour créer des API du beckend avec Python, il est utilisé ici pour développer les points d'accès du back-end, gérer les routes et retourner les résultats au frontend.

— **Pandas :** pandas est une bibliothèque Python open-source optimisée pour la manipulation et l'analyse des données. Elle repose sur des structures comme les Series et les DataFrame, permettant un traitement efficace des données. Intégrée à l'écosystème scientifique Python, elle est largement utilisée en science des données et en machine learning[1].

— **Numpy (Numerical Python) :** numPy est une bibliothèque Python optimisée pour le calcul scientifique, offrant des tableaux multidimensionnels (ndarray) et des opérations mathématiques performantes. Essentielle en data science[46].

— **itertools** : module standard Python fournissant des outils pour la manipulation efficace d'itérations, il est utilisé pour générer des combinaisons de paramètres.

4.3.3.3 Modélisation et ML

- **statsmodels** : c'est une bibliothèque python qui a été utilisée pour les modèles statistiques comme les séries temporelles pour construire des modèles SARIMA et SARIMAX.
- **Optuna** : Optuna est une bibliothèque open source de Python dédiée à l'optimisation automatique des hyperparamètres des modèles de machine learning et deep learning. Elle repose sur des algorithmes intelligents pour trouver rapidement la meilleure combinaison de paramètres permettant d'améliorer la performance d'un modèle.
- **Matplotlib :** Matplotlib est une bibliothèque de visualisation pour Python qui permet de créer une large gamme de graphiques, allant des plus basiques aux plus avancés. Elle offre une personnalisation fine des éléments graphiques, tels que les axes, les couleurs et les styles. Grâce à son intégration avec NumPy et Pandas, elle est un outil clé en data science et en analyse de données[20].
- **Scikit-learn (sklearn) :** Scikit-Learn est une bibliothèque open-source en Python, conçue pour l'apprentissage automatique. Reposant sur NumPy, SciPy et Matplotlib, elle met à disposition une vaste gamme d'algorithmes pour la classification, la régression, le clustering et la réduction de dimension. Grâce à son API simple et cohérente, elle permet une implémentation efficace des modèles et une évaluation optimisée des performances, en faisant un outil incontournable en machine learning et analyse de données[36].
  Elle est utilisée ici pour la normalisation des données (MinMaxScaler) et les métriques d'évaluation (MAE, MAPE, RMSE).
- **Tensorflow :** Tensorflow est une bibliothèque open-source axée sur le calcul numérique et l'apprentissage profond. Son architecture flexible repose sur des graphes de tenseurs, assurant des performances optimisées sur CPU, GPU et TPU. Combiné à une API intuitive et une intégration native de Keras, il permet de construire, former et déployer des modèles d'IA avec efficacité, consolidant son rôle central dans les domaines académiques et industriels[31].
  Il est utilisé ici pour construire et entraîner des réseaux GRU.
- **Keras :**Keras est une bibliothèque open-source haut niveau pour l'apprentissage profond. Intégrée à TensorFlow, elle simplifie la création de réseaux de neurones via une API modulaire (conception, entraînement, évaluation). Compatible CPU/GPU, outils avancés inclus. Utilisée en vision par ordinateur, NLP, reconnaissance vocale, bioinformatique [3].
  Le earlyStopping (Keras) est utilisé pour arrêter l'entraînement d'un modèle lorsque la performance cesse de s'améliorer afin d'éviter le surapprentissage.

4.4 Application des méthodes de prédiction sur les données réelles de l'entreprise

L'entreprise nous a fourni un jeu de données contenant la consommation mensuelle électrique estimée en kilowattheure (kWh) de 7 clients sur une durée de 4 ans (de 2016 à 2019), L graphique de la production d'électricité globale de l'entreprise durant ces années est représenté ci-dessous. Les méthodes de prédiction choisies (SARIMA, SARIMAX, GRU) ont été appliquées dans un premier lieu pour prédire les données de production future de l'entreprise.

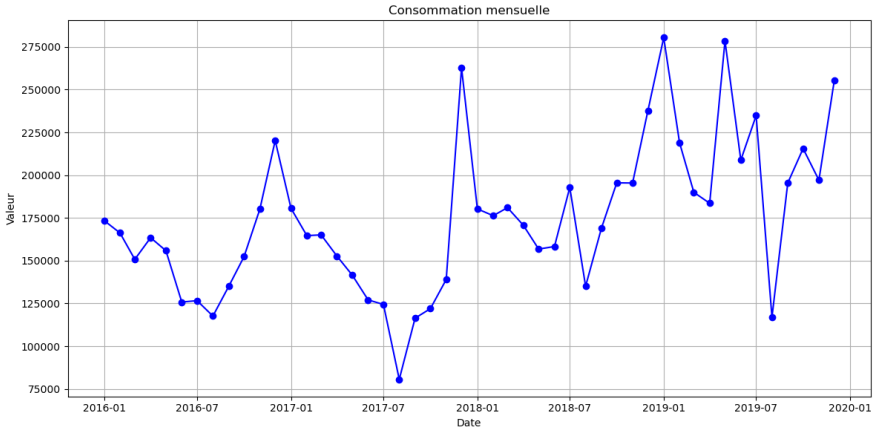


FIGURE 4.1 – Représentation graphique des données de production d'électricité de l'entreprise de 2016 à 2019

4.4.1 Analyse et décomposition STL des données de l'entreprise

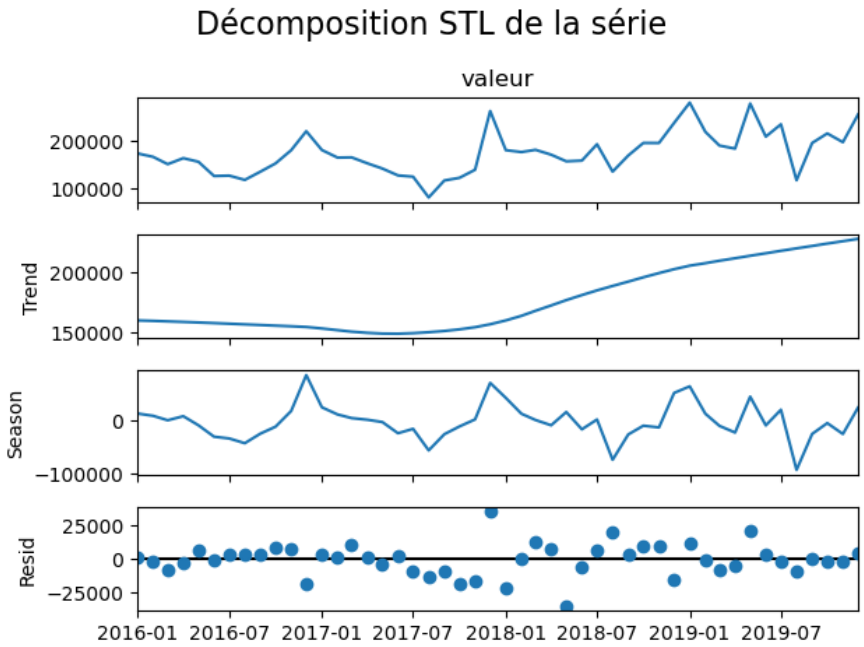


FIGURE 4.2 – Décomposition STL des données de l'entreprise

— Tendance : l'analyse des graphes de décomposition montre clairement que la production de l'entreprise présentent une **tendance haussière**.
— Saisonnalité : l'écart type de la saisonnalité et l'écart type des fluctuations (résidu) ont été calculés et leurs résultats sont respectivement : 29910.53 et 18965.97
29910.53 est supérieur à 18965.97, cela met en évidence la présence d'une **saisonnalité significative** d'une période de 12. Par conséquent, concernant les modèles statistiques **SARIMA** et **SARIMAX** sont les plus appropriés pour analyser ces données.

4.4.2 Stationnarité de la série

Le test ADF donne le résultat suivant : p-value = 0.08405

La p-value est supérieur à 0,05, et nous considérons un seuil de 5% : La série est **stationnaire** donc aucune différenciation n'est nécessaire : les paramètres de différentiation sont **d=0** et **D=0**.

4.4.3 Choix des paramètres

Afin de choisir les autres paramètres des méthodes statistiques, les graphes ACF et PACF permettent d'obtenir une première idée sur les ordres à retenir.

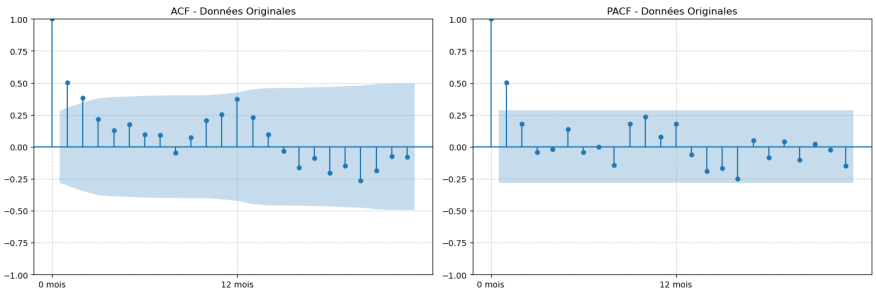


FIGURE 4.3 – Représentation graphique du ACF et du PACF des données de l'entreprise

Selon les graphiques le modèle (1,0,1)(0,0,1,12) est suggéré comme hypothèse de départ.

L'algorithme de sélection du modèle a finalement choisi le modèle (1,0,0)(1,0,0,12). Ce processus a impliqué l'évaluation de plusieurs compositions de modèles, en retenant d'abord celles présentant le meilleur AIC, puis en sélectionnant le modèle avec le meilleur MAPE parmi ces dernières.

Les résultats obtenus avec le modèle choisi sont

Indicateur	**Valeur**
AIC	700,1
MAPE	17,54%

TABLE 4.1 – Indicateurs de performance du modèle

4.4.4 Application de la méthode SARIMA

Une prédiction des 6 derniers mois du jeu de données a été effectuée afin de comparer les valeurs réelles et les données prédictives de test.

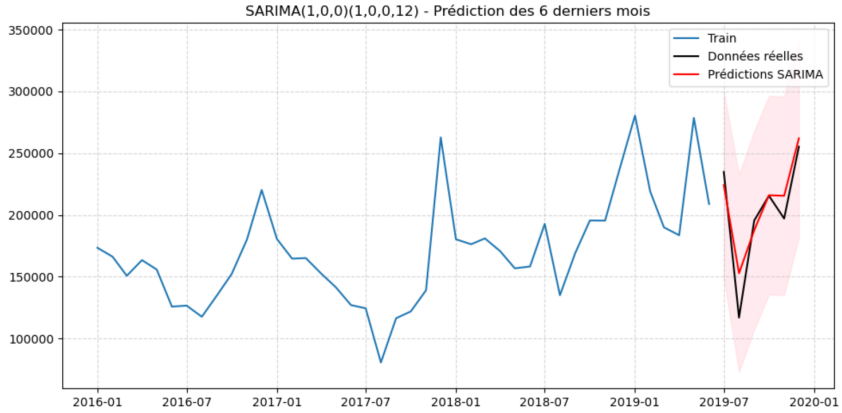


FIGURE 4.4 – Comparaison entre la prédiction SARIMA des 6 derniers mois des données de l'entreprise et ses données réelles

Le graphe montre les résultats du modèle SARIMA avec un MAPE de 17%, ce qui représente une erreur acceptable pour ce type de données. Cela indique que le modèle prédit globalement bien la consommation d'électricité et qu'il est capable de suivre les tendances et la saisonnalité des séries temporelles. Ces résultats confirment que SARIMA est un modèle fiable pour la prévision dans ce contexte

4.4.5 Application de la méthode SARIMAX

Concernant la méthode SARIMAX, ses paramètres (p,d,q)(P,D,Q,s) sont identiques à ceux du modèle SARIMA sélectionné. La différence réside dans l'intégration de variables exogènes qui améliorent la compréhension des variations et, par conséquent, la prédiction du modèle.

Les données exogènes ajoutées sont les températures moyennes mensuelles enregistrées à Bejaia de 2016 à 2019.

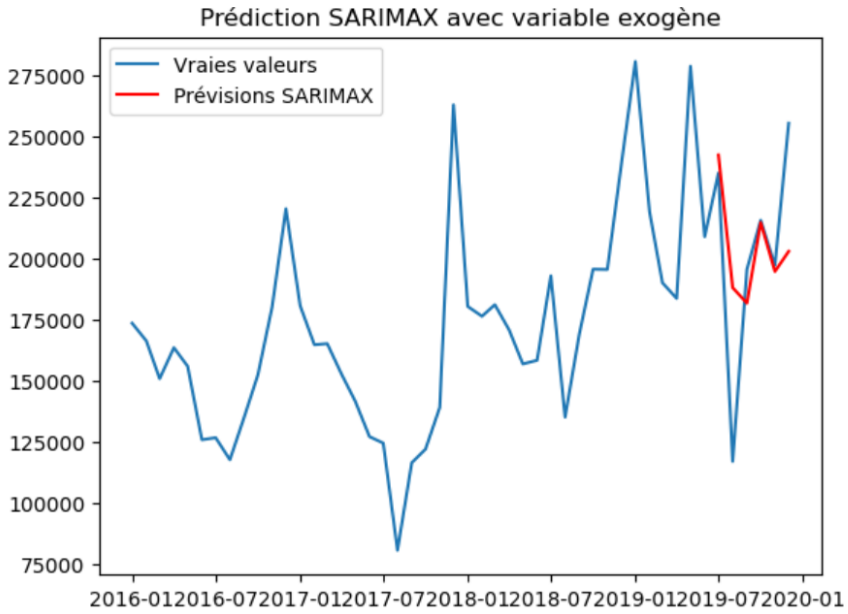


FIGURE 4.5 – Comparaison entre la prédiction SARIMAX des 6 derniers mois des données de l'entreprise et ses données réelles

Les résultats obtenus donnent :

Indicateur	**Valeur**
AIC	327,59
MAPE	12,26%

TABLE 4.2 – Indicateurs de performance du modèle

La méthode SARIMAX montrent de meilleurs résultats que ceux de SARIMA, ce qui signifie que les données exogènes on aidé le modèle à mieux comprendre les fluctuations et à fournir un résultat plus précis et plus généralisé.

4.4.6 Application de la méthode GRU

4.4.6.1 Normalisation des données

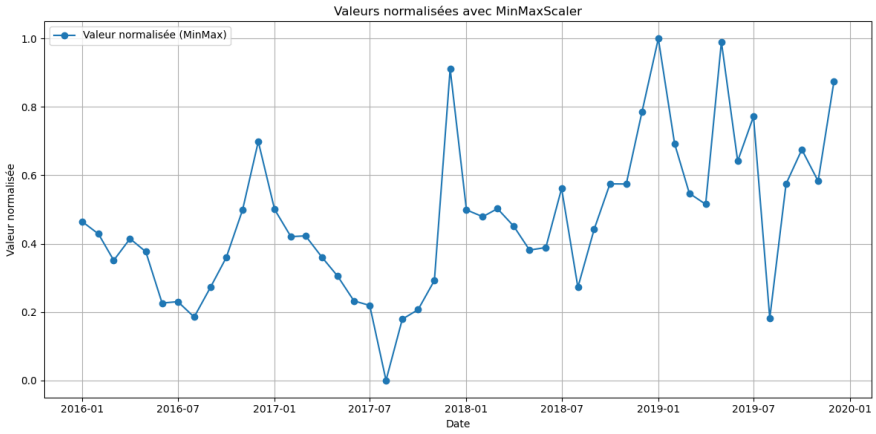


FIGURE 4.6 – Données de l'entreprise normalisées

Après la normalisation, les données varient sur une échelle de 0 à 1

4.4.6.2 Création du modèle GRU

Le jeu de donnée transmis par l'entreprise ne contient qu'un nombre limité d'observations temporelles, un modèle contenant 3 couches internes est suffisant pour capturer les dynamiques temporelles essentielles et conserver la capacité de généralisation, un modèle plus complexe augmente le risque du sur-appremtissage. Pour les mêmes raisons une couche de désactivation suffit.

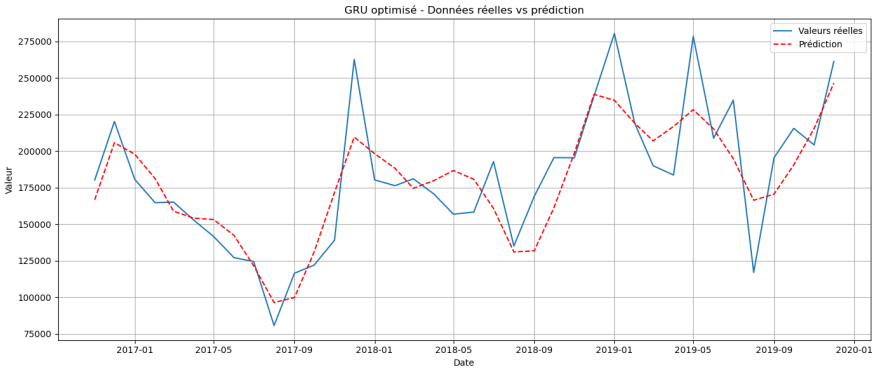


FIGURE 4.7 – Comparaison entre la prédiction GRU et les données réelles de l'entreprise de 2016 à 2019

4.4.6.3 Les hyperparamètres sélectionnés

Les hyperparamètres du modèle GRU final choisi sont :

Paramètre	Valeur
`learning_rate`	0.00691385592917535
`units`	32
`dropout`	0.2868902366349443
`epochs`	93
`batch_size`	4
`loss`	MSE

Table 4.3 – Hyperparamètres optimisés du modèle GRU

4.5 Analyse et comparaison des résultats des méthodes

Méthode	**Caractéristiques**	**Observations**	**MAPE (%)**
SARIMA	Composantes saisonnières et tendancielles	Bonne performance sur séries structurées ; ne gère pas les variables externes	∼17%
SARIMAX	SARIMA avec variables exogènes	L'ajout de variables externes améliore la précision des prévisions	∼12%
GRU	Réseau neuronal récurrent (séries temporelles)	Capture les dépendances complexes, plus flexible et performant que SARIMA/SARIMAX	∼10%

Table 4.4 – Comparaison des performances des modèles SARIMA, SARIMAX et GRU

Les résultats montrent que :

— Tous les modèles ont un taux d'erreur inférieur à 20%, ce qui confirme leur efficacité.
— L'ajout de variables exogènes (SARIMAX) améliore significativement les performances par rapport à SARIMA.
— Le modèle GRU surpasse les approches statistiques en offrant les prévisions les plus précises.

Cela confirme l'intérêt de combiner méthodes traditionnelles et deep learning pour une meilleure modélisation des séries temporelles.

4.6 Présentation de l'application

L'architecture de cette application web repose sur une séparation claire entre le front-end et le back-end. Le front-end est développé avec React, une bibliothèque JavaScript permettant de créer des interfaces utilisateur dynamiques et réactives. Le back-end est géré par FastAPI, un framework Python léger et performant pour la création d'API web. Le navigateur envoie des requêtes HTTP via l'interface React, qui sont ensuite traitées par FastAPI côté serveur. Celui-ci exécute la logique métier, traite les données,

puis renvoie une réponse que React affiche à l'utilisateur. Cette architecture permet une communication efficace entre l'interface et les fonctionnalités du système, tout en assurant modularité et maintenabilité.

4.6.1 Interface graphiques

Quelques interface graphique de l'application réalisée sont présentées ci dessous :

4.6.1.1 Interface graphique d'authentification

L'interface d'authentification illustrée dans la figure 4.8, permet à l'utilisateur de saisir son nom d'utilisateur et son mot de passe afin de pouvoir accéder aux fonctionnalitées du système.

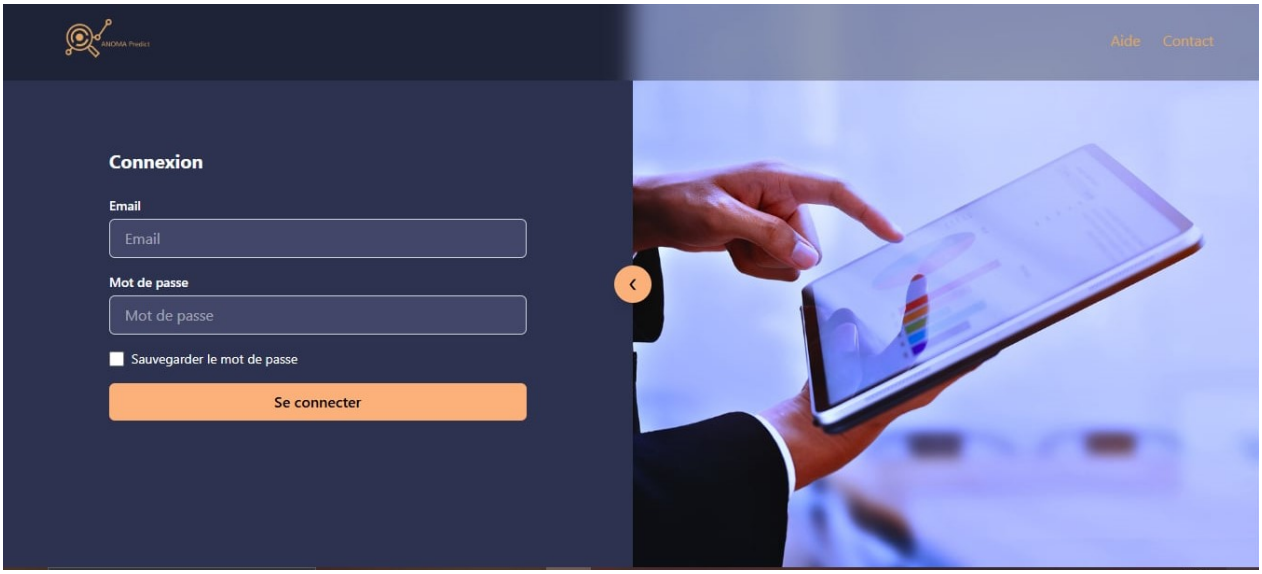


FIGURE 4.8 – Interface d'authentification

4.6.1.2 Coté client

Interface Profil du client : l'interface profil du client illustrée dans la figure 4.9, permet au client de l'entreprise de visualiser ses informations personnelles et de les modifier.

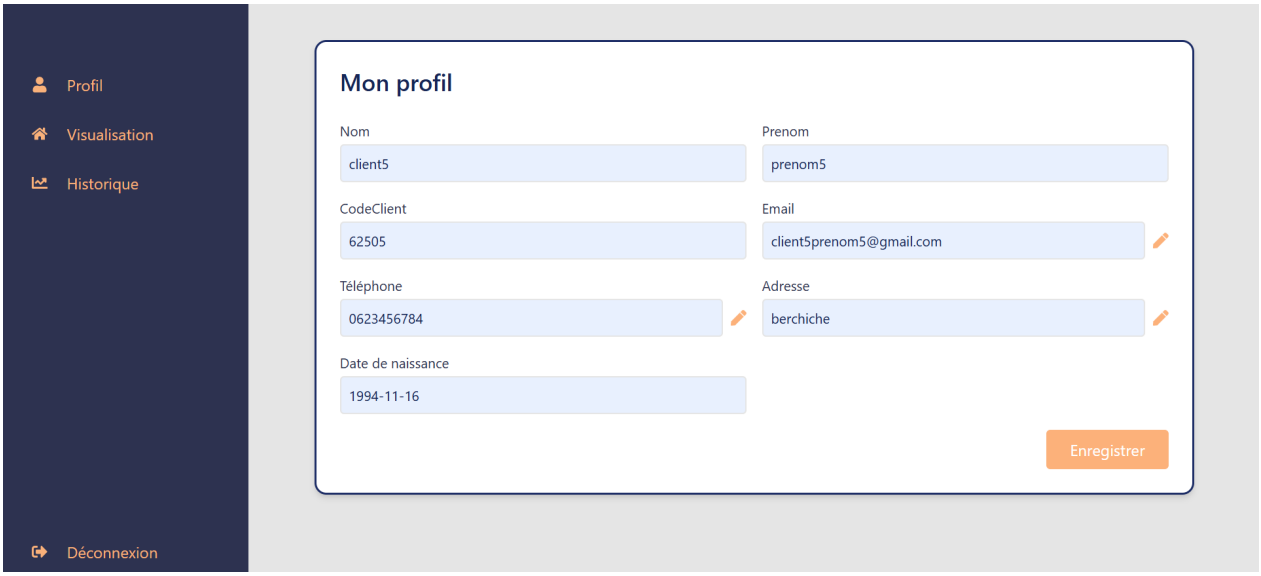


FIGURE 4.9 – Profil du client

Interface du dashboard du client : l'interface dashboard du client, illustrée dans la figure 4.10, permet au client de l'entreprise de visualiser ses consommations de l'année en cours, ainsi que la prédiction de sa consommation pour le prochain trimestre. Elle offre également une fonctionnalité de recherche permettant de filtrer les consommations directement via un tableau interactif. De plus, le client peut consulter ses consommations mensuelles, annuelles, ou détaillées par mois, à l'aide de graphiques de type histogramme et de tableaux dynamiques, facilitant ainsi une lecture claire et comparative de l'évolution de sa consommation.

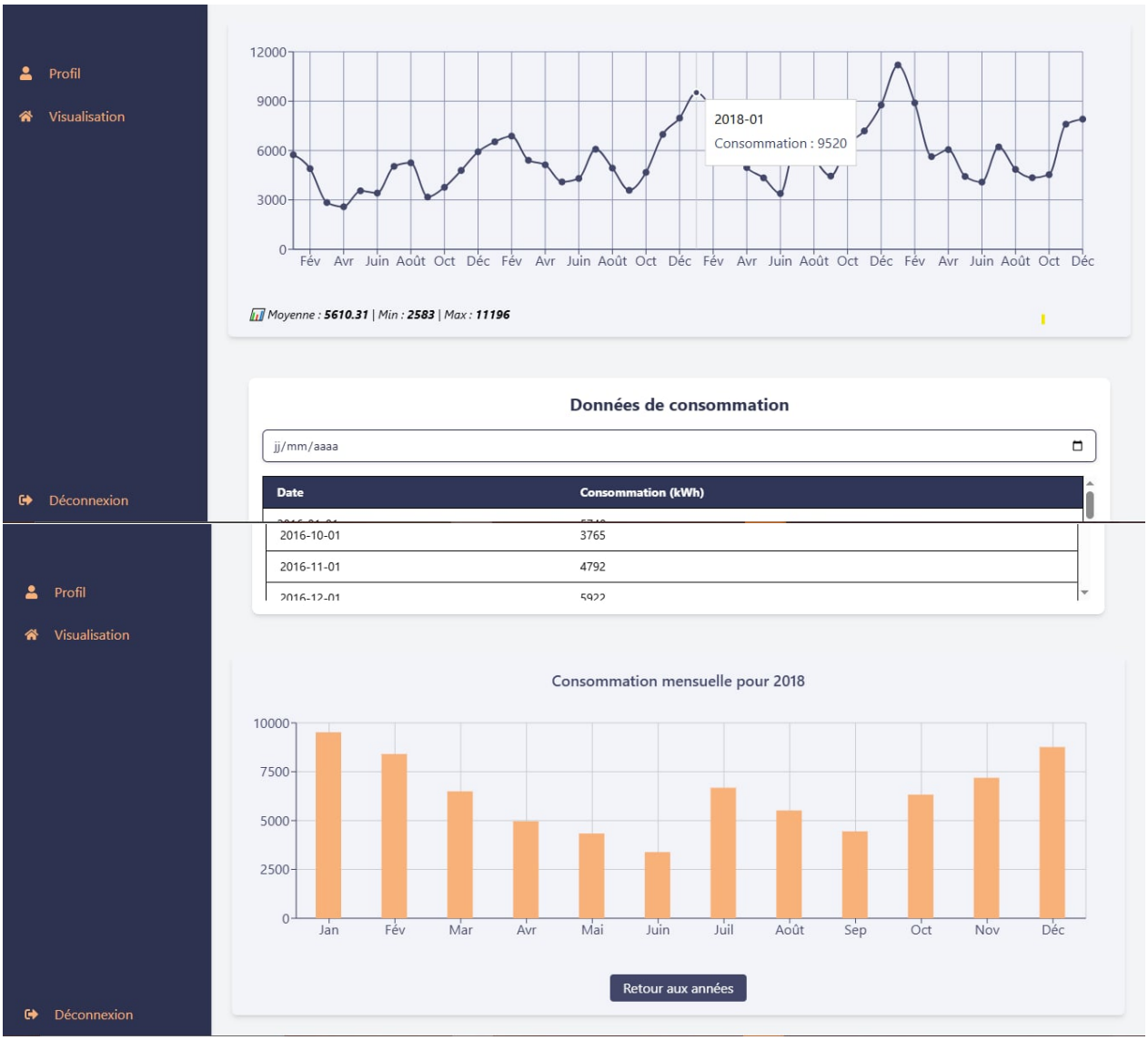


FIGURE 4.10 – Dashboard client

4.6.1.3 Coté administrateur

Dashboard : l'interface dashboard de l'administrateur illustrée dans la figure 4.11, lui permet principalement d'avoir une vue générale sur sa production d'électricité historique annuelle et mensuelle,et intègre un tableau dynamique permettant de rechercher et filtrer les données de consommation de manière efficace :

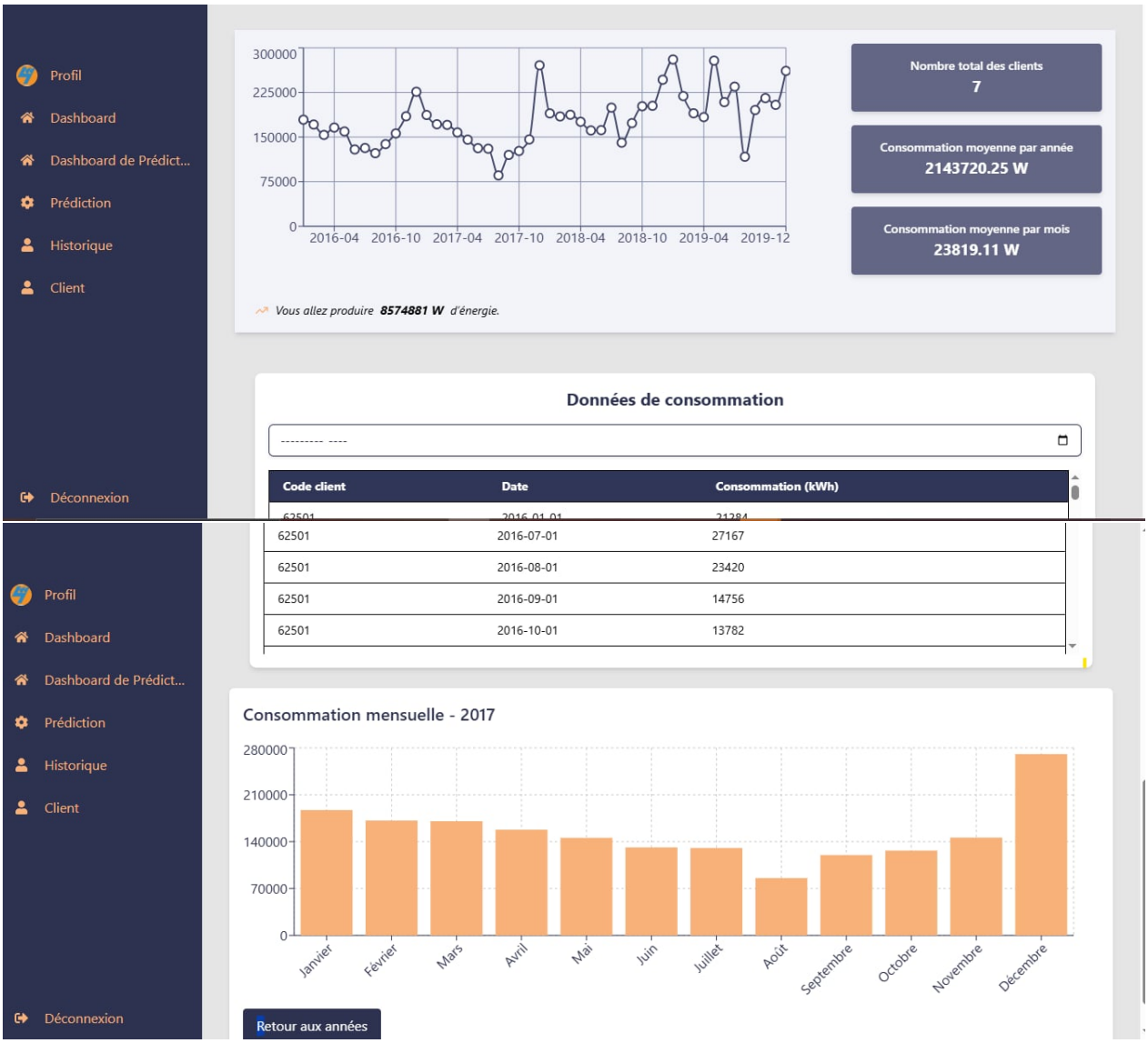


FIGURE 4.11 – Dashboard

Dashboard de prédiction : l'interface dashboard de prédiction illustrée dans la figure 4.12, permet à l'administrateur de visualiser une prédiction future et des informations qui y sont liées ainsi que la comparaison de sa dernière prédiction avec ses données réelles.

FIGURE 4.12 – Dashboard de prediction

Interface de visualisation des prédictions historiques : l'interface des prédictions historiques illustrée dans la figure 4.13, permet à l'administrateur de revoir n'importe quelle prédiction déjà réalisée et la comparer avec les données réelles

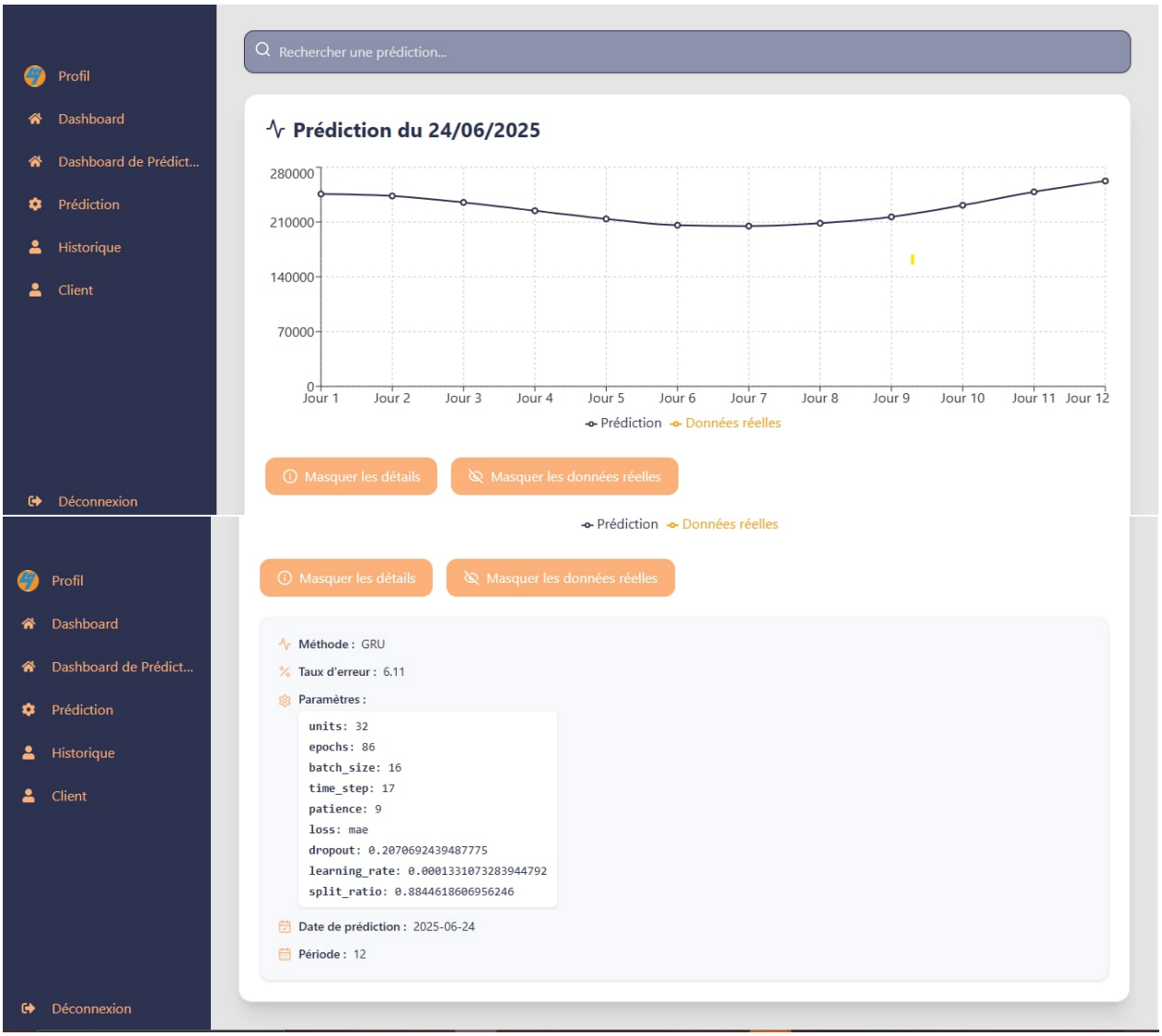
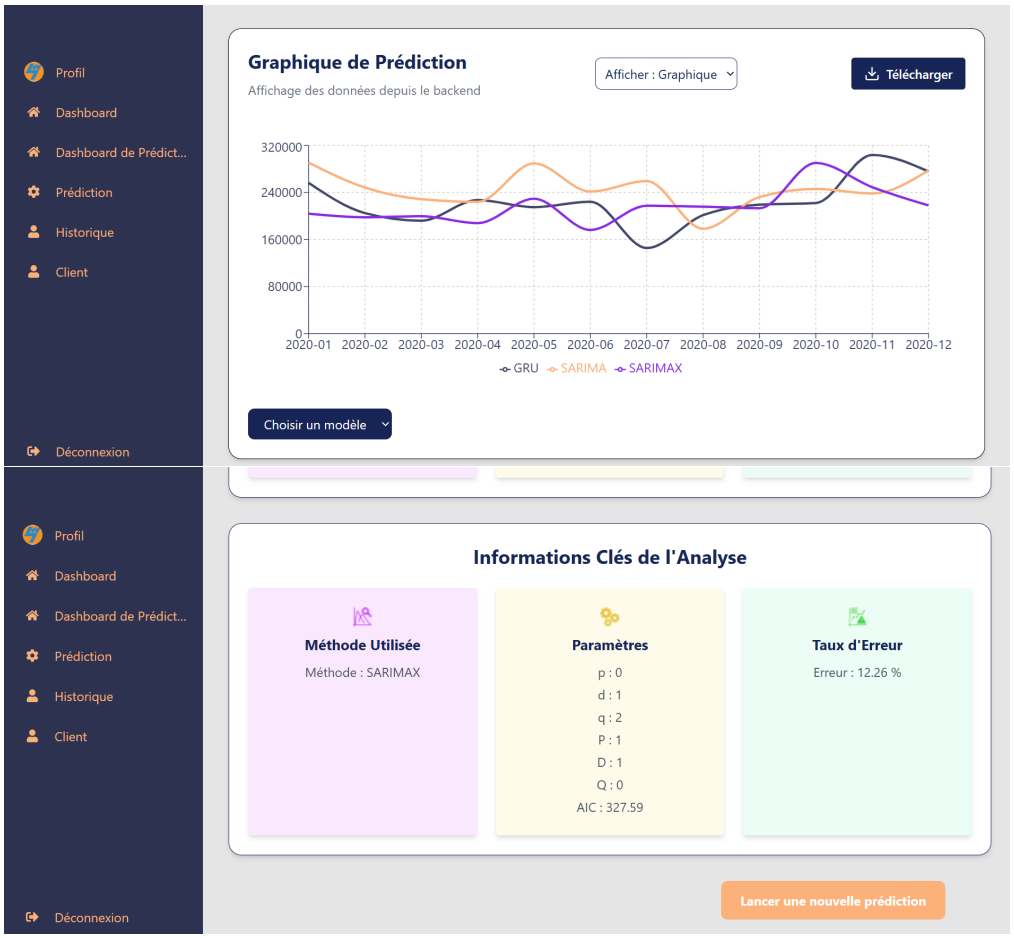


FIGURE 4.13 – Prédiction Historique

Interface gestion des clients : l'interface de gestion des client illustrée dans la figure 4.14, permet a l'utilisateur d'ajouter des clients, de modifier leurs informations, de les supprimer, d'effectuer une recherche, de visualiser la consommation et de renouveler le modèle de prédiction des clients.

FIGURE 4.14 – Gestion des clients

Interface de prédiction : l'interface prédiction illustrée dans la figure 4.6.1.3, permet a l'administrateur d'effectuer des prédictions en lencant chacune des trois méthodes (SARIMA, SARIMAX, GRU) et en indiquant la periode à prédire, de visualiser leurs résultats ainsi que leurs paramètres et leurs taux d'ereur et de choisir un modèle parmi les trois résultats proposé.



4.7 Conclusion

Ce chapitre pratique nous a permis de concrétiser les concepts abordés précédemment, en détaillant l'environnement de développement et les outils sélectionnés. Les résultats obtenus grâce à l'application des méthodes sur les données de l'entreprise ont été illustrés, ainsi que les interfaces visuelles de l'application. Cette section s'est clôturée par une analyse comparative des performances et une synthèse des apports de notre travail.

Conclusion générale et perspective

En conclusion, ce projet a facilité la conception et le développement d'une application web de prévision et d'analyse des ventes d'électricité aux clients de Sonelgaz Béjaïa. Baptisé « Approche d'apprentissage profond pour prédire la consommation d'électricité », ce travail a été réalisé dans le cadre de répondre aux besoins d'enrichissement de la prise de décision au sein de la direction commerciale.

Notre solution a eu affaire avec plusieurs problématiques concrètes de l'entreprise, à savoir : le déséquilibre entre production et consommation entraînant des surplus ou des pénuries d'électricité, la difficulté de détecter les fraudes liées aux compteurs énergétiques, avec des conséquences sur la fiabilité des factures, l'absence de système d'analyse prédictive ou d'aide à la décision, en particulier pour l'anticipation des ventes.

L'emploi de la méthode agile Scrum a privilégié une organisation structurée du projet autour de sprints clairs, facilitant la planification, la collaboration avec le client et l'adaptation progressive aux besoins fonctionnels définis dans le Product Backlog.

Par ailleurs, la modélisation UML a contribué à la bonne compréhension des flux, à la formalisation des fonctionnalités, et à une communication efficace au sein de l'équipe de développement. Sur le plan technique, l' intégration de modèles d'apprentissage automatique (SARIMA, SARIMAX) et deep learning (GRU) a permis d'améliorer la précision des prédictions.

Chaque modèle a été choisi et optimisé en fonction des caractéristiques des données disponibles, pour fournir des résultats pertinents pour la prévision des tendances de consommation et de production. En perspective, plusieurs axes d'amélioration peuvent être envisagés : no-

tamment l'intégration de modèles plus performants ou la combinaison entre deux modèles differents, ainsi l'ajout de nouvelles fonctionnalités telles que la détection d'anomalies ou les notifications intelligentes. Une autre piste importante serait de généraliser l'application, non pas uniquement pour une seule entreprise, mais pour un ensemble d'utilisateurs ou d'organisations, dans des contextes variés.

Outre une réussite fonctionnelle, ce projet a donné une expérience d'apprentissage complète, tant technique qu'organisationnelle. Il nous a permis de consolider des compétences en programmation, en modélisation prédictive, et en intelligence artificielle, tout en nous confrontant aux réalités d'un projet d'envergure professionnelle : gestion du temps, résolution de problèmes, travail en équipe.

Bibliographie

[1] Femi Anthony. *Mastering pandas*. Packt Publishing, 2015.

[2] Dor Bank, Noam Koenigstein et Raja Giryes. "Autoencoders". In : *Machine learning for data science handbook : data mining and knowledge discovery handbook* (2023), p. 353-374.

[3] Ekaba Bisong et al. *Building machine learning and deep learning models on Google cloud platform*. Springer, 2019.

[4] Michael Blaha et James Rumbaugh. *Modélisation et conception orientées objet avec UML 2*. Pearson Education, 2005.

[5] Pascale Bourbonnais. "Analyse de la performance du système portuaire de l'Arctique canadien". In : (2010).

[6] George EP Box, Steven C Hillmer et George C Tiao. "Analysis and modeling of seasonal time series". In : *Seasonal analysis of economic time series*. NBER, 1978, p. 309-344.

[7] Peter J. Brockwell et Richard A. Davis. *Introduction to Time Series and Forecasting*. Springer, 2002.

[8] Christian J Burnham et Sotiris S Xantheas. "Development of transferable interaction models for water. III. Reparametrization of an all-atom polarizable rigid model (TTM2–R) from first principles". In : *The Journal of chemical physics* 116.4 (2002), p. 1500-1510.

[9] Laurent Candillier. "Contextualisation, visualisation et évaluation en apprentissage non supervisé". Thèse de doct. Université Charles de Gaulle-Lille III, 2006.

[10] Enze Chen et Mark Asta. *Using Jupyter tools to design an interactive textbook to guide undergraduate research in materials informatics*. Consulté en janvier 2025. 2022.

[11] Kyunghyun Cho et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In : *arXiv preprint arXiv :1406.1078* (2014).

[12] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In : *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar : Association for Computational Linguistics, 2014, p. 1724-1734. url : https://aclanthology.org/D14-1179.

[13] David A. Dickey et Wayne A. Fuller. "Distribution of the Estimators for Autoregressive Time Series with a Unit Root". In : *Journal of the American Statistical Association* 74.366 (1979), p. 427-431.

[14] Norman R Draper et Harry Smith. *Applied regression analysis*. T. 326. John Wiley & Sons, 1998.

[15] Mira Drias. *Prédiction de la consommation d'énergie en utilisant les technique de machine Learning (Cas : Sonelgaz Distribution Direction de Béjaïa)*. Projet de fin de cycle, Université de Béjaïa. 2024. URL : http://univ-bejaia.dz/dspace/123456789/24885.

[16] Ian Goodfellow, Yoshua Bengio et Aaron Courville. *Deep Learning*. Consulté le 15 mars 2025. Cambridge, MA : MIT Press, 2016. URL : https://www.deeplearningbook.org/.

[17] Allan David Gordon. *Classification*. CRC Press, 1999.

[18] Alex Graves. "Long Short-Term Memory". In : *Supervised Sequence Labelling with Recurrent Neural Networks*. Berlin, Heidelberg : Springer, 2012, p. 37-45.

[19] Edward J Hannan et Barry G Quinn. "The determination of the order of an autoregression". In : *Journal of the Royal Statistical Society : Series B (Methodological)* 41.2 (1979), p. 190-195.

[20] Ivan Idris. *Python data analysis cookbook*. Packt Publishing Ltd, 2016.

[21] University of Illinois at Chicago. *CS 484 - Client-Side Web Development : JavaScript Overview*. Consulté en avril 2025. 2025. URL : https://cs.uic.edu/~i340/notes/javascript.html.

[22] Pierre Larousse. *Grand dictionnaire universel du XIXe siècle*. T. 5. Larousse, 1869.

[23] Yann LeCun, Yoshua Bengio et Geoffrey Hinton. "Deep learning". In : *nature* 521.7553 (2015), p. 436-444.

[24] Zahia Loualia et Lydia Nait slimane. *Application des techniques d'intelligence artificielle pour l'analyse prédictive*. Projet de fin de cycle, Université de Béjaïa. 2023. URL : http://univ-bejaia.dz/dspace/123456789/23223.

[25] Rania M'hamdi. *Techniques d'intelligence artificielle pour l'analyse prédictive de la consommation d'électricité*. Projet de fin de cycle, Université de Béjaïa. 2023. URL : http://univ-bejaia.dz/dspace/123456789/23484.

[26] Batta Mahesh et al. "Machine learning algorithms-a review". In : *International Journal of Science and Research (IJSR).[Internet]* 9.1 (2020), p. 381-386.

[27] B Mahut. "Diagnostic des ouvrages en béton armé : Etat, méthodes, prévision du vieillissement : Méthodes d'investigation sur ouvrages en béton précontraint". In : *Diagnostic des ouvrages en béton armé (état-méthodes-prévisions du vieillissement, Saint Rémy lès Chevreuse, 12-13 octobre 1998)*. 1998, p. 67-76.

[28] Peter Meso et Radhika Jain. "Agile software development : Adaptive systems principles and best practices." In : *Information systems management* 23.3 (2006).

[29] Valérie Monbet et Pierre Ailliot. "Sparse vector Markov switching autoregressive models. Application to multivariate time series of temperature". In : *Computational Statistics & Data Analysis* 108 (2017), p. 40-51.

[30] João Nicolau. "Processes with volatility-induced stationarity : an application for interest rates". In : *Statistica Neerlandica* 59.4 (2005), p. 376-396.

[31] Bo Pang, Erik Nijkamp et Ying Nian Wu. "Deep learning with tensorflow : A review". In : *Journal of Educational and Behavioral Statistics* 45.2 (2020), p. 227-248.

[32] Constituency Parsing. "Speech and language processing". In : *Power Point Slides* (2009), p. 20.

[33] *Partial Autocorrelation Function (PACF)*. NIST/SEMATECH e-Handbook of Statistical Methods. Accessed : [Date of Access]. URL : https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc446.htm.

[34] PJM Interconnection. *Hourly Load : Metered - Energy Consumption Data (25/12/2004 - 28/12/2004)*. https://dataminer2.pjm.com/feed/hrl_load_metered/definition. Consulté le 6 juillet 2025. Données fournies par PJM Interconnection, organisation RTO aux États-Unis. 2004. URL : https://dataminer2.pjm.com/feed/hrl_load_metered/definition.

[35] Luciano Ramalho. *Fluent Python : Clear, concise, and effective programming*. " O'Reilly Media, Inc.", 2015.

[36] Sebastian Raschka, Joshua Patterson et Corey Nolet. "Machine learning in python : Main developments and technology trends in data science, machine learning, and artificial intelligence". In : *Information* 11.4 (2020), p. 193.

[37] Jean-Claude Rault. *HastaUML – Outil de modélisation UML à visée pédagogique*. https://www.iut.univ-paris8.fr/~jcrault/HastaUML/. Consulté en mai 2025. 2016.

[38] Pascal Roques et Franck Vallée. *UML 2 en action : de l'analyse des besoins à la conception*. Editions Eyrolles, 2011.

[39] Stuart J Russell et Peter Norvig. *Artificial intelligence : a modern approach*. pearson, 2016.

[40] Ken Schwaber et Jeff Sutherland. "The scrum guide". In : *Scrum Alliance* 21.1 (2011), p. 1-38.

[41] Ian Sommerville. *Software Engineering (10th Edition)*. Consulté en avril 2025. 2020. URL : https://www.pearson.com/en-us/subject-catalog/p/software-engineering/P200000003560/9780133943030.

[42] SONELGAZ. *Site officiel de SONELGAZ*. Consulté en mars 2025. 2025. URL : http://www.sonelgaz.dz.

[43] Richard S Sutton, Andrew G Barto et al. *Reinforcement learning : An introduction*. T. 1. 1. MIT press Cambridge, 1998.

[44] Paul M Swamidass. *Encyclopedia of production and manufacturing management*. Springer Nature, 2006.

[45] Pennsylvania State University. *Autocorrelation and Partial Autocorrelation*. 2023. URL : https://online.stat.psu.edu/stat510/lesson/2/2.2.

[46] Jake VanderPlas. *Python data science handbook : Essential tools for working with data*. " O'Reilly Media, Inc.", 2016.

[47] Adam Wathan et Steve Schoger. *Tailwind CSS : A Utility-First CSS Framework for Rapid UI Development*. https://tailwindcss.com/. Consulté en mai 2025. 2023.

Résumé

Les années récentes ont vu le fulgurant essor de l'intelligence artificielle, et notamment de ses sous-branches l'apprentissage automatique (ML) et l'apprentissage profond (DL), ouvrant la voie à l'épanouissement de solutions innovantes dans tous les champs. La branche énergétique n'y fait pas exception, les entreprises mettant en place les moyens nécessaires pour utiliser ces technologies afin de prévoir de manière plus efficace la consommation des clients et augmenter leurs services.

Dans cette optique, ce projet final a consisté à développer une application web d' analyse et de prévision des consommations électriques à la direction commerciale de Sonelgaz à Béjaïa . L' application est basée sur trois modèles temporels de séries complémentaires : **SARIMA**, **SARIMAX** (variables exogènes) et **GRU**, un modèle de réseau neuronal récurrent adapté aux séquences temporelles. Il dispose d'une interface interactive pour la visualisation des données historiques et la génération de prévisions .

L'évolution du projet a été menée selon une méthode agile avec l'approche **Scrum** comme base, et les tests ont été menés sur les données de consommation mensuelle de 7 clients sur une période de 4 ans.
Mots clés : Prévision , Séries temporelles , SARIMA, SARIMAX, GRU, Application Web, Méthode Agile, Consommation électrique .

Abstract

In recent years, the rapid evolution of artificial intelligence, particularly in the fields of Machine Learning and Deep Learning, has led to significant advances in data analysis and forecasting. These technologies are increasingly used by energy companies to better understand consumption patterns and anticipate customer demand.

This study presents the development of a web application designed to analyze and forecast electricity consumption within the commercial division of Sonelgaz in Béjaïa. The application leverages three time series forecasting models : **SARIMA**, **SARIMAX** (with exogenous variables), and the **GRU (Gated Recurrent Unit)** neural network.

Through an interactive interface, users can visualize historical data and generate consumption forecasts. The project was implemented using the **Scrum** agile methodology and tested on 4 years of monthly consumption data from approximately 7 clients.

Keywords : Time Series, Forecasting, Electricity Consumption, SARIMA, GRU, Web Application, Agile Methodology.